GRAVES IN THE DIGITAL REALM

- SMARTPHONE DEVELOPMENT FOR CEMETERY EXPLORATION

Master Thesis, by
Mathias Kofoed Poulsen and Peter Christian von Westphal Thomsen
M. Sc. Geoinformatics, Aalborg University June 13th 2013

Aalborg University Copenhagen

Aalborg University Copenhagen
A. C. Meyers Vænge 15
2450, Copenhagen SV

M.Sc. Geoinformatics Master Thesis

Title:
Graves in the digital realm -
Smartphone Development for Cemetery Exploration

Project Period:
February 1st to June 13th, 2013

Submission date:
June 13th, 2013

Project Supervisor:
Thomas Balstrøm

Group members:

Mathias Kofoed Poulsen
Student number: 20082672

Peter Christian von Westphal Thomsen
Student number: 20082679

Printed copies: 4
Number of pages: 116
Number of appendixes: 11

Have you ever wondered where a historical person is buried? Or have you ever gotten yourself lost at a cemetery looking for a specific grave? The project at hand presents at solution to this problem in the form of a geoinformatic system using a native Android application as frontend.

At first the access of data regarding the location of buried persons is discussed in order to get an understanding of the possibilities for developing a service solving the before mentioned issues.

This discussion is used as the starting point to plan, build and test a service that can guide cemetery sightseers to historical persons' graves, display additional information about them, and in general be used to explore a selected cemetery.

This report documents the work that was put into the project including structuring the project through project management, developing and designing the underlying system, and testing the initial prototypes through several usability tests.

PREFACE

This Master Thesis was written during the 4rd semester as the conclusion of the M.Sc. Geoinformatics program at Aalborg University, and is rated at 30 ECTS points. The Master Thesis, from now on mentioned as the report, is one of two key elements of a project, the other being a geoinformatic system, also mentioned as the product or service with an Android application, as frontend. Use of the system is illustrated throughout the report and in addition three video examples have been produced: http://youtu.be/OKz84XqJtfo, http://youtu.be/cta5hbOAWuI and http://youtu.be/byhppB9JYKs.

The project group who made the project consists of Mathias Kofoed Poulsen and Peter Christian von Westphal Thomsen. During the project period, the consulting firm Atkins, has provided guidance in the initial concept and used development technology of the Android application. Associate professor Thomas Balstrøm at University of Aalborg has been supervisor during the project period which ran from February 1st 2013 to June 13th 2013.

The project group would like to thank Atkins for allowing us to use their office space while programming. Furthermore we would like to thank Sofie Gjedde-Simonsen, Daniel Ossipoff Jensen, Bente Kilhof, Barbara Joos, Erik Bach, Gitte Lundinge, Birgitte Foghmoes, Brian Worm Jensen, Bent Nielsen, Vagn Andersen, Anne Lise Thomsen, Morten Fuglsang and Thomas Balstrøm for their support and counselling during the project period.

REPORT STRUCTURE AND METHOD OF RESEARCH

The present report is divided into four stages, representing the stages that the project as a whole has gone through during the entire project period. These stages are inspired by (Larson and Gray 2011) and consist of a defining stage, a planning stage, an executing stage, and a closing stage.

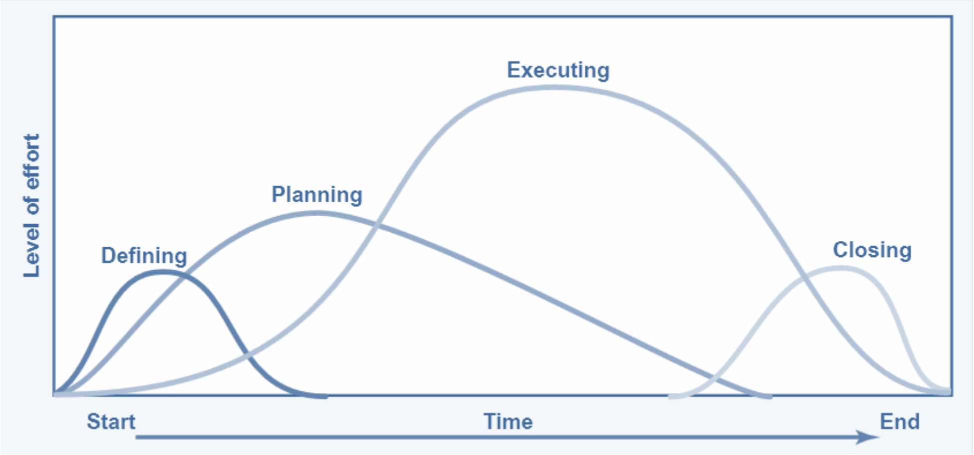


Figure 1 The flow between the four stages of the project. (Larson and Gray 2011)

As it can be seen in Figure 1, the individual stages during the project period do not typically end completely before the next stage is started, but instead follow a pattern where the main focus in the project period changes to a new stage while the previous stage is still being finished. This natural flow in between the stages of the project period is difficult to capture in a report, and instead all elements belonging to one specific stage are sorted and presented before moving on to the next stage. By doing this, it is easier for the reader to understand the process that the project has gone through, but at the same time this means that the chapters presented are not necessarily in chronological order, compared to when the described processes were executed.

The defining stage presents the research and considerations that went into specifying the focus area and problem statement of the project. Because much of the acquired information was not available in books or on websites, the collection of data and information for this stage was done primarily through interviews and mail correspondence with cemetery managers, culture centres and relevant authorities. These personal accounts are used to understand the use and limitations of existing solutions, which again are used to define the purpose of the project.

The Planning stage explains how the project group prepared for the development of the product. This was done by researching on relevant theory for the project such as programming, usability tests as well as project management. Since this stage had a heavily theoretical approach, the majority of the sources here come from theoretical text books such as *Project Management* (Larson and Gray 2011), *Interaction Design* (Rogers, Sharp and Preece 2002), and *Usability* (Gregersen and Wisler-Poulsen 2009), in order to have the backbone of knowledge needed for the stages to come.

The Executing stage documents how the product itself was built and evaluated, and how the researched theory was put into use for the development of the product. Because this stage focused on hands on developing and programming, it was difficult to find solutions to specific problems in text books, and instead internet forums for development such as stackoverflow.com, postgresql.org, and gis.stachexchange.com were used as primary sources. Here questions regarding development could be found to understand and solve issues.

The Closing stage sums up on the project, and is used to answer questions asked during the process. Because of the nature of this stage, no additional information is provided, and as such no new sources are presented, but instead knowledge gathered through the three previous stages are used to evaluate on the project as a whole.

System Data Sources

In the developed system, data from different sources was used. The network in the system was built by using FOT data from the Danish Geodata Agency. In addition Aarhus and Copenhagen municipality were so kind to deliver burial plot data for their cemeteries.

Reading Instructions

This section contains information of how references and specific definitions are used throughout the report.

References

In this report, The Chicago Manual of Style edition 15 author/date citation style has been chosen as documentation system and the sources are cited as follows:

([author] [publishing year])

The cited sources are fully described in a list of references in the bibliography section. Not all figures have a reference, the ones without were created by the authors. Certain information like images or references to websites in general are in the in the text only mentioned by the internet address and are not shown in the bibliography section.

The guideline given from Aalborg University, concerning plagiarism of making references if six or more contiguous words are used from a source is not followed in respect of code references. The reason for this is that certain functions in an Android application are only possible using multiple contiguous words, so only when several lines have been used a reference is given.

All tables, diagrams and pictures are labelled as figures and are numbered continuously throughout the text. Appendixes are numbered continuously using letters.

These elements are in the report referred to as follows:

- *Figure*: See Figure 1
- *Appendix*: See Appendix A

Examples of code are given several places throughout the report, the readability of these code snippets have been improved by the use of colour. Different languages and scripts have different notation, and the same way of colouring the code cannot be used for all. Notepad++'s use of colour coding different languages and scripts has been used in this report.

DEFINITIONS

Burial plot: Is the smallest identification unit the cemeteries use to identify a piece of land in the cemeteries. This can be a unit for a collection of graves for a whole family or just for a single person.

Grave: Is the physical place where a person is buried.

There are several places throughout the report where both words can be used to define the general area that is being referred to. In these cases the word grave will be used, as the main focus of this project is the position of individual graves and the persons buried at the specific locations.

TABLE OF CONTENTS

Defining Stage

DEFINING STAGE

The first stage in this report is the defining stage. This stage is used to expand the initial curiosity of the lack of services in the field of graves in the digital realm, into a suitable problem statement for this project. This is done through an introduction of the initial curiosity which was the reason as to why the project group first found interest in this field. At the start of the project, the curiosity is explored by acquiring knowledge about whether there actual is a lack of services, or if it is just the project group's perception. With precise knowledge of the chosen field, it will be possible see the reasons for the lack of services, and these reasons will be researched further, in order to provide the best possible position to find the goal of this project which is presented in the problem statement. These steps are presented through three main sections:

- **Introduction**
  This section introduces the field of interest that this project was created from, and determines the initial goal of the project.
- **Problem evaluation**
  The purpose of this section is to identify all of the underlying structures that need to be researched in order to understand the nature of the project.
- **Problem statement**
  This final section of the defining stage presents the problem of the project that the project group will try to solve through the report.

1 INTRODUCTION

Today's society is a knowledge society. Every single day new discoveries are made, and new information is added to the ever increasing pool of knowledge. As a result, no matter where you turn your head, you are faced with immense amounts of data ready to be collected, interpreted and used as a foundation to make the right decisions. Because of the nature of this knowledge society, there is growing demand to make existing data available to the public, so that new services which are beneficial to everyone can be developed. In other words, there is an expanding need to have information and data made available to everyone. The latest major example of this trend is the freely available basic data in several countries as Finland, Austria, Denmark, Spain and Australia, which has been released in the belief that having access to this data will strengthen the private sector. (Danish Ministry of the Environment 2012)

With this national and international trend in making data freely available, it is interesting to discover that there still exist corners in the realm of spatial data where accessibility of data is not a priority. Cemeteries house thousands of graves of both common and historical persons whose gravestones mark a distinct geographical link between the deceased and a location. This link can be used for many different types of services both internally for the cemetery administration as well as externally for visitors at the cemeteries. However, the link is not always easily accessible in the form of datasets, which makes it difficult to build services for external use.

There exist ongoing projects among certain cemeteries to develop services for external use. Despite this, it seems that many, both cemetery managers and people in general, are under the impression that data regarding the location of deceased persons is protected and that it cannot be made public, even though it is perfectly possible to determine a person's grave from the gravestone. Due to this, geoinformatic services aimed at cemetery use are very limited, and thereby there is a large market full of opportunities to be discovered.

This project will explore further into the problem of limited availability of data for graves in cemeteries, the reasons for it, and what kind of potential geoinformatic services that could be developed utilizing this data. This research will be used to develop a service that can ease the access to this type of data.

2 Problem Evaluation

To start off this report, the choice of building a geoinformatic service has to be included in a problem statement. This section breaks this process of creating a problem statement down into three sub sections. Each sub section has its individual purpose. The first section will analyse the initial curiosity and cover the potential needs and potential lack of geoinformatic services for cemeteries. The second section will identify the underlying elements of the problem by looking at society's influence in form of formal (legislation) and informal (ethical) rules. The third section will delimit the suitable parts of the identified problem and what kind of service this project will focus on.

2.1 The Existing Market of Graves in the Digital Realm

The first step in exploring further into the lack of geoinformatic cemetery services available, is an analysis of the market and the potential of geoinformatic services here. Is there a need for geoinformatic cemetery services, and what kind of needs would these services fulfil and do any solutions already exist?

Within the overarching topic of graves in the digital realm, there is a lot of interesting sub topics that could benefit from geoinformatic services. This section will identify the existing needs for geoinformatic services for cemeteries. Firstly information is presented regarding activities involving cemeteries, secondly the needs and the potential target groups involved in these activities are structured and explained in detail.

2.1.1 Information Sources

Several different sources of information were used to give an overview of which activities that took place at the cemeteries. By interviewing employees at Atkins, an insight was provided for municipalities' activities regarding administration of both maintenance and reservation of graves. These initial findings were further explored through several interviews with cemetery managers from both cemeteries managed by parish councils and city councils. See Appendix A for further details. In addition to these interviews, Assistens Culture Centre (Kulturcenteret Assistens) associated with Assistens Cemetery was also visited.

Cemetery Managers

The mangers provided insight into how both the daily maintenance provided by gardeners, as well as explaining that the individual graves could benefit from having a geoinformatic service that linked the physical grave and the cemetery's register. Another task that could benefit from a geoinformatic service was when the relatives of a deceased have to reserve a grave. This task is a time consuming process, and today it calls for an employee to be present at the cemetery to show the relatives around, which could be replaced by a digital service in the future. The cemeteries are also frequently requested to assist relatives in locating graves at the cemetery. This could be solved more efficiently with a digital self service.

Assistens Culture Centre

This institution is allegedly the only culture centre in the world connected to a cemetery. Gitte Lunding, centre manager at the facility informed that the centre arranges a number of services for the users of Assistens Cemetery in verbal, written and digital form (Lunding 2013). The verbal form includes guided tours around the cemetery where the guide is dressed as H. C. Andersen. The most recent initiative the centre was in charge of was the development of 25 short films of historical persons, which was financed by the Nordea Foundation with 1,560,000 DKK. The graves of the selected persons will have small QR codes that can be scanned with smartphones, so that the videos can be watched while standing at the grave of interest. This particular project is what the centre calls digital dissemination, and within this field there are many possibilities to use geoinformatic services to strengthen an experience while taking a walk in a cemetery. For the locals who are just passing through the cemetery or for a tourist who specifically came to the cemetery, it could for instance be possible to present all the stories that lie under each gravestone through different kinds of smartphone services. The goal for the culture centre is not to experience the cemetery through the smartphone, but to use it to broaden the real life experience with all the stories contained within the cemetery. (The Culture Centre Assistens 2013)

Digital Self Service

In regards to solutions for relatives reserving a grave, the Danish parliament has passed a bill requiring that mandatory electronic self service regarding communication with the government, regions and municipalities should gradually be implemented towards 2015. This means that relatives in the future if possible, have to report the death of a person using digital self services. This bill could influence the way funerals are going to be requested and registered, for instance by selecting the grave online as part of the process. If this was to be the case, several web services could be developed to meet the new demand. (Ministry for Economic Affairs and the Interior 2012) (The Government 2011)

2.1.2 Potential Purposes

As mentioned, there are a lot of potential for geoinformatic services, and to give a better understanding of each of them, the potential target groups need to be identified and their needs for geoinformatic services exposed in a structured way. In this field the different needs of geoinformatic services can be divided into four categories; Information, Navigation, Reservation, and Administration.

Information

Tourists and local people who go to a cemetery for recreational and sightseeing purposes can use information relevant to what they see at the cemetery. This can for instance be related to the stories behind the people who are buried. Many burial plots also have fascinating gravestones which also have interesting stories. Furthermore information about a cemetery itself can be provided.

Navigation

Navigation towards a position has several uses. It can be necessary both for relatives looking for a specific grave, sightseers looking for interesting graves and for people working professionally with managing cemeteries. Cemeteries can vary greatly in size, and the need for a tool regarding navigation within the cemetery can therefore vary, but at some of the larger cemeteries it is an issue to find ones way around. Due to this navigation can become a major issue that can be solved by creating a service that helps the user to travel through the cemetery and to find the location they are looking for.

Reservation

A cemetery receives many reservations of graves for funerals, and to integrate these reservations in a service is an obvious choice. This service can be used to let citizens make digital reservations.

Administration

It takes a lot of administration to maintain and run a cemetery. Each grave has to have a lot of information attached regarding maintenance, length of the lease, etc. Furthermore the future graves need to be thoroughly planned by dividing the cemeteries into sections that are either held free or used completely, so that the graves are not scattered throughout the cemetery, which interferes with the overall impression of the cemetery.

2.1.3 Target Groups

With the general needs of a geoinformatic cemetery service identified, it is time to indentify the potential target groups and their specific needs. Basically the potential target groups can be divided into two categories; Private Users - being people who use the cemetery for personal reasons, and Professional Users - being people whose job involve working with graves and cemeteries. The people using cemeteries for private use can be divided even further into two groups; Personal Users and Sightseers.

Personal Users

This group of users are people who are related to a deceased. One way a geoinformatic service could be utilised by personal users, would be to make it possible for them to navigate to a specific grave at the cemetery. As mentioned earlier it can become an issue to find ones way in larger cemeteries, and relatives can spend a long time wandering the cemetery trying to find a specific grave. Here a geoinformatic service would be of great help to the relatives, and save them a lot of time and worries when trying to find a grave.

When a person dies it is the family's responsibility to contact the municipality or the cemetery office to arrange the funeral including reserving graves. Because of the increased desire of digitalizing the communication with the public sector towards 2015, it is not difficult to imagine that it could be possible if not mandatory, to make the reservation of graves online through a geoinformatic service.

Sightseers

Many cemeteries house the graves of historical persons and other interesting graves. This creates the foundation for another very different group of private users, namely sightseers. Many people visit cemeteries to see these graves either to experience the beautiful surroundings or to visit some specific graves that are extraordinary, either because of the buried person or because of the gravestone or monument itself. This search could be helped with the use of a geoinformatic service that could navigate and plan trips to graves within cemeteries.

As with the private users' category, the professional users' category can be divided into two more specific types of users as well.

Administrative Users

As described previously it requires a lot of administration to run a cemetery. These tasks are maintained by different types of private or municipal cemetery offices. All of these tasks could be simplified, by integrating them in a geoinformatic service. This could both be done to organise data regarding each grave, which again could be used to simplify the work of cemetery workers, when they for example have to find out what graves that need to be maintained. In this regard, a navigation tool could also be used to make the employees work more efficient.

Cemetery related companies

The final group of users are the companies working with cemeteries. These mainly include undertakers. The undertaker is often the one who contacts cemeteries on behalf of the family of the deceased, and as such they could use a similar geoinformatic service to make reservations of graves as suggested for private users. This service could also be expanded to include data regarding each of the graves that the company is working on as an administrative tool.

2.1.4 FOCUS AREAS

The four described potential target groups and their specific needs of a geoinformatic service regarding graves have been combined with the four overall categories of needs, and have been listed in Figure 2.

	Private Users		**Professional Users**	
	Personal Users	Sightseers	Administrative Users	Cemetery related companies
Information		X		
Navigation	X	X	X	
Reservation	X			X
Administration			X	X

Figure 2 Potential target groups and their specific needs of geoinformatic related services.

2.1.5 EXISTING SERVICES AND INITIATIVES

The last section shows that there are several needs for geoinformatic services in relation to cemeteries and graves. To provide a better understanding of how new services could be developed, this section will take a closer look on what kind of existing services if any at all are available on the market, not only in Denmark but also internationally.

Information: QR codes and digital memorial pages

Digitalizing information about deceased persons is already a discipline that many people are exploring. An example of this could be that it is now possible to have QR codes engraved (Figure 3) on a gravestone, so that people can scan it and go to a website with descriptions and photos of the deceased. (stenhuggeriet.dk)

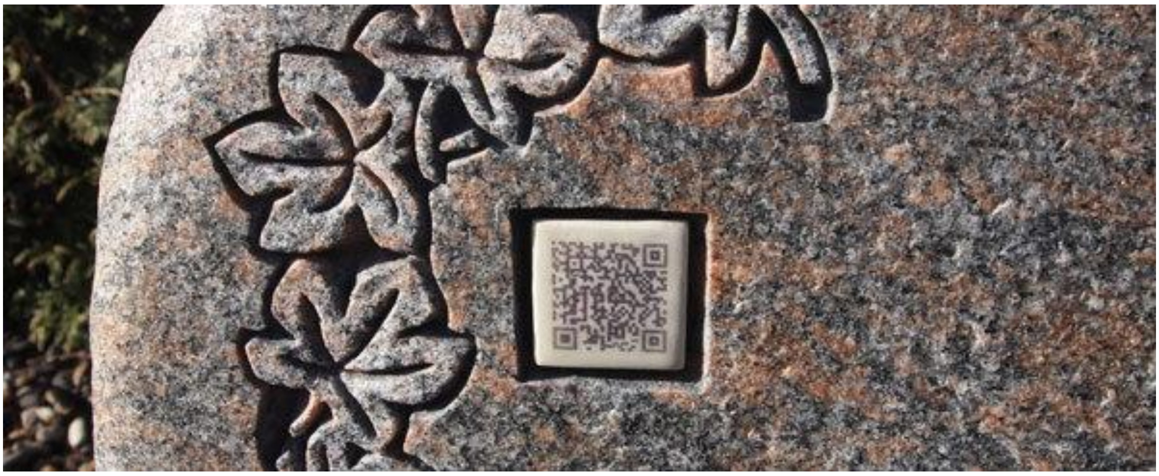


Figure 3 QR code on gravestones. (stenhuggeriet.dk)

Another example is the Facebook gravestone. Facebook now offers to have a deceased person's Facebook profile turned into an online gravestone. This is done to prevent people from befriending a deceased, but also to give friends and family an online place to meet and remember. (facebook.com)

These two websites are examples of digital memorial websites which originate from a different purpose than simply memorial, namely physical gravestones and digital social networks. Additionally websites dedicated solely to act as a memorial services exist, such as MyEternalSpace.com where a webpage for the deceased is created with photos, and relatives can post messages in memory of the deceased.

Information: Gravestone photo websites

There are a number of websites that focus on taking pictures of graves at various cemeteries. These websites are mostly run by amateur developers with an interest in cemeteries, and have no central administration connection to any cemetery offices. Due to this, the photos being taken are not coordinated, and the cemeteries covered by each websites are often random, depending on the pictures that have been collected so far, rather than having a systematic approach. As a result many of the websites overlap each other in regards to cemeteries that have already been covered, but since there is no collaboration or coordination, this is to expect. The two major Danish photo sites are Dansk Kirkegårds Index and DK-gravsten which each hold many thousand photos of gravestones. Common for the websites is the restriction of how to search for graves. It is only possible to search for graves within one cemetery at a time, so it is necessary to know what cemetery a person is buried at, to be able to see the photo of the gravestone.

An American version of a photo website is FindAGrave.com, whose main focus is American graves but it also contains information about graves from all over the world. Different from the Danish websites is that it is possible to search between millions of graves of both famous and common people by for example name, date of birth and death, or state and county within the US or other countries. This site hosts photos of the deceased persons and their graves, along with stories of their lives. Furthermore it is possible to leave a comment to express your sympathies, or share stories about the deceased.

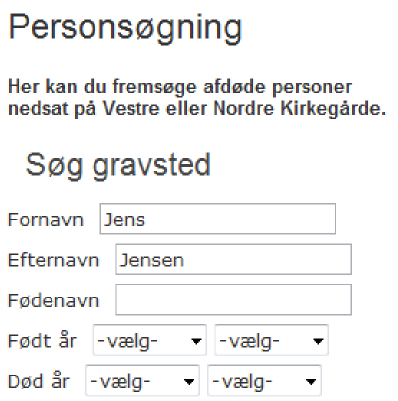
Information: Encyclopaedias

There exist many sources of information of famous persons, and even books for certain cemeteries like *"Midt i Verden i 250 år"*, Figure 4 have been published. These books not only concentrate on the stories of the persons, but also on the graves and gravestones themselves (Kulturcentret Assistens 2013). Much of this information is available online through encyclopaedias like the international Wikipedia or Den Store Danske, which opposed to Wikipedia, is written by scholars which should provide more trustworthy articles. (denstoredanske.dk) (wikipedia.org)

Figure 4 The book *"Midt i verden i 250 år"* – Assistens 1760 – 2010 which tells many of the hidden stories behind the graves and the gravestones. (Kulturcentret Assistens 2013)

Navigation: Location of, and navigating to a grave

In general the services for finding the locations of graves, and navigation within cemeteries in Denmark are limited to people having to ask the cemetery managers with one exception. One of the municipalities in Denmark which has the most advanced web services in regards to cemeteries is Aarhus Municipality. This municipality administers two cemeteries in Aarhus, and has built two websites that can be used to find the geographical location of graves. In the first service, you can search for, and locate a grave based on the grave ID, while you in the second can locate the grave of a person by typing in personal information such as name, date of birth, and date of death, Figure 5. Afterwards a map of the cemetery is presented with the position of the grave, Figure 6. (Aarhus Municipality n.d.)



Personsøgning

Her kan du fremsøge afdøde personer nedsat på Vestre eller Nordre Kirkegårde.

Søg gravsted

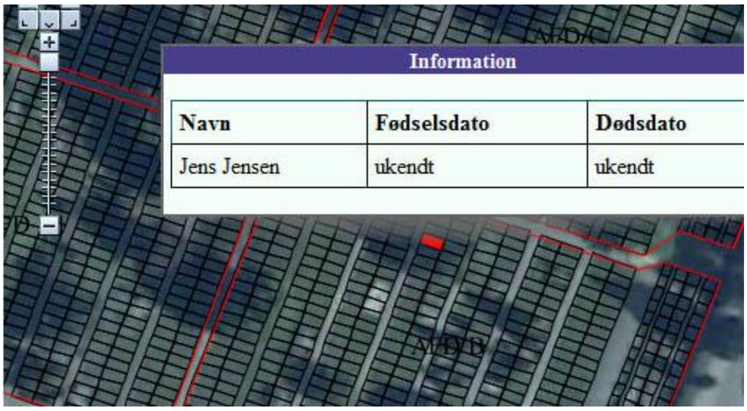
Fornavn Jens

Efternavn Jensen

Fødenavn

Født år -vælg- -vælg-

Død år -vælg- -vælg-



Information

Navn	Fødselsdato	Dødsdato
Jens Jensen	ukendt	ukendt

Figure 5 Search criteria for Aarhus Municipality's grave search service. (aarhus.dk)

Figure 6 Map view with the position of the selected grave from Aarhus Municipality's grave search service. (aarhus.dk)

Outside Denmark it seems that many more available services exist. The Geraldine cemetery in New Zealand has an online search service and cemetery maps with the surnames of people buried as seen on Figure 7 (Timaru District Council 2013). Brisbane City Council in Australia also has online search services with maps showing the position of the grave on the cemeteries (brisbane.qld.gov.au).

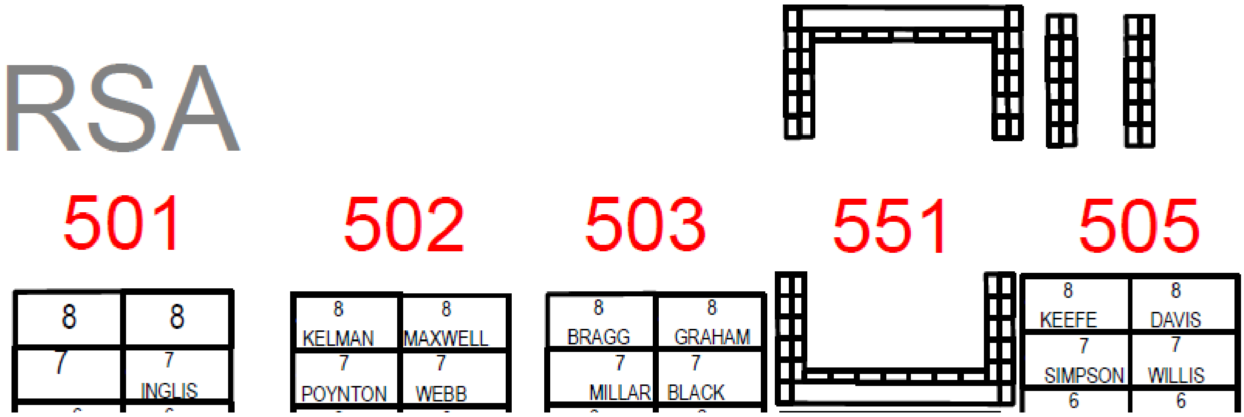


Figure 7 Grave map for Geraldine Cemetery in New Zealand. (Timaru District Council 2013)

More advanced services exist for the Recoleta Cemetery in Buenos Aires, Argentina, where many famous people are buried. This cemetery has touch screen monitors where it is possible to search for graves and see listings categorised by the deceased's occupation, see Figure 8 and Figure 9. For the different graves there are photos and a biography of the deceased. Each grave is also connected to a geoinformatic service to provide the location for the grave. (recoletacemetery.com)

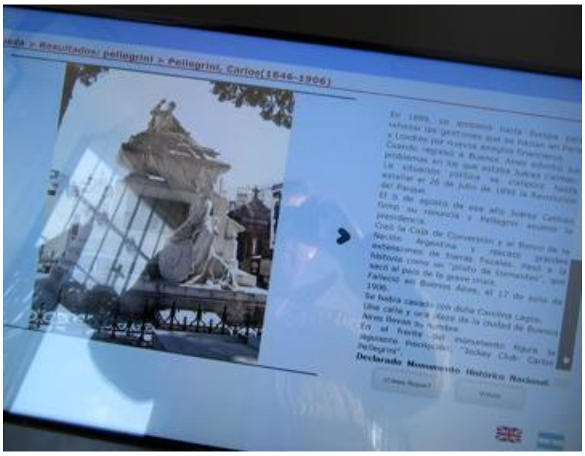

…da > Resultados: pellegrini > Pellegrini, Carlos(1846-1906)
Declarado Monumento Histórico Nacional.

Figure 8 Touch screen monitor with biography and photos. (recoletacemetery.com)

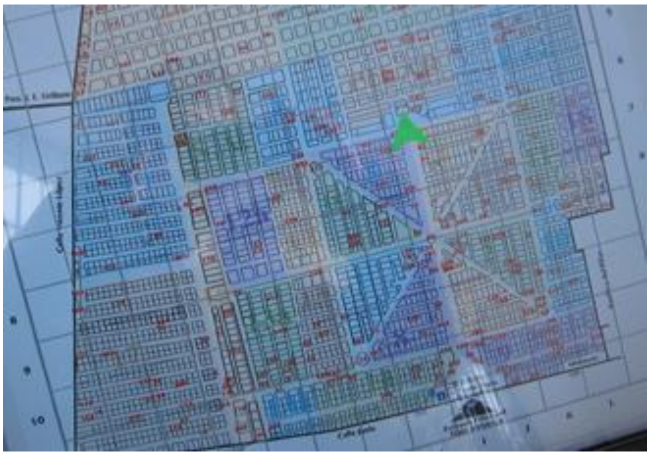


Figure 9 Touch screen monitor with map showing location of graves. (recoletacemetery.com)

According to the cemetery manager of Holmen Cemetery in Copenhagen, Birgitte Foghmoes, similar services exist in Finland, and Erik Bach from Aarhus Municipality, department of cemeteries, has also seen services like this for several Swedish cemeteries.

Reservation and administration

There exist many companies targeted towards cemeteries with a whole range of products and services. One of these is Brandsoft, which together with Skovbo Data have 99 % of the management software market within the Danish cemeteries for case management, accounting solutions and mobile products (Brandsoft 2013). The 6-city-cooperation between Copenhagen, Aarhus, Odense, Aalborg, Esbjerg and Randers municipalities are working together with Brandsoft to develop new services for tablets that can be used to see available graves, their cost, etc. (Bach 2013).

Another company is webCemeteries.com. This company has all sorts of products ranging from management systems, mapping, memorials and both mobile app and website solutions. They also offer to convert cemetery offices' records into different types of web solutions. One of these solutions could be a website where the user can search through names of deceased, and have the location of their graves displayed on a map, while another could be to have a mobile app that can be used for navigation and to present predesigned tours in the cemeteries. (webcemeteries.com)

The last two sections covering the potential focus areas and existing services and initiatives, show that there are a lot of possible services which could be developed and that there in general exist many services. For some areas however, there are few services available, and in Denmark in particular there is a lack of services for searching for locations of graves. Some services do exist but they are restricted in several ways, and in general they are not available to the public. The next section will look at formal and informal rules of the society to understand possible reluctance towards developing these services.

2.2 INFLUENCE OF THE SOCIETY

This section will identify the core elements responsible for the lack of geoinformatic service for cemeteries in Denmark. Apart from the existing formal rules the legislation sets, society also has a set of more informal rules which is the standards and norms that people accept and live by, which has a great importance in many cases, and particularly regarding deceased and their graves.

2.2.1 *FORMAL RULES: LEGISLATION AND DATA RECORDS*

As many other aspects of life in Denmark, the practice of burying people, and how the graves should be treated, is regulated through legislation. Therefore a number of laws have to be taken into account, either due to the accessibility of information or because of restrictions regarding the actual graves. Apart from the laws that deal with specific graves, it is also relevant to take a look at the existing possibilities of searching for information of the grave of a deceased, as well as how legislation for personal data is managed and controlled. By investigating these topics, it will become apparent how deceased are registered, if a central register of these records exists and if it is possible to access this information.

Funerals and the Physical Grave

In The Burial Act (Begravelsesloven) it is described how to apply for and perform funerals, as well as how burial sites and individual graves should be constructed and used. This law prescribes how citizens are supposed to contact the burial authority with information about the deceased and the burial, and how coffin burials only may take place at sanctioned cemeteries. For people cremated, it is also possible to have the ashes spread over open water, and it is possible to apply the diocesan to be buried on private property or other seemly ways of treating the ashes (Ministry for Ecclesiastical Affairs 2010), (Ministry for Gender Equality and Ecclesiastical Affairs n.d.) and (personregistrering.dk n.d.)

The Church and Cemetery Act (Lov om folkekirkens kirkebygninger og kirkegårde) concerns the many elements under the Evangelical Lutheran Church in Denmark. For this project it is the rules regarding the minimum protection period of 20 years for a grave as well as the restrictions of who has the right to be buried in a specific cemetery that is relevant. (Ministry for Gender Equality and Ecclesiastical Affairs 2009)

Accessibility of Information

It is no simple task to find the location of a deceased person's grave. All people who are buried must be registered, and for the Evangelical Lutheran Church this is done in the parish register and for sanctioned religious communities in a similar register these registers are archived in The State Archives. The Public Archives Act (Arkivloven) concerns the accessibility of data from public archives. This law is relevant as it prescribes that the burial registers are not accessible until the entries of the deceased are at least 10 years old (Ministry of Culture 2007, § 28 subsection 2)**.** Even though they are accessible after 10 years, it is necessary to know both the name and the parish of deceased person to find out what cemetery they were buried at. If the deceased has been buried after 2003, the data for the burial registers is kept in digital form and is according to the law accessible after 10 years in a digital burial register, which makes it possible to perform advanced database queries. However, this complicates the process of accessing the data, since data in digital form is regulated through The Personal Data Act (Persondataloven), and the Danish National Archives could when asked not confirm that it would be possible to access data from this burial register. (Danish National Archives 2013)

In the task of finding a person's grave, The Central Civil Register can also be helpful. This register keeps many different types of information of the citizens in Denmark. When a person dies this is also registered and every private person or company can ask for certain information like date of death and the person's address at the time. (Ministry of the Interior and Health 2006, § 38 and § 42). The Central

Civil Register contains much more information, including what cemetery that a person is buried at. Access to this information is more restricted and as such there exists no law which directly impose authorities to reveal these information as it is the case with information about date of death and address. (Folkekirke IT 2013) (Citizen Service 2013)

Even if it is possible to find the cemetery where a person is buried, it is not guaranteed that it is possible to find the specific location of the grave. The cemeteries do not have an obligation to provide this kind of information, and therefore the cemeteries can have different positions towards providing this information. The interviews with the different cemetery managers showed that some managers only give the information if the relatives accept it and others will give the information without any further requirements.

Based on this information, it is unclear if an accessible central registry of where people are buried exists. The Central Civil Register has the information, but is inaccessible, while it is unclear if the digital burial register which definitely has the information will be accessible for queries.

Private Registration of Data Related to Graves

Would a service that gave access to information of deceased persons' graves be legal? To answer this question it is necessary to take a closer look at The Personal Data Act (Persondataloven) which concerns non private use of personal data that has been partially or completely acquired and processed digitally, or non digital processed data that have been added to a register.

It is legal to make information publically available if they are non confidential and it is done if there is a legitimate interest in them for other people. Information of a deceased person's name, date of birth and death are considered non confidential. A person's religious belief is on the other hand considered sensitive and can only be used if the registered person has given permission to publication of the information. (Ministry of Justice 2000, § 6-7)

The gravestone photo websites mentioned earlier actually had the Minister of Ecclesiastical Affairs to look into the legal situation of the services, and from the response there seemed to be no unlawful matters with the service. One of the issues was, if relatives would have a grave removed from the service. In this matter it was actual concluded that since The Personal Data Act only protects the registered person, it is not possible for relatives to control or restrict access to information regarding a grave. (Datatilsynet 2007)

2.2.2 Informal Rules: The Ethical Question

The last section showed that there is no legal problem with publication of data regarding graves. On the other hand, the interviews with the cemetery managers showed that there is reluctance towards making information freely available. Furthermore, Aarhus Municipality who has the only freely available service for finding location of graves has deliberately not advertised for this solution since they fear that it might offend some. Furthermore, if relatives of a deceased want the person's grave removed from the service they will do it, and this makes it difficult for the municipality for instance to sell or give the data to a third party since they will lose control of the data. Other countries do not share the same view. In Sweden the general rule is that when a person dies any protection of data regarding the person is repealed (Bach 2013).

Dealing with deceased is a delicate matter, and there are many feelings involved, and problems cannot strictly be solved by looking at which laws that regulate the specific area and how some services could solve a specific task. The process of choosing a grave for a relative is probably the most difficult and important of all the issues involved around graves. This decision is affected by a number of factors that has a great influence on people. This could be the lighting around the grave, the surrounding features, the view, and in general the mood of the area. Most of the interviewed cemetery managers thought that these subjective choices would be difficult to include in a geoinformatic service used to choose the location of a grave for a funeral. The worst case scenario of this could be that the relatives of the deceased would not see the gravesite until the actual funeral, and that they afterwards would regret the location of the chosen gravesite. As a result of this, the relatives might want to have the deceased reburied at a new location, which is both costly and mentally tough on the relatives. A solution to this problem could be to develop the service for people with a professional insight in the process of selecting a grave, for example funeral directors.

Based on this section it is apparent that the legal and ethical issues of acquiring the locations of buried persons are rather complex. From the research conducted, it would seem that there are no laws restricting people from collecting data of deceased themselves. This could for example be done by writing down the names on, and location of gravestones. On the other hand, the authorities in charge of the official registers are reluctant to hand out the information from their archives, even though there are not laws that specifically states that this is not allowed. Furthermore the new digital register has yet to be tested for accessibility because the 10 years rule has prevented this from being possible until 2013.

In the next section the authority structure will be explained, meaning the laws that regulate the practice for cemeteries and the relevant data records that exists in this context.

2.3 Cemetery Authorities

To identify who is in charge of cemeteries it is necessary to take a closer look at the authority structure regarding the cemeteries. In Denmark the authorities who administer the cemeteries have many different levels, where each level holds different responsibilities in regards to how the cemeteries are run. There are three main different ways cemeteries are managed. Most of the about 2,200 cemeteries in Denmark are managed within the Evangelical Lutheran Church which is the national church, but in the cities some of the cemeteries are managed by the municipality. The municipalities also have the possibility to establish religious neutral cemeteries. Religious communities outside the national church have the opportunity to establish cemeteries, and such cemeteries must be sanctioned by the Minister for Ecclesiastical Affairs. (Ministry for Gender Equality and Ecclesiastical Affairs n.d.)

All cemeteries have a cemetery manager who is in charge of the daily operations. This person is responsible for solving the day to day basis problems that may occur, so that the cemetery functions well. The cemetery manager is hired by the parochial church council for cemeteries in the Evangelical Lutheran Church, and a similar committee for other religious communities. For cemeteries managed by municipalities it is the municipal council that performs this task. These committees have the daily supervision with the cemetery to make sure that the purpose of the cemetery is fulfilled. This includes providing for the necessary number of graves and the maintenance of the cemetery. The committees have the powers to make adjustments if necessary, to fulfil these requirements. In regards to management of a cemetery, the committees' most important task is to draw up a set of regulations for the specific use of the cemetery. (Ministry of Ecclesiastical Affairs 2001)

Many of the parochial church councils who manage cemeteries are quite small, and are part of the overarching hierarchy of the deaneries and dioceses which both have a counselling and a supervision role. In the following, the deaneries and dioceses key role for the parochial church councils is explained.

Above the individual cemeteries is the deanery, which includes a number of parishes, in general from 5-30, with the rural dean as the top authority. For cemeteries within the Evangelical Lutheran Church, the sets of regulations must be approved by the deanery committee. The deanery committee has a more overarching economical and administrative supervision for the operation of the cemeteries as well as supervision with the conditions of the church and cemetery. (Ministry of Ecclesiastical Affairs 2001)

Denmark is divided in 10 dioceses which each includes about 5-15 deaneries. The diocesan authority, which includes the bishop who is the top authority at this level, task is to provide legal, economical and technical advice for the parochial church councils but have also a number of supervision tasks but these are mainly of legal nature compared to the deanery committee. In matters regarding the surroundings the cemetery and the church the diocesan authority must approve changes and can also declare objection of district plans. (Ministry of Ecclesiastical Affairs 2001) and (Ministry of Environment 2009)

For the municipalities the cemeteries are managed a little differently. The administration of the municipalities is larger than the one of the parochial church councils. Because of this, the municipalities have more control of their cemeteries, and can make more independent decisions, both economically and legally. (Kilhof 2013)

On the top level of the authority structure is the Ministry for Gender Equality and Ecclesiastical Affairs, with the Minister for Ecclesiastical Affairs, who have the final saying in principal matters and can give exemption from the rules for cemeteries and funerals in a number of cases. (Ministry of Ecclesiastical Affairs 2001)

2.4 Problem Delimitation

Having investigated the potential focus areas and target groups of the project, and the written and unwritten rules of society in regards to graves and personal data, it is time to specify the goal of the project, the target group and the purpose of the service. As it has been presented, many aspects need to be considered when developing a geoinformatic service regarding graves. The previous sections clarified that certain limitations and problems are present when developing such a service which can be divided into three categories;

- Data availability
- Legal issues
- Additional research required

In order to decide what the purpose of the service should be, the four previously identified focus areas are discussed in regard to these three categories. The four focus areas are information, navigation, reservation and administration.

Information

A service focusing on providing information can have many forms and many different uses. In general much information about persons is freely available, and data availability as such should not prove to be a problem. Furthermore there are no legal issues when it comes to distributing data that is freely available, and if the data is free and already existing, little research needs to be performed. Because of this, this focus area is relatively easy to develop a service for.

Navigation

Navigational use is a widespread need for several of the potential target groups. In order to navigate to a position, it needs to be possible to know the position. This can become an issue data wise if the cemetery administrators are not willing to share data regarding the location of graves. This would not be a problem for internal administrative use, since the user and data provider would be the same person, but it could definitely become a problem for private users. On the other hand developing services for administrative use would likely require much insight into the existing administrative system, which would be very time consuming.

Reservation

In regarding to a service used for reservation there are no laws prohibiting this, and the data needed to develop such a service is available. As explained however, the process of choosing a grave for a funeral is emotionally difficult for the relatives involved. Because of this the development of a service used to make reservations for burial plots will require a lot of research in order to come up with a design and a functionality that satisfies all involved parties in the process. As a result, it is difficult to develop a service for this focus area.

Administration

The development of a service with an administrative focus is intended for people working with cemeteries as explained previously, the data can become available since the users and the data provider are the same persons. However, identifying the specific purpose for such a service would require further research of existing administrative systems, and as such this focus area will require much time to complete.

Focus Area

The administration and reservation services have a lot of considerations to take into account before the actual use of a service is clear. Furthermore it is not certain that the reservation service will meet the desired purpose, because the process of choosing a grave is a delicate matter. Based on these evaluations, reservation and administration are not desired as the focus area of this project. This leaves the information and navigation as the possible focus areas. Figure 10 shows that personal users, sightseers and administrative users all have an interest in a navigational service, while only sightseers have a use for information.

	Private Users		Professional Users	
	Personal Users	Sightseers	Administrative Users	Cemetery related companies
Information		X		
Navigation	X	X	X	
Reservation	X			X
Administration			X	X

Figure 10 The table show the focus areas which will be considered in this project, the grey areas will not be dealt with any further.

A navigation service for administrative users would require insight in the daily routines of the administration and therefore it is not optimal to develop this service. Private user navigation could be used to solve the problem of locating a deceased relative's grave within a cemetery and help navigating towards it through the roads and paths at the cemetery. For sightseers on the other hand, the service could act as a guide to graves of interest to the public, for instance a historical person's grave or a grave where the cemetery authorities have decided that the grave or grave stone itself is worthy of preservation.

With an easily understandable purpose for navigation for both personal users and sightseers, the target group should be one of these. The difference between the two target groups is mainly whether the grave of a person is of private or public interest. This means that the design and functionality of these two services can be built almost identically. The amount of data required on the other hand is quite different. A cemetery might have 5.000 graves, where only 100 of these are of historical persons. In order to make the first service useful to the public, all of these graves would have to be included in the database. The second service however, would only require the graves of the 100 historical persons to be operational and information about the persons. Furthermore data regarding the location of a historical person's grave is easier to access and there are no problematic legislation to consider compared to common persons' graves.

The two services described are very similar, and as such the first service will not require more skill to develop, though it will require much more time to acquire and prepare the datasets needed. Based on these reflections this project will revolve around creating a navigational service for sightseers. This service can easily be joined with the sightseers' second potential need for information. Here the service could navigate the user to specific graves, while providing information about the historical persons buried at the cemetery. Because of the nature of this service, it would be natural to develop it for smartphones, as users of this platform are mobile and are able to utilize the service while being at the cemetery. Such a service could be used by both Danish and international tourists. Because of this it would make sense to make the service bilingual in both a Danish and English version. However, to save time in the development period, this project will only focus on a Danish version.

Having performed the delimitation of the project, the service that will be developed in this project is:

> *A service used to find and navigate to the graves of historical persons, provide information about them and in general allow for exploration of cemeteries.*

3 Problem Statement

How can a professional service for smartphones be developed to locate and navigate the user to the graves of historical persons? Additionally, how can the service provide information about the historical persons to the user?

The answer to this problem will be found through the following three questions:

- How can the project be managed most optimally and which types of management tools and development methodology should be involved?
- Which development platform and environment will be optimal to build such a service?
- How can the user interface of the service be designed to ensure an intuitive use?

With the problem statement defined the defining phase of the project is finished. The problem statement is the goal of the project, and the following phases will try to solve the problem stated.

DEFINING STAGE: SUMMARY

The problem statement marks the end of the first stage, the defining stage. This stage provided a lot of valuable information. The identification of potential needs and existing services showed a lack of geoinformatic services in areas of locating the specific grave of persons in Denmark. This was used to choose a suitable type of service for this project. The defining stage also provided insight as to why it might be that such services do not exist, being the informal regulations by the cemetery administration themselves. The stage further looked into the authority structure of the cemetery administration.

PLANNING STAGE

With the completion of the defining stage, it is time to start the planning stage. Through this stage the management of the project will be elaborated. This will include identifying the priorities of the project, management of time and resources, and locating risks throughout the project. Besides identifying the management tools, choices will be made for how the development will be done for this project, which methods will be used and which technology will be used to build the service with. These elements are presented through the three main sections of this stage:

- **Project Management**
  This section presents the management tools and methods that were used in order to create a plan for the management of the project period.
- **Development Theory**
  The second section explain the theories behind the technologies that would be implemented in the development of the app.
- **Usability Testing**
  The final main section of the planning stage concerns usability testing theory, and presents how the needed usability tests would be conducted.

4 Project Management

In order to successfully build a service that solves the problem statement it is necessary to manage the project properly. To do this several project management tools were implemented throughout the project period which strengthened the process of defining, planning, and developing the project at hand. The first of these tools was the project priority matrix.

4.1 Project Priority Matrix

A priority matrix is an easy way to identify how the elements of a project should be managed and prioritized depending on the nature of the project at hand. The project priority matrix revolves around three criteria; the project's time frame, the performance of the project that is being managed, and the cost during the entire process. Furthermore each of these three criteria is assessed to determine which criterion that should be constrained, accepted, or enhanced respectively. These terms are explained in the following:

"***Constrain***. *The original parameter is fixed. The project must meet the completion date, specifications and scope of the project, or budget.*

Enhance. *Given the scope of the project, which criterions should be optimized? In the case of time and cost, this usually means taking advantage of opportunities to either reduce costs or shorten the schedule. Conversely, with regard to performance, enhancing means adding value to the project.*

Accept. *For which criterion is it tolerable not to meet the original parameters? When trade-offs have to be made, it is permissible for the schedule to slip, to reduce the scope and performance of the project, or to go over budget.*"

(Larson and Gray 2011)

For this project the priority matrix is seen below in Figure 11.

	Constrain	Accept	Enhance
Time	x		
Performance		x	
Cost			x

Figure 11 The priority matrix of the project.

In this project the time was constrained. Since this project had a deadline for delivery that could not be changed or postponed, the project could not be delayed beyond this point and as such time became the constraint of the project. As a result, the performance of the final product and the cost involved in creating it, were given a lower priority.

The performance of the product was not constraining the project. This was because the answer for the problem statement was not necessary a fully working service but could be a theoretical explanation, so some functions might not be incorporated in the product because of the constrained time frame. This made performance the criterion of the project that was acceptable to decrease. If there was time and resources, a fully functioning service was the goal, but if the project was running out of time it could be accepted to lessen the functionality of the product.

Finally we have the cost of the project. In this particular project there were no major resources to consider such as a budget, besides the working hours of the members of the project group. Because of this the resources available for this project were limited and every opportunity to enhance the use of these resources should be seized.

4.2 Time Frame

As explained in the preface the project period had been set to start the 1st of February and end the 13th of June, in other words the project period consisted of four and a half months. This period was been broken down to four sub periods; researching, development, writing, and correcting. This was done in order to ensure that the various parts of the project had the required time needed for completion distributed.

- Approximately two weeks were assigned to initial researching, which included exploring the existing market for similar products, collecting information regarding programming, and collecting data.
- The development period was the longest, consisting of approximately two months, which was used for developing the service, setting up databases and doing initial tests.
- The following one and a half months were used to write the project itself.
- The final three weeks were used to finalize the project. This included correcting the report, optimizing the service, and in general taking care of shortcomings through the project.

The specific period and the start and finish dates of the periods can be seen in Figure 12.

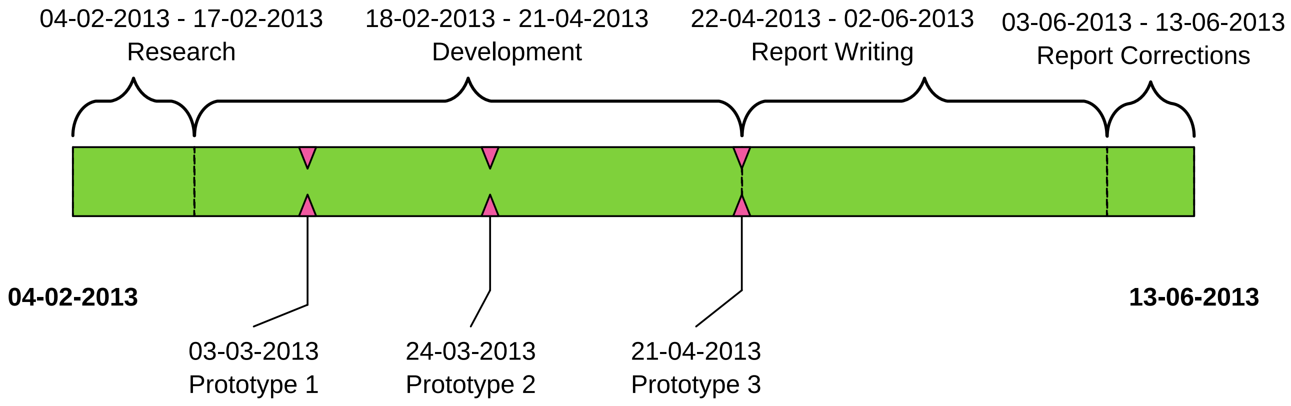


Figure 12 Project timeline and milestones.

4.3 Performance

The goal of the project was to develop a service with professional prospects, but since both time and resources available were restrained, it could be difficult to obtain this, and as a result it could be necessary to lessen the functionality of the service.

For the service planned, each function that the project group wanted to implement was identified and classified so that it was possible to determine if a specific function was fundamental for the development of the service, or if it was a specialized function that the basic service could be built without. To do this the functions of the service were divided into three categories; Framework, Basic functions, and Advanced functions. Furthermore a category called refinement was added that did not contain actual functions, but rather improvements to the service as a whole. The breakdown of the desired functions can be seen in Figure 13.

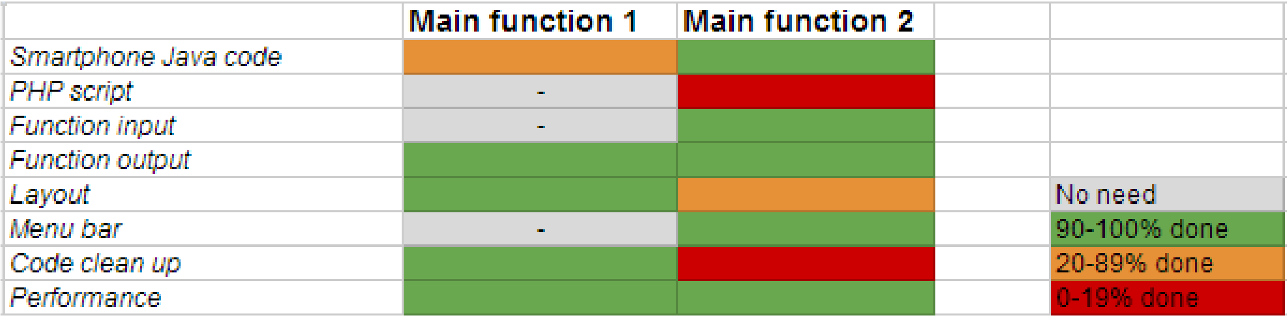
Development categories
Framework
- The least possible functionality; a map with some graves
Basic Functions
- Function to search for a grave
Advanced Functions
- All additional functionality like navigation and information functionality
Refinement
- The necessary functionality for the service to be a product ready to be released for the public

Figure 13 The four functionality categories that the development will be structured around.

By categorizing the individual functions of the service in this manner, it was possible to identify which functions that it were acceptable not to develop if time and resources ran short. For this project, the service could be developed without some or even any of the functionality contained within the "Advanced Functions" and "Refinement" categories. The functionality within the "Framework" and "Basic Functions" on the other hand should be implemented in order for the service to be considered working.

4.4 Resource Management

The purpose of this project was not only to simply develop a service, but also to learn project management from handling the whole development process. To do this the development and more importantly the implementation process was planned to be registered very closely to optimise the use of resources. During the development phase an overview of the status of the implantation was kept up to date every week in tables as shown in Figure 14. In this table, each main function was divided into eight sub classes where each sub class played a distinct role in the implementation. The implementation of these sub classes were giving three different ratings of red, orange and green, with red being very little implemented and green being almost completely implemented. Furthermore, if any bugs were found they would also be listed in this table.



	Main function 1	Main function 2		
Smartphone Java code	🟧	🟩		
PHP script	-	🟥		
Function input	-	🟩		
Function output	🟩	🟩		
Layout	🟩	🟧		No need
Menu bar	-	🟩		🟩 90-100% done
Code clean up	🟩	🟥		🟧 20-89% done
Performance	🟩	🟩		🟥 0-19% done

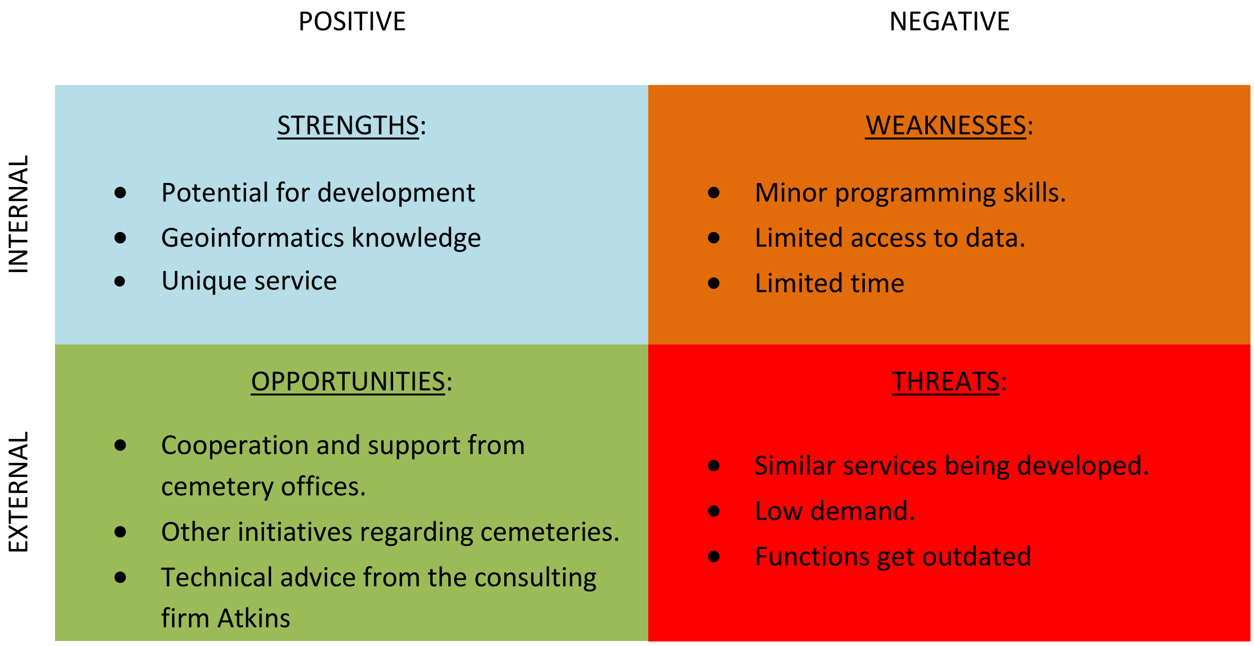
Figure 14 Example of a weekly status table. The actual tables can be seen in Appendix B.

Based on these tables it was possible to quantify the implementation process. The sub classes in the week tables were not equally difficult to implement and some of the functionalities did not have all the sub classes. Because of this implementation points were used. Here the main functions could get two to three points. One point was giving if the code responsible for actions performed internally of the smartphone was finished, one if the layout was finished and the last point was giving if the server side scripts were finished, since not all main functions used scripts, they could only receive two points. The implementation points were graphed along with the time measured in weeks, and was used to make development decisions on whether or not to continue with the chosen implementation technology during the development period.

Furthermore the work that had been made the particular day was documented every day. These notes can be seen in the project diary, in Appendix C. With these notes it was possible in detail to see the progress or lack hereof with the development.

4.1 SWOT Analysis

When planning a project it is important to identify the strengths, weaknesses, opportunities, and threats that are present to help finding the optimal strategy for the project. To do this in this project, a SWOT analysis was conducted. The four categories of a SWOT analysis represent whether a factor is internal or external, as well as positive or negative. By performing a SWOT analysis it is possible to determine which objectives in a project that will be most likely to be completed, so if there are too many major challenges that suggest that the objective of the project will be difficult to reach, the overall goal should be adjusted. All of the identified factors can be seen in Figure 15.



	POSITIVE	NEGATIVE
INTERNAL	STRENGTHS: • Potential for development • Geoinformatics knowledge • Unique service	WEAKNESSES: • Minor programming skills. • Limited access to data. • Limited time
EXTERNAL	OPPORTUNITIES: • Cooperation and support from cemetery offices. • Other initiatives regarding cemeteries. • Technical advice from the consulting firm Atkins	THREATS: • Similar services being developed. • Low demand. • Functions get outdated

Figure 15 The SWOT matrix with identified strengths, weaknesses, opportunities and threats.

Strengths

As described in the defining stage, there was a lot of potential in this project, as there were many similar demands that could be answered in solutions very similar to the product of this project. This meant that there were many ways to go, and that the work presented in this project very easily could have been adjusted to match another demand. The project group's strong knowledge about geoinformatics could be used in both the practical development as well as in theoretical understanding of how the service should function. Existing similar services were very limited which was another strength since this project's product would be the first service of its kind on the market.

Weaknesses
With the minor programming skills that the members of the project group possessed, this project had a significant weakness that needed to be handled. Furthermore the project group had limited access to the data that was needed in order to build the wanted service. Finally the project had a strict time limit which heavily influenced the work that could be done.

Opportunities
The cemetery offices in Denmark have a lot of resources that could prove beneficial for this project. Because of this it could prove very useful to cooperate with these offices, as they potentially could provide support in a number of cases. Another opportunity was the many initiatives regarding cemetery exploration. It could be possible to use the resources here, to improve the overall performance of the service. The project group furthermore had established contact with the consulting firm Atkins which could be useful in case of technical problems.

Threats
With the development of a geoinformatic service, there were a number of negative external factors that the project group could not control. One of these was that a similar service might be created simultaneously with the project groups, where more time, money and skill would be involved. Another threat that could be the aftermath of the first would be that the service developed would not generate any users. It was also possible that the functions used in the service would get outdated during the project period.

Before making the final strategy for the project it was necessary to figure out how to counter the weaknesses and threats to the project. To do this a Risk Analysis was performed where the strengths and opportunities of the project were used to uplift the potential risks.

4.2 RISK ANALYSIS

It is important to be aware of potential risks involved with a project and how to counter the risks if they become reality. To do this in this project, a Risk Severity Matrix was used to visualize the severity of the weaknesses and threats identified in the SWOT analysis. The risks were weighted depending on the likelihood that they were going to happen, and the impact that the risks would have on the project. The severity of a risk was symbolized by the colour of the box it was placed within, where red was a major risk, yellow was a moderate risk, and green was a minor risk. In this project five risks were derived from the SWOT analysis and placed into the previously mentioned matrix. This can be seen in Figure 16.

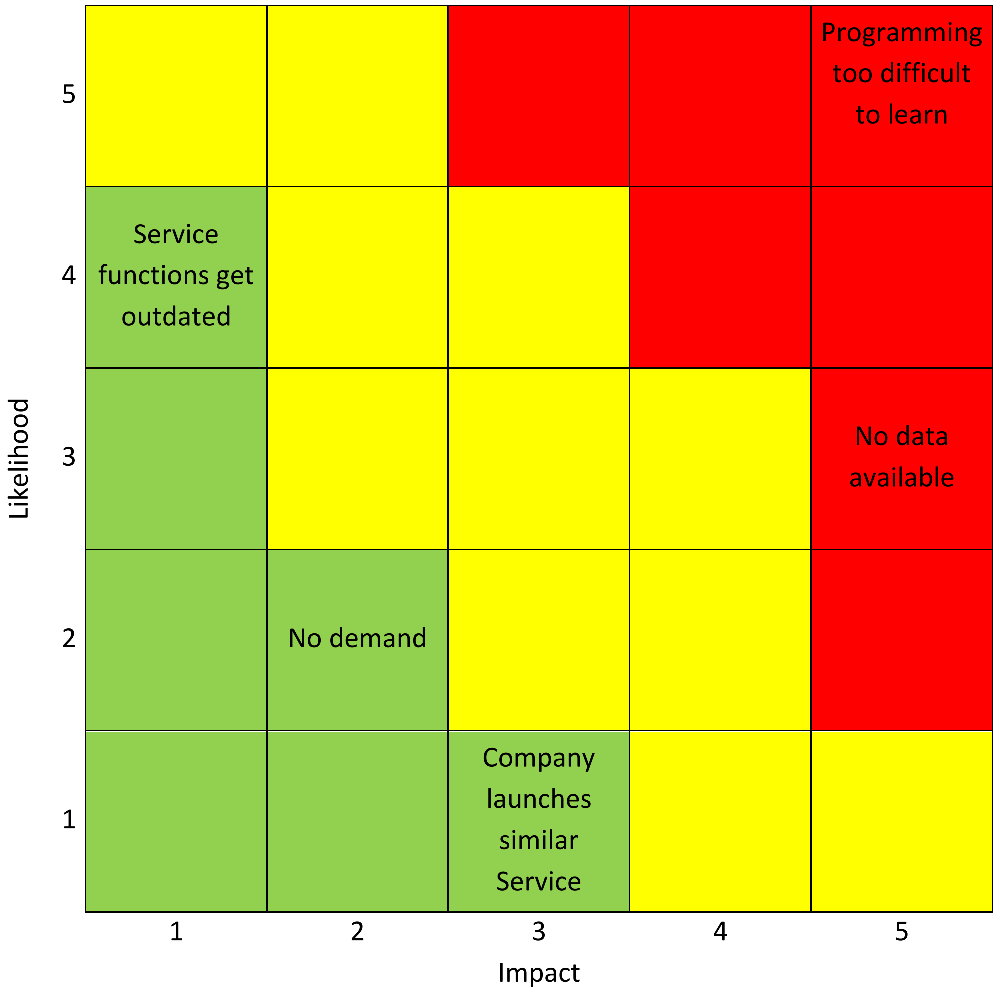

Likelihood
5
4
3
2
1
Programming too difficult to learn
Service functions get outdated
No data available
No demand
Company launches similar Service
1
2
3
4
5
Impact

Figure 16 The identified risks put into risk matrix. (Larson and Gray 2011)

As seen in the matrix, this project faced three minor risks and two major risks. These risks and the plan developed to counter them are described below:

- **Service functions get outdated.** The programming environment is constantly evolving. This means that a geoinformatic service that is developed today might be outdated tomorrow. This is of course unacceptable for a commercial product, but since the goal of this project was to build a working service, the impact of this risk was very little and would not be considered further.
- **No data available.** A major risk was, if it was not possible to acquire cemetery data, since the service then would have no data to rely on, which is unacceptable for a commercial product. For this project with no commercial interests, it would instead be necessary to build a fictional dataset which would not be a problem for the learning aspect of the project, but it would make it difficult to make user tests of the service. To counter this problem the cemetery offices would be contacted in order to gather the datasets necessary.
- **Company launches similar Service.** A minor risk to the project was if a professional company launched an app that had the same functionality as the one for this project. However, having investigated the market beforehand, this did not seem to be the case. If this risk was to happen, again, it would not affect the learning aspect of the project, but the relevancy of the entire project would be heavily decreased.

- **Programming too difficult to learn.** As no one in the project team had ever worked with programming geoinformatic services for smartphones before, it was a major risk that it was not possible to learn the needed skills to complete a working product, such as this one within the project period. Even if Atkins could provide assistance, the challenges of working with a new technology might still prove to be too much. To assess if an acceptable progress was made, the implementation points mentioned in resource management were used to make this assessment. If the development of a working service was abandoned, the project would have to be changed into having a much more theoretical angle to counter the shortcomings of the programming itself. Here the project team's knowledge of geoinformatics would be invaluable as this would be the major focus instead.
- **No demand.** A minor risk for the project would be if there is no demand for the type of service developed. For a commercial product this is crucial, but as stated before, the goal of this project was not to launch a working service onto the market, but rather to document the work being done in the creation. This risk was therefore only limited to the relevancy of the project, but as the conducted research had indicated, this risk was small.

The SWOT analysis identified a number of key characteristics for this project and the project group. Based on these characteristics the overall strategy for this project was to develop a service with emphasis on the geoinformatics aspects and external help from Atkins to overrule any limitation in programming skills. The weakness of limited time available for development could be neutralised by emphasising on theoretical knowledge. Since there were no economical interests in this project, the external threats would not play a major role in the further development.

The SWOT analysis provided strategic possibilities, and the risk analysis provided the project group with possible ways of handling the risks involved in this project. The combination of the SWOT and risk analysis defined the nature of the project which was:

> *A time constrained high risk project, with no demands on certain properties for the final product and with the overarching purpose of maximising the use of technologies and theories with professional relevance to geoinformatic services.*

This description will be used in the next section to find an appropriate development methodology for the project.

4.3 Development Methodology

For computer program development projects, it can be difficult to structure and manage the development process. If the project group had consisted of experienced developers, the need for management would have been significant less but this was not the case. Instead, having a methodology and a model was necessary to ensure a stable development framework to engineer a usable product. (Rogers, Sharp and Preece 2002, p. 183)

Within software development several different models and methods for managing a project exists. To use a very coarse generalisation there exist the waterfall and the agile approach, and in practice most of the time the used method will be a combination of these methods at different levels. These two methods were developed for different types of projects; the waterfall method for large projects in need of good documentation, and the agile method for faster client interaction with working code all the time. The methods are basically respectively document and code driven. (Boehm and Turner 2003) and (B. W. Boehm 1988)

If one takes a closer look at the fundamental nature of this project it is; a high risk project with very limited time to both learn the programming skills and to perform thorough research of the users' specific needs for the functionality of the service. Using the waterfall approach, the project might end having used all the time available on research and planning with no time for programming, or just not enough time to do all of these elements thoroughly. At the same time, the agile approach does not includes sufficient ways of handling the risk, of not reaching the end goal if one only looked at the next function to implement with less overall strategy.

To overcome some of these shortcomings most modern development strategies are based on development cycles. Introducing cycles makes it easier to manage the development phase and also faster to introduce new functions to the program and test these. Many different cycle based models exist, where each of them have certain advantages. (Rogers, Sharp and Preece 2002, p. 183)

One of these cycle models is the spiral. This model is risk driven with emphasis on prototypes. The spiral model is iterative where each iteration includes phases of researching, designing, implementing, risk analysing and prototype testing. This model was designed to be used for high risk, large scale governmental software development projects, and as such many of the aspects of the model do not apply to this project, but the overarching idea of a life cycle prototype approach fits well with this high risk project. (B. W. Boehm 1988) and (Rogers, Sharp and Preece 2002, p. 188)

The spiral model was originally designed to use the waterfall method for each iteration, and as it can be seen in Figure 17, the spiral model developed for this particular project indicated that this also was the case for this project. Each iteration included the shown elements but the design and implementation phase would not be fully restricted in two independent phases. Before the implementation phase there would be a design phase but if something under the implementation phase indicates that it would be good to change some functions, they would be changed. The specific functionality would also be implemented and tested one at a time, and in many cases a fully working service would exist after the individual implementation. This meant that while implementing, little concern would be given to the following functionalities which would be implemented later on, so in this sense it was more of an agile approach that had been chosen.

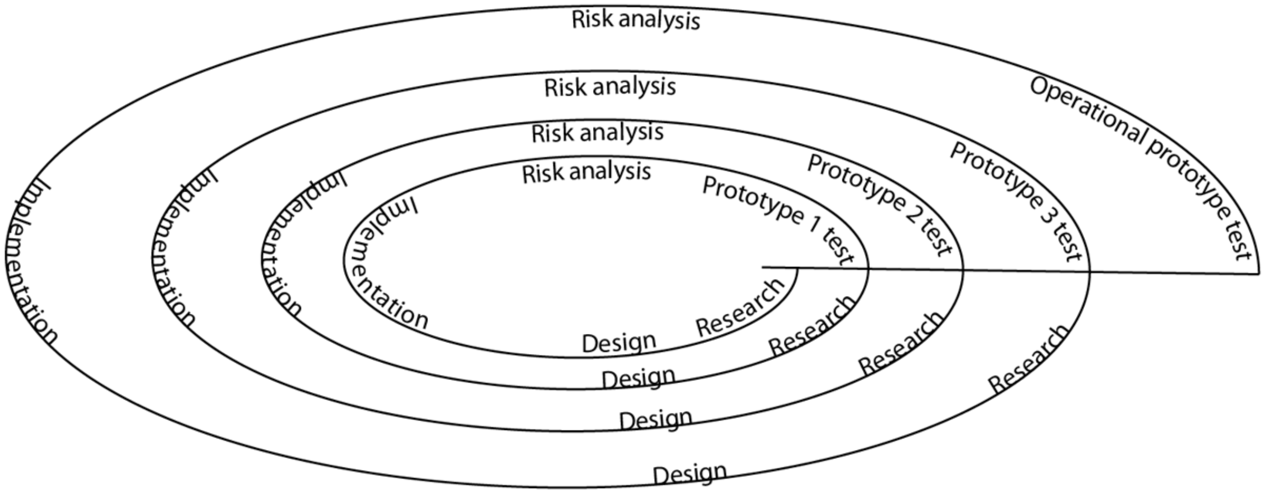


Figure 17 The prototype development model which was developed specifically for this project.

Figure 17 illustrates the idea of having a prototype in the end of each iteration, and each iteration will bring the prototype closer to the final product.

The first prototype test would only be done internally in the project group, to test and make a risk analysis for the choice of implementation platform by looking at, if the risk of failure was too high and if a different platform should be used instead, or if it would be acceptable to continue. The following prototype test would be performed to see if the functionality of the service was working as expected, or if adjustments would be needed. Since there was not enough time in the beginning of this project to do thorough research, the prototype tests were also planned to continuously gather information through the development phase. This information would concern how the user reacted and thought, and how the service should respond to this, which would be used to include new features in the original four functionality categories Each iteration or cycle would includes five overall phases:

- **Research.** Each new iteration started with a research phase where information of what functions that were necessary for this prototype.
- **Design.** The research phase provided the information necessary for the design phase where it was decided how the actual functions would work and how the layout should look.
- **Implementation.** This phase would be where all the functionality was programmed.
- **Risk analysis.** In the end of the iteration a prototype would be finished and a risk analysis assessed of whether or not the goal was reached with the specific prototype or if any changes to the project would be necessary.
- **Prototype test.** This test would have several purposes. Its main purpose was to find technical and logical errors, and to come up with ideas for improvements in the shape of new functionalities which could be used in the research phase in the next iteration.

This project was structured around four iterations and prototypes. The prototypes and the iterations were divided between the four overall categories of the functionality of the service presented in the performance section. For each category a prototype was associated as shown here:

- Framework, prototype 1
- Basic, prototype 2
- Advanced, prototype 3
- Refinement, operational prototype

At the same time, only three of the four iterations were planned to be executed, due to the lack of time in the project period. This meant that the fourth iteration focusing on the refinement of the product would not be begun.

With the weaknesses and threats identified, and strategies developed to counter these based on the strengths and opportunities of the project, the initial tools that are used to manage the project have been specified. It is now time to examine the methods and technologies which the development will be shaped around.

5 DEVELOPMENT THEORY

It is a difficult task to build a service that matches the demands specified in the problem statement. In order to successfully develop this service, it is necessary to investigate the technologies and theories that lie behind the desired functions. Based on this research, it is possible to decide what software, technology, algorithms, and methods that should be chosen and implemented for this specific service, in order to achieve the most professional outcome for the project. These elements will be discussed and presented in the following sections.

5.1 DEVELOPMENT PLATFORMS AND ENVIRONMENTS

When developing a service it is necessary to make a choice of which platforms it should support, and what technology that should be used to build it. In this section the different possibilities will be presented and a choice for the future service will be made.

To get inspiration for different technologies that the service could be built using, the project group met with Morten Fuglsang PhD student in Geoinformatics at Aalborg University. He proposed a lightweight technical solution with no or a very thin backend based on a website solution with local geographical data on the smartphone. The development technology was also discussed with technical consultants from Atkins, and they proposed two solutions; one using software to convert a website into an app, or one developing a native app for which they would be willing to assist if this was needed. (Fuglsang 2013) and (Jensen and Gjedde-Simonsen 2013)

To choose a platform and environment for the development of the service, it is necessary to take a look at what the different solutions imply. Different operating systems for which it is possible to develop apps exist. Furthermore, it is possible to just build the service as a website and it is also possible to develop a service using a mix of the proposed methods. This section will explain the different possibilities and choose the used development technology for the project.

5.1.1 DEVELOPMENT PLATFORM

There exist several operating systems (OS) for smartphones. If the purpose is to have the service available to as many users as possible it should work on all platforms. Developing native apps for all the different OSs would be very time consuming and unrealistic for this short project.

Since the project group is novices in app programming, and programming in general, it is necessary to choose a platform where it will be easy to receive technical support. Platforms with a large market share are more likely to have large user communities where technical support is easy to acquire. The two operating systems iOS and Android have the largest market share, see Figure 18, but the market share for sold smartphones is four times as high for Android than for iOS. This indicates that it is likely much easier to find information and guides about Android development than for iOS

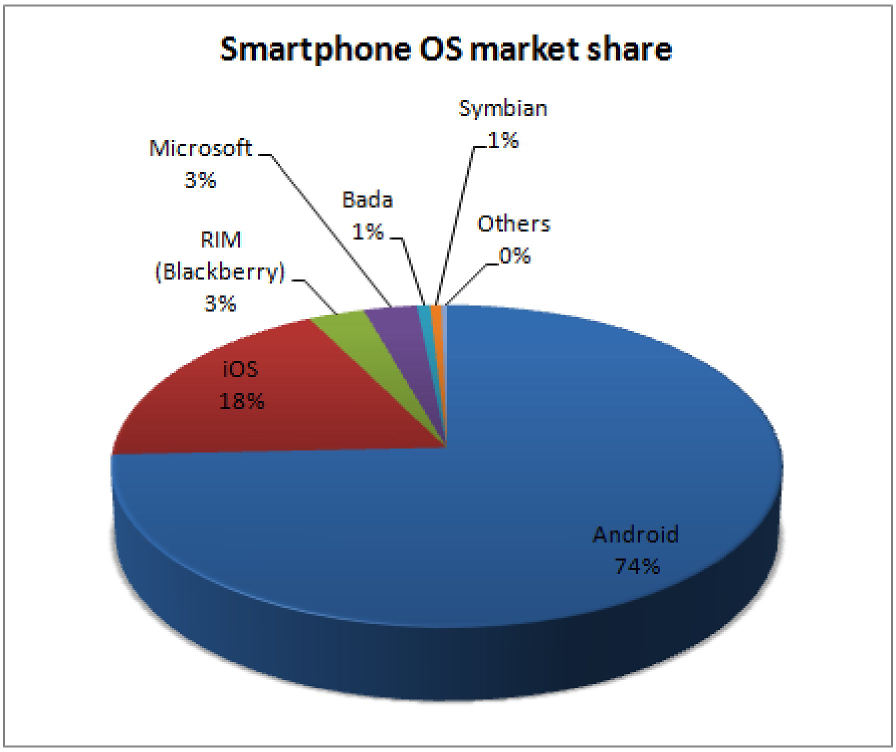


Figure 18 Smartphone OS market share for sold devices 1st quarter 2013. (gartner.com)

For both OSs there exist good development environments, but more difficulties are involved with starting developing for iOS, since it requires a computer capable of running Apple's own operating system MAC OS X. When developing for Android there is no such restriction, since Google's Android Developer Tools (ADT) is available for Windows, MAC OS X and Linux. Because of this, development of the service will concentrate on using Android.

5.1.2 DEVELOPMENT TECHNOLOGY AND ENVIRONMENT

When developing services for smartphones, there are a lot of options when it comes to the choice of technology and environment that the app will be built for. A service could both be developed as a native app or a mobile optimised webpage, or a combination of these. Each one has its own strengths and weaknesses, and each one can be very useful in the right situation. In order to choose the right development technology and environment for this project, the available options will be described in the following.

Integrated Development Environment (IDE)

One of the easiest ways to start developing apps is to use an IDE. An IDE is a piece of software that contains a number of tools that simplifies the process of building an app and provides a user friendly interface to for instance Google's ADT.

There are many different IDEs on the market, both licensed and free, which makes choosing the right one a difficult task. One of the most widely acknowledged IDEs is the free software Eclipse. Eclipse supports a wide variety of programming languages, including Java which is the main component of Android applications. Furthermore Eclipse comes with a smartphone emulator, which makes it easy to test new apps on the different versions of the android operating system. Like many other IDEs, Eclipse keeps trace of the stored files and resources used within the app, which means that the software can easily detect inconsistencies and bugs within the app. Furthermore the IDE hosts a library containing the many functions that can be called in an app which makes the IDE able to come up with a list of possible commands that matches what is being written in script. These tools of the IDE help developers getting an advantage, when programming in a new language or an entirely new environment. Because of this, using an IDE is a good option for new developers to get familiar with building apps.

Additionally, when developing an app in an IDE, it is often required to implement a number of elements from specific software packages or frameworks. To do this, a Software Development Kit, SDK, can be installed within the IDE. This is for example necessary in order to include the functionality for Android applications.

(Eclipse Foundation 2013)

Text Editors

One of the most basic ways to programme is by using a text editor together with Google's ADT, and many editors are specifically designed for programming with functions such as; displaying line number, registering tags, etc. This simplicity is favoured by some developers, but in order to programme successfully in a text editor, the developer needs to know the environment for which he is programming. Since application programming is not like writing small scripts, but uses a lot of specific commands and underlying structures, it may prove very difficult for developers with no prior experience to start programming an app in a text editor alone. Instead other options should be investigated.

Mobile Optimised Webpage

Instead of developing a native app it would also be possible to develop a web-based service where no software should be installed on the smartphone. This solution requires less specific knowledge about the app platform, and instead standard webpage coding languages and scripts such as HTML5, JavaScript and PHP would be sufficiently to have a working service. The big advantage here is that this solution would work on multiple platforms. On the other hand, the website solution may require more processing power from the smartphone, and with a native app it is possible to have greater control of the device than with a website.

As with native app developing, it is also possible among website developing to use sophisticated web editor software that helps with developing the code instead of using a plain text editor

Mobile Development Framework

A combination of the former two solutions would be to use a Mobile Development Framework. This type of software makes it possible to design and build apps using common webpage coding and then converting the code into an app package instead of using device-specific languages to build the app from scratch. This essentially means that you can build a web-based GIS service, and then afterwards convert it into an app format. In other words, it makes it possible for developers with little app programming knowledge, to create an app using existing web-based code. As a result, this software makes it possible to develop apps for several different operating systems by only writing one set of codes, since the software can convert the same web-based code to both android and apple for example. Because of the way apps are created using this type of software, the apps are hybrids between web-based and native technology. This results in some problems where certain technologies cause the app to run slowly, because the code is not completely native to the individual devices. This has previously resulted in cases where for example apps created with the Mobile Development Framework *PhoneGap* have been rejected by Apple because they are either too slow, or do not feel native in their operating system.

(Trice 2012)

Mockup Software

If designing and building an app using an IDE or a Mobile Development Framework proves too difficult, another option could be to use a Mockup Software such as *UI Blueprints* to "build" the app. A program built in a Mockup Software is a program that on the surface looks like it is working the way it is supposed to, but in reality does not have any underlying mechanics. The way this type of software works, is by displaying specific images when a certain area of the display is clicked. For instance you could have an app with a start menu consisting of a picture displaying two "buttons", one called *add*, and one called *search*. When the display is clicked where a "button" is located, the software opens a new picture that matches the name of the "button". This gives the illusion that you are operating a functional app, when in reality you are only looking at and clicking images without any underlying data or code being activated. This type of software is usually used to quickly create a hollow app that can be used to display the intended flow of an actual app that is being developed. Because of this, the software can easily be used to illustrate how the app of this project is intended to work if programming the actual app is not possible. (Mashable.com 2012)

5.1.3 CHOICE OF PLATFORM AND DEVELOPMENT TECHNOLOGY

As presented there are many ways of developing a service for smartphones. The functionality and control of the smartphone varies from complete control using a native app, to only being a simulation of a service using mockup software. However, not only the functionality, but also the performance and development time for the different methods are important, in Figure 19 a comparison of these properties is given.

	Service performance (weight 0,3)	Service functionality (weight 0,6)	Development time efficiency (weight 0,1)	Total
App, IDE	10	10	2	9,2
App, text editor	10	10	1	9,1
Webpage, web editor	8	7	8	7,4
Webpage, text editor	8	7	7	7,3
Mobile Development Framework	7	8	5	7,4
Mockup software	10	1	10	4,6

Figure 19 Comparison of the different development methods, where each property is rated from 1-10, with 10 being the highest, and the total score after weighting.

The total rates of the different development methods were very similar except for the Mockup software, but the individual properties were very differently. Based on the different types of development environments, there were many ways in which to build a service for this project. Since the project group had no prior experience in building apps, it could be a good option to start working with a Mobile Development Framework. Since it in this project had been decided to spend two months on the developing period, time was not the essential factor. Further the goal with this project was to develop a service with professional prospects. Because of this the scores of the total rates were weighted 0,6 for functionality, 0,3 for performance and 0,1 for development time efficiency. Looking at the total ratings using an IDE for native app development was the best choice and therefore it was decided to develop a native app. The platform chosen was Android since this OS gave the best possibilities of finding help for developing.

The risk of failure using unknown methods was very high. In the project management section it was described how this risk would be minimised using a prototype driven spiral model for the development. To make this assessment the implementation points from the resource management section would be used. For app development it was expected that the ratio between time and points would be exponential since it would take time to get familiar with a new technology. If the ratio signalled that it was not possible to learn native Android application development, the second choice of method was to build a website since the project group was more familiar with this technology.

5.2 WEBGIS

When a large amount of geographical data has to be made available to a large number of people, the obvious choice is to do it using a WebGIS service. Depending on the nature of the data, the spatial data infrastructure (SDI) can be modified, and even though the SDI and general architecture behind a WebGIS service can vary greatly, there are some basic elements that all such services should include in order to function properly. As seen in Figure 20, these elements are a database to contain data, a server that hosts the service itself and the client platform which is the users' access point to the service. The choices for these basic elements, as well as more advanced ones needed for this project will be explained in the following chapters.

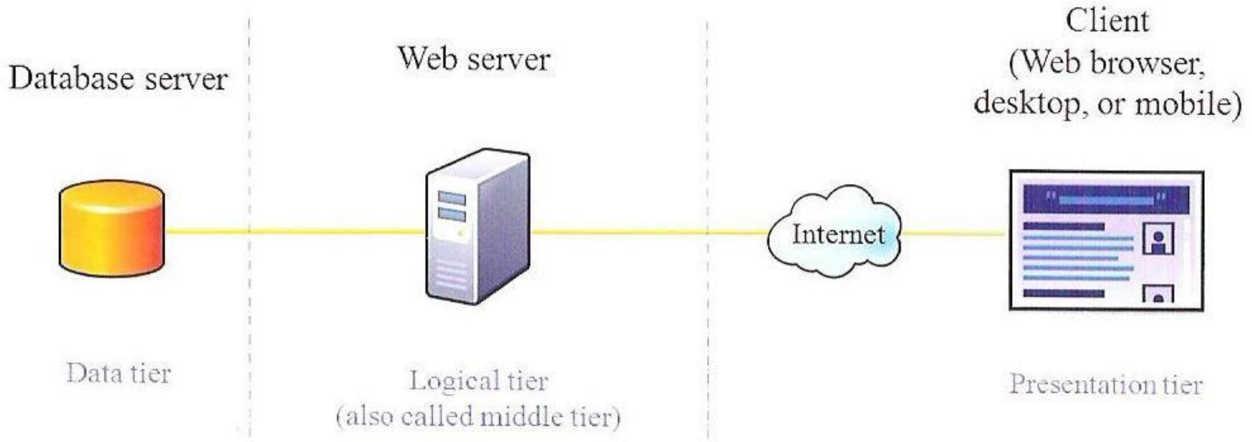


Figure 20 WebGIS principles. (Fu and Sun 2011)

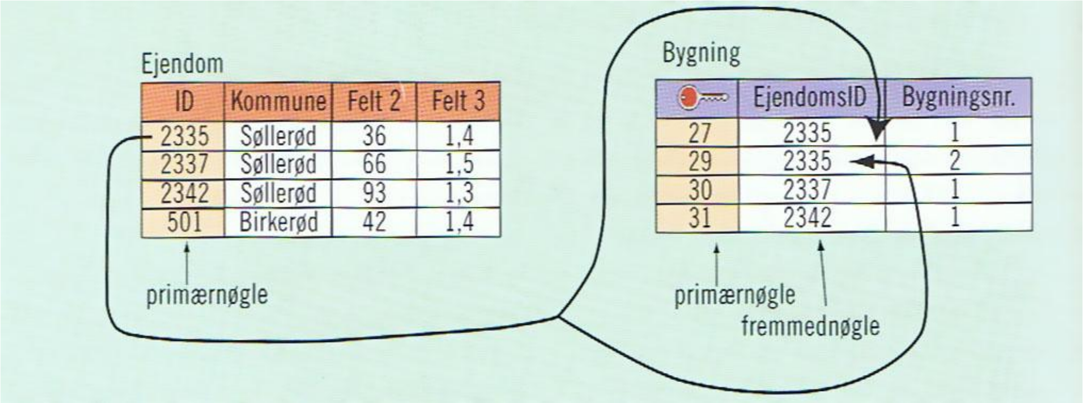
When working with Web services, the underlying architecture can generally be divided into two parts, the front and back end. The front end is the part of the service that the user can see, typically through the client. The back end on the other hand is usually hidden from the user, and contains all of the databases and servers that are used in order to solve the request made by the user at the front end. Depending on whether the majority of the processing of a request is done at the front end or the back end, a GIS service is referred to as being thin or thick clients. Thin clients are users who rely on the servers of the service to do the work, while thick clients access data from the servers, and then do the work on the device being used themselves, often through advanced installed software such as ArcGIS. Because of this most thick clients are able to function without an internet connection if the data has already been downloaded, since all of the processing is done through the device the client is located on.

When building a WebGIS service for smartphones, it needs to be done as a thin client. This is due to the simple fact, that smartphones do not have the memory or CPU to process the large amounts of data that a WebGIS service such as the one being designed, needs. This means that the WebGIS for this project has to be constructed in a way where the service on the smartphone requests the servers to access data from the databases, perform the needed calculations, and then send the result to the smartphone itself.

(Fu and Sun 2011)

5.2.1 DATABASES

A key element in any GIS system is the databases. The databases contain the collected and structured data in tables or even plain text files, and the databases themselves can be organized using a *Database Management System,* DBMS. A DBMS makes it possible to make queries or requests in a database for specific data, such as all entries starting with the letter h. These requests are performed using a *Structured Query Language*, SQL which can be implemented in server scripts. A more advanced version of DBMSs is the Relational DBMS, RDMBS. Here the data is structured in tables with several columns that each represents specific information about an entry in the database. These tables and columns serve as the relation between entries where data from two different tables can be joined using keys, Figure 21.



Ejendom

ID	Kommune	Felt 2	Felt 3
2335	Søllerød	36	1,4
2337	Søllerød	66	1,5
2342	Søllerød	93	1,3
501	Birkerød	42	1,4

Bygning

🔑	EjendomsID	Bygningsnr.
27	2335	1
29	2335	2
30	2337	1
31	2342	1

primærnøgle
primærnøgle
fremmednøgle

Figure 21 Examples of tables in a database.

This type of Management System allows the developer to organize and regroup individual databases, which otherwise can become a difficult task when dealing with several hundred databases each containing a thousand entries for instance. With the DBMS it becomes possible to find and access specific data in a specific database, which can be used to edit, extract or even delete entries.

(Balstrøm, Jacobi and Bodum 2006)

A variety of databases are the spatial databases. These databases can in addition to basic information about the contained data such as values and metadata, also store spatial data about an entry. This could be the geometric shape of an entry, whether it is a point, a line, or a polygon or the geometry or projection that the dataset uses. Because of their ability to store spatial data, spatial databases offer some very useful tools that can enhance the performance of a GIS system tremendously. These systems are also called Object-Relational DBMSs, ORDBMS. Since this project will need a lot of spatial data to function, for instance the geometry and location of graves at cemeteries, a spatial database needs to be configured.

(Shekhar og Chawla 2003)

One of the most common ORDBMSs is PostgreSQL together with its spatial extension PostGIS. This open source database was developed by the creators of ORDBMSs, and has a large group of users. The reason for PostgreSQL's popularity is that it despite being open source, easily can manage databases with terabytes of data, is easy to use, offers support, and has many built in functions ready to use as well as the possibility to implement additional functions. One of the functions that can be implemented is a Network Modelling tool. Because of these advantages of PostgreSQL, this management system would be used as the foundation for the databases of this project.

(Obe and Hsu 2011)

5.2.2 WEB MAP SERVICE

A basic element of the desired app is to display a map with graves, a common way to generate such a map is to use a Web Map Service, WMS. Here the full map is located on a server, and when a user enters a site with a WMS, a request is sent to the server. This request contains a bounding box which is specified by the user, usually by the actual screen view of the service, as well as information as to which layers that the user wants to see. The server then sends back an image that matches the request which is then used as the background map. A drawback of the WMS is that it can take time to receive the image because a new request has to be sent and a new image has to be received every time the screen view changes. To counter this, tile cashing can be implemented. Tile cashing concerns sending multiple requests to the server at one time and receiving the corresponding images regarding the possible future screen views. This way, a new image is already available when the user for example zooms in, or pans the screen view slightly north, because the request has already been sent.

(OGC 2012)

To host the WMS for this project, GeoServer has been chosen. GeoServer is an open source software that is the implementation reference of the Open Geospatial Consortium (OGC) regarding Web Feature Services, and is highly regarded concerning generating WMSs. This together with the fact that GeoServer is very easy to use makes it the obvious choice for hosting the WMS for this projects service.

(GeoServer n.d.)

5.2.3 *SERVERS*

Web Servers are an essential part of a WebGIS service. Technically, a server is a computer that contains vital information for the service, such as data or scripts. This computer can be anything from a local computer, an overseas located computer, or a server service such as the *XAMPP* software. (Frantzell 2004) In this project a virtual private server (VPS) in the USA has been leased, here the hosting company have several virtual servers installed on one physical server, which is a lot cheaper to lease than leasing a whole physical server. The VPS will contain the databases and WMSs that is needed for the service to function, but also all of the PHP scripts that will perform the necessary calls to the databases. The processing power and storage capacity of the leased server can vary greatly depending on the price one is willing to pay. Since this project is a university project with no economic agenda, the cheapest server available has been chosen, even though this could prove to slow the final product.

5.2.4 *EXTERNAL WEB SERVICES*

When developing a WebGIS service, it can often save time to implement External Web Services to your own service. An example of such a service is Google Maps. Through the Google Maps Application Programming Interface, API, it is possible to implement a Google Map to the service, and thereby generating a fast map for reference. Another example is Wikipedia.com, which also has an API. This API makes it possible to parse Wikipedia articles directly to a website, which can be used to gather information about buried historical persons easily.

5.2.5 *COMMUNICATION METHOD OF THE WEBGIS SYSTEM COMPONENTS*

For a WebGIS to successfully interact with each of the individual elements, a strong internal and external communication system consisting of requests and responses needs to be developed. This can be done in a number of different languages and scripts, the different choices will be described in the following and the used ones for the development will be chosen.

External Communication

Communication between the server, smartphone and the used services from Google and Wikipedia is referred to as external communication.

A common way for a web client and a web server to communicate is through the Hypertext Transfer Protocol, also known as HTTP. This protocol includes a number of methods for the client to make requests to the server. In this project, the POST method will be used for exchanging data between client and server.

External Communication Objects

Many different ways of exchanging information exist, and three of the most common methods are presented in the following.

The simplest way is to include data as key-value pair with the POST method as shown in Figure 22. This is very easy to implement, but for large amounts of data it is not very efficient. In the example only data for one person is shown, but it is not simple to add another person. Formatting the data in a structured way can overcome this problem.

```
name=Jens Jensen&
streetAddress=Hollændervej 1&
city=Copenhagen&
postalCode=2000
```

Figure 22 Example of key-value formatted data.

Extensible Markup Language, XML, is a simple exchange format for structuring, storing and transporting data between a client and a server, see Figure 23. Because of its simple self describing structure, it is easy to understand and use, and as such, it is one of the most common exchange formats used.

```
<?xml version="1.0" encoding="UTF-8"?>
<persons>
   <person>
      <name>Jens Jensen</name>
         <address>
            <streetAddress>Hollændervej 1</streetAddress>
            <city>Copenhagen</city>
            <postalCode>2000</postalCode>
         </address>
   </person>
   <person>
   ....
   </person>
</persons>
```

Figure 23 Example of XML data.

Another exchange format is JavaScript Object Notation, JSON, Figure 24. Like with XML, the purpose of JSON is to exchange data between the client and the server, but because of the structure, the same amount of data requires less storage capacity. Because JSON is a native data type in JavaScript, this exchange format is very easy to parse when using JavaScript and is widely used for web services and products like Google Directions, and Openlayers. Because of the several advantages using JSON, this will be used

```
{"persons":
   [
      {"name": "Jens Jensen",
         "address": {
            "streetAddress": "Hollændervej 1",
            "city": "Copenhagen",
            "postalCode": 2000
         }
      }
      ...
   ]
}
```

Figure 24 Example of JSON data.

Database Communication

When communicating with databases, the Structured Query Language SQL can be used. This language is used to retrieve and manipulate data stored in a tabular form on a database. A SQL query can be made in several ways either programmatically from software like GeoServer, directly from a client using PostgreSQLs frontend/backend protocol or from server side scripts. The advantage using server side scripts is that the communication between a script on the server and to the database is much faster and processing heavy calculations like network routing can be performed by the server instead of by the client. Because of this server side scripts handling the queries will be used in this project.

(Fu and Sun 2011) and (PostgreSQL 2013)

Internal Server Communication

When a client requests a server for data a common way to handle these requests are through server side scripts. The most common, being Hypertext Preprocesser also called PHP, and Python, these scripts can be used to write files on the server that are used elsewhere in the communication process. This could for example be an XML file that is written on a server to contain information needed for service that cannot access information from the server by itself.

Both scripting languages have all the needed functionality for this project and share many functionalities. But there are also some differences. A very coarse generalisation is to classify Python as the more professional strong scripting languages while PHP just emerged to serve a need for website programmers. Because of this the user community is much larger for PHP and technical support is easier to find online. Since in term of functionality there is no advantage between the two scripting languages for this project, PHP with the larger online community will be used for writing server side scripts.

(Python 2012) and (PHP.net 2013)

5.3 NETWORK DATA MODELLING

As part of the developed service it is desired to implement direction functionality. In order to do this, it is necessary to construct a Network Data Model that can be used to calculate the optimal route for the users of the app. A Network Data Model consists of edges and nodes, where the edges represent stretches that can be travelled along, while the nodes represent junctions where you can move from one stretch to another. What that means is that it is not possible to travel between two edges just because they cross, but only when they share a node in the crossing. This is an important element in network modelling, since it makes it possible to take bridges and tunnels into account in the Network Data Model. This can be seen in Figure 25. Here, in the network to the left, it is only possible to travel from point A to D, and from point B to C even though the two edges cross but do not share a node. In the network to the right it is possible to travel in between all nodes, because a fifth node, E, has been added where the two edges meet. This makes it possible to travel from A to B or C as well, via E.

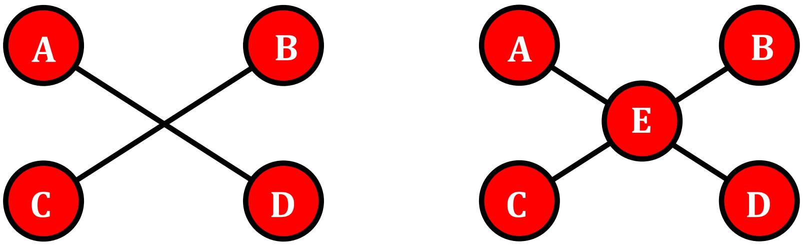


Figure 25 Examples of simple network models.

When creating a Network Data Model, it is important that the model represents the reality as closely as possible. This is of course a challenge, but if the model is supposed to be useful when it is complete, this is of high importance. For instance some roads only allow traffic in one direction. This piece of information is highly important for the network model, and should be implemented if the network is to be trusted. Otherwise the model will contain false information, and the entire structure is useless.

5.3.1 Shortest Path Algorithm

A typical problem within a network is to calculate the optimal route between two points. This problem is called the Shortest Path Problem, and can be solved through a number of algorithms such as Dijkstra's algorithm, A* algorithm or the shooting star algorithm, which are supported by pgRouting.

Dijkstra was one of the first shortest path algorithms developed, and is a widely used method for calculating the optimal route between two nodes in a graph. This algorithm calculates the cost of travelling from a starting point and outwards to all adjacent points, until the destination point has been reached. In order to determine the route that can be called optimal, the algorithm needs a cost for each edge, called impedance. The Impedance represents some kind of cost associated with travelling along the edge. The units used for the impedance vary depending on the scenario, but the most common ones are time and distance. By using different values for the impedance, it is possible to find the fastest or the shortest route, by using time or distance respectively as the impedance.

A* is an improvement of Dijkstra's algorithm which by incorporating heuristics, which is a guess or a rule of thumb based on previous experience, should be able to find the optimal route faster. A used heuristic could be to choose the geographically closest node instead of traverse all of the nodes.

The last algorithm pgRouting supports, is the shooting star algorithm whose difference is that it is edge based instead of node based as the two previous algorithms. This algorithm supports rules such as left turn restrictions in junctions or added delays at traffic lights.

To find the optimal algorithm depends on many things, especially the size and type of network can have a significant influence. The special rule features for shooting star is not necessary at cemeteries, and in this project the networks that are traversed are very small and as such a possible performance gain from using the A* algorithm would likely be very small. Therefore the more precise Dijkstra's algorithm is chosen to be implemented.

5.3.2 Travelling Salesman's Problem

If the app for this project is going to support a tour planner, using Dijkstra's Algorithm is not enough, as it is only used to find the optimal route between two points, while the tour planner is supposed to make the optimal route between several points. This problem is the key element of the *Travelling Salesman's Problem*, TSP. As mentioned earlier PostgreSQL can be extended by libraries including new tools, where one is a routing tool called pgRouting, which can be used to solve the TSP. This tool is not an exact solution, as the algorithm uses Euclidean distance in between nodes rather than pre calculating the distance along the edges beforehand. Furthermore this algorithm has been designed for larger scale networks, such as cities, where the network for this project will have a fraction of that size. This could mean that the results from the implemented algorithm could be faulty, but since the project has a strict deadline, this option will be used to save time rather than trying to implement a different algorithm.

(Balstrøm, Jacobi and Bodum 2006) and (pgRouting n.d.)

5.4 Approximate String Matching and Semantic Search

If the user of the app is supposed to be able to search the databases for the grave of a specific historical person, a search function needs to be added to the app. But what if the user is not able to spell the name of the historical person correctly or if he does not know if a name is spelled with a C instead of a K? To combat this issue, Approximate String Matching also known as Fuzzy String Searching, can be used. Here, an algorithm is used to identify a substring that matched the entered string approximately. This is typically done using three types of operations:

Insertion:	Grave	→	Grave**s**
Deletion:	G**r**ave	→	Gave
Substitution:	Gra**v**e	→	Gra**p**e

where the substring with the least variety from the original string gets the highest value and is presented as the most likely result. The three operations can be interchanged and combined in order to find entries in a database that approximately matches the search criteria entered by the user in the search option. The most efficient method of achieving a useful string search is being researched, and while many functional algorithms exist it is a challenging task to understand which one to use. However, as with the TSP, PostgreSQL has a built in tool, that serves as a string matching algorithm that can easily be implemented when accessing the databases. This tool allows for a number of different search criteria as well as options for ranking the found entries. Because of this, it is possible to include an Approximate String Matching function to the app without having to install and calibrate additional software to the architecture.

(Baeza-Yates and Navarro 1996) (PostgreSQL n.d.)

Another issue when implementing a search option is the fact that databases and search systems cannot know the intent of a search like humans. For example if a human said "I am thirsty because it is really hot outside", another human would recognise this as a request for something cold to drink, but a database would not be able to understand the intent of the sentence in the same way. In this project, this becomes relevant when the users type in names of historical persons with their initials, such as H. C. Andersen. Here the app needs to recognise H. C. Andersen as being the same as Hans Christian Andersen in order to make the right query to the database and give the user the wanted result. To solve this problem Semantic Search can be implemented. Here the search engine is given a list of rules to follow in order to understand the intent of the user as closely as possible. Unfortunately no easy way was found to implement real semantic search so the search function is limited to only contain fuzzy string search.

(John n.d.)

5.5 User Interfaces

Besides the technical considerations when building a WebGIS, it is also important to address the user interface of the service since this can be said to be the communication between the developer and the user. As it was explained in the first stage of the report, the problem statement is a response to a lack in the market of WebGIS services regarding cemeteries. This problem will be solved through the app being developed, but in order to make this happen, the purpose of the app and the solution needs to be presented to the users in a way that they will understand. This is the significant role that user interfaces play; to transfer the ideas of the developer into an understandable tool for the user.

(Brodersen 2007)

5.5.1 Communication and presentation

When designing a user interface, there are a number of issues that needs to be attended for the product to live up to its purpose. Basically it can be said that the user interface has to promote satisfied users as opposed to frustrated users. In order to do this, elements that promote frustrated users need to be identified and eliminated. There are many such elements, but some of the most common are; program crashes, users cannot solve task and do not know why, illogical build design, users' expectations are not met, vague explanations of how functions work, as well as counter intuitive functions. To do this, the developer and the user need to have the same understanding of what is important for the user, and to do that, the developer has to know the user's *frame of understanding*. A person's frame of understanding is created from his knowledge, experience, skills and abilities, and if the developer's frame of understanding differs too much from the user's frame, then misunderstandings can occur. For instance, a developer might find it completely logical that his service's search function is found by clicking on a button with a pictogram of Sherlock Holmes, but if the user does not get the same relation by looking at the pictogram, then the search option might never be found. To counter this problem, the developer needs to make sure that the users in the target group understands the service they have been provided, and then and only then, the developer is able to communicate and present his purpose of the GIS service.

(Brodersen 2007)

5.5.2 Interaction Design

In order to build an app that takes the users' frame of understanding into account, the *Interaction Design* approach can be used. This approach has three characteristics that should be followed in order to understand the users. These are:

1. **Focus on the users**. Since the product is being developed for a specific target group, it is important to keep this group in mind when developing. This includes trying to match the app with the needs of the target group, but it also including the future users in the development process, so that feedback is used to enhance the existing solution. Because of this, a user test of

the product will be conducted in order to establish whether or not the target group is able to use the developed app.

2. **Identify goals and use of the product**. If the product is to be a success, it needs to have a goal that it can meet. Furthermore the specific use of the product including when and where it will be used, needs to be identified, so that the product can meet the demand. In order to fulfil this demand, a document will be executed stating exactly what the goal and purpose of the product will be.
3. **Iteration**. As explained with the development methodology in the project management section, it is important to repeat the processes of development several times in order to achieve a successful product. Because of this, this project will consist of four iterations, where each iteration will produce a prototype that will then be evaluated.

(Rogers, Sharp and Preece 2002)

5.5.3 *Visual and Functionality Interaction*

When the actual product most be implemented it is helpful to have a set guidelines to help with the visual look and feel as well as how the product should interact with the user and how the overall functionality should be.

One way is to use a set of heuristics. There are many types of heuristics, and many ways to use heuristics depending on the product that is being evaluated, but an acclaimed set of heuristics developed for web development is the one developed by Jakob Nielsen, which consists of 10 different heuristics. These heuristics cover a wide range of aspects regarding usability and by using them, the developer are able to develop a more useful product. Even though they were original designed for web development they are equally helpful for Android application development.

1. **Status updates**. A website should give the user feedback regarding the status of the performed request so that the user does not think that the site has crashed for example.
2. **Connection in between the real world and the website**. A website can easily alienate itself from its users by using words that are too technical. Instead the website should use words familiar to the user, and present information in a way that is recognizable by the user.
3. **Control of movement**. On larger websites, users are more likely to pick a wrong option or enter a wrong webpage. Here the website should allow the user to easily go one step backwards to the desired position to start over.
4. **Consistency and standards**. In order to lessen the confusion of the user when using a website, specific words or actions should not mean different things at different places of the site.
5. **Avoid errors**. The developers should spend more time preventing errors from happening rather than to make detailed error messages to the users that they will not understand anyway.
6. **Recognition over memory**. The user should intuitively understand the use, tools and functions of the website, rather than having to memorize what every single button does.
7. **Room for flexibility and efficiency**. As the user becomes more familiar with a website, the website should have a number of tools that can improve the experienced user's use of the site.
8. **Minimalism**. In order to improve the user's understanding of the site, all unnecessary information should be removed so that the user does not spend time looking at this.

9. **Errors should not scare users away**. When error messages have to be presented to the user, they should contain information that specifically explains to the user what went wrong, so that the user can easily solve the problem.
10. **Help**. A website might become too advanced for users to use without help or documentation of how to use the site. In that case, it should be easy to find the relevant topics in the help option.

(Gregersen and Wisler-Poulsen 2009)

6 Usability Testing

An important part of the chosen development methodology and model for this project is the prototypes. To evaluate on these prototypes usability tests are necessary. In this section the need and how to perform these tests is explained.

This project will pass through several iterations, and a number of prototypes of the app will be developed before a finished product can be presented. These prototypes need to be evaluated in order to identify what changes that should be implemented in the further development of the app. A tool for evaluating products is using usability tests with test subjects involved. Having the testing process split between several prototypes ensure that many more problems will be found than using only one test. (Krug 2006)

However, in order to understand the usefulness of usability tests and user tests, it is interesting to ask the question, “Why are user tests important?” It is clear that the objective of the user tests is to find the logical and technical flaws within the app, but why cannot the developers themselves do that?

There are three reasons why it is important to conduct usability tests on a product:

- Designers and developers cannot forget, or pretend to forget the knowledge they have acquired during the process of development.
- The Designers are not necessarily part of the target group
- There is no such thing as an average user, meaning that people are different, and thereby it is impossible to predict every single person’s actions and thoughts.
  (Kuniawsky 2003)

These three reasons can in general be narrowed down to the designer’s inability, to pretend to not know about the underlying structures and ideas of a product as well as the inability to foresee the user’s reactions to it. Because of this the three reasons explain the shortcomings that designers and developers have, when it comes to seeing their product with new eyes. As a result, user tests have to be conducted, in order to have the product evaluated by a person with no prior knowledge or insight about the project. (Hansen 2005)

Another question that is also both interesting and important to ask is, “How many test subjects should be recruited?” In the 1990’ies it was discovered through research that by using five test subjects about 75 - 80 % of the errors and issues of a product would be detected. This percentage however, did only increase slightly after the fifth test subject, meaning that it would take ten additional test subjects in the usability test to discover the remaining 20 – 25 % of errors in the product, while the cost of the test would increase sharply due to the higher number of participants. These numbers can be derived from Figure 26.

(Gregersen and Wisler-Poulsen 2009)

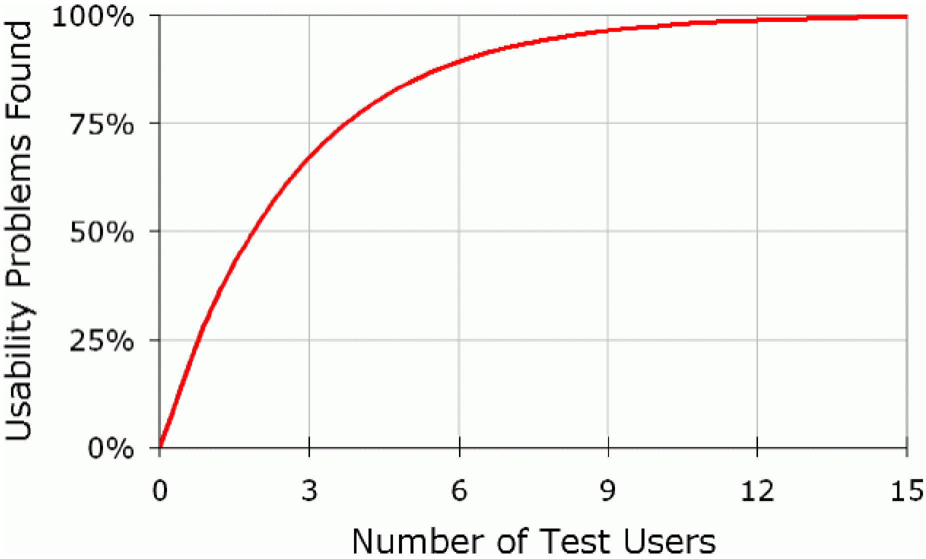


Figure 26 The graph shows the ratio between the number of test users and the number of problems found. http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

This claim of five test subjects being the golden number has in recent years been challenged by a number of new studies. In one of these, 49 participants were invited to test a product, and here it was suggested that five participants in a user test only discover as little as 35 % of the usability problems. Furthermore this research claims that in several cases, serious issues were not discovered until the 13th and 15th participant in the test.

From this it is clear that there exists contradicting studies within this field, and as such it is clear that the right answer to the question regarding the number of test subjects to use, has yet to be found. However, the rule of five test subjects is still an accepted theory that is used by many companies today.

(Spool and Schroeder 2001)

6.1 Set Up

According to (Bastien 2010) there are nine steps that can be followed when designing a usability test in order to achieve a successful outcome. Some of these steps are more important than others while others are self explanatory, and as such it is not a requirement to plan or use all of these steps in great detail for a single test. However, the more that are implemented, the higher is the chance of acquiring a test result the can enhance the overall performance of the product. The possible steps are:

- **Objectives.** The user test needs to have a specific objective which the test is supposed to solve; otherwise the testing does not have a purpose, and thus loses its meaning of existence.
- **Recruitment and testing.** In order to receive a useful outcome of the test, the test subjects have to match the target group that is needed to be tested. As a result, the recruitment of test subjects needs to consider who will be selected as well as under which circumstances.
- **Description of scenarios.** The scenario in which the test subject is physically put, or asked to imagine to be put in, can be planned and specified to make the test more realistic.
- **Tasks.** In most cases the test subject will need to be presented with one or more specific tasks that he needs to solve in order to gather information about the usability of a given product.

- **Observation and recording method.** It is necessary to plan how the results of the test will be recorded.
- **Design of test protocol.** The tester, or facilitator, needs to be chosen and instructed in the plan and design of the usability test.
- **Preparation.** A test can involve a number of materials and items that of course need to be collected and prepared before a test can begin.
- **Data analysis procedures.** When the data from the test has been collected, it is important that a structure for the analysis of the data has been prepared, so that an organized result can emerge.
- **Evaluation of test results.** The final step is to present the result of the test as well as evaluating on the results.

To be able to design a user test that can follow as many of the mentioned steps above, it is necessary to investigate further as to which methods and observation techniques that can be used for testing.

6.2 TESTING METHODS

When it comes to usability testing there are a number of different testing methods that can be implemented. These methods either focus on how the test subjects are recruited, how they are tested, or in some cases both. The four most common testing methods have a great variety in their practices, and all work well for specific tasks. These methods are described in the following.

Hallway Testing Method gets its name from the original strategy which involved finding a number of random test subjects in a hallway, and introducing them to a usability test of a product. The strength of this method is that it allows for the developers to identify major issues with a product, in the sense that if random people cannot complete a given task with the product, then it is too complicated to use.

In some cases of testing, the facilitators and test subjects are not physically able to meet. This can be because they are not in the same country, are in different time zones, or in two different university buildings for example. To combat this, Remote Usability Testing can be used, which either can be synchronous, typically in the form of a video link, or asynchronous which can be a test on a website.

Another method of usability tests is Expert Reviews which uses specialists as test subjects. The advantage of this method is that the experts are able to identify and comment on issues and performance problems that are relevant to the product based on their level of expertise. These comments are usually presented using the beforehand mentioned heuristics about the visual and the functionality interaction with the user.

The A/B Testing method is often used for website design, and as such it can prove useful when developing an app. When using this method, the test subject is presented with two different versions of a product, and the test subject is then asked to explain which version he thinks works better and why. Another version of this method is to have two user groups that each is testing one of two different version of a product. Based on the test result, it will be possible to see if one group and thereby one version of the product performs better.

(Gregersen and Wisler-Poulsen 2009)

6.3 Observation methods

When the testing method has been chosen, it is equally important to decide how the data from the tests should be observed and collected. Like with the testing methods, there are many types of observations, but some of the most common ones are as follows.

The Think Aloud Protocol method is also known as the Talk Aloud Protocol, because it revolves around having the test subjects to tell what they think, while they are performing a given task. While the test subject thinks aloud, observers write down all of the test subject's comments of what they are doing, why they are doing it, and what they are trying to do. This gives the observers a firsthand experience of how people who have not been a part of the development phase react and interact with the product at hand. The method can be enhanced further by including a facilitator that asks the test subjects non leading questions, such as "what are you thinking?" or "now what?" in order to understand the test subjects choice of actions.

A much more technical method is the Eye Tracking method. This method uses advanced equipment to track the test subject's eye movements or point of gaze, and by doing so the observers are able to detect what is taking the test subject's eye's attention. This method is useful in many different situations, and web development is definitely one of them. Because there are many potential fields of interest on a screen, it is interesting to perform an eye tracking test, in order to see if there are elements on the screen that take all of the test subject's attention, leaving specific buttons or functions unattended. By monitoring the sight of the test subject, it can be possible to detect patterns in his gaze that he might even be unaware of himself, which thereby would not be detected using the think aloud protocol.

Another very common method of collecting data is by using a Questionnaire. This can either be done electronically or on paper, and allows for observations of a large group of participants relatively easily. In the questionnaire the test subject is asked a number of questions that are related to either a specific situation or an opinion on some matter that has relevance for the observers. This could be after having performed a number of tasks using a specific product. In order to produce a successful questionnaire there are many considerations to be made. Since the questionnaire is likely to be the only communication the observers have with the test subject, all relevant questions to the test have to be included in the questionnaire. Furthermore, the questions need to be expressed in the right way, so that the questions do not influence on the test subject's answers, and at the same time makes sure that the answers that are received are meaningful for the evaluation of the test.

(Lewis 1982)

6.4 Prototype Test Planning

As explained the development period would run through three iterations that each would produce a prototype. In order to conduct usability tests on these prototypes, it was necessary to evaluate on which methods and techniques that would be implemented in the individual iterations.

6.4.1 Usability Test in Iteration 1

The first prototype that would be developed in the project of course had to be tested. However, because the scheduled functionality of this prototype was very shallow, it was decided that it would not make sense to plan an actual usability test for this prototype, and instead only the project group members would test the raw functionality. This meant that none of the presented recruitment or observation techniques would be put to use.

6.4.2 *Usability Test in Iteration 2*

For the second usability test, it was decided that it would be useful to conduct an Experts Review, where the specialists could identify logical and visual issues with the prototype based on their expertise. This usability test was therefore planned to be held at a status seminar for geoinformatics students writing their master thesis. Here, fellow student as well as the project supervisor could test and evaluate the presented prototype, and based on the Think Aloud Protocol, the project group could document the reviews.

6.4.3 *Usability test in Iteration 3*

The prototype in this iteration was planned to include many advanced functions, and as such a more detailed usability test had to be conducted. Because of this, the recruitment of test subjects in iteration 3 would be done using the Hallway Testing method. This was decided because it, as stated, allowed for the developers to locate major issues in the product. The Experts Review method was not relevant to this stage of the testing, as experts should already have tested the product in iteration 2. The Remote Usability Testing method was considered as the main tool for recruitment and testing. Here it was suggested that the test subjects could be invited to visit a website from where they could download the app, a task description, and a questionnaire. However, it was apparent that this method would be very complex, and would require a lot from the test subjects, which most likely would scare of the majority of the participants, jeopardizing the test. The A/B Testing method was also considered, but it was decided that this type of test would be better for a fourth iteration, after the major issues had been uncovered.

In order to observe the test subjects, the Think Aloud Protocol would once again be used. This method was chosen because it was simple to use and require less preparation and resources than the other two mentioned. The Eye Tracking Technique required equipment that the project group did not have access to, and it would take time to understand all of the theories and techniques involved. As stated, the Questionnaire method was good for administering a large group of participants, especially for Remote Usability Testing, but since this test would only involve five test subjects, a questionnaire would require a large amount of work compared to the number of participants. Due to this, the Think Aloud Protocol was chosen.

Planning Stage: Summary

The discussion and presentation of the theories and methods that will be used to develop and evaluate the app built in the project conclude the second stage of the report, the planning stage.

The management part presented the overall plan of the project product, including a SWOT and risk analysis of the possible issues that can occur while developing. This part concluded that this development project could be categorised as a high risk project and that it was necessary to handle these risks in a proficient way. It was found that a prototype driven spiral model would provide the project with sufficient ways of handling the limited time available, as well as the major risk of problems that might occur with the chosen development environment.

Furthermore the theories and technologies that would be implemented were presented which provided the overall plan for the design and development of the service. Here the key elements of how to design a user interaction product were found together with the specific technology which would be used.

The third stage of the report will explain how the actual development of the app was executed from idea to product, as well as presenting the prototypes of the service.

EXECUTION STAGE

The third stage of the report covers the execution of the product development during the project period. During this stage, the development of the product will be explained, from the first initial idea to the final version of the produced app. In the end of the defining stage the goal for the project and thereby the development phase is defined to:

> *Development of an app used to find and navigate to the graves of historical persons, provide information about them and in general allow for exploration of cemeteries.*

The development of the product will be done using the five phases of the developed spiral model; research, design, implementation, risk analysis and prototype testing. As described in the development methodology section, this project ran through three iterations, however this does not mean that the presentation of the five phases will be repeated three times. Instead, the individual phases of all three iterations' will all be explained in main sections regarding each of the five phases. In several phases, little additional development happened during the later iterations, and in these phases the progress within a specific phase throughout all three iterations is presented in one section. In other stages however, there was much progress from every iteration. In theses stages, the development done in each individual iteration is separated, so that the progress is presented clearly. This is done to provide the reader with the best overview and knowledge of the development process, as well as to avoid repetitions.

The five phases of the spiral model and their content described in this stage are:

- **Research**
  In this section, the findings of the three iterations regarding data availability and the scenarios for potential use of the app, are presented.
- **Design**
  Here the considerations regarding the functional and visual design of the app are discussed.
- **Implementation**
  The implementation of the chosen technologies and methods in the system design, including examples of information flow for certain tasks are presented in this section.
- **Risk Analysis**
  Here the implemented technologies of each iteration are evaluated, in order to assess if the risk of not completing the development of the app is too great, and if so what new technologies to use instead.
- **Prototype Testing**
  Here the individual prototypes developed are presented as well as the result of the performed usability tests.

7 Research Phases

The research performed at the start of each iteration was used to define the desired functionality of the app. During the first iteration, a brainstorm resulted in the definition of three user scenarios that could occur in which the app could be useful. Furthermore the accessibility of data was investigated and available data was acquired.

In the later iterations, the research phase was used to analyse the results from the prototype test of the previous iteration in order to assess, if the functionality of the app should be modified.

7.1 User Scenarios

Before the actual development of the app could begin, it was necessary to reconsider the situations where the app would be used. The initial purpose of the app was for the users to be able to be guided to the grave of a historical person, and to receive basic information about the life of the person. This basic function revolved around the idea that the user was at the same cemetery as the grave of interest, and was in need of help locating the specific grave due to the size of the cemetery. In this case, the ability to be able to learn about the life of the historical person might not be of importance, but it would still a useful and interesting option to have available.

Even though the scenario described above was the initial focus for the app, it was still possible to imagine situations where the user was not at the same cemetery as the grave of the historical person that was being searched. If fact it was quite possible that the user did not have the slightest idea where the historical person was buried. For instance a user might be interested in the whereabouts of H. C. Andersen's grave. In this case the app could be used to provide the user with information about the location of the grave, and when it has been located, the user could acquire more information about the person if wanted. Based on this it did not matter for this functionality if the user wished to go to the cemetery that housed the grave or not, but the app should of course support users who wanted to get directions to a cemetery that they would not close to.

Finally it might be possible that some users were not searching for the grave of a specific person, but rather wanted to learn about the historical persons buried at a specific cemetery that they were not currently at. In this case the user should have the possibility to search freely from cemeteries supported by the app. This way the users would be able to explore cemeteries that had particular interest to them such as the cemetery in their home town or famous cemeteries in the cities.

The three scenarios described above were the foundation for three distinct overarching functions that should be included in the app. These three scenarios are:

1. Users wanting to explore the cemetery they are currently visiting.
2. Users searching for the location of a historical person's grave regardless of the geographic position of the grave
3. Users searching for a specific cemetery regardless of the geographical position of the cemetery

These three scenarios and the additional research performed in the research phases, made it possible to develop a more detailed list of the functionalities that the app should support. These functionalities were still sorted under the four main categories functions; Framework, basic functions, advanced functions and refinement. During the three iterations, the list was expanded and refined. The final list can be seen in Figure 27 while the first draft can be seen in Appendix D.

Categories
Framework
- A map of the cemetery showing the individual positions of the graves.
Basic Functions
- Function to search for a historical person with the result showing which cemetery and the position at the cemetery the person is buried.
Advanced Functions
- On the map different colours for historical persons' graves will be used based on their profession.
- Function to show a map of all the historical persons' graves on a specific cemetery.
- Function to find and exploring nearest cemetery.
- When a grave is clicked on the map an information window will show with information specific for this grave and the person who is buried there.
- Implementation of navigation to a grave on road/path level within the cemetery.
- The navigation functionality must include walking mode
- The navigation functionality must include bicycling mode
- The navigation functionality must include driving mode
- Navigation towards the cemetery.
- Tour guide for a number of selected graves on the cemetery.
Refinement
- The grave search can be enhanced by auto complete of the name and fuzzy/semantic search.
- Support of UTF-8 character set.
- Support for different screen sizes.
- More advanced representation for cemeteries. Markers instead of WMS and clustering of data shown on the map.
- The app should work smoothly without any performance problem.
- Any graphic problems should be solved.
- Any functionality problems should be solved.

Figure 27 The final desired functionality for the app divided in categories corresponding to the four iteration cycles of the development.

7.1 DATA AVAILABILITY

Before the system could be implemented, data to represent the real world was necessary. With the project's focus of developing an WebGIS service to historical person's graves, a number of data sets would be useful for information regarding the persons, the position of the grave, the infrastructure of the cemeteries and the infrastructure outside them as well.

Information of historical persons is available from many sources. For the development in this project, Wikipedia was an obvious choice since this site has an API that makes it possible to extract certain parts of the information available in their articles. The information at Wikipedia is not necessary completely correct, and does not cover as many aspects as other encyclopaedias. In particular Wikipedia lacks specific information found in "*Midt i verden i 250 år*", about the stories of interesting grave stones at Assistens Cemetery for example. For a project with more resources it would have been possible to obtain information from other encyclopaedias and acquire information from the copyright holder of "*Midt i verden i 250 år*".

The positions of graves were not easy to acquire, but with some work it was possible. Many cemeteries have this data digitalised but as mentioned earlier, no central registration exists, so it was necessary to ask every cemetery for this information individually. Furthermore, there still are cemeteries that do not have cemetery maps with burial plots digitally, and instead the data for these cemeteries had to be digitalised from orthogonal photographs. For this particular project, the City of Copenhagen provided datasets for Assistens Cemetery and Vestre Cemetery, and the Municipality of Aarhus provided datasets for their two cemteries, Nordre and Vestre. Additionally data from The Cemetery of Holmen was also included in the project but here it was necessary to geo reference an image of a map to include the position of the graves from this cemetery. Each of these five cemeteries have a list of interesting graves at their particular cemetery, but none of the cemetery offices would provide a digital list of this information. This however, was due to the time it would take to collect the data and not because it was restricted information. Regardless, this data was obtained from the cemeteries' websites, where a list of graves of interest was freely available. Here the graves of interest were noted, for then to have the information typed in manually for all of the graves at Assistens Cemetery and only a number of graves for the other cemeteries. This was done because the success of this project did not rely on having all graves of interest present in every cemetery present in the developed app. Instead only a handful of graves were needed to test the developed functionalities.

The publically available directions services in Denmark only have a few if any of the roads and paths available in the cemeteries. Some cemeteries though, are covered better than others, and Assistens Cemetery is almost covered completely by Google Directions with bicycle paths and walking paths. Unfortunately many other cemeteries are not covered at all. The FOT data set, now freely available, covers close to 100 % of the roads and paths at the cemeteries but does not necessary explain if it is a road for cars, a path for bicyclists or just a walking path. The Open Street Map data set covers this information for some cemeteries, but not enough to make a navigational service available internally at the cemeteries. This lack of information about the type of roads meant that it was necessary to acquire information about the type of road or path manual from the individual cemetery. This information was added to the FOT dataset that had a precise position of the roads on the cemetery, and a network was generated from this data. Only network data from Assistens Cemetery and Vestre Cemetery have been created in this project, because it like with the graves, is not a requirement for the success of the app to have a functional network at every included cemetery.

If navigation has to be performed outside of a cemetery another service could be used. Many services exist for getting directions between two addresses in Denmark, and in general the quality of these is very good. Google Directions is one of these which also have an API so this service would be very easy to implement in the app.

8 DESIGN PHASES

The design of the app had two aspects; the visual design and the functional design. In the first iteration of the execution stage, only the very basic visual design of the map with graves was decided upon. In the second iteration a number of overall visual design considerations were made. This concerned what graphic elements that the maps should display, what colours to use and how the buttons and layout of the app should be designed. Additionally a wireframe regarding the functional design was created, explaining how the functionalities of the app should interact and how the user should move from a menu to a map for example. In the final iteration, this phase was used to enhance the existing concept of visual and functional design based on the results from the previous research phase.

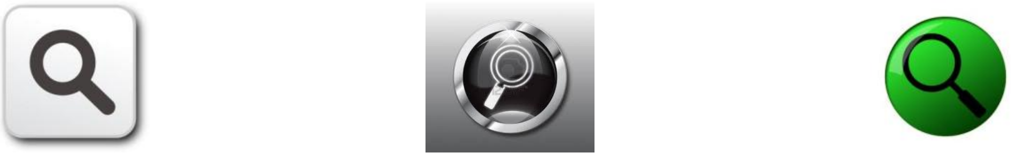
In the process of designing the visual and functionality aspect of the app, the 10 heuristics from Jacob Nielsen explained in the development theory, were used as inspiration and guidelines.

8.1 VISUAL DESIGN

Before the implementation of the WebGIS system could begin, it was necessary to consider the visual design aspect, so that the user would know what to do and what he was looking at when they start the app. To do this, the future user's frame of understanding had to be taken into account. Because of this, the icons and symbolism that would be used, had to either be common so that the user had seen them before, or create a distinct association to the wanted function or object.

8.1.1 PICTOGRAMS

The app that was developed contained a number of buttons, and a good way to distinguish these from one another, is to use pictograms. Many functions have common standard pictograms associated with them, such as the magnifying glass for search options as seen in Figure 28. These buttons use very different graphic designs, but the overall pictogram is the same, and that makes the user recognize the function of the button without having used the specific web service before.



Clker.com 123rf.com Shibashake.com

Figure 28 Examples of search button pictograms.

This type of recognition can be linked to many other pictograms such as; a house representing the home menu, a question mark representing the help menu, a bicycle representing directions for bicyclists. By using as many of these classic pictograms in the visual design the user would be able to feel familiar with the functionality of app to be developed from the very first time they would use it.

The heuristic of recognition was also used for colouring of directions. In the map the directions are shown with green colour for walking paths, blue for bicycle paths and purple for roads where cars are allowed to drive. These colours the user would have seen on the direction icons seen in Figure 29, and it was thought that the colours would be recognised when the map with the same colours was shown for the user.

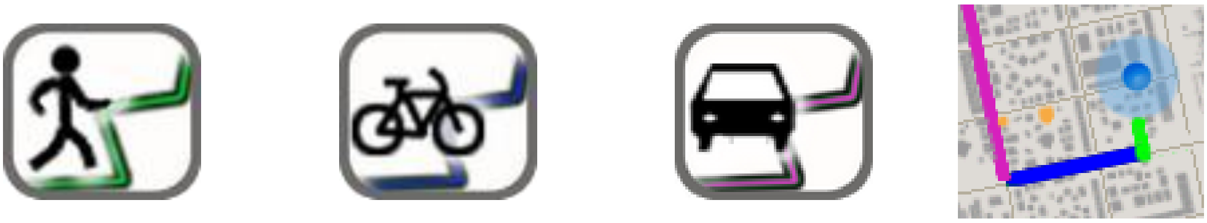


Figure 29 Direction icons, and an example of the same colours used in a map.

8.1.2 MAP LEGENDS

The second major design consideration that needed to be taken into account was how the map legends should be designed. Here it was difficult to rely on the user's ability to recognize what he saw, because colours and shapes can mean different things on different maps. For this app the features of interest on the maps were graves. Here many cemeteries already have a printed map where the graves are colour coded depending on the profession of the historical person. Because the user most likely do not has a specific association between the colour of a grave and the buried person's profession, these predefined colour codes were optimal to use because it made the app compatible with the existing printed map from the cemetery offices. These colours can be presented in a number of ways, for instance through a full colour or only by colouring the edge of the specific graves as seen in Figure 30.

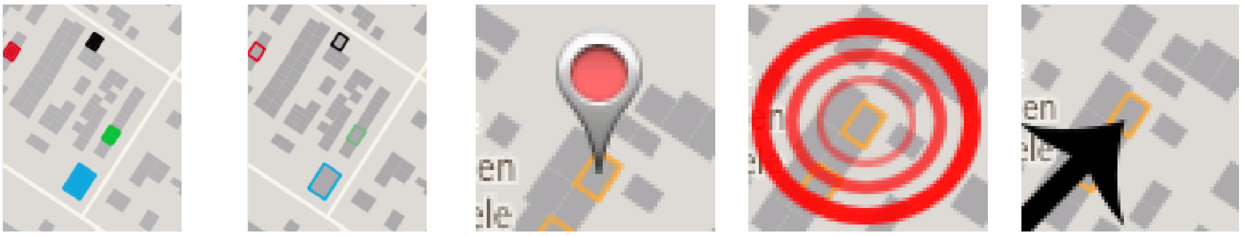


Figure 30 Examples of graves and grave markers.

Additionally graves that are selected can be displayed using a marker. Here it is possible implement Google Maps markers, that are easily recognizable by most people, but other more original markers can be designed as well, as seen Figure 30.

The final grave map markers shown in Figure 31 were inspired by a combination of the Google map marker, and the colours used on the official maps on cemeteries managed by the Municipality of Copenhagen. Each colour represents a specific profession, except the black one which represent an interesting grave, not necessarily because of the person but because of the gravestone or monument itself.

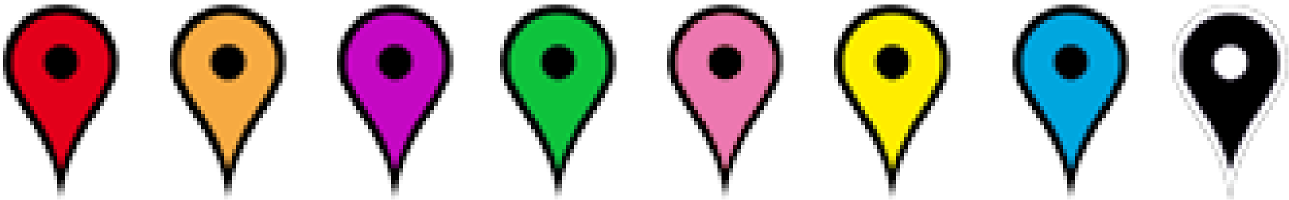


Figure 31 The used grave map markers for this project.

8.2 Functional Design

While the visual design discussed how the graphic elements of the app should be designed, it was also necessary to create a design of the functionality of the app, and how all of the functions should be connected to each other.

The general idea of minimalism was used throughout the design phase. The different possible actions were kept at a minimum. Some actions like the directions guide was not implemented at all of the cemeteries, and were therefore removed. This might conflict with the heuristic of consistency, but here the heuristic of minimalism was favoured.

Whenever possible, a progress bar was to be shown on the screen while the app was fetching online data. This was done to assure the user that the app was still working, while for instance searching for directions towards a grave.

Errors were of course sought to be avoided, but whenever an error screen was necessary, an explanation for what the user could do to avoid it, would be given.

8.2.1 Wireframe

With the overall considerations regarding the functional design in place, these guidelines should be put to use. This was done by creating an initial layout of the desired functionality and flow of the app. Here not only the design of the screen views that the user could encounter were considered, but also the functions of the individual screen view's elements like buttons and popup boxes. This layout can be seen in Figure 32. The layout was revised after every iteration based on the usability tests of the prototypes.

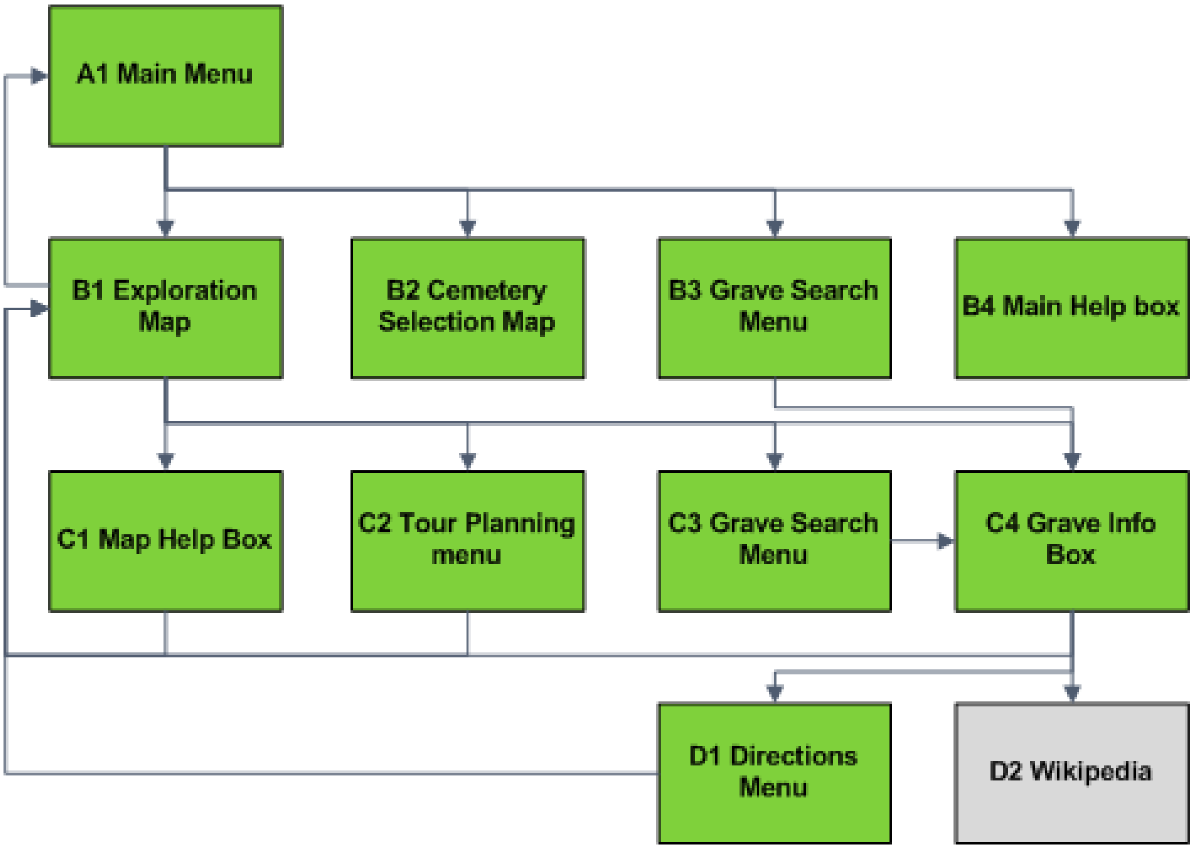


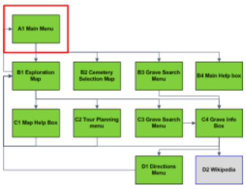
Figure 32 Wireframe of the eleven main functions of the app.

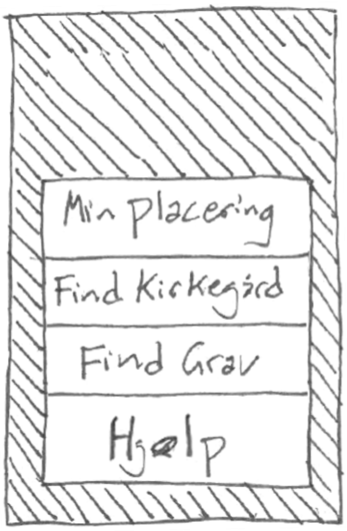
The diagram displaying the functionality layout is designed so that each arrow represents a click to a new screen view. The movement in between the main functions of the app was restricted to buttons which brought the user forward, while the back button on the smartphone would bring the user backwards in the wireframe structure. In addition a home button was placed in the exploration map view that would bring the user back to the main menu of the app. The letters in the boxes represent the levels of the interface describing how deep in the app you are while the numbers explain which one of the interfaces of the level that is being used.

The flow of the functionality has been designed to form a loop, meaning that you will never reach a dead end in the app where you have to revisit the previous site to continue your exploration. This design helps the user to make new searches without having to worry about finding their way back to the search menus.

In the following sections, the flow of the app is presented using a wireframe. Here sketches and descriptions of each screen view the user can find in the app is presented, along with a guide explaining what screen views that can take the user to the specific screen view, and where the user can go from there. The first sketches and wireframe can be seen in Appendix E. The screen view for Wikipedia - D2 will not be explained since it is an extern website which the project group has not developed, but only refer to.

A.1 Main Menu
Predecessor – B.1
Successor – B.1; B.2; B.3, B.4



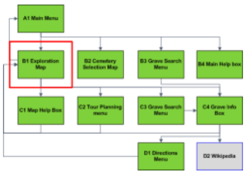


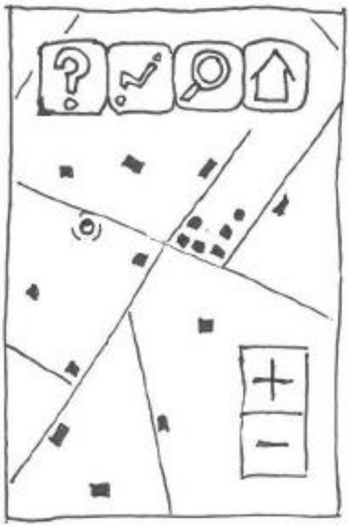
The Main Menu is the welcome page that greets the users when they start the app, and is designed for the user to quickly being able to start exploring cemeteries. This menu consists of a button to answer each of the three user scenarios described in the previous section. These being finding the closest cemetery based on the user's position, finding a specific cemetery, and finding a specific grave. Furthermore, a button representing a help menu is present. When a button is clicked the user is taken to either the Help Box - B.4, the Exploration Map - B.1, Cemetery Selection Map - B.2, or Grave Search Menu - B.3 and the exploration has begun.

When the Main Menu is revisited all previous searches are reset, making it possible to start a new clean search.

B.1 Exploration Map

Predecessor – A.1; B.2; C.1; C.2; C.3. C.4; D.1
Successor –A.1; C.1; C.2; C.3; C.4;





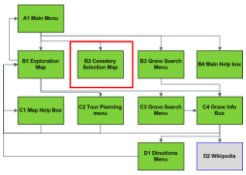
The Exploration Map is the key element in the app, meaning that most of the functionality of the app revolves around this screen view. This becomes apparent when looking at the predecessors where all screen views but two can bring the user directly to the Exploration Map. The basic function of the Exploration map is to display the graves at a cemetery so the user quickly can start exploring graves of interest, but the Exploration Map can be manipulated in a number of ways and made more specific through the use of various search options found in other screen views.

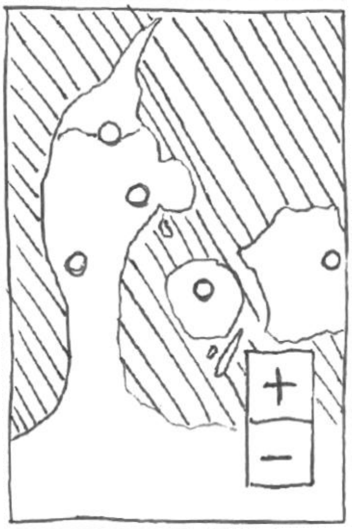
The screen view of the Exploration Map will vary depending on the predecessor, but will in its simplest form, which is found through the Main Menu, only include the user's position as well as the graves at the nearest cemetery. The graves will be colour coded depending on the profession of the buried person. From here the user has five ways to go. If the user clicks on a grave, a pop up box will appear, C.4, which can be read to gain information about the buried person whose grave was clicked. In the upper part of the screen view four buttons are located that when clicked will take the user a new screen view, either the help and info box - C.1, tour planner - C.2, search for grave - C.3 or back to the Main Menu, A.1. When returning to A.1 by clicking the associated button, all saved data displayed on the Exploration Map is reset.

B.2 Cemetery Selection Map

Predecessor – A.1

Successor – B.1





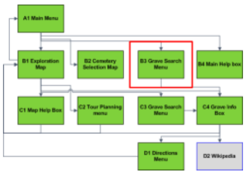
This screen view is meant for users who want to search cemeteries that they are not close to, as opposed to the default functionality of the Exploration Map . The screen view displays a map of Denmark with markers at cemeteries that are found in the app's database. From here the user can zoom in on the map, and when a specific zoom level has been reached, the markers will disappear and a polygon of the cemetery will be displayed instead as well as the names of the cemeteries.

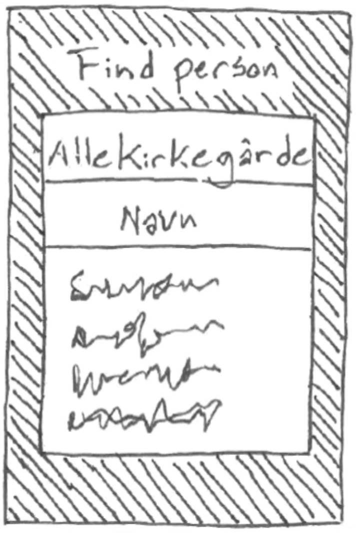
The user can then click a cemetery which will start a new screen view, B.1. This new view will be identical to the basic Exploration Map with the difference that the user has chosen which cemetery to go to, and that the user's position is not visible on the cemetery.

B.3 Grave Search Menu

Predecessor – A.1

Successor – C.4

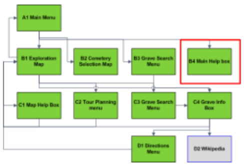


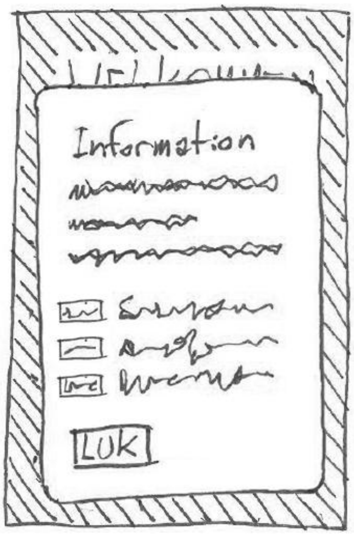


The Grave Search Menu is available for users who want to find the grave of a specific historical person regardless of the location. To do this the user can make a search by typing in the name of the person that wants to be found in the search field. Alternatively the field can be left blank, which will provide a list of all registered persons.

When the desired search criteria have been filled in the search button can be clicked, resulting in a list of all historical persons that match the search. The user can then click a name on the list of matching persons, which will take the user to a new screen view of an info box, C.4, with further information about the selected person.

B.4 Main Help Menu
Predecessor – A.1
Successor – A.1

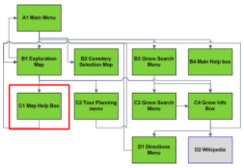


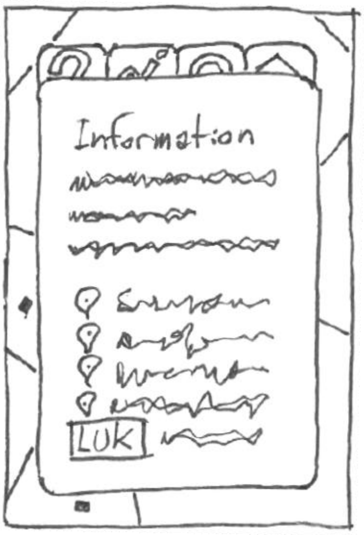


From the main menu it is possible see a help box with no new buttons except a close button. The function of this screen view is to introduce the user to the concept of the app and explain when to use the three different modes of the app.

When the help has been read the box can be closed which take the user back to the Main Menu (A1).

C.1 Help and Legend Box
Predecessor – B.1
Successor – B.1

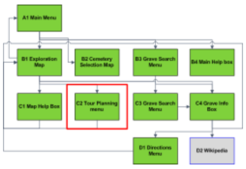


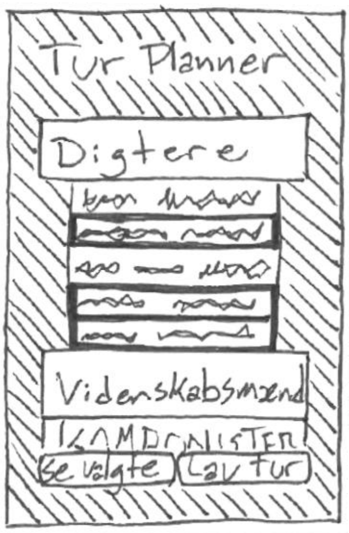


The Help and Legend Box is a simple box on top of the Exploration Map with no new buttons except a close button. This screen view's only function is to explain how to use the Exploration Map as well as describing the colour code of the graves.

When the box has been read it can be closed which will take the user back to the Exploration Map, B.1, ready to explore.

C.2 Tour Planning Menu
Predecessor – B.1
Successor – B.1

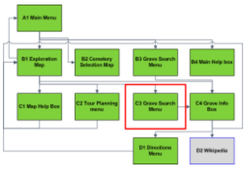


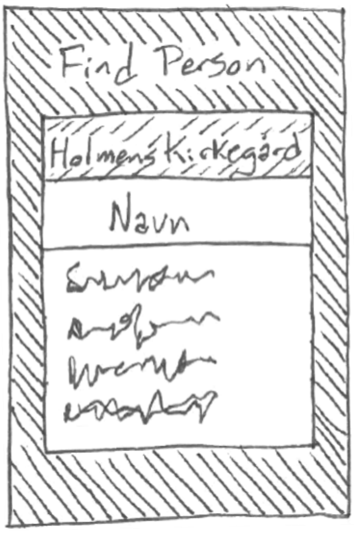


The Tour Planning Menu can be used to plan a tour on a cemetery if the user wants to visit several graves. When this menu is opened seven coloured buttons are visible, representing the seven categories that the historical persons are divided into according to their profession. When a button is clicked a drop down menu opens which reveals all the names of the historical persons within the category. From here the user can click a name, which will be highlighted in order to select which persons that should be added to the tour.

In the bottom of the screen view, two additional buttons are present. The first one makes it possible for the user to see a list of the persons that have already been selected, while the other takes the user back to the Exploration Map where a tour that is calculated from the list of selected persons is added.

C.3 Grave Search Menu
Predecessor – B.1
Successor – C.4



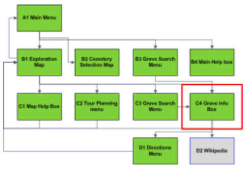


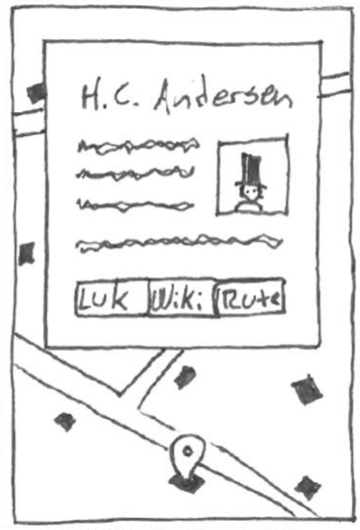
This screen view is similar to the B.3 Grave Search Menu with the difference that the search is limited to the specific cemetery that has been chosen at a previous stage in the app.

Besides this difference, the screen view works as explained earlier, and when a name on the search list is clicked the user is taken to an info box, C.4.

Execution Stage

C.4 Grave Info Box
Predecessor – B.1, B.3, C.3
Successor – B.1, D.1, D.2

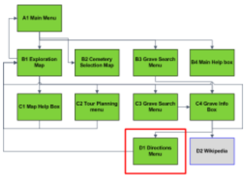


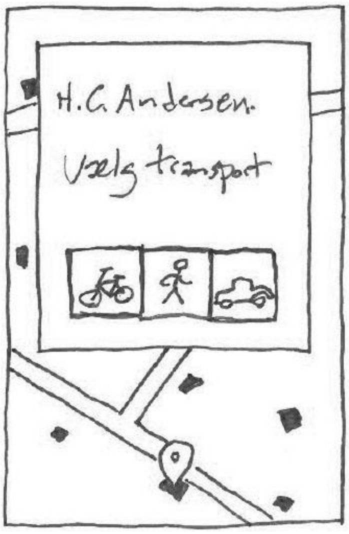


The Grave Info Box is a pop up box on top of the Exploration map that contains information about the person whose grave has been selected. This selection is either done by clicking on a grave in the Exploration map, or by clicking on a name in one of the two Grave Search Menus. The box contains text, and if available a photo, that is parsed from Wikipedia, and is meant to give the user basic information about the person of interest.

From this screen view the user can either return to the Exploration Map, B.1, or get directions to the grave from the user's position, D.1. Furthermore the user will have the option to be transferred to the corresponding Wikipedia page if they want to know more about the target person.

D.1 Directions Menu
Predecessor – C.4
Successor – B.1





The purpose of the Directions Menu is for users to choose the means of transportation that is preferred between the users' position and the previously selected grave. Because of this the layout of the Directions Menu will depend on the location of the user.

If the user is not at the cemetery that houses the previously selected grave, a box will open over the previous Grave Info Box with two buttons that can be clicked depending on if the user wants to travel to the grave by car or by public transit. If the user on the other hand is at the cemetery of interest, a similar box will open on top of the Grave Info Box, only in this case the buttons are not fixed. Instead the app will determine what kind of transportation that is allowed at the specified cemetery, and depending on this up to three buttons will be available that each represent getting directions by either walking, by bike or by car.

No matter what layout that is being displayed and what button that is clicked, the user will be directed back to the Exploration Map, B.1, where a route will be added to the map, displaying the suggested route calculated by the app.

9 Implementation Phases

In this section, the system behind the app that was developed throughout the implementation phases based on the design considerations is documented. During the three iterations many functions were added. As it has been explained, the goal for the first iteration was to built the framework for the app, while the second and third iteration should be used to develop the basic and advanced functions respectively. This plan was followed, and as a result, the amount of code contained in the front- and backend of the app grew considerably after each iteration. Because of this, the complexity of the system increased tremendously with every iteration, and as such it is difficult to capture every single event that happened during these phases in the report. In order to avoid repetition, only the system developed during the third iteration will be described.

To explain how the system was built, how it works and how the functionalities were implemented, this section is divided into four parts.

1. In the first part, the development process during the three iterations is presented.
2. The second part presents the system design of the app, and explains the complexity of the system created, including how many lines of code that was written and how it was distributed across the entire system.
3. The internal components of the system are presented in the third part
4. In the fourth and final part, examples of the key functions of the app are presented, including the communication throughout the system to solve the function.

9.1 Development Process

The research and design phases in the first iteration created the foundation for a list of desired functionalities and a wireframe of how the main functions of the app should be connected. However, since the project group had no previous knowledge of Android application development, the first task was to get to know the development environment.

On http://developer.android.com/training/index.html, Google has a number of getting started guides for new developers. These guides were first followed, along with Google's MapView tutorial which shows how to integrate a Google Map in an app. Other guides were used in the following days in particular guides about map overlays and ways of incorporate a WMS with a Google Map. After nine days of research, a framework using Google Maps Android API v2 to overlay a WMS was found which provided acceptable performance, http://www.azavea.com/blogs/labs/2013/01/wms-on-android.

With the framework in place, it was just a matter of implementing the rest of the functionality. This was done by making heavily use of Google's search engine and when the right question was asked, a useful answer often would be given. Especially at sites like http://stackoverflow.com/ for general Android code questions, http://gis.stackexchange.com/ and http://www.postgresql.org/docs/ for database related questions and http://www.php.net for help writing the necessary PHP scripts. In addition to these internet sources two books covering PostGIS (Obe and Hsu 2011) and an introduction to Java programming (Barnes 2000) were used.

During the development a code diary was kept, see 0. In this diary it is possible to get a deeper understanding of the different problems involved in implementing the functionality along with the daily progress or lack hereof.

Since none of the project group's members were skilled Android programmers it was necessary to rely on other people's knowledge, and as explained, this was often done using various "questions and answers" sites for small pieces of code. Often this code was heavily changed before it could be used for this app, and because of this, it was not thought necessary to provide references for every part of the code.

In two cases several lines of contiguous code were used. This was to return the network data status of the smartphone and code for implementing a WMS overlay in a Google Map. In these two cases the source is mentioned in the code. No other references are given in the code, but in the daily dairy, references to 71 webpages containing code examples which provided inspiration are mentioned.

9.2 System Design

The final system completed during the third iteration consists of a number of components besides the smartphone with the app installed. These components are servers which through a number of different software programs performs several services for the system. Only the programs used on the server controlled by the project group is shown since the internal structure of Google's and Wikipedia's servers are not know.

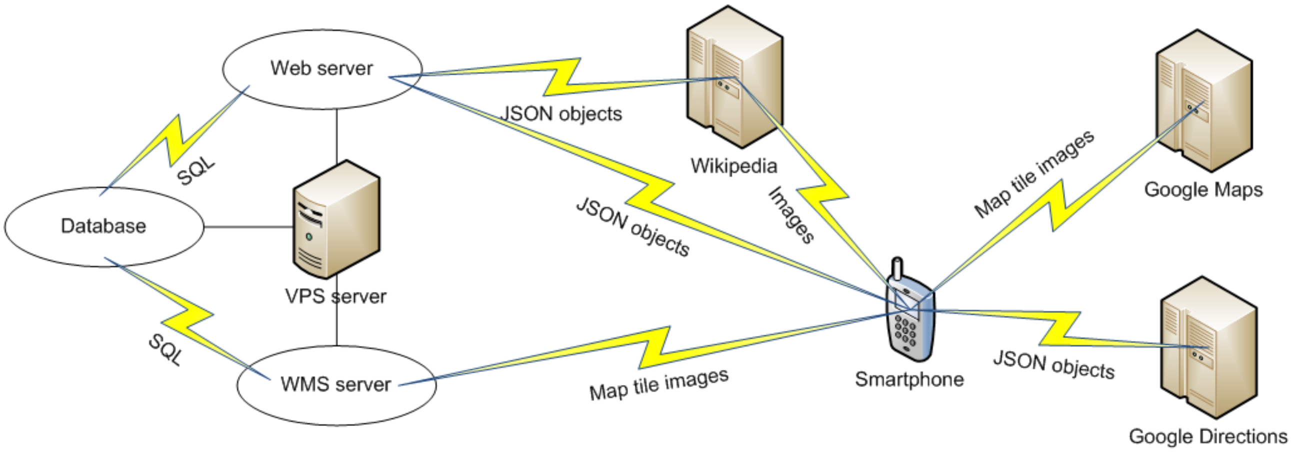


Figure 33 The different components and the communication links possible in the developed system. The Google Directions component was not implemented in the last prototype.

The framework of the system which can be seen in Figure 33 consists of a smartphone as frontend, a virtual server as backend and a number of services from Wikipedia and Google. The frontend is a native Android smartphone app that makes use of map data from Google Maps, images from Wikipedia and additional data provided from the virtual server. The frontend was also supposed to use Google Directions for directions between a user's position and the nearest cemetery but this was not implemented in the app because the fourth iteration was never run. The backend is a virtual server running a PostgreSQL database with the PostGIS extension to handle spatial data queries. The map data is built as a WMS delivered through GeoServer based on grave polygon objects from the database. A web server running on the virtual server process requests from the smartphone both to the internal database and for text from Wikipedia. The connections between the frontend the other components will be implemented through the HTTP protocol and text data will be transferred as JSON objects.

To get an understanding of the complexity of the system, Figure 34 can be viewed. Here the amount of codes written, along with its distribution between the different types of scripts is illustrated. A total of 4803 lines of code has been written in addition to this, about 600 lines of comments. Because the total number of lines of code is so vast, the code has not been included in the report. Instead examples of each type of script can be seen in Appendix F, while the precise number of lines within each category can be seen in Appendix G. Additionally the scripts containing all of the code and the app itself can be found on the CD in Appendix H.

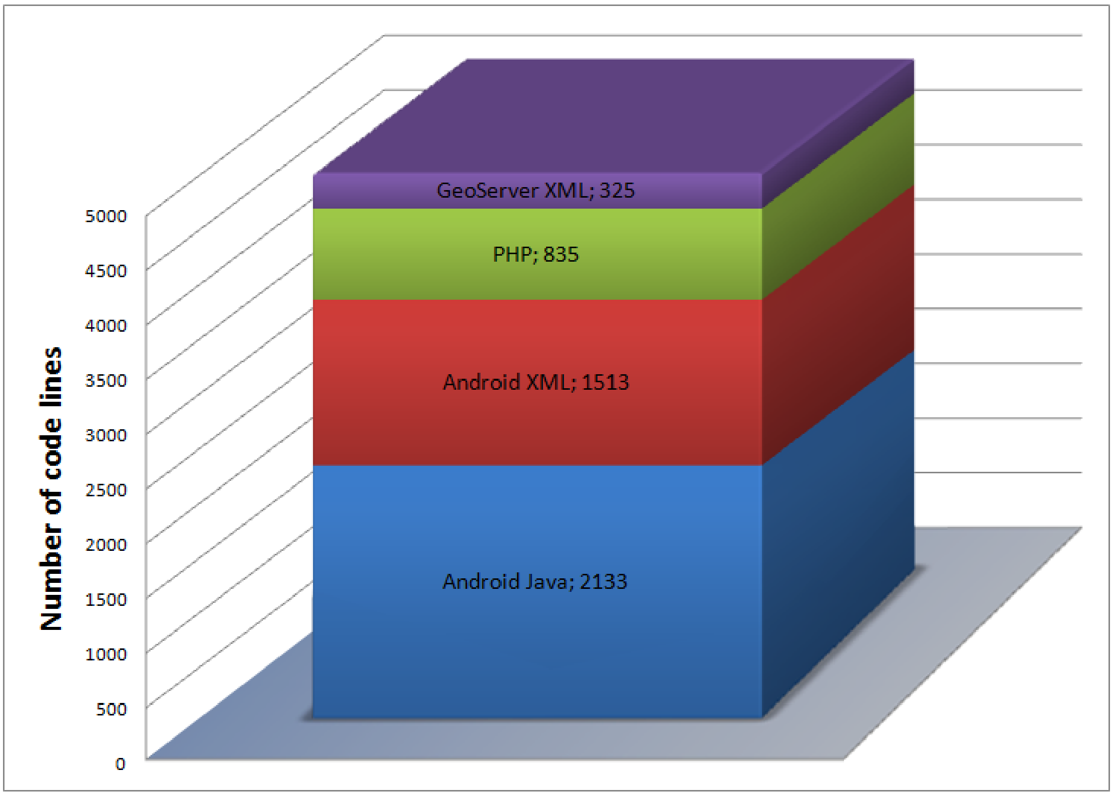


Figure 34 The number of code lines that has been written divided between the different programming languages and scripts.

9.3 THE INTERNAL STRUCTURE OF THE SYSTEM COMPONENTS

With the overall system design of the different components explained, it is time to take a look at how these individual components are structured. What files and resources were developed for the components and how is the relation between the main functions shown in the wireframe and these different files and resources?

Smartphone

The frontend of the developed system is an Android application for smartphones. The environment is based on a number of files, which can be Java, XML or images. The Java files contain the object oriented java code which defines the fundamental behaviour of the app. Graphic elements of an Android application can be defined through the XML files and images used for these graphical elements are contained in a number of images JPG, PNG BMP etc.

The code of an Android application is structured around a number of activities. An activity is an independent function of an app that is designed to interact with the user. When structuring an app around activities many tasks are done automatically by the development environment making it less resource demanding for the developer to program the app. Such task is the graphical layout of the screen view, the navigation between activities using back button and turning off the GPS when switching to another activity. The activities along with other key functions of the app are defined in the AndroidManifest.xml file.

Java code is structured around classes which define objects. An activity is one example of an object defined in a class. The code of classes can be shared and used by several activities. TileProviderfactory.java and WMSTilerovider.java is examples of files containing classes that are used by both Map activity and the FindCemetery activity.

In the developed app there are five activities as seen in Figure 35, the layout of the activity is defined in the corresponding activity XML style file, for the Main activity the style file is for instance called activity_main.xml .The layout can also be changed dynamically, also called programmatically layout, which is used for the Map activity. Additional styling files can be used by the activities and the activities can share the styling files. The styling file list.xml is both used by the FindGrave activity and the TourPlanner activity.

The code in front of the XML styling file corresponds to the code in the wireframe diagram of the app shown in Figure 32. The XML style file only contains layout information, the actual code where the actions are implemented is written in the Java files as seen above the XML files in Figure 35.

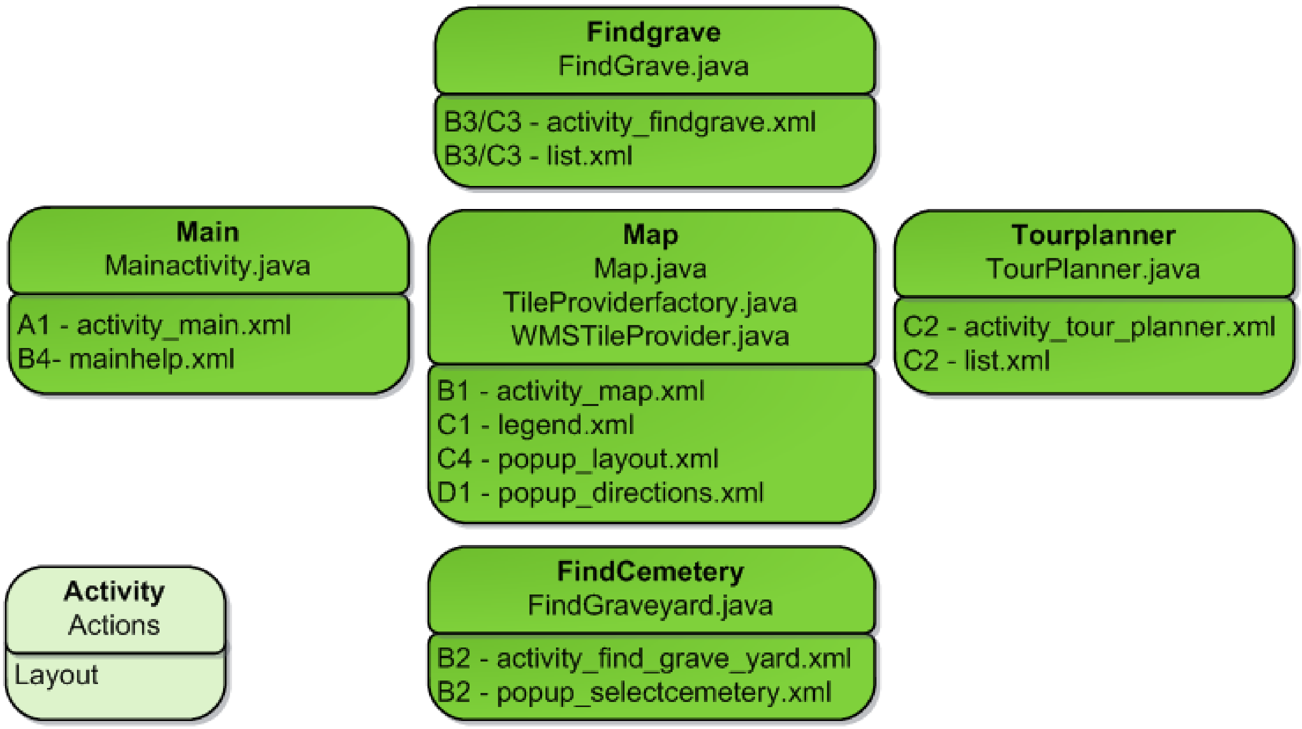


Figure 35 The five activity elements that the app is composed of. In general the Java files contain the actions of the app while the XML files define the layout of the app. The alphanumeric code corresponds to the wireframe diagram of the functionality.

WMS

The WMS overlay map is generated using GeoServer. GeoServer publish spatial data stored in the database using XML styling files. Three different WMSs have been developed, each one using its own separate styling file. The three files are shown in Figure 36 with corresponding code for the wireframe diagram of the app.

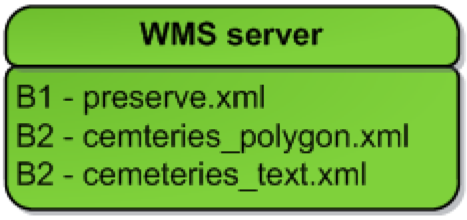


Figure 36 XML style files used for WMS.

Web server

The web server controls the execution of the PHP scripts which are the communication bridge between the smartphone and the database and Wikipedia. The developed PHP scripts and their corresponding wireframe diagram code can be seen in Figure 37.

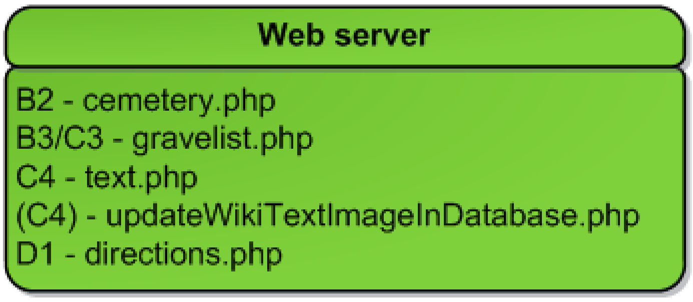


Figure 37 PHP scripts executed on the webserver on basis of connection from the smartphone.

Database

Most of the data used by the app is stored in a database on the server controlled by the project group. The data representation in the database is essential a model of the real world. The process of designing the database consisted of defining three underlying representation of the real world. The structure is as shown in Figure 38:

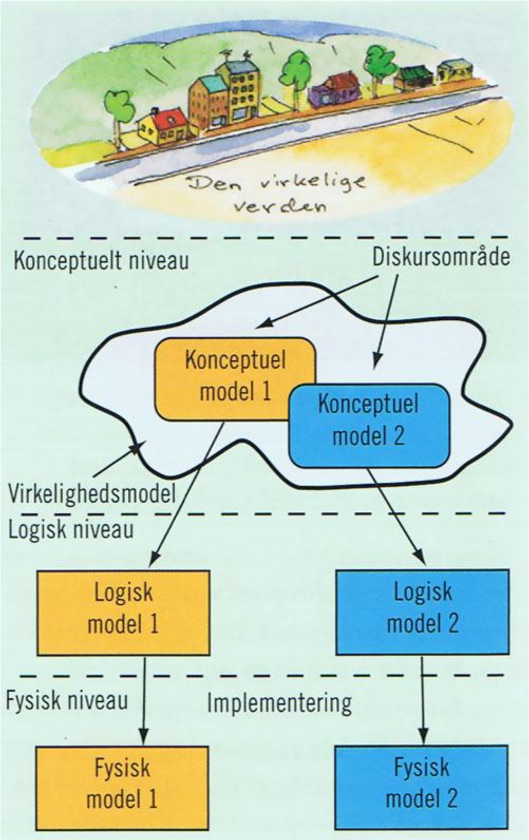


Figure 38 The structure of getting data from the real world into a database. (Balstrøm, Jacobi and Bodum 2006)

The conceptual model is an interpretation of how the real world can be generalised using standardised diagrams. The logic model is a data representation based on the data model used for the specific database that will be used for the implementation. The last level is the physical level where the precise implementation is done like which data types to use for the different data to store in the database. (Balstrøm, Jacobi and Bodum 2006, p 140)

When dealing with relational databases it is normal to use an Entity-Relationship (ER) diagram to show the representation of the conceptual level. This diagram consists of entities which are objects like the cemetery and the persons buried there. Between these objects there are relations which are shown in the diagram. The diagram also includes information about the different entities in a number of attributes to each of them. Since a spatial enabled relational database is used for this project, attributes can also be geographical objects. This simplifies the process of including information about a road. Instead of having attributes defining length, start- and end point it is now possible to just having one attribute including all of the geographical representation. In Figure 39 geographical attributes are shown as polygons, lines or points. (Balstrøm, Jacobi and Bodum 2006, p 145)

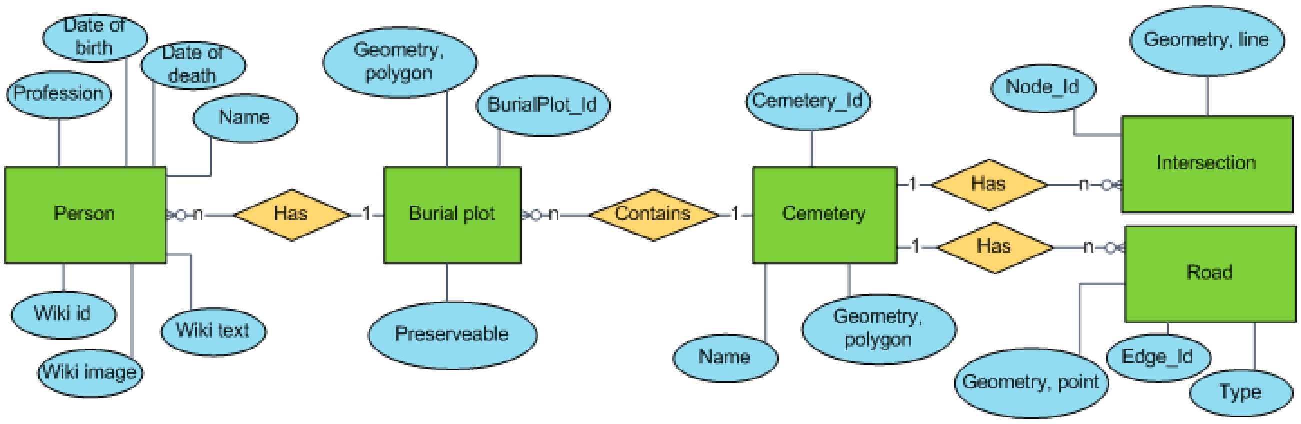
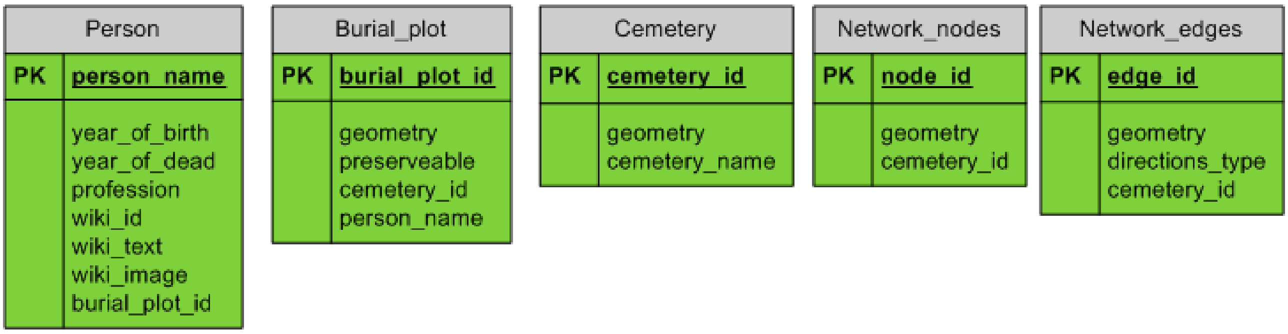


Figure 39 The conceptual model shown as an ER diagram. The squares are the entities, the parallelograms are the relations and the ellipses are the attributes including spatial objects.

With the conceptual model in place it is possible to continue with the next level the logical model. This model is based on the data model of relational databases but not on any physical restrictions for the specific used database. Here the tables are taken from the ER diagram as the entities and a field is defined for each attribute connected to the entity. Further primary keys are chosen and the relations between the entities in form of a number of foreign keys. (Balstrøm, Jacobi and Bodum 2006, pp 146-147)



Person	
PK	person_name
	year_of_birth year_of_dead profession wiki_id wiki_text wiki_image burial_plot_id

Burial_plot	
PK	burial_plot_id
	geometry preserveable cemetery_id person_name

Cemetery	
PK	cemetery_id
	geometry cemetery_name

Network_nodes	
PK	node_id
	geometry cemetery_id

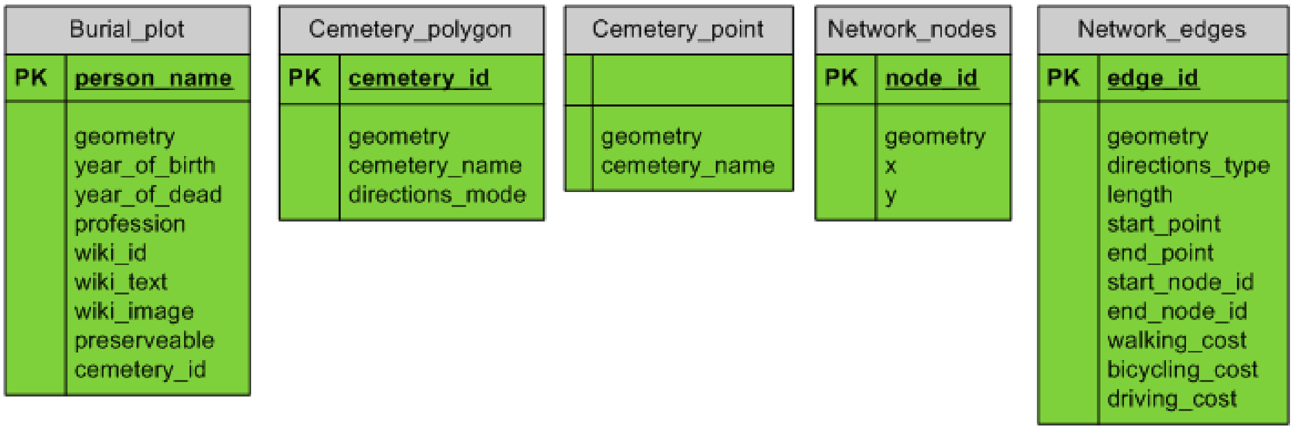
Network_edges	
PK	edge_id
	geometry directions_type cemetery_id

Figure 40 The logic model of the database structure.

The actual implementation defines the physical model. Here it is necessary to take any restrains the specific database has on the data model into account. It is also necessary to look at the overall efficiency. The conceptual model shows a generalisation of the real world, but this might not be the representation which will be easiest to implement. This project focuses on a digital guide for historical persons and even though several persons are buried at one burial plot, it is often only one of these persons who is historically interesting. To simplify the work of registering the graves in the database, it was decided to join the person and burial plot tables into only one. This has the limitation that if two or more historical persons are buried at the same burial plot only one can be found. The ideal logic model is shown in Figure 40 and in Figure 41 the actual used logic model is shown.

One of the functions of the app is to show the name of a cemetery when the user has zoomed into a specific level on the map of Denmark. This is done through the use of a WMS through GeoServer but because of the way that the WMS consists of a number of small tiles, the name sometimes becomes displaced twice. To overcome this problem, the name is based on a point dataset for each cemetery and the additional table *Cemetery_point* is therefore necessary. The only time this table is accessed is through GeoServer that uses the geographical index to identify where the name should be drawn on a map. Therefore it is not necessary to provide a primary key for the different cemeteries in this table.

The implemented internal routing functionality at the cemeteries is done using the pgRouting library, which is extending the functionality of the database to include SQL direction queries. PgRouting uses the two tables *Network_nodes* and *Network_edges* and there are a number of requirements for the fields of these tables. The nodes of the network must be represented by x and y coordinates. The different edges must have a cost field for the impedance. Because this project works with both driving, bicycling and walking modes, the cost for each of these modes must be included, as well as the type of the road and length. The routing queries are also depending on the start and end point and node for each edge in the network.



Burial_plot	
PK	person_name
	geometry year_of_birth year_of_dead profession wiki_id wiki_text wiki_image preserveable cemetery_id

Cemetery_polygon	
PK	cemetery_id
	geometry cemetery_name directions_mode

Cemetery_point	
	geometry cemetery_name

Network_nodes	
PK	node_id
	geometry x y

Network_edges	
PK	edge_id
	geometry directions_type length start_point end_point start_node_id end_node_id walking_cost bicycling_cost driving_cost

Figure 41 The logic model of the physical implementation that was used.

To give an overview of the link between the main functions in the wireframe and the individual resources and files developed for this main function a diagram can be seen in Figure 42. This figure can be seen enlarged in Appendix I.

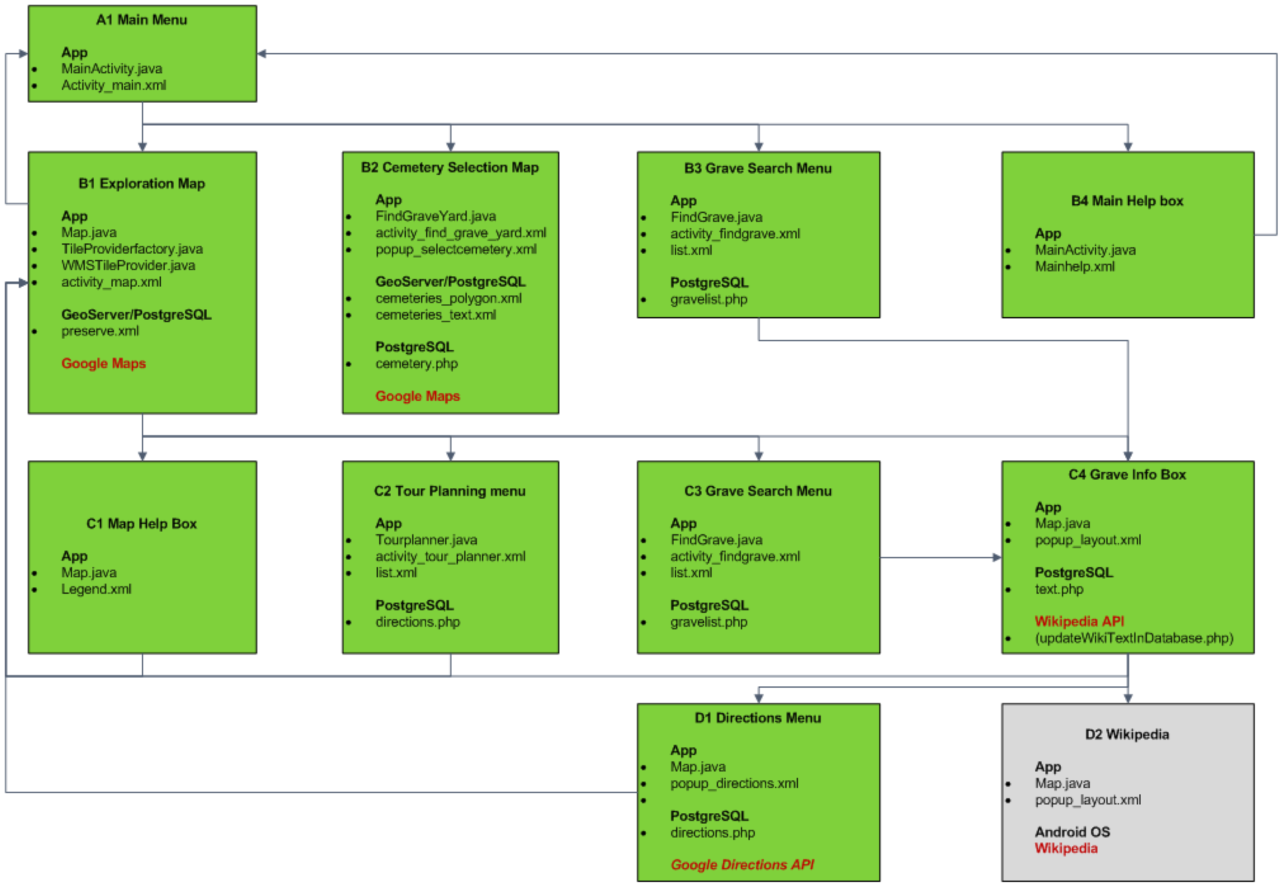


Figure 42 Main functions with used resources and files.

9.4 Key Functions

To provide the reader with an understanding of how some of the key functions included in the app works, this section will explain what services that these functions use, and how the communication is preformed between the involved components in the system.

The total number of functions implemented in the app is vast, so it is of little purpose to give a detailed explanation of all of them, since not all functions are equally interesting in terms of the core purpose of the app. Instead five distinct functionalities of the app have been identified, which presents most of the fundamental elements of the app. These five functionalities are:

- Find nearest cemetery
- Find a person from a text string
- Show a person's grave on a map
- Show information popup box for a person's grave
- Give directions between two points

9.4.1 *FIND NEAREST CEMETERY*

To find the nearest cemetery, a number of tasks must be performed. At first the smartphone will identify its position using GPS, network triangulating or WIFI positioning. If the smartphone's share position option has been disabled, the user will be guided to the location dialog of the smartphone to enable sharing.

With the position found, the smartphone sends this position by a HTTP POST request with key-value formatted data (Figure 43) to a PHP script on the server. The PHP script makes a spatial query to find the closest cemetery within 1km of the smartphone's position. If no cemetery is found the radius is increased to 5km, then 25km, and finally 125km and if no cemetery is found within this radius, the search is stopped. If a cemetery is found, another spatial query is performed that asks for the bounding box containing the cemetery. The PHP script now sends the name of the cemetery, *cemetery_id*, centre coordinates and the directions mode of the cemetery back to the phone as JSON object, see Figure 44.

```
Lon=12.536430&lat=55.701140
```

Figure 43 Cemetery request from smartphone formatted as key-value pairs.

```
{"name":"Assistens Kirkeg\u00e5rd","id":"Assistens",
"graveyardLat":"55.6911091967571",
"graveyardLon":"12.5495926676622",
"directionsMode":"2"}
```

Figure 44 Response from server formatted as a JSON object.

The smartphone will then use the *cemetery_id* to select what WMS to display. The centre coordinate and the position of the smartphone will be used to display the minimum map view that contains these two positions.

9.4.2 *FIND A PERSON FROM A TEXT STRING*

When the user searches for a person's grave, the name searched is sent using HTTP POST request key-value formatted data, Figure 45. In addition to the name, a cemetery id is also sent if the user is searching within a specific cemetery. If the search is general, the cemetery id key will contain the text "all". It is also possible to restrict the search to certain professions which are used by the tour planner, this is done with the *searchProfession* key. If no restrictions are used this key is set to "all" as well.

```
name=Karl Nielsen&cemeteryId=all& searchProfession=all
```

Figure 45 Name request from smartphone formatted as key-value pairs.

The PHP script now makes a query in the database. A fuzzy search is performed in the database, meaning that the persons in the database whose names look the most like the name searched for, will appear at the top of the search list that is sent back to the smartphone. The response from the database is then sent to the smartphone formatted as multiple JSON objects inside an array, see Figure 46.

```
[
{"gravename":"Carl Nielsen",
"cemeteryid":"kbh_vestre",
"dead":"1931",
"profession":"KM",
"graveLat":"55.6602597062387",
"graveLon":"12.5290644026336",
{"gravename":"Georg Nielsen",
"cemeteryid":"Assistens",
"dead":"1797",
"profession":"V",
"graveLat":"55.6905669141299",
"graveLon":"12.5527995789289"},
...
]
```

Figure 46 The search for graves response from server formatted as a JSON object. Only the first two graves are shown here the actual response contains twenty graves.

The received data is presented on a list on the smartphone displaying the name of the deceased and the year of death. Each item on this list has a function attached and when activated, the additional information cemetery id, profession and position of the grave will be used. The use of this particular information is identical to some of the functionality described in the next two key function examples, and due to this, these operations will not be explained any further here.

9.4.3 SHOW A PERSON'S GRAVE ON A MAP

The exploration map view in the app consists of two layers. The first layer is Google Maps, Figure 47, whose functionality is implemented through Java code in the smartphone. The implementation of Google Maps has the option to add overlays on top of its own map. The second layer is the layer containing the different graves within a specific cemetery, Figure 48. This overlay is created through a WMS and after further processing on the phone, it is possible to display this on top of the Google Maps layer, Figure 49. The smartphone requests for the WMS map tiles by sending a URL with information of what cemetery, projection type and bounding box to retrieve the tiles, Figure 50.

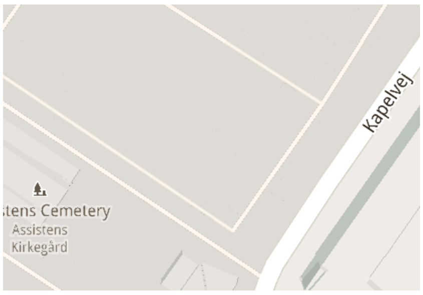


Figure 47 Google Map layer.

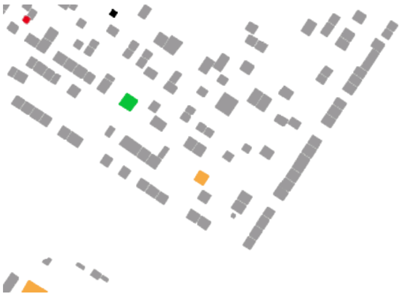


Figure 48 Graves WMS layer.

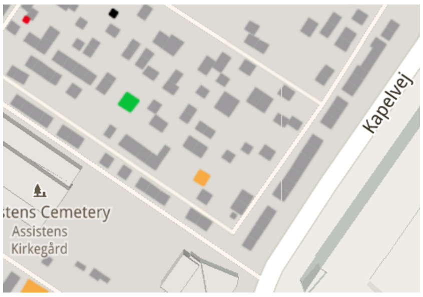


Figure 49 The final map shown in the smartphone.

Based on the profession that the deceased used to have, the graves are given a colour. It takes a lot of processing power each time GeoServer has to generate the necessary map tiles, and using a server with limited processing power therefore causes delay, before the user receives the tiles. Since the same map tiles are used over and over again, it is much more efficiently to pre generate the tiles on the server. This means that when the user requests for certain parts of the WMS, GeoServer does not generate any new tiles, but instead sends the already generated tiles.

```
http://SERVER_IP:8080/geoserver/gwc/service/wms?
service=WMS&version=1.1.0&
request=GetMap&
layers=grave:CEMETERY_ID&
styles=&
bbox=1397421.751465,7496637.988006,1397574.625522,7496790.862063&
width=256&
height=256&
srs=EPSG:900913&
format=image/png&
transparent=true
```

Figure 50 Example of the structure of WMS map tile request URL for pre generated tiles.

9.4.4 SHOW INFORMATION POPUP BOX FOR A PERSON'S GRAVE

The information popup box is shown, either if the user clicks on the exploration map near a grave or when the user performs a search for a deceased and clicks on the name at the list. The process of showing the pop up box is basically the same in these two examples; a position is sent to server to find the nearest grave. The only difference is that the position used if the user did a name search will be the exact position for the grave which already was known, while in the case where the user clicks on the map, the position will be where ever the map was clicked on.

Whichever method is used, this will result in a position and the id of the cemetery will be sent as key-value pairs to a PHP script on the server, as seen in Figure 51.

```
lon=12.5290644026336&lat=55.6602597062387& cemetery=kbh_vestre
```

Figure 51 Position request for information popup box from smartphone formatted as key-value pairs.

In the PHP script, a spatial query is made asking the database for the closest grave within 10m of the position clicked. If no grave is found, the search radius multiplied by five, until the final radius exceeds 6.25km. If a grave is found, the name, year of death, profession and the position of the grave are retrieved and if available, the Wikipedia id as well. The Wikipedia id is a unique text string which identifies an article. This id will be used to extract text and image URL from Wikipedia's server.

The text for a Wikipedia article can be retrieved by using an URL as shown in Figure 52. In This example the first 240 characters of an article are extracted. This response can be formatted in several ways, but the JSON format response used in this project is shown in Figure 53. The extracted text contains HTML formatting codes and other unwanted information like small icons. This information is removed before sending the information back to the smartphone.

```
http://da.wikipedia.org/w/api.php?
action=query&
prop=extracts&
exchars=240&
titles=H.C._Andersen&
format=json
```

Figure 52 Example of a Wikipedia API request for the Danish article of H. C. Andersen.

```
{
    "query": {
        "normalized": [
            {
                "from": "H.C._Andersen",
                "to": "H.C. Andersen"
            }
        ],
        "pages": {
            "12872": {
                "pageid": 12872,
                "ns": 0,
                "title": "H.C. Andersen",
                "extract": "<dl>\n<dd>[Disambig bordered fade.svg] <i>Der er
                 flere personer med dette navn, se H.C. Andersen (flertydig).
                </i></dd>\n</dl>\n<p><b>Hans Christian Andersen</b> (2.april
                 1805 i Odense \u2013 4. august 1875 p\u00e5 Rolighed i
                 K\u00f8benhavn) var en dansk digter og forfatter"
            }
        }
    }
}
```

Figure 53 JSON formatted response for a Wikipedia article extract.

The retrieval of the image URL requires two steps, at first an image id must be found before the actual image URL can be retrieved in second step. In Figure 54 the request for images for an article is shown. The response is shown in Figure 55.

```
http://da.wikipedia.org/w/api.php?
action=parse&
page=H.C._Andersen&
prop=images&
format=json
```

Figure 54 Example of a Wikipedia API request for images of the Danish article for H. C. Andersen.

Wikipedia articles contain other image files than portraits of the person in question, these are often svg and png files, see Figure 55. Since jpg file format is most suitable for portraits the first jpg image is expected to contain an image of the person. If the article contains an image of the person, it is almost always the first image and the first jpg file is therefore used.

```
{
    "parse": {
        "title": "H.C. Andersen",
        "images": [
            "Padlock-silver-light.svg",
            "HCA_by_Thora_Hallager_1869.jpg",
            "Disambig_bordered_fade.svg",
            "HansChristianAndersen_House.jpg",
            ...
        ]
    }
}
```

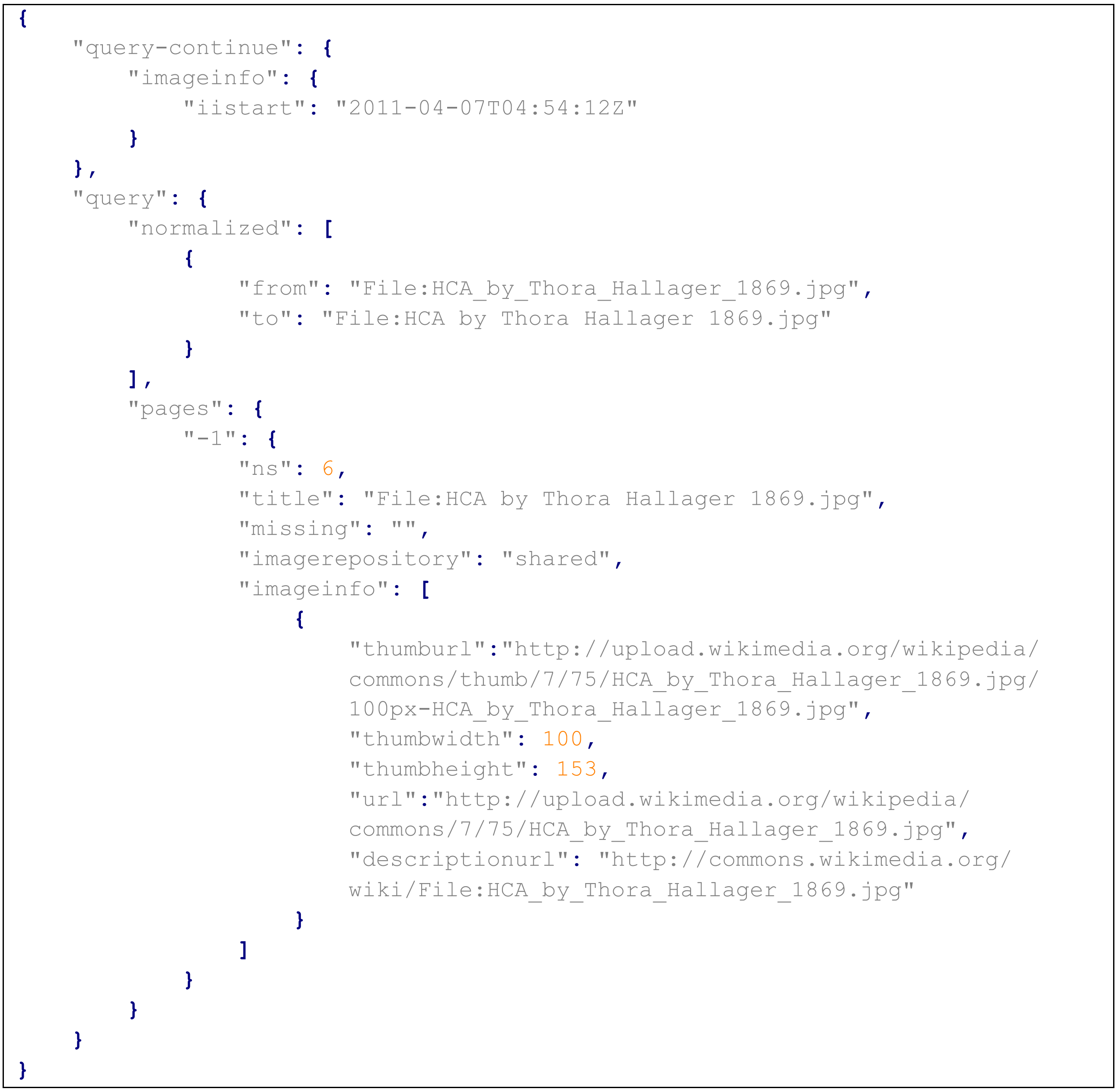
Figure 55 List of images used in the Danish article for H.C. Andersen. (only the first four images is shown here)

The file name of the image is not enough to form an actual URL and another request has to be made. This request concerns specific information about the image file. This request is shown in Figure 56.

```
http://en.wikipedia.org/w/api.php?
action=query&
titles=File:HCA_by_Thora_Hallager_1869.jpg&
prop=imageinfo&
iilimit=1&
iiprop=url&
iiurlwidth=100&
format=json
```

Figure 56 Example of a Wikipedia API request for images information of a specific image file.

This request results in another JSON object response as shown in Figure 57. Here at last the image URL can be gathered and used later on. It is actually possible to have the Wikipedia server to resize the image and in this example the image is resized to a width of 100 pixels.



```
{
    "query-continue": {
        "imageinfo": {
            "iistart": "2011-04-07T04:54:12Z"
        }
    },
    "query": {
        "normalized": [
            {
                "from": "File:HCA_by_Thora_Hallager_1869.jpg",
                "to": "File:HCA by Thora Hallager 1869.jpg"
            }
        ],
        "pages": {
            "-1": {
                "ns": 6,
                "title": "File:HCA by Thora Hallager 1869.jpg",
                "missing": "",
                "imagerepository": "shared",
                "imageinfo": [
                    {
                        "thumburl":"http://upload.wikimedia.org/wikipedia/
                        commons/thumb/7/75/HCA_by_Thora_Hallager_1869.jpg/
                        100px-HCA_by_Thora_Hallager_1869.jpg",
                        "thumbwidth": 100,
                        "thumbheight": 153,
                        "url":"http://upload.wikimedia.org/wikipedia/
                        commons/7/75/HCA_by_Thora_Hallager_1869.jpg",
                        "descriptionurl": "http://commons.wikimedia.org/
                        wiki/File:HCA_by_Thora_Hallager_1869.jpg"
                    }
                ]
            }
        }
    }
}
```

Figure 57 Response for an image information request sent to Wikipedia API.

The information is then sent from the PHP script, formatted as JSON object, see Figure 58.

The process alone of acquiring the information from Wikipedia takes 2-3 seconds and this delay could be removed by storing the information in the database on the server. Because of this it was decided for the current implementation of the system to have the text and image URL stored in tables of the database which will be fetched instead of the wiki id of the person and used directly in the PHP script. A PHP script is instead written to update the content in the database from Wikipedia if necessary.

```
{"text":"\nCarl August Nielsen (9. juni 1865 i Sortelung, N\u00f8rre Lyndelse,
p\u00e5 Fyn \u2013 3. oktober 1931 i K\u00f8benhavn) var en dansk komponist og
dirigent...",

"imageurl":"http:\/\/upload.wikimedia.org\/wikipedia\/commons\/thumb\/f\/f7\/C
arl_Nielsen.jpg\/100px-Carl_Nielsen.jpg",

"name":"Carl Nielsen",
"dead":"1931",
"profession":"KM",
"graveLat":"55.6602597062387",
"graveLon":"12.5290644026336"}
```

Figure 58 The search for graves response from server formatted as a JSON object. (only a small part of the text value is shown)

On the smartphone a popup box is opened with the text and the URL for the image is used to retrieve an image directly from Wikipedia. A marker is placed on the position of the grave, with the colour matching the profession type, and the map view is animated to move the map so the marker on the screen is located just under the popup box.

9.4.5 GIVE DIRECTIONS BETWEEN TWO POINTS

The directions generated internally of a cemetery, are created by sending information about start and end position, the cemetery id, the desired directions mode e.g. walking/bicycling/driving and final information about if the directions are part of a tour from the tour planner. Example of a request can be seen in Figure 59.

```
{"startLon":"12.55276",
"startLat":"55.689621",
"endLon":"12.5525171589379",
"endLat":"55.6901410445725",
"cemeteryId":"Assistens",
"isTourPlanner":"0",
"mode":"walking"}
```

Figure 59 Example of a directions request send to the server.

The PHP script handling the directions task makes numerous queries to the database, to generate the final route. The most important of these queries, is the one that finds the shortest path between two nodes based on the pgRouting Dijkstra's algorithm. However, the problem of finding the shortest route between two positions is much more complicated than just finding the one between two nodes. In the example in Figure 60 there are four possible routes between point 1 and the grave at point 2. The pgRouting library provides the length between the nodes, but here it is also necessary to know the length from the position (1) to the nearest point on the nearest edge, and from this point to the two possible nodes. But what if the user is asking for direction between point 1 and 3? Most likely the user will not do this because the distance is so short, but the directions should be able to handle this. In this case, the shortest route will not be to one to the nearest node. In general the directions given try to find a route that minimise the length needed for walking outside the roads. This is done by only giving directions directly to the grave, if the direct distance is smaller than the distance to the nearest node.

The last property that also needs to be taken into account is which directions mode was chosen and if bicycling or driving is allowed on the particular edge.

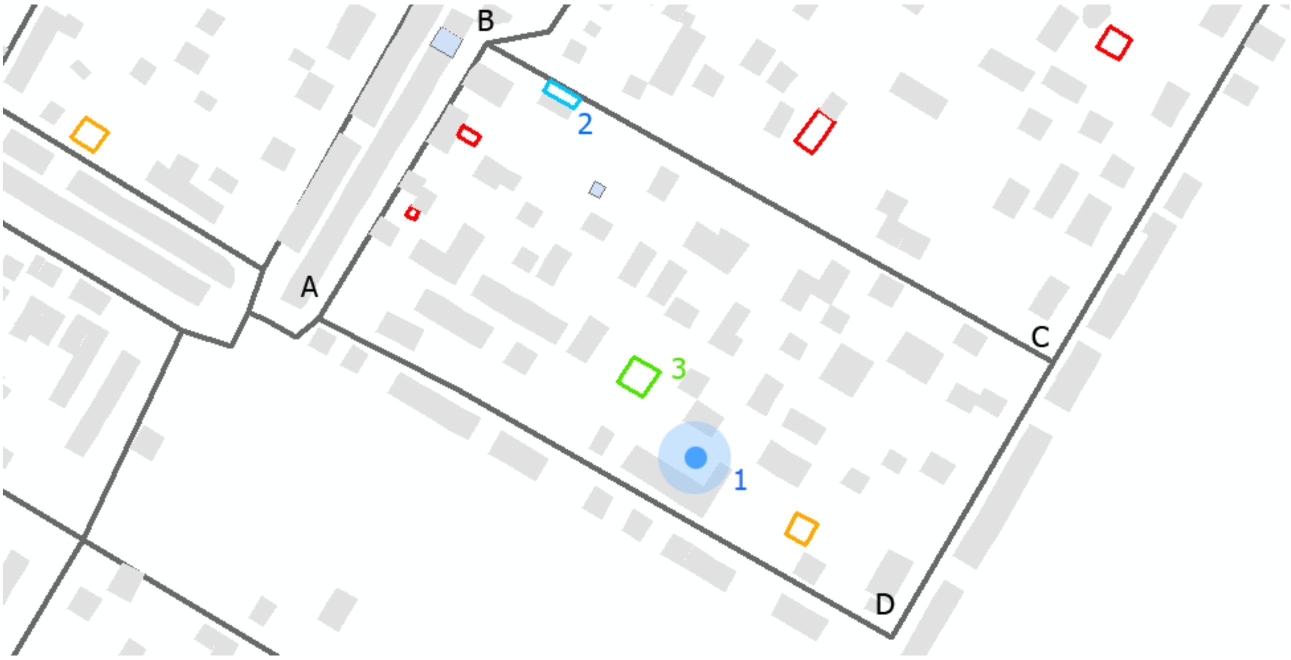


Figure 60 Network example of edges (ab, bc, dc and ad) and nodes (A, B, C and D).

All of these problems require quite many spatial queries to be solved, but when the fastest route is found, every line segments of the route is sent to the smartphone with information about start/end point, length in meters, cost will be walking=length, bicycling=length/4 and driving=length/5. Example of a direction response to the smartphone can be seen in Figure 61.

```
[
{"lat1":"55.689621538350",
"lon1":"12.5527538308906",
"lat2":"55.6896478225003",
"lon2":"12.5527905538307",
"type":"walking",
"cost":"3.55188838212894",
"length":"3.55188838212894"},
{"lat1":"55.6896478225003",
"lon1":"12.5527905538307",
"lat2":"55.6899130814011",
"lon2":"12.5520310482869",
"type":"walking",
"cost":"56.1709041084521",
"length":"56.1709041084521"}
...
]
```

Figure 61 Example of a direction response formatted JSON object, containing two line segment from the server.

When the smartphone receives the JSON object, it will be parsed, and every line segments are drawn on the map with different colours depending on whether it is a walking path, bicycling path or roads allowed for driving, see Figure 62.

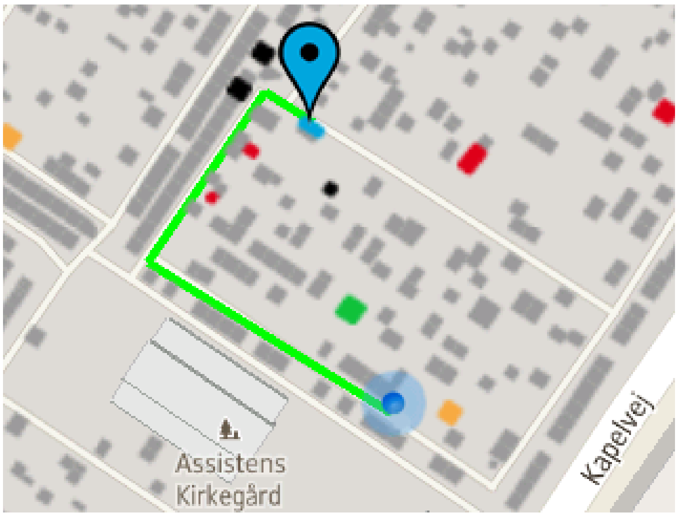


Figure 62 The final directions route drawn on the map.

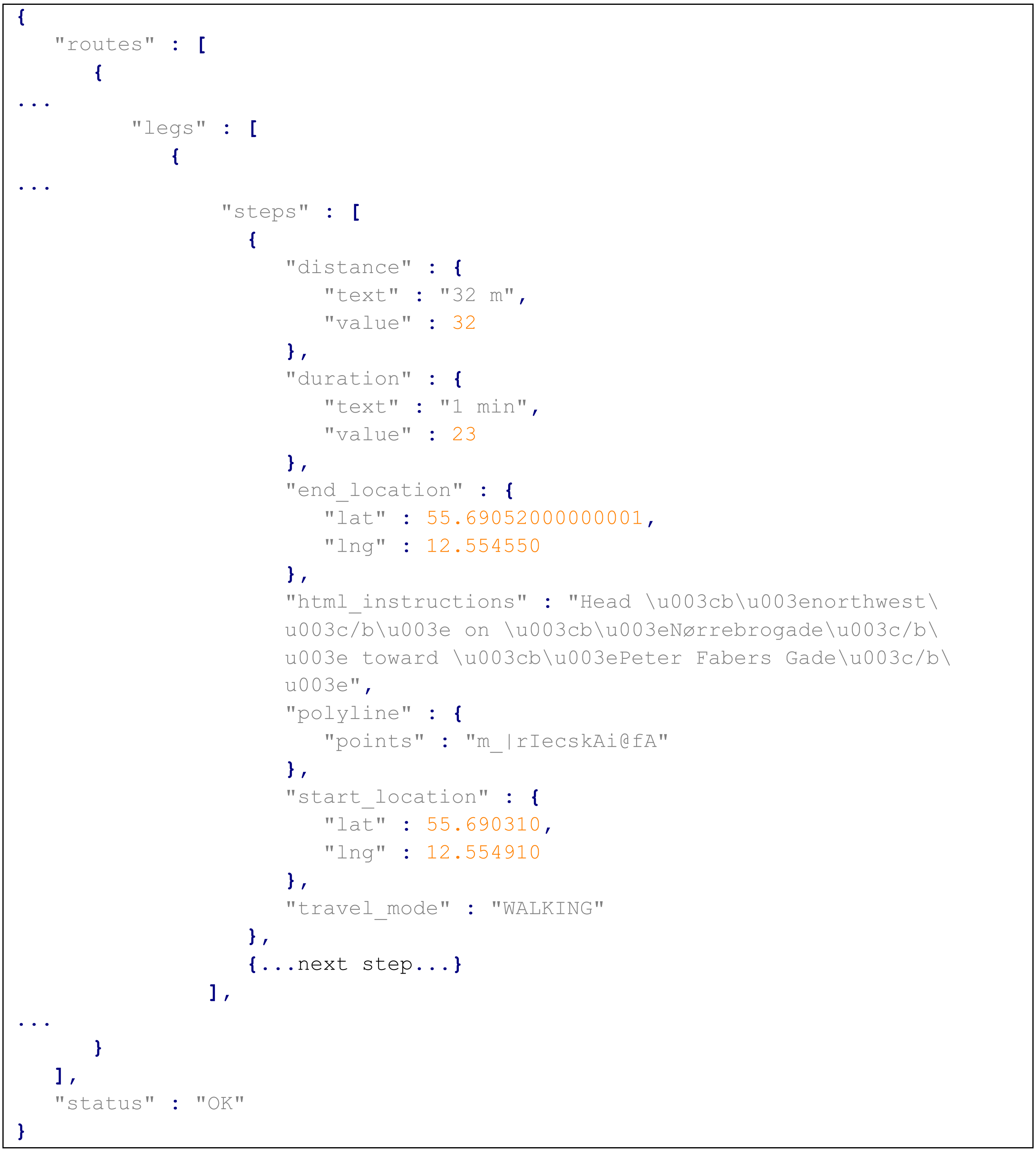
It was also planned to implement directions to a cemetery if the user was not yet at the cemetery using Google Directions API, but due to lack of time this was not implemented, but a description of how it could be implemented is presented here.

Google Directions API gives developers access to the routing functionality known from Google services like Google Maps. A request is made using a URL like the one in Figure 63, with origin, destination and directions mode.

```
http://maps.googleapis.com/maps/api/directions/json?
origin=55.690307,12.554905&
destination=55.690392,12.554305&
sensor=false&
units=metric&
mode=walking
```

Figure 63 Example of Google Direction route request between two positions.

The response from Google Directions is JSON formatted. This response includes much information, the start and end position of the route, how long the route is in total, how long time it will take to travel and the individual segments of the route. These segments are called steps, which includes information about length, start and end point among others, see Figure 64.



```
{
   "routes" : [
      {
...
         "legs" : [
            {
...
                "steps" : [
                  {
                     "distance" : {
                        "text" : "32 m",
                        "value" : 32
                     },
                     "duration" : {
                        "text" : "1 min",
                        "value" : 23
                     },
                     "end_location" : {
                        "lat" : 55.69052000000001,
                        "lng" : 12.554550
                     },
                     "html_instructions" : "Head \u003cb\u003enorthwest\
                     u003c/b\u003e on \u003cb\u003eNørrebrogade\u003c/b\
                     u003e toward \u003cb\u003ePeter Fabers Gade\u003c/b\
                     u003e",
                     "polyline" : {
                        "points" : "m_|rIecskAi@fA"
                     },
                     "start_location" : {
                        "lat" : 55.690310,
                        "lng" : 12.554910
                     },
                     "travel_mode" : "WALKING"
                  },
                  {...next step...}
               ],
...
      }
   ],
   "status" : "OK"
}
```

Figure 64 Example of JSON formatted response from Google Directions. Only the first direction step is shown and other parts of the response has also been removed.

Instead of having the PHP script to request the database for directions as it was used internally for the cemeteries, it would be possible to request directions from Google Directions. Some adjustments would be necessary in the PHP script and the code for the app itself, but it would not be very difficult before this functionality could be implemented. It is simply a question of parsing the response JSON object, like the response from Wikipedia, and change the directions to the form which was used for the internal cemetery directions.

10 RISK ANALYSIS PHASES

When a prototype had been built during an iteration's implementation phase, a risk analysis was performed. During the first two iterations the risk analysis was performed in order to assess if would be possible for the project group to develop the app with the chosen technology and platform, or if it was too risky to continue, meaning that a less demanding development technology should be used in the following iterations. The risk analysis phases during iteration three on the other hand were not used to assess if a new technology should be used. Instead this phase was used to provide feedback for the further process, and to decide how much further the project would go.

As mentioned, the development period was structured around a number of prototypes and iteration cycles as seen in Figure 65. The purpose of the individual iteration was both to act as risk analysis, milestone, as well as to provide feedback for the development process.

Development Cycles			
	Prototype 1 (internal)	**Prototype 2 (presentation seminar)**	**Prototype 3 (first test with real users)**
Iteration time	2 weeks	3 weeks	4 weeks
Functionality category	Framework	Basic	Advanced

Figure 65 The different prototypes and their iteration time.

10.1 PROTOTYPE 1

The purpose of the first prototype was to review the chosen development environment. Would it be realistic to implement the functionality in a native app or should the project use a different technology? After two weeks of development, a native app was produced that basically only could show a map with the position of a number of graves, by integrating a GeoServer WMS into a Google Maps on the smartphone. This progress was given ½ an implementation point, see Figure 66, which was not much, but enough to meet the requirement for this iteration. The risk of failure with the rest of the functionalities was certainly there, but the risk analysis concluded it would likely be possible to develop at least the basic functions and possibly also some of the advanced functionality, and therefore the implementation through native app programming continued.

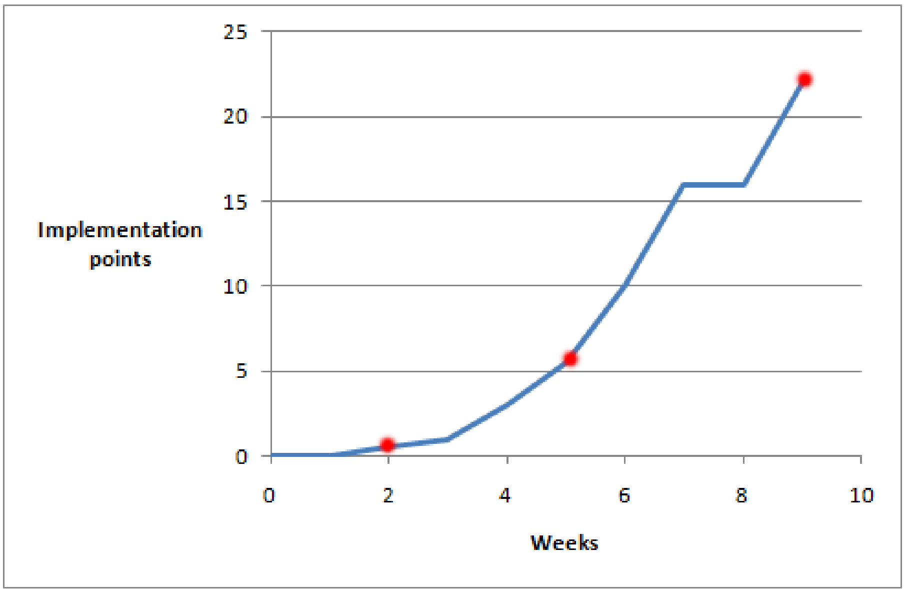


Figure 66 The relationship between development time and implementation points.

10.2 Prototype 2

The purpose of the second prototype besides marking the second risk analysis, was also to provide feedback for the desired functionality of the app. Concerning the risk analysis; 5 ½ points were given. The development phase was halfway through, so if a change of technology was necessary, a lot of resources would have been wasted. However, it would still be possible to make a product, which could illustrate the concept of the desired app by using a less demanding development environment.

Some of the functionalities at this point were implemented, not without problems, but a working prototype with limited functionality was produced. By looking at the total ratio between time and implemented functionalities, not many new functions could be predicted to be implemented in the remaining development period. On the other hand starting from scratch with a new technology would probably also results in few functions to be implemented in this new system. There was tendency to the ratio between implemented functions and time was changing, see Figure 66, and this slight change was thought to be more pronounced later on since the fundamental problems with understanding the object oriented Java language became fewer and fewer. Based on these observations, it was therefore decided to continue implementing the functionalities in a native app.

10.3 Prototype 3

Prototype 3 was completed at the end of the development period for this project. After this iteration, the number of implementation points was 22, and Figure 66 indicates that the development progress showed exponential trends as expected, and that the operational prototype which the fourth iteration would result in without problems could be built using the chosen technology.

10.4 Operational prototype

The last iteration focusing on the functionality refinement category was never performed, but it was also never expected from the beginning to reach further than the advanced functions. With an additional iteration, the app would be very close to a final product that would be possible to release for public testing with all the desired functionality as seen in Figure 27.

11 PROTOTYPE TESTING PHASES

The final phase of an iteration was the prototype testing. As explained, this project completed three iterations, where these iterations have been concluded by evaluating the prototype developed during that iteration using usability tests. All of these usability tests except the one conducted for prototype 1 were evaluated by performing user tests, which will be elaborated in the following where the three prototypes and their respective usability tests will be presented. Each of the usability tests have been given an individual section. This has been done to display how the usability tests of each prototype affected the further development of the app throughout the project period.

11.1 PROTOTYPE 1

As described earlier, the first proto type was completed two weeks into the programming period, and the purpose of the app was to see if it would be possible for the project group to code and develop the framework for the app in a native environment.

11.1.1 PRESENTATION

As seen in Figure 67, the first prototype of the app was nothing more than a simple screen view displaying a number of features. This prototype consisted of a background map implemented from Google Maps, as well as a WMS displaying all of the graves of historical persons at a cemetery.

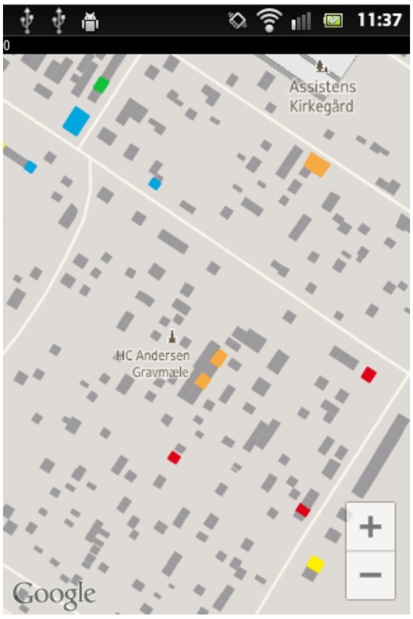


Figure 67 The first prototype map view.

The colours of the graves represented a certain category in which the historical person belonged, but otherwise it was impossible to learn more about a specific grave or the person buried at it, since a grave could not be selected – The app could only display the Google Maps service and a WMS.

11.1.2 Usability Test Design

As explained in the introduction to this section, prototype 1 was the only prototype that was not tested by anyone but the project group members themselves. This means that no actual usability test was conducted on prototype 1, but instead the project group used this prototype to evaluate the risk assessment concerning the likelihood that the project group would not be able to develop an app as specified in the problem statement.

11.1.3 Evaluation

Prototype one did not contain a single one of the functions that were needed in order for the app to meet the identified demands, and as such this prototype was useless. However, prototype 1 contained the framework that was needed for the further development of the app, and the project group had through this prototype proved that building a simple app with limited knowledge was possible.

11.2 Prototype 2

The second prototype developed during the project period was completed 1½ month into the development phase and contained a number of tools that started to make it possible to solve the tasks that the app was being developed for. More specifically this meant that it was possible to search for a grave by typing in a name in a menu, and every grave had an attached info box that could be opened by selecting the specific grave. This functionality can be seen in Figure 68.

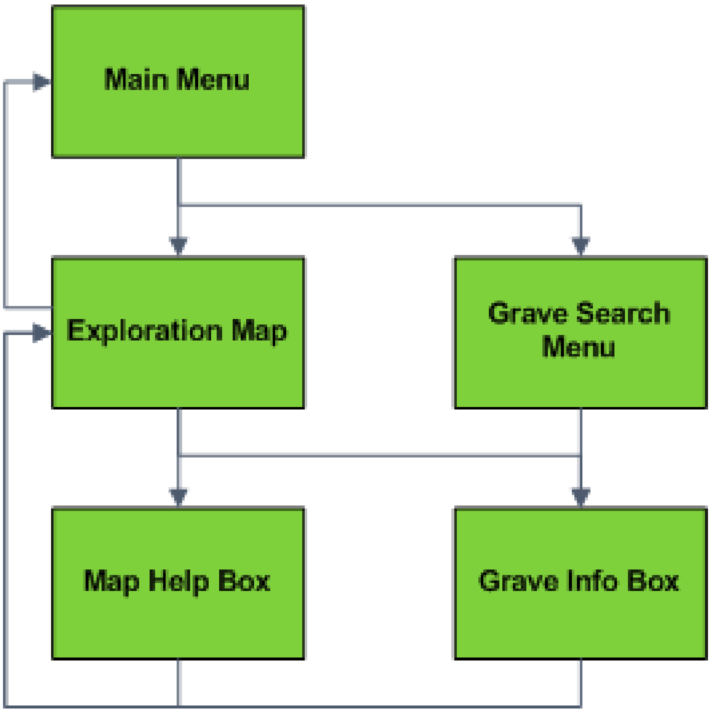
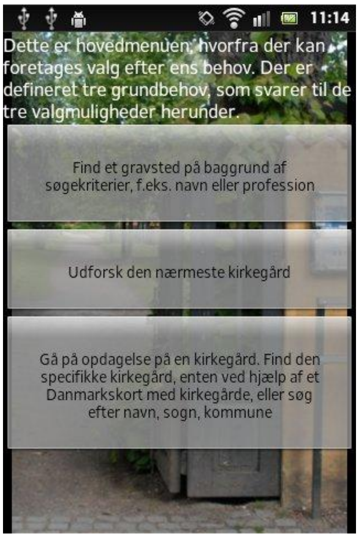


Figure 68 The functionality of the second prototype.

11.2.1 Presentation

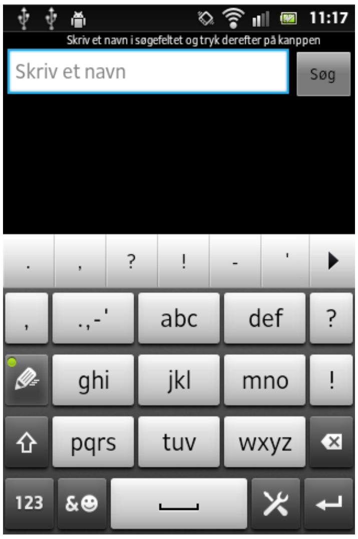
The development of the additional tools meant that the app's start screen view was not a map any more. Instead a Main Menu was implemented as seen in Figure 69, from where the user was supposed to choose one of the three options in the main menu, that each should answer one of the three identified scenarios that a user could find himself in as explained in the User Scenarios section. In prototype 2 however, only data from Assistens Cemetery had been added to the database, and therefore this was the default cemetery that was chosen when the user selected the "Nearest Cemetery" option. Furthermore it was possible to enter the "Search" option, Figure 70, from where the user could type in a name as seen in Figure 71.



Dette er hovedmenuen, hvorfra der kan foretages valg efter ens behov. Der er defineret tre grundbehov, som svarer til de tre valgmuligheder herunder.

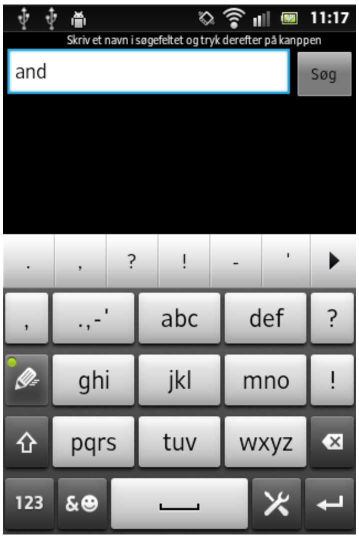
11:14
Find et gravsted på baggrund af søgekriterier, f.eks. navn eller profession
Udforsk den nærmeste kirkegård
Gå på opdagelse på en kirkegård. Find den specifikke kirkegård, enten ved hjælp af et Danmarkskort med kirkegårde, eller søg efter navn, sogn, kommune

Figure 69 The main menu from where it is possible to chose the search option.



Skriv et navn i søgefeltet og tryk derefter på kanppen

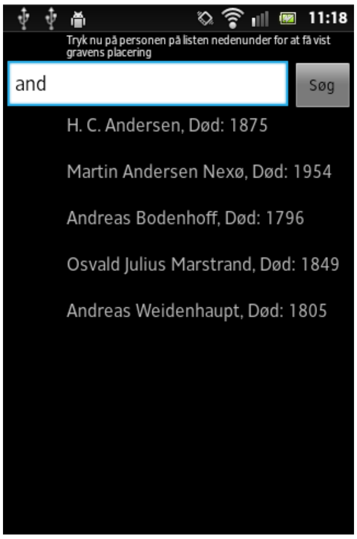
Figure 70 The search function.



Skriv et navn i søgefeltet og tryk derefter på kanppen

Figure 71 The name that is being searched for can be typed in the text field.

When a name had been typed and the search button clicked, a list of the names that matched the entered name best would appear which can be seen in Figure 72. When a name was clicked, the Exploration Map view would appear with an info box opened matching the selected grave. The info box consisted of text from Wikipedia which the user could scroll in, and an image of the person which was always visible in the box. When the box was closed, the Exploration Map would become fully visible, and a new info box could be opened by clicking a new grave or by making a new search. The Exploration Map with and without an open info box can be seen in Figure 73 and Figure 74.



Tryk nu på personen på listen nedenunder for at få vist gravens placering

and

Søg

H. C. Andersen, Død: 1875

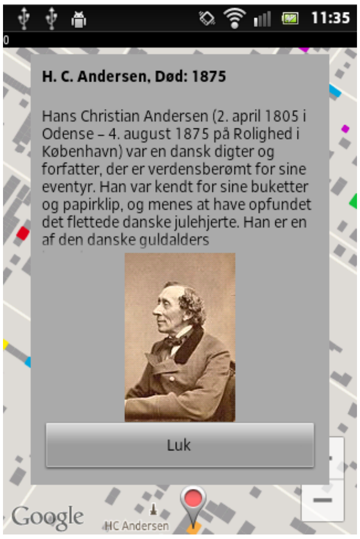
Martin Andersen Nexø, Død: 1954

Andreas Bodenhoff, Død: 1796

Osvald Julius Marstrand, Død: 1849

Andreas Weidenhaupt, Død: 1805

Figure 72 Search results.



H. C. Andersen, Død: 1875

Hans Christian Andersen (2. april 1805 i Odense – 4. august 1875 på Rolighed i København) var en dansk digter og forfatter, der er verdensberømt for sine eventyr. Han var kendt for sine buketter og papirklip, og menes at have opfundet det flettede danske julehjerte. Han er en af den danske guldalders

Luk

Figure 73 The information popup box.

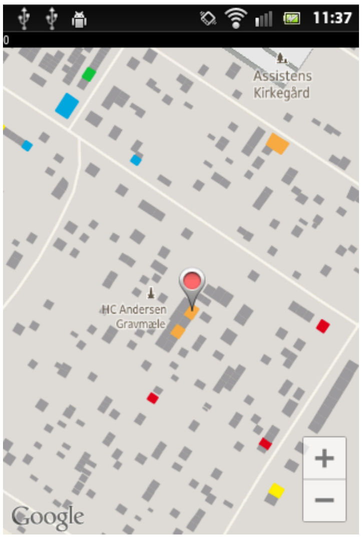


Figure 74 Position of the selected grave.

Additionally when entering the Exploration Map using the "Nearest Cemetery" option, the app wrote a message explaining that the user should click near a grave to receive information regarding the specific grave.

11.2.2 Usability Test Design

The testing of prototype 2 was performed at a status seminar April the 3rd where the app was presented and subsequently evaluated by fellow geoinformatics students as well as the project group's supervisor, by displaying a slideshow with images of the app in use. Furthermore a number of slides displaying future plans for development of the app were presented. The status seminar produced useful feedback from the attendances that was implemented in the third prototype.

11.2.3 Evaluation

The overall review of the prototype was positive. The functionality of the app was accepted as being well considered, and the majority of the comments received concerned the graphical design. Here the design of the pin marking the selected grave was discussed, and the attendances of the seminar concluded that the gray outline of the pin made it difficult to spot when the background was in gray scale as well. The design of the info box was also discussed. Several of the attendances found the scrolling in the box troublesome because the scrolling could only happen by touching the textbox, and not when touching the picture. Furthermore the general opinion was that the picture should not be visible in the info box at all times, but rather be able to be scrolled along with the text. The idea of providing navigation between the user’s position in the cemetery and a grave was also presented at the status seminar and was well received. Some of the attendances would have liked if the navigation functionality also had provided navigation to a cemetery if the user was not already at the specific cemetery. The attendances also suggested that a tour planning tool should be implemented in the app, so that it was possible to either follow a predefined tour, or to create a new tour between several graves. Finally some of the attendances commented that the message when entering the Exploration Map, telling the user to click on the map to open info boxes was unnecessary as this was self explanatory.

11.3 PROTOTYPE 3

The final prototype that was developed during the project period was prototype 3. This prototype was the conclusion of the implementation phase of the project period, which means that it represents all knowledge and skills acquired during the nine weeks of programming. Within those nine weeks the functionality of the app had increased dramatically as explained in the status and limitations section, to a point where almost all of the desired tools and functions of the app were implemented. The graphic design suggestions from the status seminar were put into use, and the text explaining what to do when opening the Exploration Map was removed Furthermore an additional screen view was implemented, consisting of a help box that when opened informs the user of the purpose and use of the app as well as describing how the options in the main menu works. The previously suggested tour planner was also implemented in this version.

Though all of the functions discussed throughout the report was implemented in prototype 3, there were some shortcomings that need to be explained. As described in the wireframe, the idea of the Directions Menu, D.1, was that it would always be available when a grave's info box had been opened. The options regarding the means of transportation on the other hand would change in the menu depending on whether the user was located at the cemetery of a selected grave, or further away. In prototype 3 however, the directions menu button was only made available when the user would be standing at the specific cemetery, while the button will be missing from the info box if the user was located elsewhere. This problem was also present in the tour planner, C.2. Here it was only possible to access the tour planning menu if located in the cemetery that was being explored, and if this was not the case, the tour planning menu button would be missing from the Exploration Map. Finally prototype 3 only detected the user’s position when the Exploration Map had been accessed directly from the main menu. This meant that neither the route tool nor the tour planner would be accessible if a cemetery's exploration map had been opened through the Search or Select Cemetery option in the main menu.

11.3.1 PRESENTATION

Because of the complexity of prototype 3, the presentation of the functions will be done through four examples that are typical tasks that the user could use the app to solve. In these examples the user is standing at Assistens Cemetery and will solve the tasks accordingly. These tasks and the way to solve them using the app are explained in the following.

Finding H.C. Ørsted's Grave

For this task we assume that the user does not know where H. C. Ørsted is buried. When the user starts the app, he is greeted by the Main Menu seen in Figure 75. The Search option which is the first from the top is then selected. In this menu the user types in the name he wants to search for as seen in Figure 76, and a list of possible persons appear on a list, Figure 77.

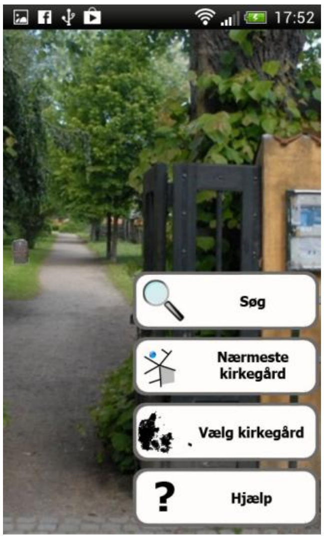
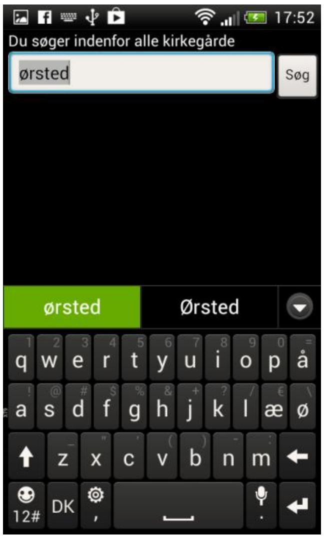


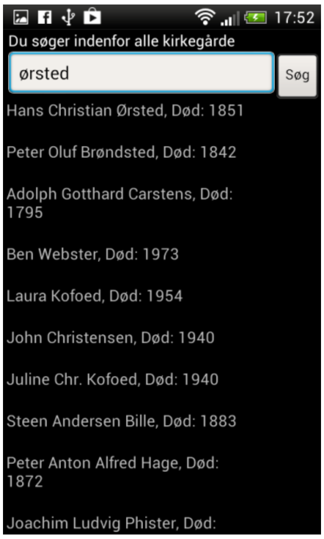
Figure 75 The main menu of the app



Du søger indenfor alle kirkegårde

ørsted

Figure 76 A name being written in the search field of the search function.



Du søger indenfor alle kirkegårde

ørsted

Hans Christian Ørsted, Død: 1851

Peter Oluf Brøndsted, Død: 1842

Adolph Gotthard Carstens, Død: 1795

Ben Webster, Død: 1973

Laura Kofoed, Død: 1954

John Christensen, Død: 1940

Juline Chr. Kofoed, Død: 1940

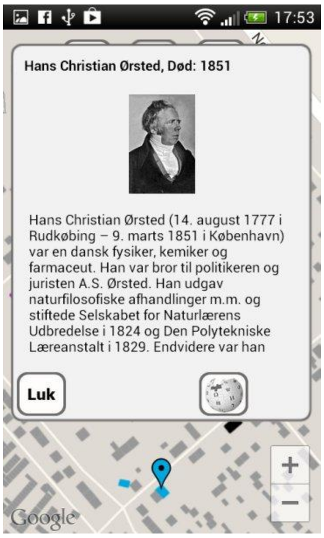
Steen Andersen Bille, Død: 1883

Peter Anton Alfred Hage, Død: 1872

Joachim Ludvig Phister, Død:

Figure 77 Search result.

When the user has selected a name, the user is taken to the Exploration Map with an open info box matching the name that was selected, Figure 78. Here the picture of the person as well as basic information is provided by Wikipedia, and the user can choose to see the person's entire Wikipedia article by clicking the Wikipedia logo in the bottom of the info box. Notice that the Route icon is missing from the info box because the Exploration Map was entered through another entry than the "Nearest Cemetery".



Hans Christian Ørsted, Død: 1851

Hans Christian Ørsted (14. august 1777 i Rudkøbing – 9. marts 1851 i København) var en dansk fysiker, kemiker og farmaceut. Han var bror til politikeren og juristen A.S. Ørsted. Han udgav naturfilosofiske afhandlinger m.m. og stiftede Selskabet for Naturlærens Udbredelse i 1824 og Den Polytekniske Læreanstalt i 1829. Endvidere var han

Figure 78 A PopUp box of the selected person.

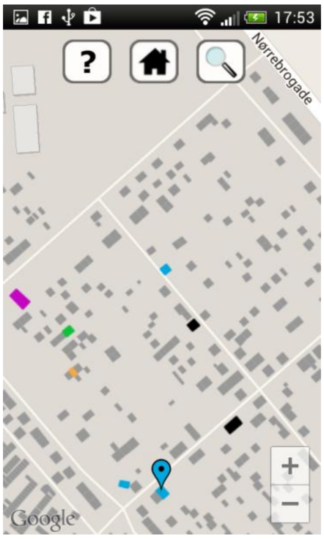


Figure 79 The selected grave.

When the user is done reading the info box, it can be closed to display the Exploration Map only, and from here the user can make a new search, or return to the Main Menu by pressing the main menu icon in the top of the screen, Figure 79.

Locate a Grave at Sct. Knud's Cemetery on Bornholm

For this task we imagine that the user has an interest in a specific cemetery. The user once again starts at the Main Menu, Figure 80. Now the third option from the top is chosen, "Select Cemetery", which takes the user to the cemetery map that displays all cemeteries found in the database, Figure 81.

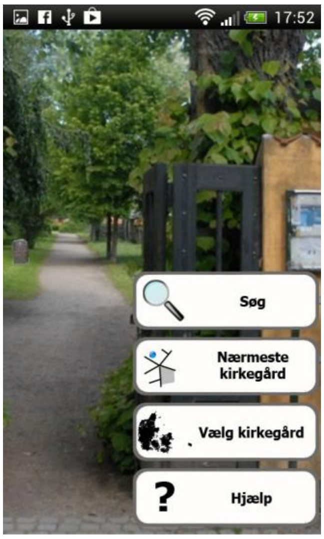
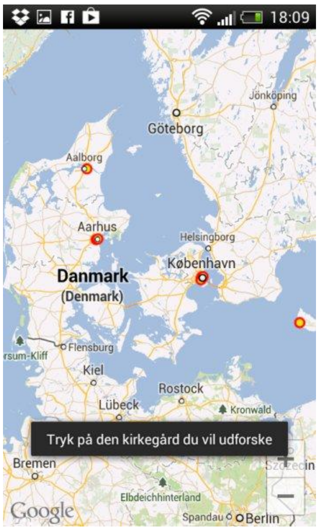


Figure 80 Main menu.



Tryk på den kirkegård du vil udforske

Figure 81 Select Cemetery map opened.

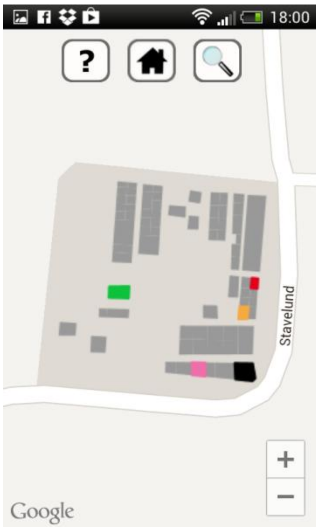
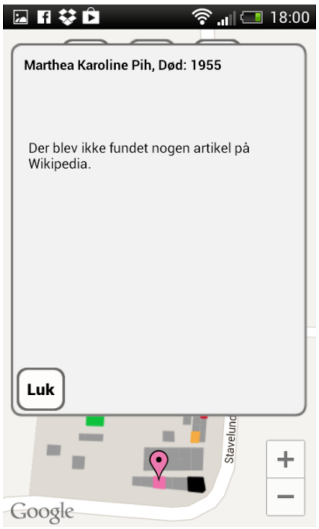


Figure 82 The selected cemetery.



Marthea Karoline Pih, Død: 1955

Der blev ikke fundet nogen artikel på Wikipedia.

Figure 83 PopUp box opened.

When the user has found, clicked and selected the cemetery of interest the user is brought to the Exploration Map of the selected cemetery, Figure 82. Here the user can freely click on a grave in order for the info box to appear so that the user can learn more about the selected grave, Figure 83.

Create a Tour in Between Four Graves at Assistens Cemetery

The third task is solved by entering the second option from the top in the Main Menu, "Nearest Cemetery", Figure 84. This brings the user directly to the Exploration Map as seen in Figure 85. Notice how the Tour Planner icon has appeared in the top of the screen, as opposed to the previous tasks because the Exploration Map has been selected directly from the "Nearest Cemetery" option.

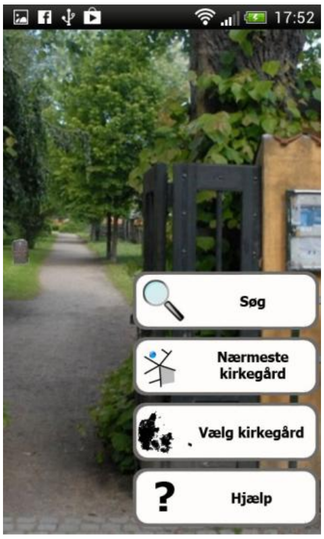


Figure 84 Main menu.

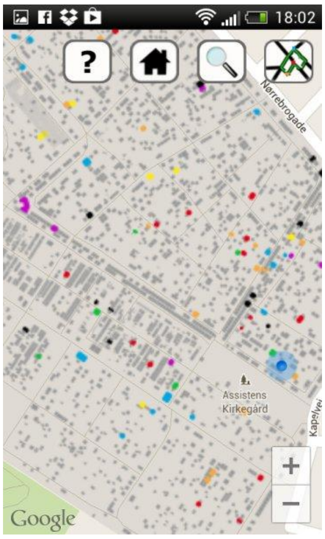


Figure 85 A map of the nearest cemetery has been displayed.

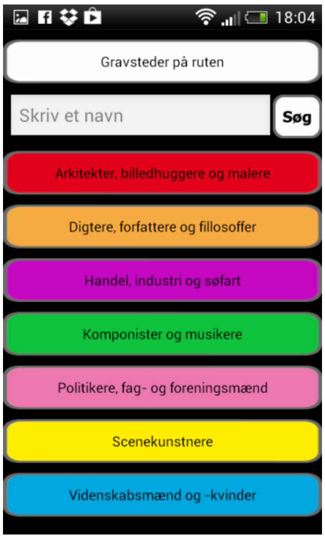

18:04
Gravsteder på ruten
Skriv et navn
Søg
Arkitekter, billedhuggere og malere
Digtere, forfattere og fillosoffer
Handel, industri og søfart
Komponister og musikere
Politikere, fag- og foreningsmænd
Scenekunstnere
Videnskabsmænd og -kvinder

Figure 86 Tour Planner tool opened.

When the Tour Planner icon is clicked, the Tour Planner Menu opens, Figure 86. From here the user can open the individual profession tabs and select historic persons within a specific category, Figure 87. When the user has selected all the graves he wants to add to his tour he clicks the "View graves on the route" button in the top of the screen, Figure 88. From here the list of selected persons can be seen and persons can be deselected if wanted. When the user is ready he can choose the means of transportation for the tour depending on the allowed types of transportation at the cemetery.

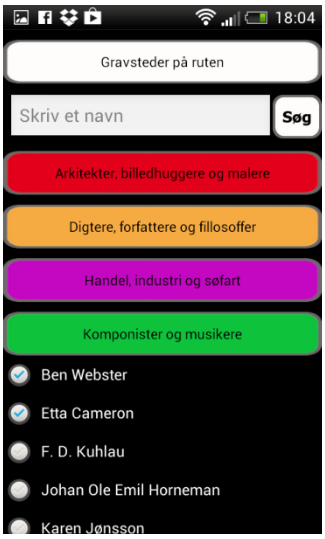


Figure 87 A profession tab is opened, and graves are selected.

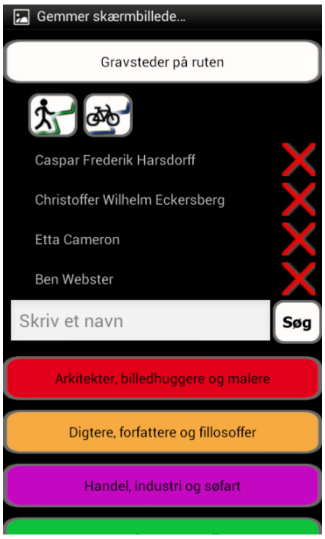
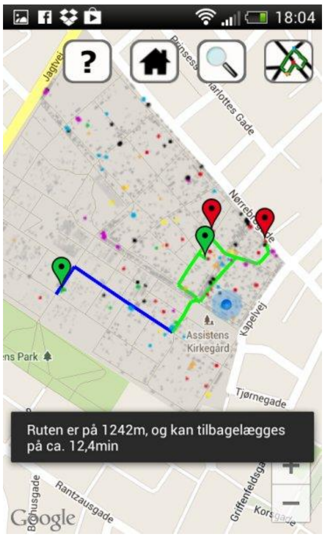

Gemmer skærmbillede...
Gravsteder på ruten
Caspar Frederik Harsdorff
Christoffer Wilhelm Eckersberg
Etta Cameron
Ben Webster
Skriv et navn
Søg
Arkitekter, billedhuggere og malere
Digtere, forfattere og fillosoffer
Handel, industri og søfart

Figure 88 The selected graves are displayed on a list.



Ruten er på 1242m, og kan tilbagelægges på ca. 12,4min

Figure 89 A tour is generated.

This adds the created tour to the Exploration Map, Figure 89. The pins at the graves are coloured depending on the profession, and the route is coloured depending on the allowed means of transportation along the path. Finally a message is displayed for 5 seconds, explaining how long the tour is, and how much time it will take to travel.

Find Out How Much Time it Will Take to Walk to H. C. Andersen's Grave.

To solve the final task, the user is once again at the Main Menu, Figure 90. The user is aware that H. C. Andersen is buried at Assistens Cemetery, where the user is standing himself, and therefore he selects the "Nearest Cemetery" option in order to access the Exploration Map, Figure 91. The user now clicks the search icon in the top of the screen, and finds H. C. Andersen by tying in his name in the search menu, Figure 92

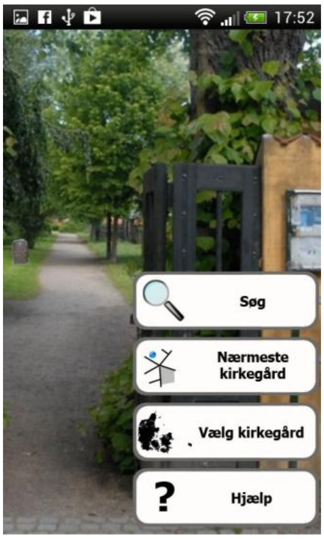


Figure 90 Main menu

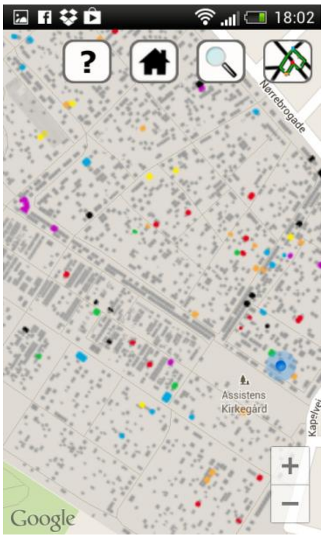
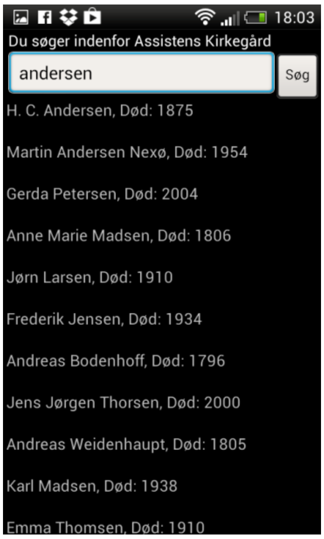


Figure 91 The map of the nearest cemetery is opened.



Du søger indenfor Assistens Kirkegård

andersen Søg

H. C. Andersen, Død: 1875

Martin Andersen Nexø, Død: 1954

Gerda Petersen, Død: 2004

Anne Marie Madsen, Død: 1806

Jørn Larsen, Død: 1910

Frederik Jensen, Død: 1934

Andreas Bodenhoff, Død: 1796

Jens Jørgen Thorsen, Død: 2000

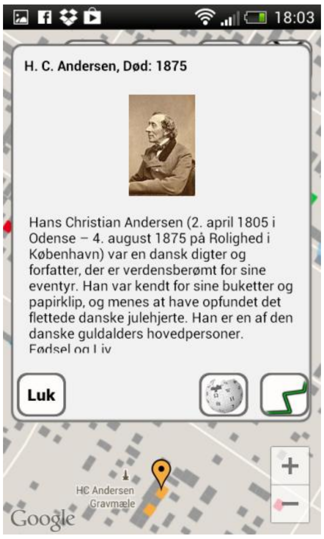
Andreas Weidenhaupt, Død: 1805

Karl Madsen, Død: 1938

Emma Thomsen, Død: 1910

Figure 92 A search within the specific cemetery is performed.

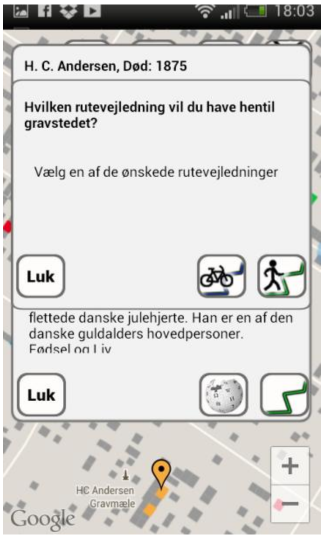
When H. C. Andersen's name is clicked and selected, the user returns to the Exploration Map with the info box of H. C. Andersen open, Figure 93. Notice how the route icon is now available in the bottom of the info box because the Exploration Map was entered through the "Nearest Cemetery" option. When this icon is clicked, a new directions box appears on top of the info box, Figure 94. Here the user can choose to generate the route depending on whether he is walking or biking. When a means of transportation has been selected, both the info box and the directions box close, and a route is added to the Exploration Map, Figure 95. Furthermore a message is displayed for 5 seconds, explaining how far the route is and how much time it will take to travel to the grave.


18:03

H. C. Andersen, Død: 1875

Hans Christian Andersen (2. april 1805 i Odense – 4. august 1875 på Rolighed i København) var en dansk digter og forfatter, der er verdensberømt for sine eventyr. Han var kendt for sine buketter og papirklip, og menes at have opfundet det flettede danske julehjerte. Han er en af den danske guldalders hovedpersoner.
Fødsel og Liv

Figure 93 An info box.


18:03

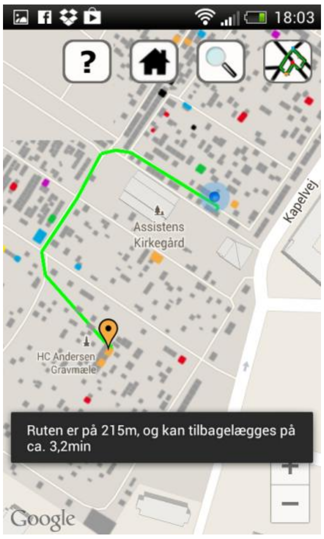
H. C. Andersen, Død: 1875

Hvilken rutevejledning vil du have hentil gravstedet?

Vælg en af de ønskede rutevejledninger

flettede danske julehjerte. Han er en af den danske guldalders hovedpersoner.
Fødsel og Liv

Figure 94 The route tool is chosen.


18:03

Ruten er på 215m, og kan tilbagelægges på ca. 3,2min

Figure 95 The route is displayed.

Having explained how prototype 3 works by describing four examples, it is time to document how the usability test of prototype 3 was designed.

11.3.2 *USABILITY TEST DESIGN*

The usability test of prototype 3 was the first test that was meant to bring the app to test subjects with no geoinformatics or programming skills. This was done because prototype 3 as mentioned was the first prototype that represented a nearly functional app that could be released. Because of this, much more time was put into thoroughly planning the usability test in order to collect and evaluate the opinions and experiences of the test subjects, so that this could be integrated in the future development of the app. To do this, the nine previously mentioned steps defined by Bastien were used. In the following, the thoughts and decisions as to how to the user tests should be executed are described using these nine steps.

As it has been stated throughout the report, the objective of this usability test, and for that matter any of the other tests done during the iterations, was to identify the logical and technical issues of the app, as well as getting inspiration to further development. Because of this, the goal of this user test was to collect data from the user test that could be used to enhance the usability of the app, as well as identifying technical problems.

Even though the Hallway Testing method was used, the test subjects would not be recruited directly from a cemetery in respect for the privacy of visitors at the cemeteries. Instead test subjects were recruited elsewhere, and because of this they were not located at a cemetery during the test. Furthermore, only five random test subjects were asked to participate in the usability test, since this as stated earlier should make the project group able to identify the majority of the issues of the app. During the test the participants were using a handed out a preconfigured smartphone with the app installed on it, in order to complete a number of given tasks.

As the test subjects were not located at a cemetery at the time of the user test and had no prior knowledge of the app's use or existence, the test subjects needed to be instructed as to what situation they should imagine being in, when performing the test. Because of this all test subjects were asked to imagine that they were standing at Assistens Cemetery, and that they were sightseers interested in knowing more about the historical persons buried at various cemeteries.

For this project, five tasks were designed. Each of these tasks were designed so that the test subject would have to try using most of the functions available in the app. The tasks the test subjects were given are as follows:

- **Can you locate the grave of H. C. Ørsted?** This task is solved by entering the first of the three search options. Here the user can type in the name and hopefully find grave of the person of interest.
- **In which cemetery is Nicolai Dajon buried?** This task is solved just like the first, but the question has its focus on the cemetery the grave is located at. This has been done to see how the user reacts when the question includes a cemetery, and to see if they instead of the first search option select the third called "Find cemetery". Furthermore Nicolai Dajon has been selected to see the user's reaction trying to spell a difficult name, and to see if the Approximate String Matching works.
- **Can you find the grave of a person buried in Sct. Knud's Cemetery on Bornholm?** The purpose of this task is to see if the user selects the search option "Find cemetery", and from there is able to find the correct cemetery and find a grave located at the cemetery.
- **Are you able to create a tour by bike to four graves, where one belongs to a scientist?** This task has been designed so that the test subject has to find and use the tour planning tool. The user will have to specifically enter the scientist category to select a scientist, and figure out how to create the route.
- **How many minutes would it take you to walk to H.C. Andersen's grave from where you stand?** For this task, the test subject needs to remember that he is imagining to be standing at Assistens Cemetery. Because of this the test subject should choose the second search option "Nearest cemetery", locate H.C. Andersen's grave and select the route tool.

This user test's observation and recording method was the Think Aloud Protocol. The test was conducted by one member of the project group who acted as the observer and facilitator at the same time, both taking notes of the event, as well as asking relevant questions to the test subject.

Since the facilitator was a member of the project group, the facilitator was already well aware of the context that the user test was a part of, and therefore he did not need any additional briefing prior to the tests. Because both the testing and observation methods were relatively simple, there was no need for any specific preparations before a test. The facilitator had to be well aware of the tasks the test subject was supposed to perform, and make sure that the smartphone that would be handed out was working.

To be able to compare each of the test subject's results with one another directly, the time taken was noted besides the comments from the Think Aloud Protocol. Based on this, the level of difficulty on a scale from 1 to 5, where 1 was "very easy" and 5 is "could not complete" was assessed by the facilitator. Furthermore the test subjects were asked for their age, as well as their smartphone experience from 1 to 3, where 1 is no experience, 2 is some experience, and 3 is regular smartphone user. This was done in order to have a reference in between test subjects to see if the level of experience for example, affected the results of the test.

When the user tests were complete, they were evaluated based on the rating from 1 to 5 for each task. Furthermore, the collected comments were examined in order to identify the logical and technical issues, which was the objective of the entire test.

After having designed the usability test for the third prototype, the test was performed. The specific results from the test can be seen in Appendix J.

11.3.3 *EVALUATION*

With usability tests completed, the result can be presented by comparing each test result in order to evaluate whether the app is working properly, or if it contains bugs and issues. This will be done firstly by investigating the result of each of the five tasks individually. Secondly the evaluation of the general impression of the app will be reviewed, and lastly a list of the logical and technical issues and problems that have been identified will be presented.

Task 1

For the first task, the test subjects were supposed to locate the grave of H. C. Ørsted. All of the test subjects managed to solve this task without any significant difficulties, and on an average they spent less than a minute to solve the task.

All of the test subjects selected the correct option in the Main Menu, "Search", as their first action, and were ready to start searching. Two of the test subjects were in doubt where to click in order to start the search once a name had been typed in, while one test subject wanted an auto finish function for typing names. Finally one of the test subjects wondered why so many irrelevant names appeared on the list of results.

Task 2

This task required the test subjects to find out what cemetery that Nicolai Dajon was buried at. Though the solution of this task was similar to the first task, one of the test subjects did not complete the task, while another reported it being very difficult.

All of the test subjects were able to find the grave, but the main problem was that the name of the cemetery that the grave was located at did not appear clearly anywhere. Additionally, two of the test subjects identified the cemetery correctly only by using local knowledge, rather than finding the small label provided by Google Maps.

Task 3

The assignment for this task was to locate a grave at Sct. Knuds Cemetery on Bornholm. This all of the test subjects managed to do by going back to the main menu and selecting the "Find Cemetery" option.

Two of the test subjects however did have difficulties in figuring out how to access information regarding a grave when it was not done through a direct search function.

Task 4

For this task, the test subjects were asked to create a tour to four graves at Assistens Cemetery where one of the graves was of a scientist. This task proved to be very difficult for all of the test subjects, and though they all managed to generate a tour, it took a considerable amount of time for everyone to complete the task.

One of the main issues that four of the test subjects struggled with, was that the tour planner was only available when the Exploration Map had been entered through the "nearest cemetery" option, and not though any of the two other options in the main menu. Because of this the test subjects could not find the tour planner tool, though they still could read about it in the help menu, which they found frustrating.

Another major issue was the tour planner menu itself, where all of the test subjects struggled in one way or another with creating a tour. All but one of the test subjects did not find the layout to be intuitive, and often wondered what to do next, how to see the selected graves, or how to display a tour. Furthermore the app did not display the progress bar when creating a tour, which had several of the participants wondering if the app was responding. One test subject also wondered why one of the profession tabs did not open when it was clicked, and thought it to be a mistake, though this was because no one with that profession was buried in the cemetery.

Finally two of the test subjects did not succeed in creating a tour even though they solved the task. Instead the app said that the tour could not be shown and did nothing more, which the test subjects did not understand the reason for.

Task 5

In the final task, the test subjects had to figure out how much time it would take them to walk to the grave of H.C. Andersen. Four of the test subjects did this without any problems, while the final test subject gave up after trying to generate the route using the tour planner.

Like with task 4, a major issue with the route function is that is also only available when the Exploration map has been entered from the "nearest cemetery" option. This tricked most of the test subjects at first, and was the reason why the one test subjects did not succeed in completing the task. Two of the test subjects also commented that they would have liked if the text presented in the popup message regarding distance and time could be seen again after it disappears when they click somewhere else.

An overview for the average difficulty rating and average time used for the five tasks can be seen in Figure 96.

	Average difficulty rating	Average time used
Task 1	1.2	>1 minute
Task 2	2.8	1 minute
Task 3	1.4	1 minute
Task 4	4.2	3.30 minutes
Task 5	2.0	2 minutes

Figure 96 The average difficulty rating and average time used for each tasks.

Besides the comments related to performing the specific tasks, the test subjects also had a list of comments that were related to the app in general. Besides a number of random crashes throughout the testing, all of the test subjects thought that the app overall seemed useful and logical.

The main menu seemed to work well for the test subjects. There was some confusion about when to use the “nearest cemetery” and “select cemetery” options. Furthermore a number of the test subjects requested some kind of basic welcome information either when starting the app or in the main menu. Additionally only one test subject entered the help menu in the main menu, and did not bother to read the help because it had too much text.

The “Search” option worked well for all test subjects, but only one test subject read and realized that there were two different “search” menus, depending on whether the main menu search or the exploration map search was selected. This caused a number of problems, and none of the test subjects understood why the tour planner and route option were only available some times for instance.

The “Select Cemetery” option worked well, though some of the test subjects wondered why there was no home button in that view.

The exploration map confused many of the test subjects. Some were wondering if the map was done loading, some were wondering what the shapes on the map were, and everyone was wondering what the colours represented. Everyone eventually realized that the shapes were graves, but few of the test subjects thought of that instantly. A number of the test subjects furthermore expected to see an image of the graves when zooming in on the exploration map.

The help menu in the Exploration map was used by all of the test subjects. However, few of them found what they were looking for. Many of the participants expected the help menu to contain basic information regarding the cemetery they were at, such as the cemetery name. Furthermore few of the test subjects actually read the text in the help menu, but only looked at the icon descriptions to find out what they meant, especially the colours of the graves.

Having successfully finished the user tests, the third and final iteration in this project has been completed. This iteration has identified a number of interesting technical, logical and visual issues within the developed app. Through this, the project group has been enlightened in regard as to what a pair of fresh eyes with no programming or developing skills thinks of the developed product.

12 STATUS AND LIMITATIONS OF THE SYSTEM

This section describes the problems that were discovered during the development period and the status of the app implementation, which functions that were implemented and which ones that were not.

Identified Issues

Based on the performance and comments from the participants, as well as the observations done during the user tests, a number of problems with the design of the app have been revealed. These issues have been divided into three categories; Technical Issues born from bugs in the coding; Logical Issues where the app structure does not make sense; Visualisation Issues being problems regarding visually presenting information. These issues and an immediate suggestion for a solution can be seen in Figure 97 below.

Location	Issue	Solution
	Technical Issues	
Tour Planner C.2	Tour cannot be generated with some graves selected	The TSP algorithm implemented in pgRouting only works with four different nodes, instead a manual algorithm most be implemented for routes with less than four nodes
Exploration Map B.1	App crashes randomly	Needs thoroughly testing with debugger function in Eclipse to find the reason
Displaying Route or Tour B.1	Once toast displaying distance and time to travel disappears, it cannot be seen again	A text field with this information could be placed on the screen
Info Box C.4	The picture is not always of the person and sometimes there is none even if there is one on Wikipedia	The picture is limited to show the first jpg files from the article. To change this the images should be from a source under control.
Cemetery Selection Map B.2	Sometimes the name of the cemetery is not shown	Text in a WMS from GeoServer will not be drawn across tiles and there is no way around this
Exploration Map B.1	Sometimes there is no response from a click on the map	If a click is made on top of a city name, other names or objects like a transport icon the click is not detected. This is the way Google Maps currently functions and there is no way around it
Exploration Map B.1	Some roads/paths from directions are slightly off	The internal network is based on FOT which does not 100 % match Google Maps. There is no easy way to change this

Logical Issues		
Tour Planner C.2	No progress bar is present when generating a tour	Use asynchronous task
Displaying Route or Tour B.1	Once toast displaying distance and time to travel disappears, it cannot be seen again	A text field with this information could be placed on the screen
Search functions B.3 & C.3	Both functions are virtually identical, so it is difficult to know that there are two functions and that they work slightly differently	Emphasise the already existing text message of which cemetery the search will be done for, or maybe use a toast message when the user starts the search
Exploration map B.1 **Infobox C.4**	Function buttons disappear from the screen views when they are not available	To avoid confusion, buttons could just be made gray
Route button C.4 **Tour Planner button B.1**	The functions are only available when the Exploration Map has been entered through the "Nearest Cemetery option"	Enable tour planner for all modes of the app
Tour Planner B.2	All Profession tabs are available even when no graves are contained within some of them.	Remove tabs with no graves or maybe show a number of graves on the tab
Visualisation Issues		
Help menus B.4 & C.1	The help menus have too much text, and are too detailed.	Rephrase and simplify
Exploration map B.1	It is not clear what is displayed on the Exploration map.	When entering the map a toast could inform the user that the polygons on the cemetery are graves

Figure 97 Identified issues.

Status of Implementation

The status for the implementation can be seen in Figure 98, all of the functionalities from the first two categories were implemented, and almost all from the third category while only some from the last category. There were some minor details with the general directions functionality which had some logical errors but these can easily be fixed once detected.

Categories
Framework
- A map of the cemetery showing the individual positions of the graves.
Basic Functions
- Function to search for a historical person with the result showing which cemetery and the position at the cemetery the person is buried.
Advanced Functions
- On the map different colours for historical persons' graves will be used based on their profession.
- Function to show a map of all the historical persons' graves on a specific cemetery.
- Function to find and exploring nearest cemetery.
- When a grave is clicked on the map an information window will show with information specific for this grave and the person who is buried there.
- Implementation of navigation to a grave on road/path level within the cemetery.
- The navigation functionality must include walking mode
- The navigation functionality must include bicycling mode
- The navigation functionality must include driving mode
- Navigation towards the cemetery.
- Tour guide for a number of selected graves on the cemetery.
Refinement
- The grave search can be enhanced by auto complete of the name and fuzzy/semantic search.
- Support of UTF-8 character set.
- Support for different screen sizes.
- More advanced representation for cemeteries. Markers instead of WMS and clustering of data shown on the map.
- The app should work smoothly without any performance problem.
- Any graphic problems should be solved.
- Any functionality problems should be solved.

Figure 98 Implementation completions for the desired functionality. Text with green colour shows fully implemented functionality or functionality with only minor problems, orange indicate only partial implemented and red indicate no or almost no implemented functionality.

In Figure 99 the problem regarding the network not handling edges with multiple points can be seen. To solve this, the directions should use the points of the different edges using the PostGIS function *ST_DumpPoints* instead of using line segments from node to node. The directions are also missing the last stretch from the path to the grave, which happens if there are less than two nodes between the start and end position. The cause of this problem is most likely because of a small error in the directions script which it should be possible to correct.

Figure 100 shows the zigzag problem that occurs, if the two graves in a tour planner route are on opposite sides of the edge. Right now the directions are a direct line if the nearest edge is the same. This is only true if the graves are on the same side. An additional check should be implemented by looking at the coordinates of the two graves and of the edge.

Directions for driving mode were implemented in the same way as bicycle directions. However, directions for driving should work fundamentally different, meaning that instead of finding the fastest overall route, the route which followed roads nearest to the grave should be chosen, and to implement this, more coding is needed before this will work as desired. The same is the case with external directions towards a cemetery using Google Directions. This is possible, and most of the needed code is already written, but still some work is left before this will work.

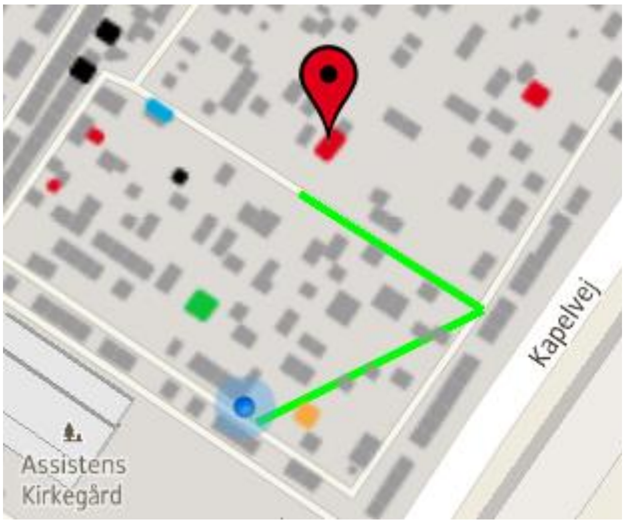


Figure 99 Example of problem when the route is described as straight lines between nodes and start point, in the network.

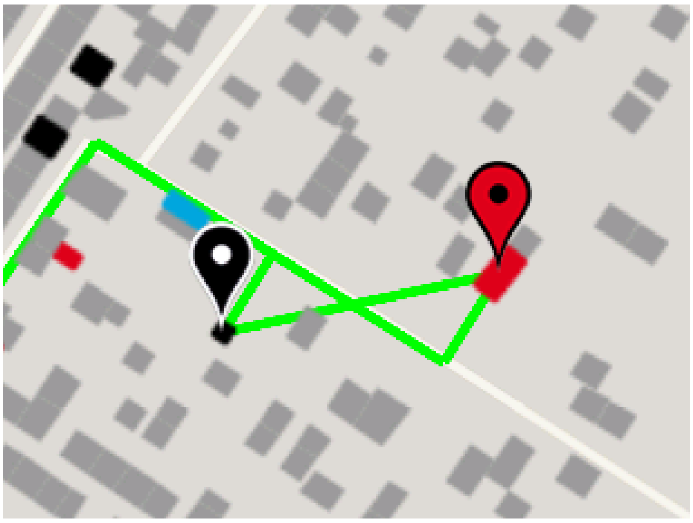


Figure 100 The Zigzag error when two following graves is closest to the same edge but on opposite sides.

In the refinement category some of the functionalities are completely implemented, some half implemented and others not at all implemented.

The auto complete and semantic functions for grave name search were only implemented with a fuzzy algorithm. For a final product, a semantic algorithm would be necessary and today most people are expecting auto complete functionality while searching for text. To implement the lacking functionality of a semantic search, it would probably be necessary to develop an algorithm for this specific purpose that handles abbreviations. The problem now is not necessarily the algorithm itself, but more how to index the searches in the database. Without a prober indexing, the searches could be slow if the database was to include more graves than the few hundred it has now. One solution for the auto complete could also quite easily be coded, but here the problem would be to connect to the database fast enough. It is without doubt possible to implement this function, but it might take some time to find a way to do this fast enough for the user not to become impatient.

A clustering function of the cemeteries at the “Select Cemetery” map would be useful. Some solutions for this were found, but unfortunately they were developed for Google Maps Android API v1 so a lot of changes would be required before this would work with this app which is build with Google Maps Android API v2. Of course it is possible to build the function from scratch but this would take some time.

Most of the time the app works smoothly and without any performance problems, but when the app has been running for some time, it can stop working and even crash. The reason for these apparent randomly crashes are unknown. However, the Eclipse development environment includes a debugger function, and by having the app connected to this, should provide insight to what the cause for these random crashes is.

The app is dependent on an internet connection and a positioning from both GPS and network triangulation. Since map data is very demanding, a fast internet connection is preferred. Right now a WMS has been created for every cemetery included in the app, and this forces the app to know a relatively precise position of the user, before it can select the right WMS to display.

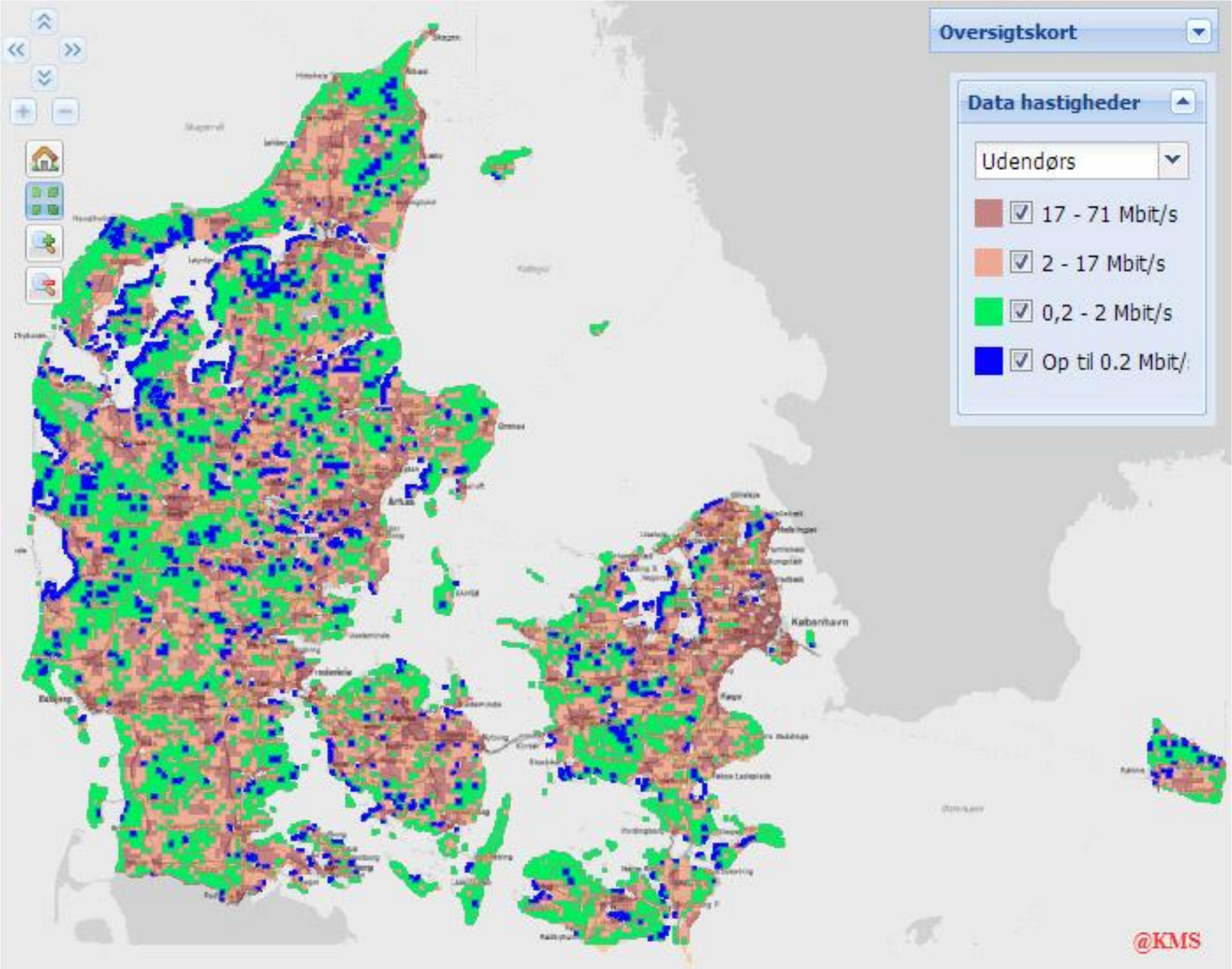


Figure 101 Theoretical mobile coverage map for TDC's network. http://daekning.tdc.dk/privat.html

In Figure 101 a mobile coverage map is shown using data speed. Within urban areas, both the data speed and network triangulation used to find the closest cemetery are excellent. Some problems can occur when trying to acquire an accurate GPS position while standing at a cemetery with many trees, but often the position is still useful. In the blue areas of the figure, the internet speed is so slow that it would possibly make the app unusable for impatient persons, and since there normally is a correlation between slow data speed and the number of cell towers, the network triangulation is probably also very imprecise. In fact testing in the rural parts of Bornholm showed positions 2-3 km off.

To overcome the imprecise positioning, the app could force the use of GPS positioning, but this could increase the time of loading the map with up to one minute, before a position was found. Another way of overcome the problem, could be to only have one WMS covering all of Denmark, This solution was deliberately avoided, because it requires a lot of processing power and a lot of disc space to pre generate such a large area. However, for the final system, a faster server than the one used by project group would probably be used, and thereby it would be possible to use one WMS covering all of Denmark.

Execution Stage: Summary

This stage covered the process going from idea to product, and how the developed service functions. The final screen views of the app are presented in Appendix K. As the project group foretold, native app programming is difficult to learn, and it was useful to have a development strategy, which in the beginning concentrated on the very basic core functionality.

The last developed prototype nearly has all the functionality which was the goal for the final product, and many more than it was thought possible in the beginning of the project. There was no way of knowing this in the beginning of the development phase, but it was thought that there would be an exponential correlation between time and the number of implemented functionalities. During the development, it can be very difficult actually to believe this, and having a weekly status table with almost only blank or red fields after four weeks can be mentally tough. Here it was important to reassure oneself of that it was actually going accordingly to the plan.

The prototype driven model was very efficient in providing the necessary input to the research and design part of the development. Many aspects of the way that people think and interpretive the behaviour of the app would never have been realised without these prototype tests. As it was the case with this project, just a couple of tests with three to five persons actually discovered many, both technical and logical problems of the product very fast and in an inexpensive way.

With the Execution Stage finished, it is time to look at the project period as a whole, and evaluate the entire project period, and the work that has been performed throughout the past four and a half months. This will be done in the final stage the Closing Stage.

CLOSING STAGE

The final stage of the project, is used to sum up on the work done and choices made, during the project period. During the previous three stages, many decisions had to be chosen, and many questions were answered as well. In the closing stage, these elements will be presented through three sections in order to close the project:

- **Discussion**
  Here the methods and theories that were implemented will be discussed.
- **Conclusion**
  In this section, a conclusion of the project as a whole will be presented.
- **Perspectives**
  The third section will describe how the project could evolve further.

13 DISCUSSION

Throughout the project many decisions and choices were made. Due to this, it is interesting to discuss how these choices and decisions have affected the project and the developed product, and how other decisions could have changed the entire process.

Before the development phase the project group was advised by Morten Fuglsang and consultants from Atkins. Morten Fuglsang proposed a lightweight solution consisting of a website with local geographical data on the smartphone so a heavy backend could be avoided. The technical consultants from Atkins proposed a mobile development framework or development of a native app for which they could assist if this was needed. With different suggestions from the two technical advisors, it was clear that there were different ways of developing the intended service. As a result of this, a closer study of the possible development environments was initiated.

In the end an IDE for Android applications was chosen as the development environment. As stated, this method was the most relevant for developing the service, but at the same time it was also the most difficult way to go. Because of this, a lot of time went into learning and understanding how to program an Android application in its native environment - time that could have been used to implement, enhance and test more functions in the product.

As an alternative, one of the other suggested methods could have been chosen, such as developing the service for a website able to adapt to the browser of smartphones. Since the members of the project group have worked with WebGIS for websites before, this solution would have required less time to become familiarized with, and as such this would lead to more time available for implementing more functionality. On the other hand it is not certain that a smartphone optimised website would work perfectly on all smartphones, and different OSs and browsers should have been tested. The main reason to avoid this solution was that the project team favoured maximum performance and control of the functionalities of the smartphone, which is possible with a native app. However, the product could likely have been developed for mobile optimised browsers with only a slight loss of performance and functionality.

The simplest method for developing the service would have been to use mockup software. This solution would not have been an actual service, but rather a working presentation of how the service was meant to function in its complete form. Because of this, this method would have been perfect for designing the layout, flow, and graphics of the service, but it would not have given any "hands on" experience in building the actual WebGIS service for a smartphone. Choosing this method would have given the project a very different approach where the main focus could have revolved around how users see, observe, and interact with WebGIS by building many different prototypes of the service. However, because the project group wanted to focus on the actual development of the service, this option was never considered as a real possibility.

Based on the development methodology it was decided to use a spiral model with four iterations, of which three were performed in this project, where the functionality and design of the product were revised at the beginning of each iteration, for then to be enhanced and finally tested in a prototype. The number of iterations could easily have been increased in order to perfect the product even further, but this would have required additional time to complete, which would have strained the timeline elsewhere in the project period. Conversely the project could have been built up around a single extensive iteration where each phase in the process was worked through before the next one was started. This method however would most likely not be very successful because the final product would not have been tested in any way, which would make it likely that logical and technical issues would only be discovered in the end. Furthermore it would not have been possible to evaluate the risk of failure throughout the project period in the same way, which could have jeopardized the outcome of the entire project.

By choosing the method with the best performance and the most possible functionality for developing the desired service, the project group pushed themselves as far as possible within the project period. But this was no easy task, and by looking at the graph of time and implemented functionalities it becomes clear that the implementation progress in the first month barely was tolerable. Retrospective the project turned out to be a success, but risk of failure was present for a long time. The project group handled this with the confidence of having a second familiar technology to fall back on. But was the success just pure luck, or was it a bad decision to develop a native app?

The choice of native app development could be regarded as a safe choice, since the project group was promised assistance, but this help would be very limited and it could not be expected that Atkins would solve every single problem that would occur. Because of this, the choice between technologies was difficult, but it was tempting to try developing for a native app but at the same time risky, because it was a completely new development environment for the project group. As explained however, the risk of failure was removed by having a familiar technology to fall back on. With this in mind, it was a very bold move to decide to develop a native app, and without the familiar web technology as a second option, it would have been a daring and maybe even a reckless choice.

Based on these considerations, the project group advises others to carefully consider the option of developing a native app with no previous experience before starting such a project. As it has been presented in this project it is definitely possible to complete such a task, but requires a lot of time, strength and determination, and other simpler options are available which will require less work to learn to use.

14 CONCLUSION

With the end of the project period, it is time to conclude on the work completed in the past four and a half months. When the project was started, the goal was to create a geoinformatic service regarding graves and cemeteries. This has been completed with the creation of a WebGIS system with an Android application for smartphones as frontend that makes it possible to explore cemeteries in order to find the graves of historical persons. To do this, the first stage of the project defined the specific goals and objectives of the project including the problem statement. This was done by investigating the market for existing solutions, and interviewing people working with management, registration, and mapping of cemeteries. This research was the foundation for the problem statement:

> *How can a professional service for smartphones be developed to locate and navigate the user to the graves of historical persons? Additionally, how can the service provide information about the historical persons to the user?*
>
> *The answer to this problem will be found through the following three questions:*
>
> - *How can the project be managed most optimally and which types of management tools and development methodology should be involved?*
> - *Which development platform and environment will be optimal to build such a service?*
> - *How can the user interface of the service be designed to ensure an intuitive use?*

The second stage of the project was used to do research on the different methods, theories and technologies that could be used in the development of the intended service. Here a management plan to structure the project period including a development strategy was created. This management plan presented a SWOT analysis that was used to identify the risks of the project, which again were used to define how the development stage should be executed in order to combat these risks. Furthermore it became apparent that the service should be developed for Android using an IDE to build a native app, since this implementation would perform best on smartphones and provide the greatest functionality. The solution was to follow a prototype driven spiral model consisting of four iterations that each was concluded by a prototype usability test in order to eliminate logical and technical errors and issues.

It was not until the third stage of the project that the actual development of the product was initiated. Here the technologies and methods chosen in the previous stage of the project were implemented in a WebGIS system, and although only three iterations were completed, the end product was very close to being fully operational.

When looking back at the problem statement, it is interesting to note that the answers to all of the sub questions were provided in the second stage of the project. However, it was not possible to know this until the third stage of the project had been completed, because this stage tested the prototypes that were based on the choices made in the second stage. By performing usability tests on the prototypes it was revealed that the average user was able to use the developed app successfully, and that is why the result of these tests truly showed that the second stage had answered the sub questions, and ultimately the overall question of the problem statement.

From these observations, it can be concluded that the app and its underlying system lives up to the task that it was developed for. The combination of providing information and assisting with navigation increases the use of the service, and with the three types of entrances for exploring cemeteries, the product is able to serve a variety of purposes. The usability tests showed that users were able to understand the design and functionality of the app serving as frontend, which was essential for the usability of the app and thereby the entire developed system. With this in mind, it can be said that the service is definitely serving its purpose, but there still are some functionalities that need to be adjusted before the service is fully operational.

The one major element in the developed service that is not working optimally is the network functionality. As stated, the algorithms used were a function extension for PostGIS. These algorithms had a number of limitations that affected their use in this project. First of all the Dijkstra algorithm was designed for calculating routes and tours from one node to another. When working within a cemetery, there are a lot of graves between each node, which causes challenges when the tour has to stop between two nodes. Furthermore the TSP algorithm needs the generated route to travel between four nodes before it can calculate a route. This caused problems meaning that the algorithm had to be altered to be able to work with less than four nodes. With the end of the project, it is now clear that the selected network algorithms were far from perfect for this specific purpose, but in order to build the developed service it was an easy option. Hindsight, a new algorithm should have been developed that could answer the needs of the app, but this would have been a significant task that would not have been realistic to accomplish along with the other functions developed.

Based on these reflections, it can be concluded that the development of the service lives up to the goal of the project and answers the problem statement through the results of the usability tests. Because of this, the project can be regarded as a success. Furthermore the members of the project group have acquired a considerable amount of knowledge within a number of disciplines, such as geoinformatic system development for smartphones, functionality design, usability testing as well as project management. These skills as well as the experience gained throughout the project period have strengthened the competences of the group members, giving them a wider understanding of the use and possibilities of geoinformatic technology.

15 PERSPECTIVE

As it has been stated in the report, the final product developed for this project was close to being fully operational in terms of the implemented functionality and overall design of the app. It has also been stated in the report that there still are several issues that need to be resolved. However, the implemented technology and the way that the underlying system has been structured, is of a level that could be used as the foundation of a professional product. On the other hand, even if all the technology and functionality of the service had been completed, additional data content still has to be collected, before the service can be released for public use. Only the data coverage of Assistens Cemetery was close to being complete, while the remaining cemeteries had little data attached, and the majority of the cemeteries in Denmark were not present in the app at all. To acquire and structure this data is the real challenge in a service like this, and if the ultimate goal of the service is to have a nationwide coverage of historical persons' graves, a lot of work is left.

The reason for this is that the process of acquiring all the data would require a lot of manual work. The location of the burial plots would likely be easy to acquire in digital form and in general the different cemetery managers do not seem to have any problems with releasing this kind of information. However, when it comes to collecting specific information about historical persons, such as where they are buried some cemetery managers do not have the information digitally or it is time consuming to extract information only for historical persons, while others are not willing to share the data, which complicates the process. In addition to this, it is necessary to construct networks used for directions in each cemetery. This is difficult because not all paths and roads at every cemetery are mapped, and the data regarding paths that are mapped, might not reveal if the path is only for pedestrians or bicyclists as it is the case with some of the FOT data used in this project. Because of these obstacles and time consuming tasks, a nationwide service like the one planned in this project is not likely to be released in the near future. It would definitely be technical possible, but with the work that would have to be invested, it would require a considerable financial support.

Even though a nationwide service is not likely to be developed, the prospects for a product only covering Assistens Cemetery are very different. Assistens Cemetery has a unique position with the attached culture centre which is financially supported from various foundations, lately with 1,560,000 DKK for development of 25 short films about historical persons buried at the cemetery. A project like the one documented in this report, with a focus on historical persons, could very likely be realised in cooperation with this culture centre. Here the app could complement the existing services provided by the culture centre, by serving as a digital guide to select tours and explore the cemetery.

Another direction that the developed product could follow with few adjustments, could be to focus on searching for common people's graves instead. The research performed in the defining stage showed that there is a market for a product with such a purpose, but here the problem would once more become the process of acquiring data from the cemetery managers. In order to develop a national service, all cemeteries in Denmark would have to be included in the service. As it was discovered, the general opinion of Aarhus Municipality regarding when to release geographical data is to do so if they believe that the data can be used to create a service that will benefit the citizens. This opinion is likely to be shared with most of the municipalities in Denmark, while some of the local cemeteries managed by parochial church councils, on the other hand might not be of the same conviction, instead valuing privacy of the deceased more. Additionally, these cemeteries are part of a larger hierarchical system, meaning that the parochial church council managed cemeteries unlike the municipalities might not be able to decide if they can release data by themselves. Instead they would have to ask permission up through the hierarchy, where many levels of church authorities could put a stop to the release of the data. This could result in having some of the cemetery offices denying the usage of their cemetery data. If this would be the case, the national service could end up not having data for all Danish cemeteries which would damage the overall impression of the service. In this case the users would not be sure if the cemeteries that they wanted to explore would be available in the service, which in turn could reduce the popularity of the app.

As the research throughout the project has shown, it unfortunately seems that there is a high degree of self restriction from the cemetery administrations that makes it difficult to make cemetery data publically available. Furthermore it remains unclear if a central registry of deceased persons exists and if it can be accessed, as not even the central cemetery authorities, the IT department of the Evangelical Lutheran Church nor the Danish National Archives that were contacted could answer this with certainty. With the increased focus on collecting, digitalizing and publishing data to the public in general however, it will hopefully become possible in the near future to gain access to data regarding where people are buried without difficulties. Aarhus Municipality at least, has made information related to people buried at their cemeteries public, and other cemetery offices might follow this trend in the future. Until this change in data availability happens, the prospect of having a nationwide service for searching graves is years into the future, but the potential for the service exists today.

16 BIBLIOGRAPHY

Aarhus Municipality. http://gis.aarhus.dk/natur/kirkegaard/ (accessed February 25, 2013).

Bach, Erik, interview by Mathias Kofoed Poulsen. *Cemetery Administrator, Aarhus Municipality* (10 May 2013).

Baeza-Yates, Ricardo, and Gonzalo Navarro. "A faster algorithm for approximate string matching." *Combinatorial Pattern Matching 96*, 1996: 1-23.

Balstrøm, Thomas, Ole Jacobi, and Lars Bodum. *GIS og Geodata.* København: Forlaget GIS og Geodata, 2006.

Barnes, David J. *Object-Oriented Programming with JAVA An Introduction.* Department of Computer Science University of Kent at Centerbury, 2000.

Bastien, J. M. C. "Usability testing: some current practices and research questions." *The International Journal of Medical Informatics*, 2010.

Boehm, Barry W. *A Spiral Model of Software Development and Enhancement.* 1988.

Boehm, Barry, and Richard Turner. *Observations on Balancing Discipline and Agility.* 2003. http://people.cs.aau.dk/~jeremy/SOE2011/resources/Boehm.pdf (accessed February 10, 2013).

Brandsoft, interview by Mathias Kofoed Poulsen. *Sales Manager* (13 May 2013).

Brodersen, Lars. *Geokommunikation.* Forlaget Tankegang a-s, 2007.

Citizen Service, City of Copenhagen, interview by Mathias Kofoed Poulsen. (13 May 2013).

Danish Ministry of the Environment. *Frie data gavner miljø og vækst.* 2012. http://www.mim.dk/Nyheder/2012/20121012_grunddata.htm (accessed May 05, 2013).

Danish National Archives. "Interview by Mathias Kofoed Poulsen." 13 May 2013.

Datatilsynet. 2007. http://dk-gravsten.dk/images/data.pdf (accessed February 12, 2013).

Eclipse Foundation. 2013. http://www.eclipse.org (accessed May 20, 2013).

Folkekirke IT. "Interview by Mathias Kofoed Poulsen." 13 May 2013.

Frantzell, Nils-Erik. *Install XAMPP for easy, integrated development.* 30 November 2004. http://www.ibm.com/developerworks/linux/library/l-xampp/ (accessed May 11, 2013).

Fu, Pinde, and Jiulin Sun. *Web GIS, Principles and Applications.* Redlands: ESRI PRESS, 2011.

Fuglsang, Morten, interview by Mathias Kofoed Poulsen and Peter Christian von Westphal Thomsen. (18 February 2013).

GeoServer. *Welcome.* http://geoserver.org/display/GEOS/Welcome (accessed May 13, 2013).

Gregersen, Ole, and Ian Wisler-Poulsen. *Usability.* København: Forlaget Grafisk Litteratur, 2009.

Hansen, John Paulin. *Forelæsningsnoter: Usability.* IT-Universitetet, 2005.

Jensen, Daniel Ossipoff, and Sofie Gjedde-Simonsen, interview by Mathias Kofoed Poulsen and Peter Christian von Westphal Thomsen. (11 February 2013).

John, Tony. *Semantic Search and how it works with Google search.* http://www.techulator.com/resources/5933-What-Semantic-Search.aspx (accessed May 20, 2013).

Kilhof, Bente, interview by Mathias Kofoed Poulsen. *Head of Divison, Center for Cemeteries* (5 June 2013).

Krug, Steve. *Don't Make Me Think, A Common Sense Approach to Web Usability.* New Riders, 2006.

Kulturcentret Assistens. *Midt i verden i 250 år – Assistens 1760 – 2010.* 2013. http://assistens.dk/butikkens-boger-og-kataloger/ (accessed April 27, 2013).

Kuniawsky, Mike. *Observing the User Experience.* Morgan Kaufmann publishers, 2003.

Larson, Erik W, and Clifford F Gray. *Project Management, The managerial process.* New York: McGraw-Hill/Irwin, 2011.

Lewis, C. H. *Thinking Aloud, method in cognitive interface design.* Yorktown Heights: Watson Research Center, 1982.

Lunding, Gitte, interview by Mathias Kofoed Poulsen. *Centre manager at Assistens Culture Centre* (30 April 2013).

Mashable.com. 2012. http://mashable.com/2012/06/07/mockup-tools/ (accessed 02 15, 2013).

Ministry for Ecclesiastical Affairs. *Bekendtgørelse af lov om begravelse og ligbrænding.* 2010.

Ministry for Economic Affairs and the Interior. *Forslag til Lov om ændring af forskellige love som følge af indførelse af obligatorisk digital selvbetjening for borgere.* 2012.

Ministry for Gender Equality and Ecclesiastical Affairs . *Bekendtgørelse af lov om folkekirkens kirkebygninger og kirkegårde.* 2009.

Ministry for Gender Equality and Ecclesiastical Affairs. http://miliki.dk/kirke/andre-trossamfund/begravelsespladser/ (accessed May 08, 2013).

—. "Urnenedsættelse på privat ejendom." http://miliki.dk/kirke/livets-begivenheder/doedsfald/kirkegaardsvalg/urnepaaprivatgrund/ (accessed February 25, 2013).

Ministry of Culture. *Bekendtgørelse af arkivloven.* 2007.

Ministry of Ecclesiastical Affairs. *Vejledningen Kirker og Kirkegårde.* Kirkeministeriet, 2001.

Ministry of Environment. "Vejledning om planloven." 2009.

Ministry of Justice. *Lov om behandling af personoplysninger.* 2000.

Ministry of the Interior and Health. *Bekendtgørelse af lov om Det Centrale Personregister.* 2006.

Obe, Regina O, and Leo S Hsu. *PostGIS in Action.* Stamford: Manning Publications Co., 2011.

OGC. *Web Map Service.* 16 November 2012. http://www.opengeospatial.org/standards/wms (accessed May 13, 2013).

personregistrering.dk. https://www.personregistrering.dk/index.php?id=95 (accessed February 20, 2013).

pgRouting. *pgRouting Documentation.* http://pgrouting.org/docs/1.x/index.html (accessed May 3, 2013).

PHP.net. 2013. http://php.net/ (accessed February 27, 2013).

PostgreSQL. 2013. http://www.postgresql.org/docs/ (accessed March 9, 2013).

—. *F.35 pg_trgm.* http://www.postgresql.org/docs/9.1/static/pgtrgm.html (accessed April 10, 2013).

Python. 2012. http://wiki.python.org/moin/PythonVsPhp (accessed February 27, 2013).

Rogers, Yvonne, Helen Sharp, and Jenny Preece. *Interaction Design.* Crawfordsville: John Wiley and Sons ltd, 2002.

Shekhar, Shashi, and Sanjay Chawla. *Spatial Databases, a Tour.* Upper Saddle River: Prentice Hall, 2003.

Spool, J., and W. Schroeder. "Testing Web sites: five users in nowhere near enough." *Proceedings of the Conference extended abstracts on Human Factors in Computing Systems*, 2001.

The Culture Centre Assistens. 2013. http://assistens.dk/ (accessed May 01, 2013).

The Government. *Den Digitale Vej Til Fremtidens Velfærd.* Regeringen, 2011.

Timaru District Council. 2013. http://www.timaru.govt.nz/cemetery-database2.html (accessed February 16, 2013).

—. 2013. http://www.timaru.govt.nz/dmdocuments/Geraldine_Sheet2.pdf (accessed February 16, 2013).

Trice, Andrew. *Adobe.* 29 October 2012. http://www.adobe.com/devnet/phonegap/articles/apple-application-rejections-and-phonegap-advice.html (accessed April 20, 2013).

Master thesis in M. Sc. Geoinformatics

Aalborg University - Copenhagen, June 2013

17 APPENDIX

Appendix A. TELEPHONE CONVERSATIONS WITH CEMETERY MANAGERS

Telephone interviews conducted on the 7th of February by Mathias Kofoed Poulsen

Overview of relevant aspects/advantage/disadvantage in relation to increased digitalisation based on five interviews with cemetery managers from:

Holmens Kirkegård: Cemetery manager Birgitte Foghmoes**,**

Farum Kirkegård: Cemetery manager Brian Worm Jensen**,**

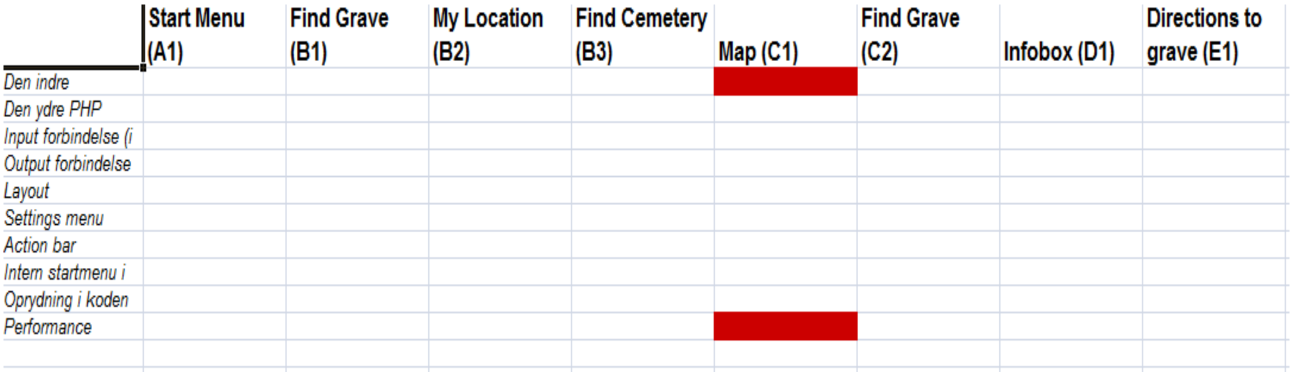
Slagelse Kirkegård: Cemetery manager Bent Nielsen**,**

Hørsholm Kirkegård: Cemetery manager Vagn Andersen

Herning Kirkegård: Secretary Anne Lise Thomsen

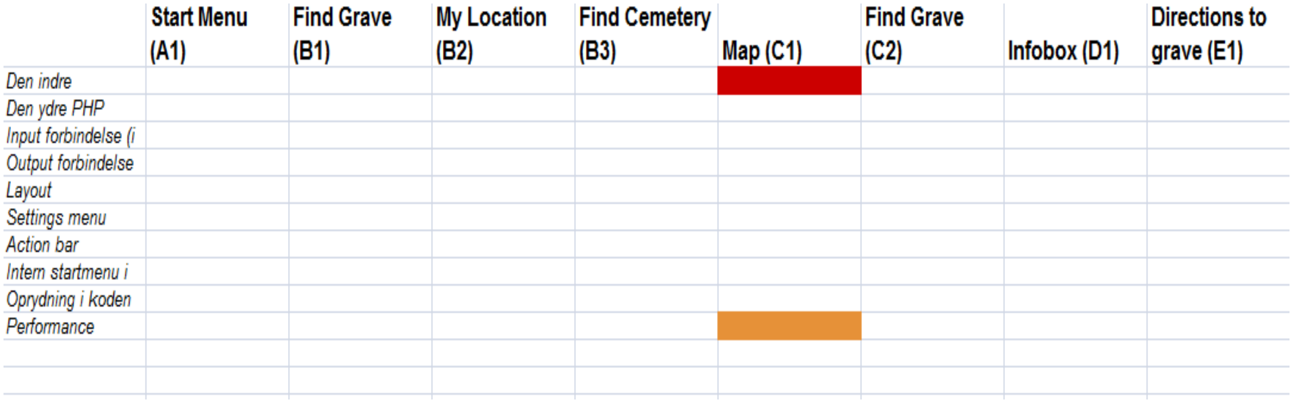
- Larger administrative overview, more simple case work and less resource demanding in the long run
- Cemeteries are planned and managed for many years in the future, it is no easy task to identify which graves are vacant
- Graves worthy of preservation
- Brandsoft has almost monopoly for cemetery related software
- A digital solution to locate historical/famous people will fit with the present state of society with that there are many people who are famous right now but maybe not in twenty years time. It is easier to update a digital service with these data than to create new physical maps all the time.
- Grief process and choice of grave. The process of a dead relative starts with the undertaker, then a selection of the gravestone and final the selection of the actual burial plot. This whole process can be a way of handling the grief for many, and the relatives get exhausted in the end. Without prober guidance to choose the right burial plot is it possible for the relatives to make a wrong choice, and later on regret the burial plot and want to move the grave to a different burial plot. With a digital service this problem would according to the cemetery managers probably be very likely to occur. It is important with an initial tour around the cemetery before specific burial plots are shown, in order to sense the locations atmosphere.

Appendix B. WEEKLY PROGRESS



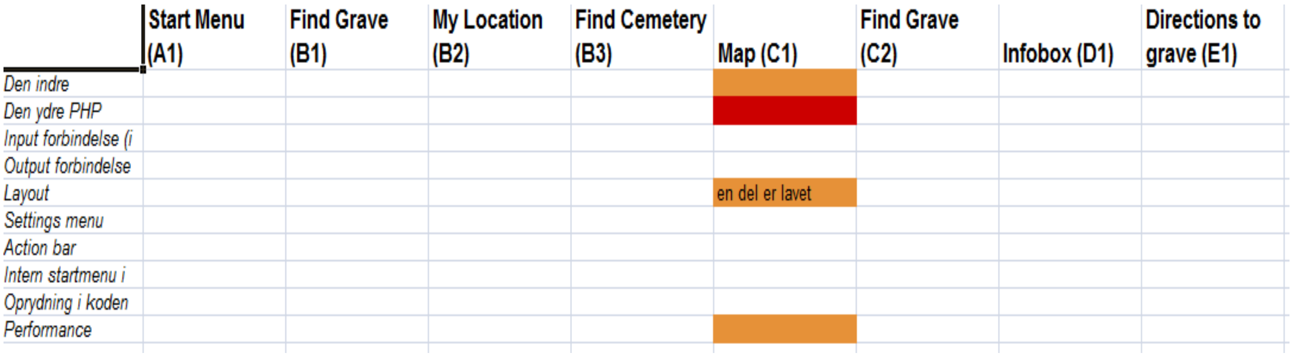
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre					🟥			
Den ydre PHP								
Input forbindelse (i								
Output forbindelse								
Layout								
Settings menu								
Action bar								
Intern startmenu i								
Oprydning i koden								
Performance					🟥			

Figure 102 Week 1



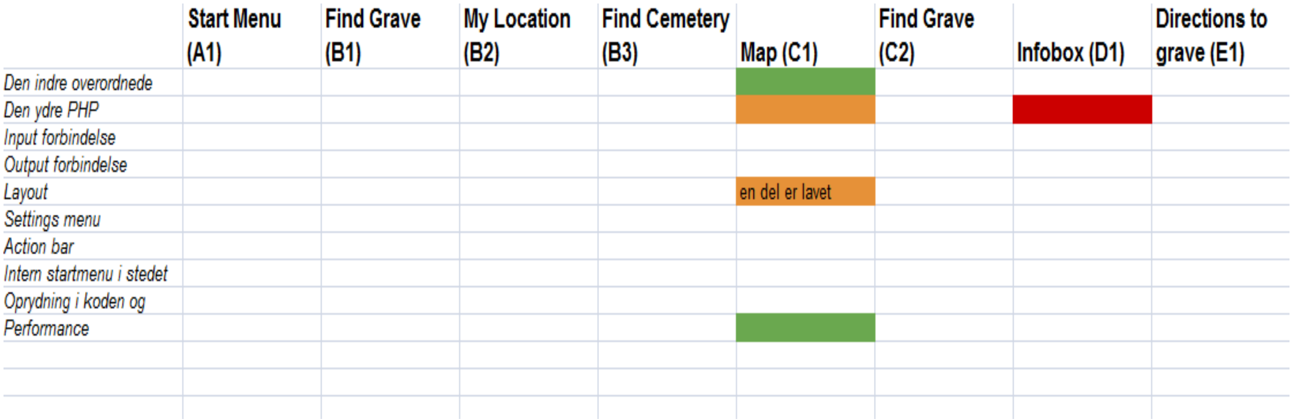
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre					🟥			
Den ydre PHP								
Input forbindelse (i								
Output forbindelse								
Layout								
Settings menu								
Action bar								
Intern startmenu i								
Oprydning i koden								
Performance					🟧			

Figure 103 Week 2



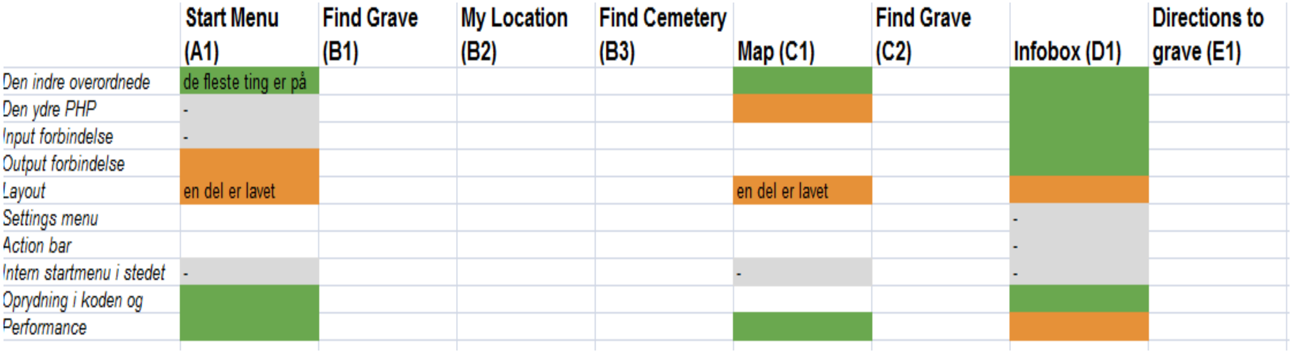
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre					🟧			
Den ydre PHP					🟥			
Input forbindelse (i								
Output forbindelse								
Layout					🟧 en del er lavet			
Settings menu								
Action bar								
Intern startmenu i								
Oprydning i koden								
Performance					🟧			

Figure 104 Week 3



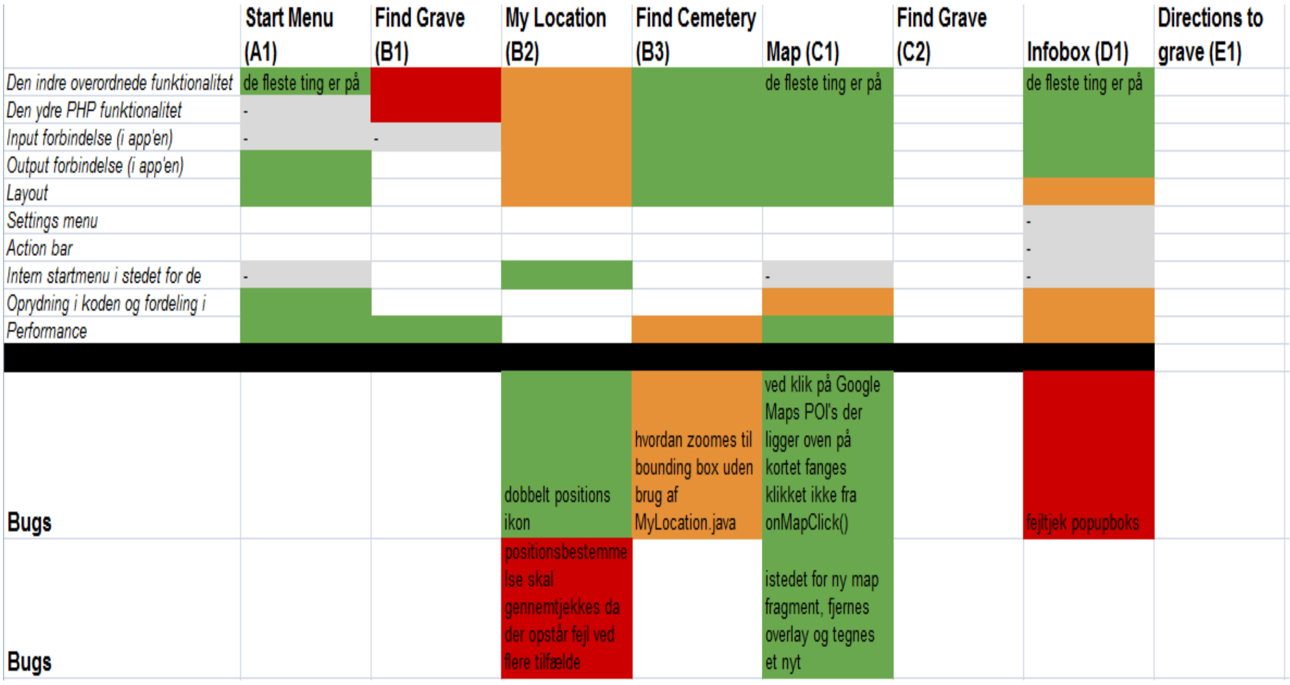
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre overordnede					🟩			
Den ydre PHP					🟧		🟥	
Input forbindelse								
Output forbindelse								
Layout					🟧 en del er lavet			
Settings menu								
Action bar								
Intern startmenu i stedet								
Oprydning i koden og								
Performance					🟩			

Figure 105 Week 4



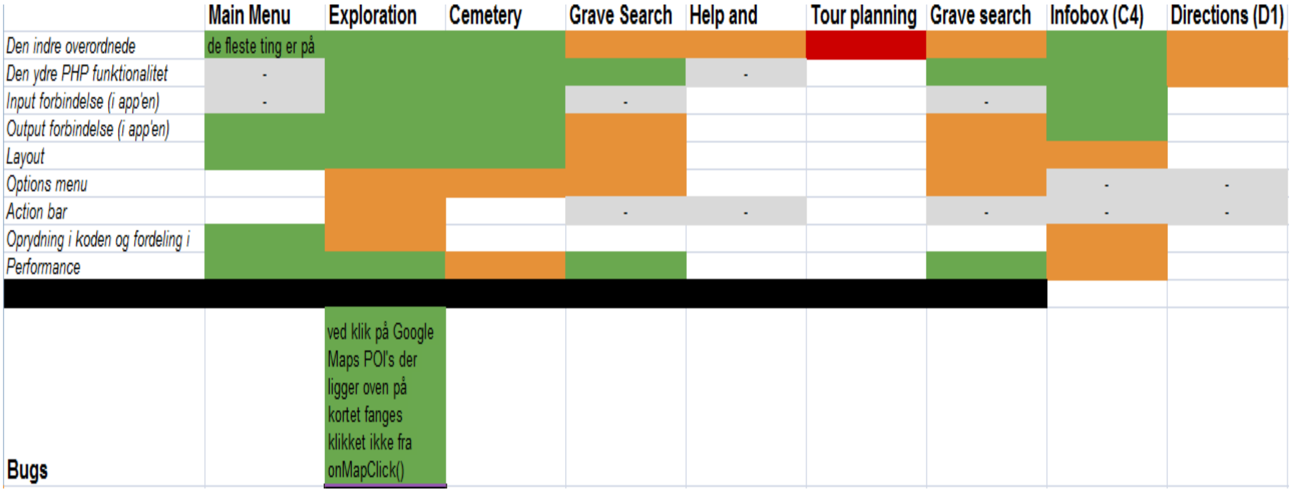
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre overordnede	🟩 de fleste ting er på				🟩		🟩	
Den ydre PHP	-				🟧		🟩	
Input forbindelse	-						🟩	
Output forbindelse	🟧						🟩	
Layout	🟧 en del er lavet				🟧 en del er lavet		🟧	
Settings menu							-	
Action bar							-	
Intern startmenu i stedet	-				-		-	
Oprydning i koden og	🟩						🟩	
Performance	🟩				🟩		🟧	

Figure 106 Week 5



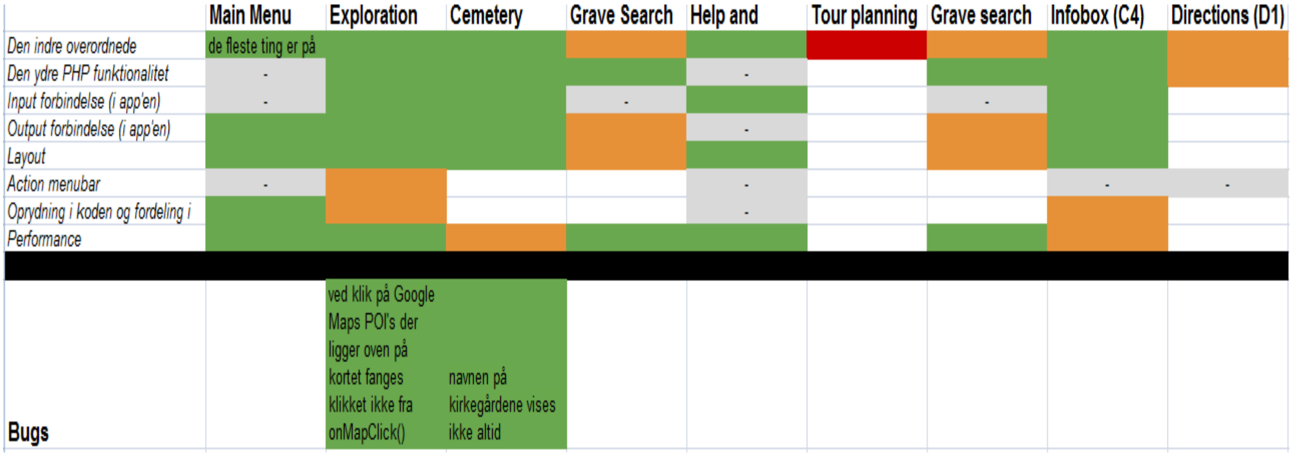
	Start Menu (A1)	Find Grave (B1)	My Location (B2)	Find Cemetery (B3)	Map (C1)	Find Grave (C2)	Infobox (D1)	Directions to grave (E1)
Den indre overordnede funktionalitet	🟩 de fleste ting er på	🟥	🟧	🟩	🟩 de fleste ting er på		🟩 de fleste ting er på	
Den ydre PHP funktionalitet	-	🟥	🟧	🟩	🟩		🟩	
Input forbindelse (i app'en)	-	-	🟧	🟩	🟩		🟩	
Output forbindelse (i app'en)	🟩		🟧	🟩	🟩		🟩	
Layout	🟩		🟧	🟩	🟩		🟧	
Settings menu							-	
Action bar							-	
Intern startmenu i stedet for de	-		🟩		-		-	
Oprydning i koden og fordeling i	🟩				🟧		🟧	
Performance	🟩	🟩		🟧	🟩		🟧	
Bugs			🟩 dobbelt positions ikon	🟧 hvordan zoomes til bounding box uden brug af MyLocation.java	🟩 ved klik på Google Maps POI's der ligger oven på kortet fanges klikket ikke fra onMapClick()		🟥 fejltjek popupboks	
Bugs			🟥 positionsbestemmelse skal gennemtjekkes da der opstår fejl ved flere tilfælde		🟩 istedet for ny map fragment, fjernes overlay og tegnes et nyt			

Figure 107 Week 6



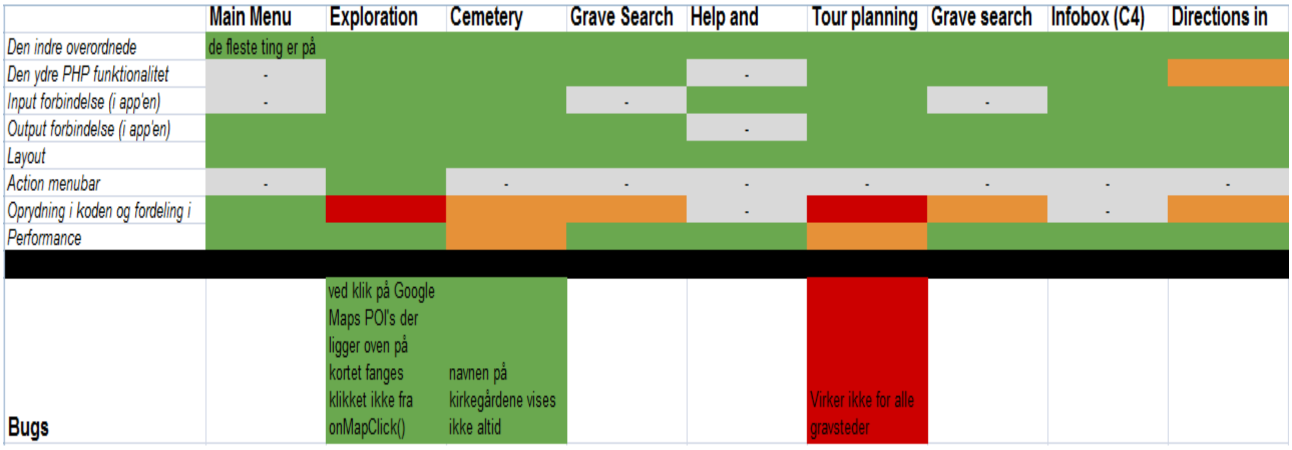
	Main Menu	Exploration	Cemetery	Grave Search	Help and	Tour planning	Grave search	Infobox (C4)	Directions (D1)
Den indre overordnede	🟩 de fleste ting er på	🟩	🟩	🟧	🟧	🟥	🟧	🟩	🟧
Den ydre PHP funktionalitet	-	🟩	🟩	🟩	-		🟩	🟩	🟧
Input forbindelse (i app'en)	-	🟩	🟩	-			-	🟩	
Output forbindelse (i app'en)	🟩	🟩	🟩	🟧			🟧	🟩	
Layout	🟩	🟩	🟩	🟧			🟧	🟧	
Options menu		🟧	🟧	🟧			🟧	-	-
Action bar		🟧		-	-		-	-	-
Oprydning i koden og fordeling i	🟩	🟧						🟧	
Performance	🟩	🟩	🟧	🟩			🟩	🟧	
⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛		
Bugs		🟩 ved klik på Google Maps POI's der ligger oven på kortet fanges klikket ikke fra onMapClick()							

Figure 108 Week 7



	Main Menu	Exploration	Cemetery	Grave Search	Help and	Tour planning	Grave search	Infobox (C4)	Directions (D1)
Den indre overordnede	🟩 de fleste ting er på	🟩	🟩	🟧	🟩	🟥	🟧	🟩	🟧
Den ydre PHP funktionalitet	-	🟩	🟩	🟩	-		🟩	🟩	🟧
Input forbindelse (i app'en)	-	🟩	🟩	-	🟩		-	🟩	
Output forbindelse (i app'en)	🟩	🟩	🟩	🟧	-		🟧	🟩	
Layout	🟩	🟩	🟩	🟧	🟩		🟧	🟩	
Action menubar	-	🟧			-			-	-
Oprydning i koden og fordeling i	🟩	🟧			-			🟧	
Performance	🟩	🟩	🟧	🟩	🟩		🟩	🟧	
⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
Bugs		🟩 ved klik på Google Maps POI's der ligger oven på kortet fanges klikket ikke fra onMapClick()	🟩 navnen på kirkegårdene vises ikke altid						

Figure 109 Week 8



	Main Menu	Exploration	Cemetery	Grave Search	Help and	Tour planning	Grave search	Infobox (C4)	Directions in
Den indre overordnede	🟩 de fleste ting er på	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Den ydre PHP funktionalitet	-	🟩	🟩	🟩	-	🟩	🟩	🟩	🟧
Input forbindelse (i app'en)	-	🟩	🟩	-	🟩	🟩	-	🟩	🟩
Output forbindelse (i app'en)	🟩	🟩	🟩	🟩	-	🟩	🟩	🟩	🟩
Layout	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Action menubar	-	🟩	-	-	-	-	-	-	-
Oprydning i koden og fordeling i	🟩	🟥	🟧	🟧	-	🟥	🟧	-	🟧
Performance	🟩	🟩	🟧	🟩	🟩	🟧	🟩	🟩	🟩
⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
Bugs		🟩 ved klik på Google Maps POI's der ligger oven på kortet fanges klikket ikke fra onMapClick()	🟩 navnen på kirkegårdene vises ikke altid			🟥 Virker ikke for alle gravsteder			

Figure 110 Week 9

Appendix C. PROJECT DEVELOPMENT DIARY

February 18th: Development start

February 20th: Android Google Maps API v1: https://developers.google.com/maps/documentation/android/v1/hello-mapview

February 22th: GeoServer setup

February 25th: Android Google Maps API v1 with a WMS overlay, too slow

http://androidgismap.blogspot.dk/2011/07/adding-wms-to-google-maps.html

February 27th: Android Google Maps API v2 with a WMS overlay

http://www.azavea.com/blogs/labs/2013/01/wms-on-android/

February 28th: Modifying the emulator to run Google Play Services apps

http://stackoverflow.com/questions/13691943/this-app-wont-run-unless-you-update-google-play-services-via-bazaar/13734937#comment19100235_13869332

http://nemanjakovacevic.net/blog/2012/12/how-to-make-android-google-maps-v2-work-in-android-emulator/

March 6th: PostgreSQL/PostGIS->GeoServer->WMS different colour based on values from database

March 8th: Go to my location with map

http://www.vogella.com/articles/AndroidLocationAPI/article.html#locationapi

March 10th: send/receive data Android->PHP->PostgreSQL->PHP->Android

Android

http://webtutsdepot.com/2011/11/15/android-tutorial-how-to-post-data-from-an-android-app-to-a-website/

http://adilmukarram.wordpress.com/2011/01/29/sending-and-receiving-data-from-a-php-web-application/

PHP/PostgreSQL

http://www.techrepublic.com/blog/howdoi/how-do-i-use-php-with-postgresql/110

http://www.sitepoint.com/accessing-postgresql-php/

http://unserializableone.blogspot.com/2007/02/using-postgis-to-find-points-of.html

March 11th: Touch event toast and textview

http://stackoverflow.com/questions/14013002/google-maps-android-api-v2-detect-touch-on-map-solved og http://android-er.blogspot.dk/2013/01/google-maps-android-api-v2-example.html

LatLng, get latitude and longitude: http://stackoverflow.com/questions/15264884/getting-latitude-and-longitude-from-latlng-object-in-android and https://developers.google.com/maps/documentation/android/reference/com/google/android/gms/maps/model/LatLng

March 12th: Closest objects problem, PostGIS

PostGIS In Action page 136

March 13th: GeoServer Tile Caching (GeoWebCache: http://docs.geoserver.org/stable/en/user/geowebcache/index.html)

digitising historical graves for Assistens Cemetery

March 14th: UTF-8

UTF-8 (to/from PHP<->Android): http://stackoverflow.com/questions/10029193/special-chars-between-android-and-php

UTF-8 from Shape File to PostGIS Importer to database:

http://postgis.17.n6.nabble.com/converting-to-postgis-td3570432.html

ArcGIS enable UTF-8 tables support: http://support.esri.com/es/knowledgebase/techarticles/detail/21106

March 15th: Wikipedia extract image link and rescale

http://en.wikipedia.org/w/api.php

http://da.wikipedia.org/w/api.php?action=parse&page=H.C._Andersen&prop=images&format=jsonfm

http://en.wikipedia.org/w/api.php?action=query&titles=File:HCA_by_Thora_Hallager_1869.jpg&prop=imageinfo&iilimit=1&iiprop=url&iiurlwidth=150

March 18th: wikipedia extract plain text first paragraph

http://da.wikipedia.org/w/api.php?action=query&prop=extracts&exchars=1000&titles=H.C._Andersen&format=jsonfm

March 19th: PopUp Window and scroll view

pop up window

http://android-er.blogspot.dk/2012/03/example-of-using-popupwindow.html layout with dynamical changes: http://stackoverflow.com/questions/7299962/change-text-of-textview-in-popupwindow-not-working scroll window

http://davidwparker.com/2011/08/25/android-fixing-window-resize-and-scrolling/

March 20th: JSON send receive

http://codeoncloud.blogspot.dk/2012/11/android-php-json-tutorial.html

lock screen to portrait:

android:configChanges="keyboardHidden|orientation", android:screenOrientation="portrait"

March 21st: Wiki titles for historical graves

March 22nd: Wiki-data into database. Various methods for data-gathering to speed up popup_window.

- PHP include wiki PHP files
- PHP with all code
- Wiki data in database

No significant difference in speed

March 24th: Starting new activities and some layout/design

Remove title bar: http://stackoverflow.com/questions/2591036/how-to-hide-the-title-bar-for-an-activity-in-xml-with-existing-custom-theme

background image: http://stackoverflow.com/questions/10679366/android-background-image-to-fill-screen

starting a new activity: http://developer.android.com/training/basics/firstapp/starting-activity.html

March 25th: GeoServer labels

GeoServer labels: http://docs.geoserver.org/stable/en/user/styling/sld-reference/labeling.html?highlight=labelplacement

problem with labels not shown for the same zoom levels: http://osgeo-org.1560.n6.nabble.com/labels-omitted-when-they-would-extend-across-tiles-td4995215.html

March 26th: separate get location class

get location class: http://stackoverflow.com/questions/3145089/what-is-the-simplest-and-most-robust-way-to-get-the-users-current-location-in-a

March 27th: gotoGraveyard, set zoomlevel from lat/lon-bounds

http://stackoverflow.com/questions/5114710/android-setting-zoom-level-in-google-maps-to-include-all-marker-points

March 28th: options menu

options menu for changing location info

http://mobileorchard.com/android-app-development-menus-part-1-options-menu/

March 30th: Wiki text in database: minimise load time for popup box

dynamically change options menu: use onPrepareOptionsMenu

hide button: http://stackoverflow.com/questions/6173400/how-to-programmatically-hide-a-button-in-android-sdk

Control action on back key pressed: http://stackoverflow.com/questions/3141996/android-how-to-override-the-back-button-so-it-doesnt-finish-my-activity and http://androiddev.orkitra.com/?p=30513

March 31st: "ingengravsteder_i_nærheden" image
http://us.123rf.com/400wm/400/400/jara3000/jara30001209/jara3000120900004/15070839-hand-cursor-clicks.jpg

search for one person in database

April 1st: search results as list, hide keyboard

list results http://www.ezzylearning.com/tutorial.aspx?tid=1351248

http://www.vogella.com/articles/AndroidListView/article.html

http://stackoverflow.com/questions/11466188/android-how-to-get-the-json-array-to-the-listview

hide keyboard: http://stackoverflow.com/questions/10550290/best-way-to-hide-keyboard-in-android

April 2nd: fragment layout, network on main thread

fragment layout: http://stackoverflow.com/questions/14694119/how-to-add-buttons-at-top-of-map-fragment-api-v2-layout

network call in Main UI Thread, workaround - a better solution would be to use AsyncTask

http://stackoverflow.com/questions/13136539/caused-by-android-os-networkonmainthreadexception

April 3th: pgRouting

install: https://github.com/sanak/pgrouting4w/downloads

og http://gis.stackexchange.com/questions/26816/installing-pgrouting-in-windows-7

enable a-star: http://workshop.pgrouting.org/chapters/shortest_path.html beginner guide: http://anitagraser.com/2011/02/07/a-beginners-guide-to-pgrouting/

php kald:
http://workshop.pgrouting.org/chapters/php_server.htmlhttp://anitagraser.com/2011/02/07/a-beginners-guide-to-pgrouting/

http://anitagraser.com/2011/02/07/a-beginners-guide-to-pgrouting/

April 4th, 5th and 6th: pgRouting: PHP and SQL

poly line in android: http://stackoverflow.com/questions/13699856/google-maps-android-api-v2-route-overlay

April 7th: icon, layout, new simple location, fix bounding box bug

bounding box bug: http://stackoverflow.com/questions/13702117/how-can-i-handle-map-move-end-using-google-maps-for-android-v2

check for no location providers: http://stackoverflow.com/questions/10311834/android-dev-how-to-check-if-location-services-are-enabled

April 8th: install pg_trgm and fuzzy string search, install: CREATE EXTENSION pg_trgm

http://www.postgresql.org/docs/9.1/static/pgtrgm.html

April 10th: marker icons, animate to marker when click on map, fix bug with pg_trgm for øæå (uft8_encode()), background for popup box

http://upload.wikimedia.org/wikipedia/commons/3/3e/Wikipedia-logo-v2-da.png

April 11th: icon different screen size and resolution

April 12th: fix popupwindow size to wrap content (-2), legend

April 13th: legend, main help box

April 14th: walking and bicycling pgrouting implementation

polyline to a list http://stackoverflow.com/questions/14853084/how-to-remove-all-the-polylines-from-a-map

April 15th: small bug fixes, testmode

April 16th: Tour Planner

dynamically add textview to linearlayout: http://stackoverflow.com/questions/13969878/click-event-on-dynamically-added-textviews-in-android

set margins for twxtview http://stackoverflow.com/questions/3696843/how-to-set-textview-attributes-programmatically

April 17th: Tour Planner day2

Set focus: http://stackoverflow.com/questions/2743559/set-initial-focus-in-an-android-application

dynamically add elements to array: http://stackoverflow.com/questions/5144549/append-values-dynamically-into-an-long-array

April 18th: Tour Planner day3, layout

April 19th: Tour Planner, day4

send json http://www.codeproject.com/Articles/267023/Send-and-receive-json-between-android-and-php

http://stackoverflow.com/questions/4685534/android-json-to-php-server-and-back

tsp: http://gis.stackexchange.com/questions/20299/how-to-use-the-pgrouting-tsp-function
http://pgrouting.org/docs/1.x/tsp.html

April 20th: Tour Planner day5

asynctask: http://mobiledevtuts.com/android/android-http-with-asynctask-example/

April 21st: Tour Planner day6

remember php: utf8_encode

integrating tourplanner with directions, general functions in php file

April 22nd: network check,

http://stackoverflow.com/questions/6357603/problem-in-detecting-internet-connection-in-android

updating of user position: http://www.vogella.com/articles/AndroidLocationAPI/article.html

April 23th: Development end

May 3th: Tour Planner bug fixed,

split the routing between tours with four different nodes and tours with less nodes and included directions from road to grave it self

May 20th: fixed bug when closing popup not on main thread, from directions http://stackoverflow.com/questions/5161951/android-only-the-original-thread-that-created-a-view-hierarchy-can-touch-its-vi

Appendix D. First Desired Functionality Overview

Categories
Framework
- A map of cemetery showing the individual positions of the graves.
Basic Functions
- Function to search for a historical person with the result showing which cemetery and the position on the cemetery the person is buried.
Advanced Functions
- On the map different colours for historical persons' graves will be used based on their profession.
- Function to show a map of all the historical persons' graves on a specific cemetery.
- When a grave is clicked on the map an information window will show with information specific for this grave and the person who is buried there.
- Implementation of navigation to a grave on road/path level within the cemetery.
- The navigation functionality must be able to distinguish between driving, bicycling and walking mode.
Refinement
- The grave search can be enhanced by auto complete of the name and fuzzy/semantic search.
- Support of UTF-8 character set.
- Support for different screen sizes.
- The app should work smoothly without any performance problem.
- Any graphic problems should be solved.
- Any functionality problems should be solved.

Figure 111 The first draft of desired functionality for the app divided in categories responding to the four iteration cycles of the development

Appendix E. FIRST WIREFRAME AND SKETCHES DESIGNED

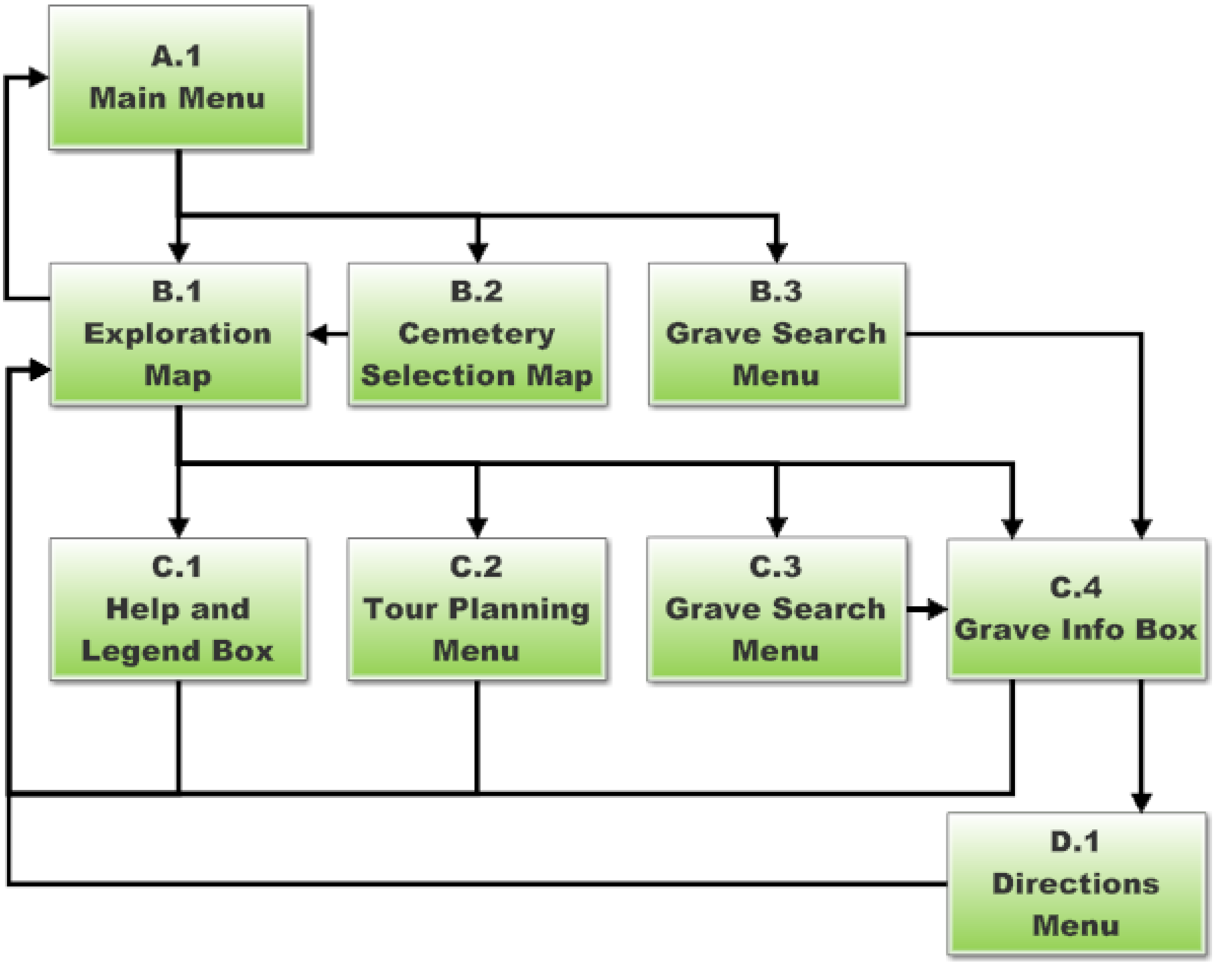


Figure 112 First wireframe designed

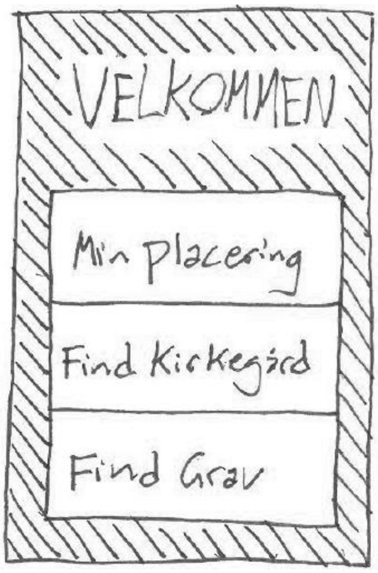


Figure 113 First main menu designed

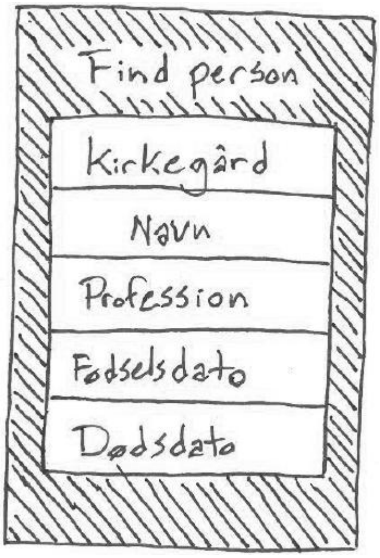


Figure 114 First search menu for all cemeteries

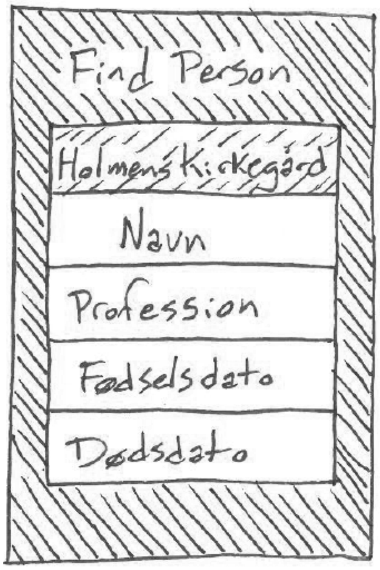


Figure 115 First search menu for a specific cemtery

Appendix F. Code Examples

Android Java code

The functionality of an Android application is defined through object oriented Java code. In Figure 116 a small part of code is shown. This piece of code gets executed every time the select cemetery function of the app is chosen, and sets the starting conditions for the activity. A specific XML styling file is chosen which defines the layout.

In the layout file a Google Map fragment is defined which contains information about how to incorporate the map, this fragment can then be connected to a variable (mMap) which now contains the map object that later on is used to control what the user sees on the screen.

The map object is controlled by changing its fields which contain information about how the map looks and acts like. Fields can be changed by calling the methods of an object. In the example the mMap object is changed using the moveCamera method to centre the map over a specific position and go to a certain zoom level.

There exist many predefined objects which are specific elements in an Android application. One of these elements is the ability to show a little message on top of the screen for a short period of time. This object is called Toast and the last line in the example it is shown how to have a short message shown on the screen.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //Use specific layout file
    setContentView(R.layout.activity_find_grave_yard);
    //Initialise Google Map variable mMap, using the map element from the
        layout file
    mMap = ((SupportMapFragment)
        getSupportFragmentManager().findFragmentById(R.id.map2)).getMap();
    //Move camera and zoom to level 6 at the centre of Denmark
    mMap.moveCamera( CameraUpdateFactory.newLatLngZoom(new
        LatLng(55.677, 11.601),6));
    //Display message on load

    Toast.makeText(mapContext, "Tryk på den kirkegård du vil udforske",

        Toast.LENGTH_LONG).show();
}
```

Figure 116 Example of a small part of Java code for an Android application.

Android XML Styling

The main part of the layout for an Android application is defined in XML styling files. In Figure 117 a small part of the XML styling that creates the main menu of the app is shown.

```
<TableRow
    android:id="@+id/tableRowhome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="4dp" >

    <ImageView
        android:id="@+id/imagesearch"
        android:layout_width="80dp"
        android:src="@drawable/barsearch" />

    <TextView
        android:id="@+id/textsearch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"

        android:text="Søg efter personers gravsteder uafhængig af placering i

                      landet på baggrund af søgekriterier"

        android:textColor="#000000" />
</TableRow>
```

Figure 117 Part of an Android XML styling file

XML files are supposed to be self descriptive using tags surrounded by < and >, between the start/end tags the actual data is written but can also be written as attributes for the individual tag which saves the end tag and this is how the Android XML files are composed.

The example shows one row which is part of a table that covers the whole screen view. This row contains two view objects. As the tags hint the objects are an image and a text field. The attributes defines information like width of the image and colour of the text.

PHP Scripts

PHP scripts are executed by the Apache webserver running on the server. They can perform numerous tasks, tasks that are too processor demanding for the smartphone and other tasks as connecting to the PostgreSQL database. In the developed service they are particular used for connecting to the database.

In Figure 118 a SQL query is defined which ask the database for the closest object from a position defined by lon/lat and within the distance dist. The structure for the SQL query is first to write the five information which we need: cemetery name, cemetery id, cemetery latitude, cemetery longitude and distance from the position. Then an inner query is made which results in a list of the cemeteries which are located within the distance dist from the point. The outer query orders the list by distance and limit the list to one, which then will be the closest cemetery.

The actual query is first made with the line "$result = pg_query($dbh, $sql);" where database connection from the variable $dbh is used. The results are looped over with the following while loop which save information about the closets cemetery in PHP variables.

```
...
$sql = 'SELECT name,
cemeteryid,
(point( transform(  the_geom ,4269 )))[1],
(point( transform(  the_geom ,4269 )))[0],
directionsmode,
ST_Distance(the_geom, loc.geom) As dist_m

FROM "cemeteries" inner join

(SELECT
ST_Transform(ST_SetSRID(ST_Point(' . $lon . ', ' . $lat . '), 4326),32632)
As geom) As loc
ON ST_DWithin(the_geom, loc.geom, '. $dist .')

ORDER BY ST_Distance(the_geom, loc.geom)
LIMIT 1;';

$result = pg_query($dbh, $sql);

while ($row = pg_fetch_array($result)) {
           $name = $row[0];
           $id = $row[1];
           $graveyardLat = $row[2];
           $graveyardLon = $row[3];
           $directionsMode = $row[4];
}
...
```

Figure 118 Example of a small part of the cemetery.php script

GeoServer XML Styling

When a WMS map is published using GeoServer the visual output is controlled by an XML file, in this file a number of rules are defined. In Figure 119 an example of such a rule is given. This rule changes how a grave will look like with the WMS.

In the example the first part <ogc:Filter> defines which data from the database should use this rule, and in this case it is polygons where the column profession contains the text KM. Every grave for a historical person in the database has a profession abbreviation connected, in this case komponist/musiker (KM) or in English composer/musician.

The last part <PolygonSymbolizer> defines the stroke and fill of the polygon for the grave. In this case a stroke with the colour #0ec23c, width 1, and the opacity from 0-1 where 0 is 0 % and 1 is 100%, so a non transparent green colour and a fill with the same green colour.

```
<Rule>
      <ogc:Filter>
            <ogc:PropertyIsEqualTo>
                  <ogc:PropertyName>profession</ogc:PropertyName>
                  <ogc:Literal>KM</ogc:Literal>

            </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <PolygonSymbolizer>
            <Stroke>
                  <CssParameter name="stroke">
                        <ogc:Literal>#0ec23c</ogc:Literal>
                  </CssParameter>

                  <CssParameter name="stroke-width">1
                  </CssParameter>
                  <CssParameter name="stroke-opacity">
                        <ogc:Literal>1.0</ogc:Literal>
                  </CssParameter>
            </Stroke>
            <Fill>
                  <CssParameter name="fill">#0ec23c
                  </CssParameter>
            </Fill>
      </PolygonSymbolizer>
</Rule>
```

Figure 119 GeoServer XML rule

Appendix G. OVERVIEW OF THE CODE

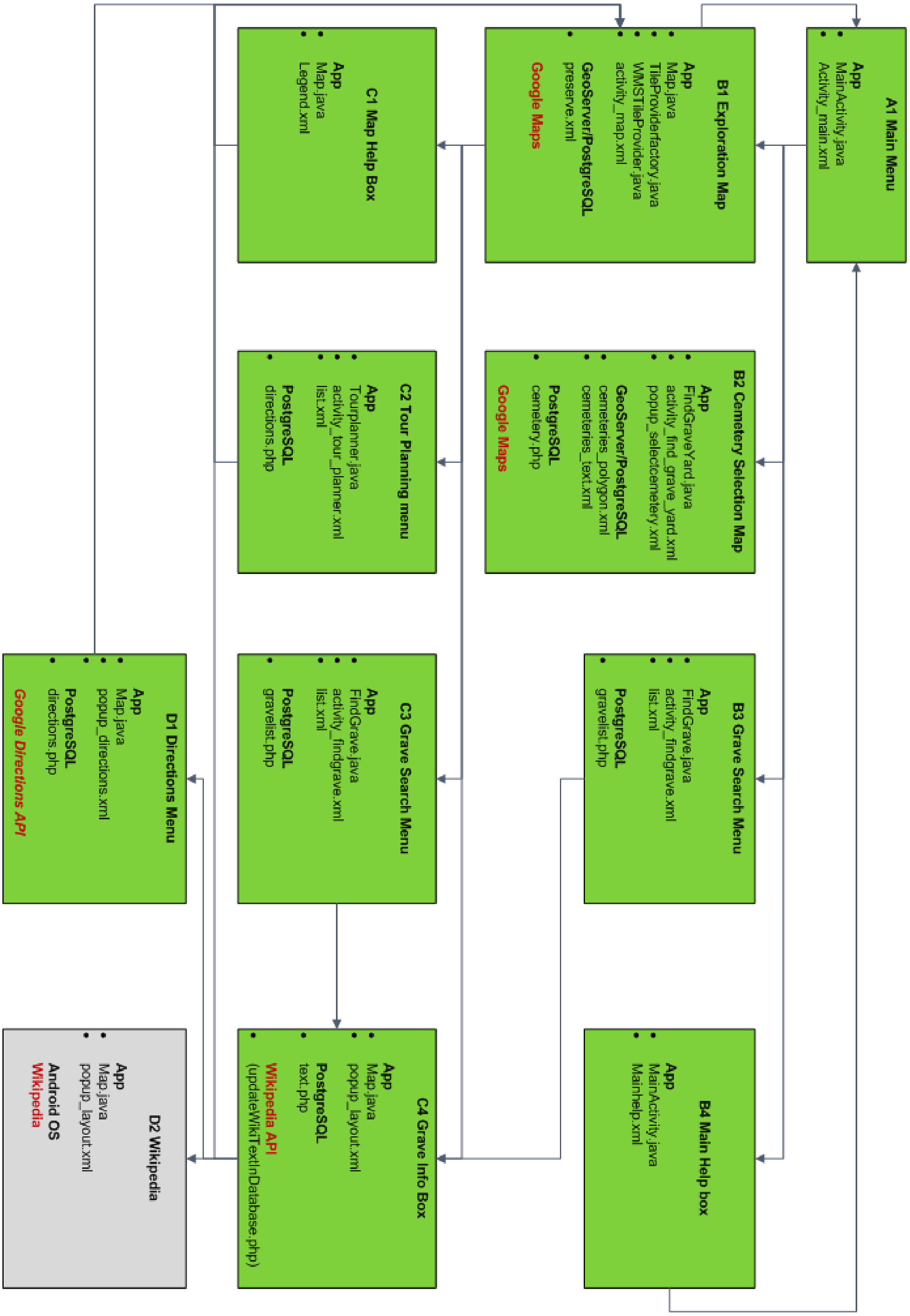
Type	File	Total	Comments	Code
PHP	cemetery.php	77	14	63
	directions.php	630	95	535
	grave.php	69	12	57
	text.php	87	18	69
	updateWikiInDatabase.php	147	36	111
		1010	175	835
GeoServer XML	cemeteries.xml	71	9	62
	cemeteries_text.xml	61	9	52
	preserve.xml	224	13	211
		356	31	325
Android Java	FindGrave.java	191	34	157
	FindGraveYard.java	236	48	188
	MainActivity.java	250	36	214
	Map.java	1180	191	989
	TourPlanner.java	545	87	458
	TileProviderFactory.java	84	6	78
	WMSTileProvider.java	58	9	49
		2544	411	2133
	AndroidManifest.xml	105	6	99
Android XML	activity_find_grave.xml	92	6	86
	activity_find_grave_yard.xml	33	2	31
	activity_main.xml	112	6	106
	activity_map.xml	105	3	102
	activity_tour_planner.xml	390	22	368
	legend.xml	331	2	329
	list.xml	8	1	7
	mainhelp.xml	140	2	138
	popup_directions.xml	94	2	92
	popup_layout.xml	113	2	111
	popup_selectcemetery.xml	44	3	41
		1462	51	1411
	Total	**5477**	**674**	**4803**

Appendix H. CD Containing Source Code

The content of the CD is structured around the three categories in the resource overview in Appendix I. The three categories are *App*, *GeoServer* and *PostgreSQL*. The *App* category is further subdivided between *App – Java* and *App – XML*. In addition to the three categories the CD also contains the graphics used in the App. The different numbers of files are shown here:

- App: 19 files
  - Java : 7 files
  - XML: 12 files
- GeoServer: 3 files
- Graphics: 36 files
- PostgreSQL: 5 files
- Report and Poster: 2 files

Appendix I. Main Functionality and Resources



Appendix J. USER TEST

Test subject: 1	**Age:** 25	**Smartphone Experience:** 3
Task 1	**Difficulty:** 1	**Time:** >1 minute
Comments: Chooses the search category. Writes name. Clicks on name. Finds grave. Easy		
Task 2	**Difficulty**: 4	**Time:**2 minutes
Comments: Clicks the home button. “Must be same as before” Chooses the search category. Writes name. Clicks name. Finds grave. Cannot find name of the cemetery. Clicks question mark. Expects to find the name of the cemetery by clicking the question mark. Cannot find it. Zooms out on the map and finds the name		
Task 3	**Difficulty:** 2	**Time:** 1 minute
Comments: Clicks the home button. Selects search option 3 – Search Cemeteries. Zooms to Bornholm. Finds cemetery and selects it. Enters exploration map of the cemetery. Is confused as to what is displayed. A lot of colours. What do the colours mean? What are graves at the map? Tries to click on a coloured square. Realises that it is a grave.		
Task 4	**Difficulty:** 5	**Time:** 3 minutes
Comments: Clicks the home button. Enters the information box for help. Tries to write Assistens Kirkegården in the search graves menu. No result. Enters Find Cemetery menu. Finds and click the cemetery. Presses the question mark. Clicks the tour button icon in the help menu, nothing happens. “Where is the button?” Cannot find it. Home. Enters nearest cemetery. Finds icon for tours and clicks on it. Clicks on the scientists’ tab. Cannot see that a list is open outside the screen view, and does not understand why nothing is happening. Clicks search. Finds names. Solves the task afterwards, but the app cannot create a tour. Names selected: Anne Margrete Stahll, Marie Schrøder, Frito Jürgensen, Herman Wilhelm Bissen.		
Task 5	**Difficulty:** 1	**Time:** >1 minute
Comments: Clicks home. Enters nearest cemetery. Clicks search button. Types in H.C. Andersen. Clicks name. Finds grave, and clicks route button. Done.		
General comments: Would like to know what the colours of the graves mean. The Tour planner is difficult to use.		

<table>
<tr><td><b>Test subject:</b> 2</td><td><b>Age:</b> 26</td><td><b>Smartphone Experience:</b> 3</td></tr>
<tr><td><b>Task 1</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> >1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Selects the Search option. Misses an auto complete name function. Types in name. Clicks on name. Finds grave. Has a feeling that the map is not done loading. Why is Ørsted located at several places? Because there are multiple blue dots. What does to colour mean? Would like to see the actual grave when zooming in on the map.</td></tr>
<tr><td><b>Task 2</b></td><td><b>Difficulty:</b> 2</td><td><b>Time:</b> >1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks on the search function on the exploration map view. Does not know how to spell name. Tries typing in Nicolaj. Finds the person. Clicks name. Believes that the colours are some kind of category after seeing the colour of Dajon's grave. Zooms out and finds the name of the cemetery.</td></tr>
<tr><td><b>Task 3</b></td><td><b>Difficulty:</b> 2</td><td><b>Time:</b> 1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks search in exploration map. Types in Sct. Knud. No result. Clicks the back button to the main menu. Selects Find Cemetery. Finds the cemetery and selects it. Clicks the question mark in hope of finding an index of the buried persons. Finds the colour legend. Understands the colours completely now. Tries to click on a grave. "Nice that it finds a grave even when you don't click directly on it"</td></tr>
<tr><td><b>Task 4</b></td><td><b>Difficulty:</b> 5</td><td><b>Time:</b> 3 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks home. Wonders why there was not a home button on the search cemetery map. Selects "nearest cemetery". Clicks tour planner. Clicks scientists' tab. "Why is nothing happening?" Thinks that Yellow and Orange, and Blue and Purple are very alike. The Tour Planner is confusing. Clicks the green tab. Something happens. "Why is this different than the blue tab? Why is something happening here?" Wonders if the names that are being clicked are added to a list. "How do I make a tour?" Click all the buttons available and ends up making a tour. The App does not show the route. The names selected was: Familien Bohr, Etta Cameron, Henry Heerup, Søren Gyldendal</td></tr>
<tr><td><b>Task 5</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> >1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks Search in the exploration map. Types in H.C. Andersen, and clicks name. Grave appears. A button that the test subject has not tried before is present. The route button. Clicks it. Would like an image to appear when zooming in. It would be nice with Google Street Map?</td></tr>
<tr><td colspan="3"><b>General comments:</b>
Miss title and welcome text. Too much text in the help functions. Won't bother reading it. It includes a lot of long words and is very technical. Spelling errors. Nice background in the main menu.</td></tr>
</table>

Test subject: 3	**Age:** 57	**Smartphone Experience:** 2
Task 1	**Difficulty:** 2	**Time:** 1 minute
Comments: Clicks search. Types in name. Does not know if you are supposed to click on the small search button to get a result. Finds name and finds grave.		
Task 2	**Difficulty:** 5	**Time:** 2 minutes
Comments: Clicks home. Clicks search. Tries to spell the name. Finds grave. Read the info box. "It does not say anything about where the person is buried". Does not know how to figure out what cemetery the person is buried at.		
Task 3	**Difficulty:** 1	**Time:** 1 minute
Comments: Clicks home. Select Search cemetery. Finds cemetery on Bornholm. Selects it. Clicks on the search button. Clicks the small search button. Gets a list with the names of the buried persons.		
Task 4	**Difficulty:** 4	**Time:** 4 minutes
Comments: Clicks home. Search Cemetery. Finds Assistens Kirkegården and selects it. Clicks question mark. Finds out that blue is scientists. Does not see the explanation for the tour planner button. Clicks home. Clicks search. Expects it to search for graves where you are at. Clicks help. Clicks home. Clicks Nearest cemetery. Finds the tour planner. Clicks the scientists' tab and scrolls down. Selects four names. "What do I do now?" "How do I continue?" Finds the list of selected graves. Clicks the Cycle icon. Creates tour.		
Task 5	**Difficulty:** 2	**Time:** 3 minutes
Comments: Clicks Search in Exploration map. Types in H. C. Andersen. Clicks name. Closes info box. Clicks home. Clicks nearest cemetery. Clicks Tour planner. Tour planner says that you need minimum four names for a tour. "Why do I have to choose four?" clicks search in exploration map. Finds H.C. Andersen. Closes info box. Clicks on the grave. Sees the route button in the info box. Presses it.		
General comments: Missed a guide. Had a lot of questions during the test that the test subject could not find the answers for.		

<table>
<tr><td><b>Test subject:</b> 4</td><td><b>Age:</b> 24</td><td><b>Smartphone Experience:</b> 2</td></tr>
<tr><td><b>Task 1</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> 1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks search. Is wondering if you are searching for all cemeteries. Types in name. Is looking for the button to continue. Finds it, and clicks the name. Understands what the shapes represent, and imagines that the colours are some kind of category.</td></tr>
<tr><td><b>Task 2</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> >1</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks search in exploration map. Tries to type in name, though it is difficult to spell. “Nice” that the app find the name even when you spell it wrong. Knows that it is Assistens Kirkegården because the test subject lives next to it, but cannot find a name saying so. Expect it to be written in the info box. Enters Wikipedia. Cannot find it written down anywhere.</td></tr>
<tr><td><b>Task 3</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> >1</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks home. Clicks find Cemeteries. Finds the cemetery and selects it. Clicks on a grave.</td></tr>
<tr><td><b>Task 4</b></td><td><b>Difficulty:</b> 3</td><td><b>Time:</b> 4 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks home. Clicks search cemetery. Zooms in. Finds and selects Assistens Kirkegården. Clicks the question mark to find out what the colour code for scientists is. Plans a route from the screen view. Does not expect the app to be able to do it. Tries afterwards. Clicks the tour planner icon in the help menu. Nothing happens. Cannot find the icon. Clicks home. Clicks the nearest cemetery. “I haven’t tried this option yet” Selects four graves and create a tour with no problems. “Why does the tour have two different colours?” “Maybe it has something to do with the category?”</td></tr>
<tr><td><b>Task 5</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> 2 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks tour planner. Wants to add H.C.Andersen to the existing tour. Finds the grave through the tour planner. Deletes the other graves in the tour planner. App says to at least select four. Is wondering what to do. Clicks H.C.Andersen’s grave. Finds the route button. Clicks it and solves the task. Did not have time to read how long it would take to travel the distance because the accidently clicked on another location.</td></tr>
<tr><td colspan="3"><b>General comments:</b>
Seems logical. Is not sure what the app can do when you start it. Would rather just use it as a map. It makes sense when you know about the different functions.</td></tr>
</table>

<table>
<tr><td><b>Test subject:</b> 5</td><td><b>Age:</b> 56</td><td><b>Smartphone Experience:</b> 3</td></tr>
<tr><td><b>Task 1</b></td><td><b>Difficulty:</b> 1</td><td><b>Time:</b> >1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Nice welcome screen. Clicks Search. Wonders how precisely you have to spell the name. Notices that the search is looking within all cemeteries. Thought that it would only concern Assistens Kirkegården, since that is where the test subject is located. Wonders why all the other names besides Ørsted appear. Most of them have nothing to do with what was written.</td></tr>
<tr><td><b>Task 2</b></td><td><b>Difficulty:</b> 2</td><td><b>Time:</b> 2 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks Search in exploration map. “How do you spell the name?”. “Why is Carl Nielsen the first hit?” Finds the grave. “Why is his grave red, when the one before was blue?” “What do the colours mean?” Goes to Wikipedia believing that it will tell which cemetery the grave is located at. Can see that it is Assistens Kirkegården, but cannot find it written anywhere.</td></tr>
<tr><td><b>Task 3</b></td><td><b>Difficulty:</b> 1</td><td><b>Time: ></b>1 minute</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks home. Select Cemetery. Finds and selects the cemetery. Clicks search, and then the small search button. A list appears.</td></tr>
<tr><td><b>Task 4</b></td><td><b>Difficulty:</b> 4</td><td><b>Time:</b> 4 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Home. Select cemetery. Clicks the question mark. Finds the description of the tour planner. Cannot see the button on the exploration map. Home. Nearest Cemetery. Sees the tour planner button has appeared. “That is strange” Clicks Tour Planner. Too many colours. Clicks a couple of names in the different tabs. Politician tab is not working. Figures out that you can scroll. Clicks back button instead of the tab once more to close it. The selected names disappear. Annoying. Would like a count of selected graves. Selects four graves. Is not sure how to finish. Tries to click the names on list button. Makes tour. Thinking circle is missing while making tour. Toast disappears too fast. Did not have time to read. Would be nice with names above markers when more graves are selected.</td></tr>
<tr><td><b>Task 5</b></td><td><b>Difficulty:</b> 5</td><td><b>Time:</b> 2 minutes</td></tr>
<tr><td colspan="3"><b>Comments:</b>
Clicks the tour planner. App says to select at least four. Test subject does not understand what to do when you only want to select one. Clicks home. Search. H.C. Andersen. Looks at the info box, but the route button is not available. Does not know what to do.</td></tr>
<tr><td colspan="3"><b>General comments:</b>
Seems clever. Believes that it can be put into use. Seemed intuitive with the exception of the last task.</td></tr>
</table>

Appendix K. Presentation of the Final App

Examples of each screen view of the app based on the wireframe.

Main Menu A.1 & Help Box C.4

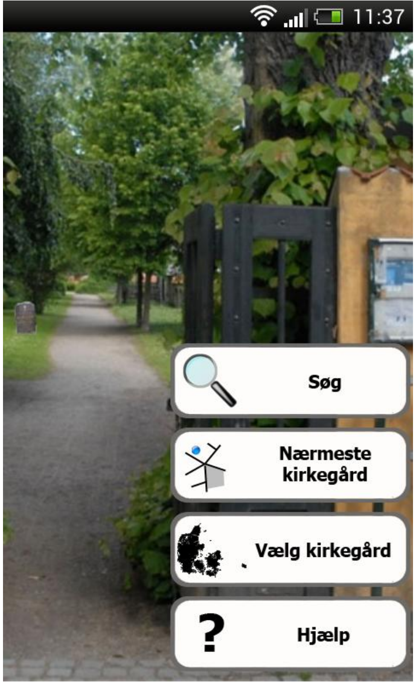
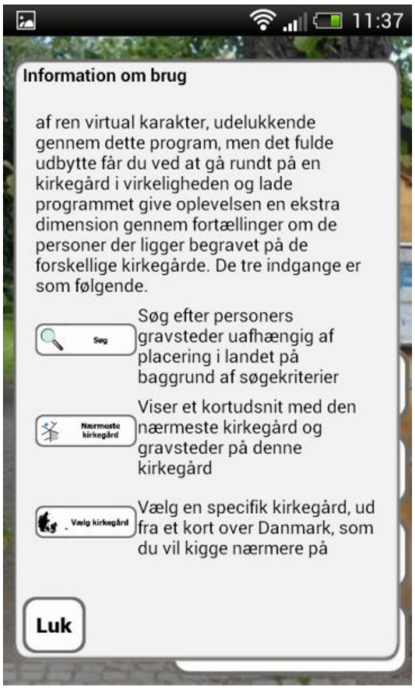


Figure 120



Information om brug

af ren virtual karakter, udelukkende gennem dette program, men det fulde udbytte får du ved at gå rundt på en kirkegård i virkeligheden og lade programmet give oplevelsen en ekstra dimension gennem fortællinger om de personer der ligger begravet på de forskellige kirkegårde. De tre indgange er som følgende.

Søg — Søg efter personers gravsteder uafhængig af placering i landet på baggrund af søgekriterier

Nærmeste kirkegård — Viser et kortudsnit med den nærmeste kirkegård og gravsteder på denne kirkegård

Vælg kirkegård — Vælg en specifik kirkegård, ud fra et kort over Danmark, som du vil kigge nærmere på

Luk

Figure 121

Exploration Map B.1

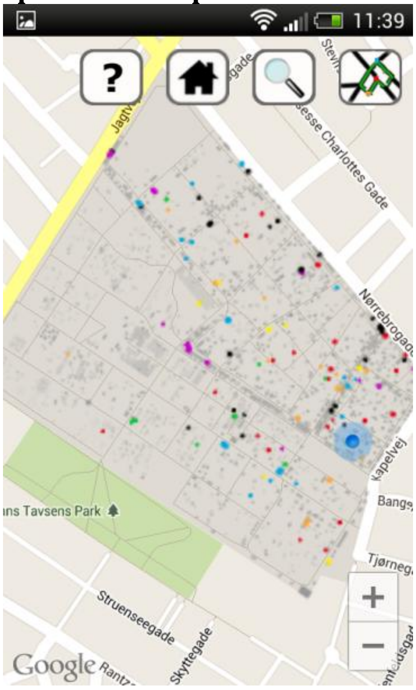


Figure 122

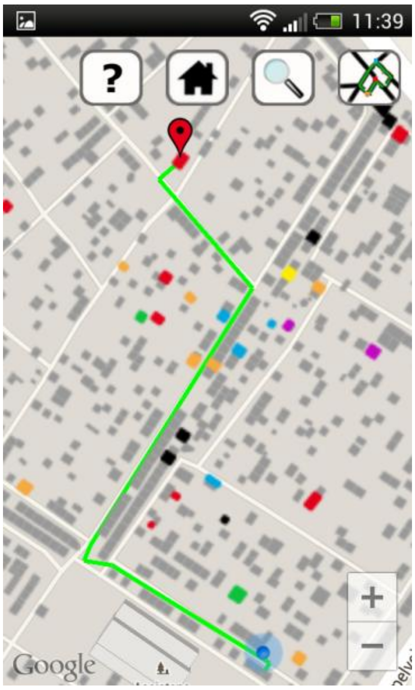


Figure 123

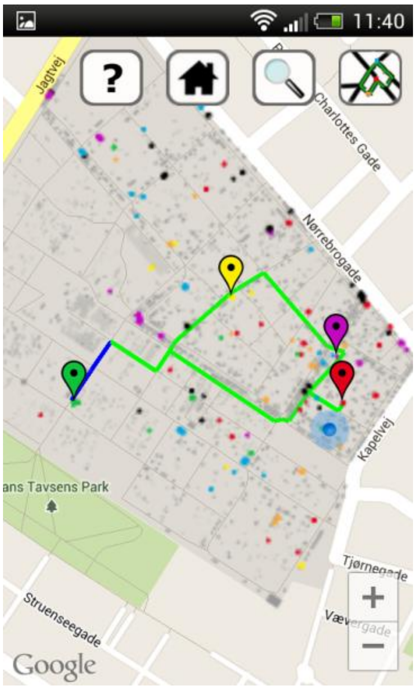
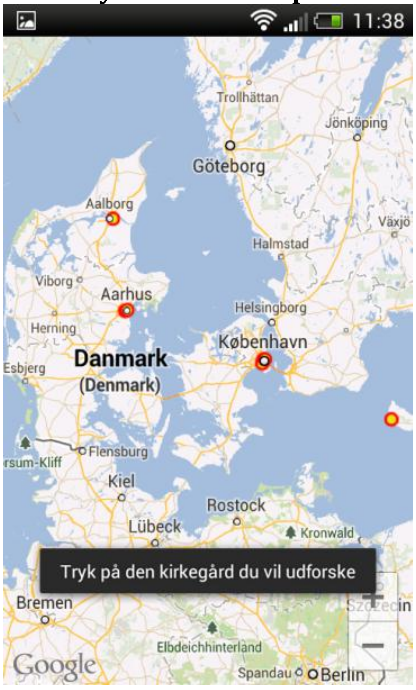


Figure 124

Cemetery Selection Map B.2



Tryk på den kirkegård du vil udforske

Figure 125

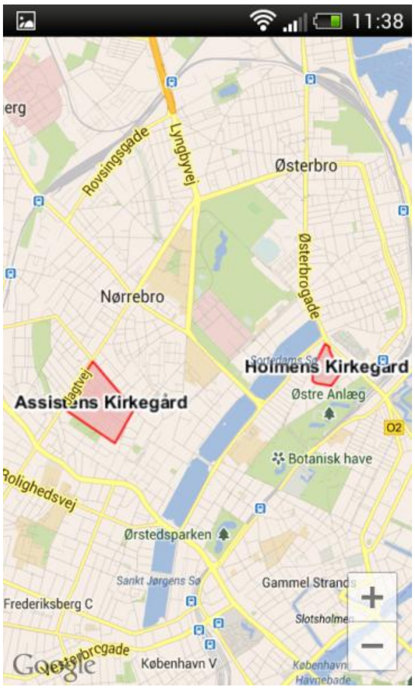
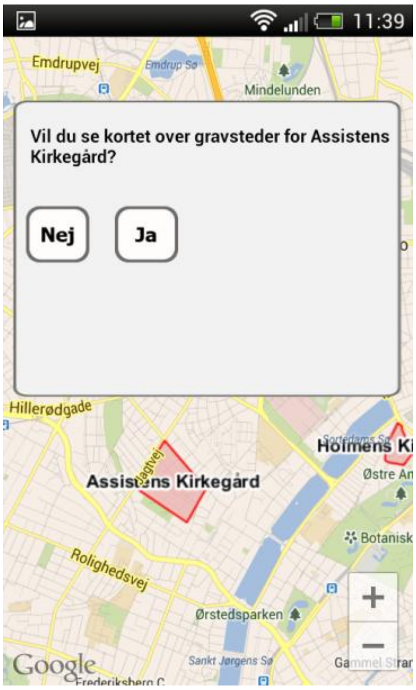


Figure 126

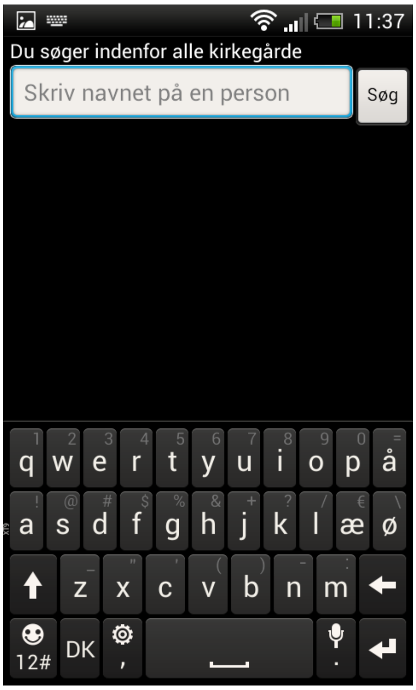


Vil du se kortet over gravsteder for Assistens Kirkegård?

Nej Ja

Figure 127

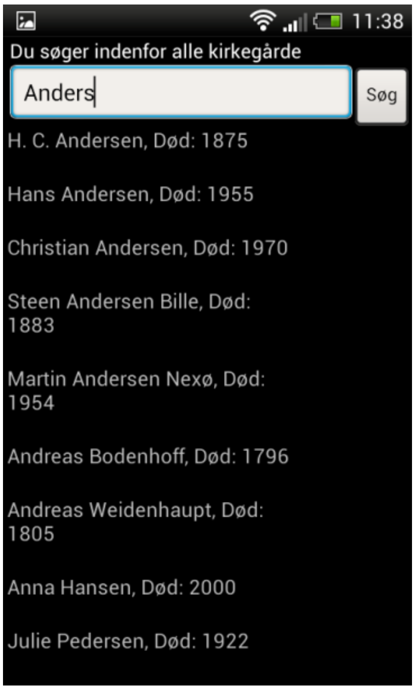
Grave Search Menu B.3 & C.3



Du søger indenfor alle kirkegårde

Skriv navnet på en person

Figure 128



Du søger indenfor alle kirkegårde

Anders

H. C. Andersen, Død: 1875

Hans Andersen, Død: 1955

Christian Andersen, Død: 1970

Steen Andersen Bille, Død: 1883

Martin Andersen Nexø, Død: 1954

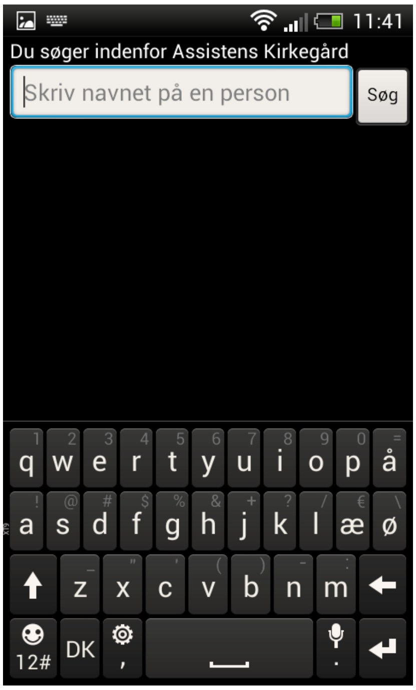
Andreas Bodenhoff, Død: 1796

Andreas Weidenhaupt, Død: 1805

Anna Hansen, Død: 2000

Julie Pedersen, Død: 1922

Figure 129

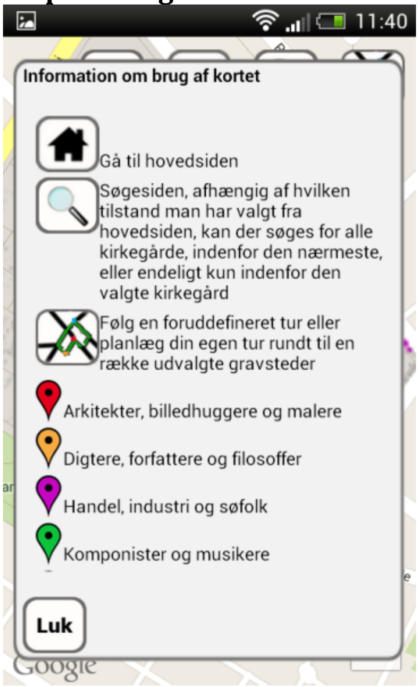


Du søger indenfor Assistens Kirkegård

Skriv navnet på en person

Figure 130

Help and Legend Box C.1



11:40

Information om brug af kortet

Gå til hovedsiden

Søgesiden, afhængig af hvilken tilstand man har valgt fra hovedsiden, kan der søges for alle kirkegårde, indenfor den nærmeste, eller endeligt kun indenfor den valgte kirkegård

Følg en foruddefineret tur eller planlæg din egen tur rundt til en række udvalgte gravsteder

Arkitekter, billedhuggere og malere

Digtere, forfattere og filosoffer

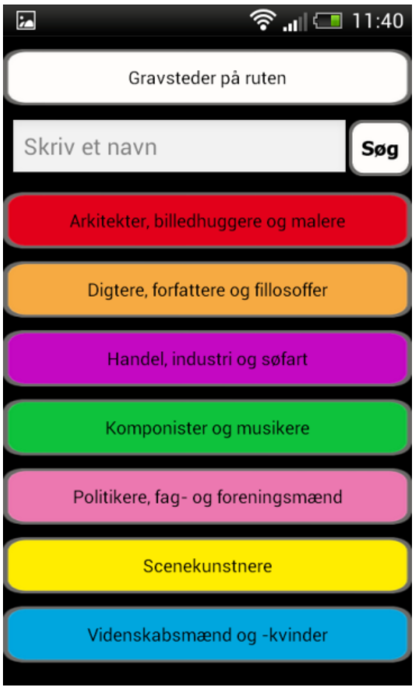
Handel, industri og søfolk

Komponister og musikere

Luk

Figure 131

Tour Planner C.2



11:40

Gravsteder på ruten

Skriv et navn | Søg

Arkitekter, billedhuggere og malere

Digtere, forfattere og fillosoffer

Handel, industri og søfart

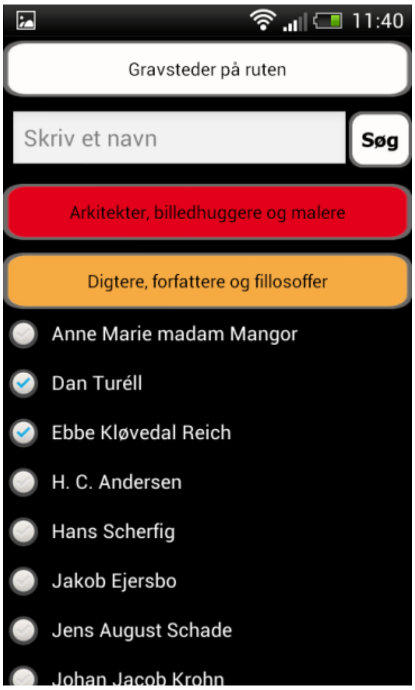
Komponister og musikere

Politikere, fag- og foreningsmænd

Scenekunstnere

Videnskabsmænd og -kvinder

Figure 132



11:40

Gravsteder på ruten

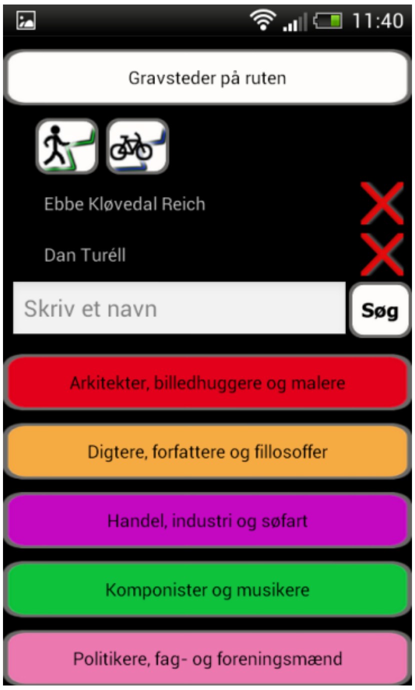
Skriv et navn | Søg

Arkitekter, billedhuggere og malere

Digtere, forfattere og fillosoffer

- ○ Anne Marie madam Mangor
- ☑ Dan Turéll
- ☑ Ebbe Kløvedal Reich
- ○ H. C. Andersen
- ○ Hans Scherfig
- ○ Jakob Ejersbo
- ○ Jens August Schade
- ○ Johan Jacob Krohn

Figure 133



11:40

Gravsteder på ruten

Ebbe Kløvedal Reich ✗

Dan Turéll ✗

Skriv et navn | Søg

Arkitekter, billedhuggere og malere

Digtere, forfattere og fillosoffer

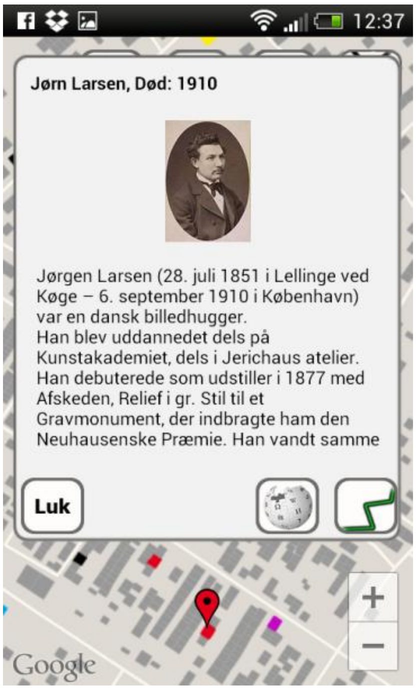
Handel, industri og søfart

Komponister og musikere

Politikere, fag- og foreningsmænd

Figure 134

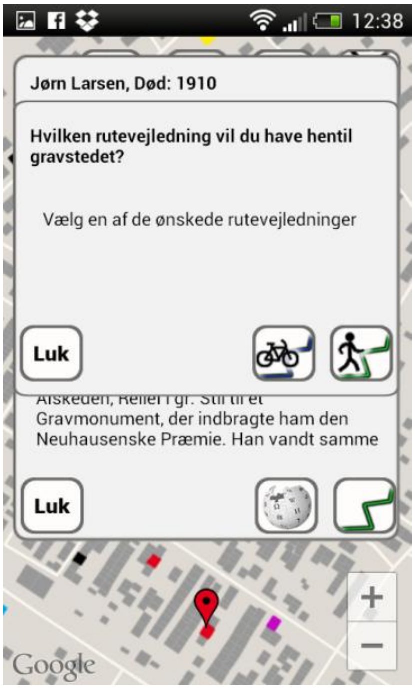
Info Box C.4 and Directions Box D.1



Jørn Larsen, Død: 1910

Jørgen Larsen (28. juli 1851 i Lellinge ved Køge – 6. september 1910 i København) var en dansk billedhugger.
Han blev uddannedet dels på Kunstakademiet, dels i Jerichaus atelier. Han debuterede som udstiller i 1877 med Afskeden, Relief i gr. Stil til et Gravmonument, der indbragte ham den Neuhausenske Præmie. Han vandt samme

Figure 135



Jørn Larsen, Død: 1910

Hvilken rutevejledning vil du have hentil gravstedet?

Vælg en af de ønskede rutevejledninger

Afskeden, Relief i gr. Stil til et Gravmonument, der indbragte ham den Neuhausenske Præmie. Han vandt samme

Figure 136