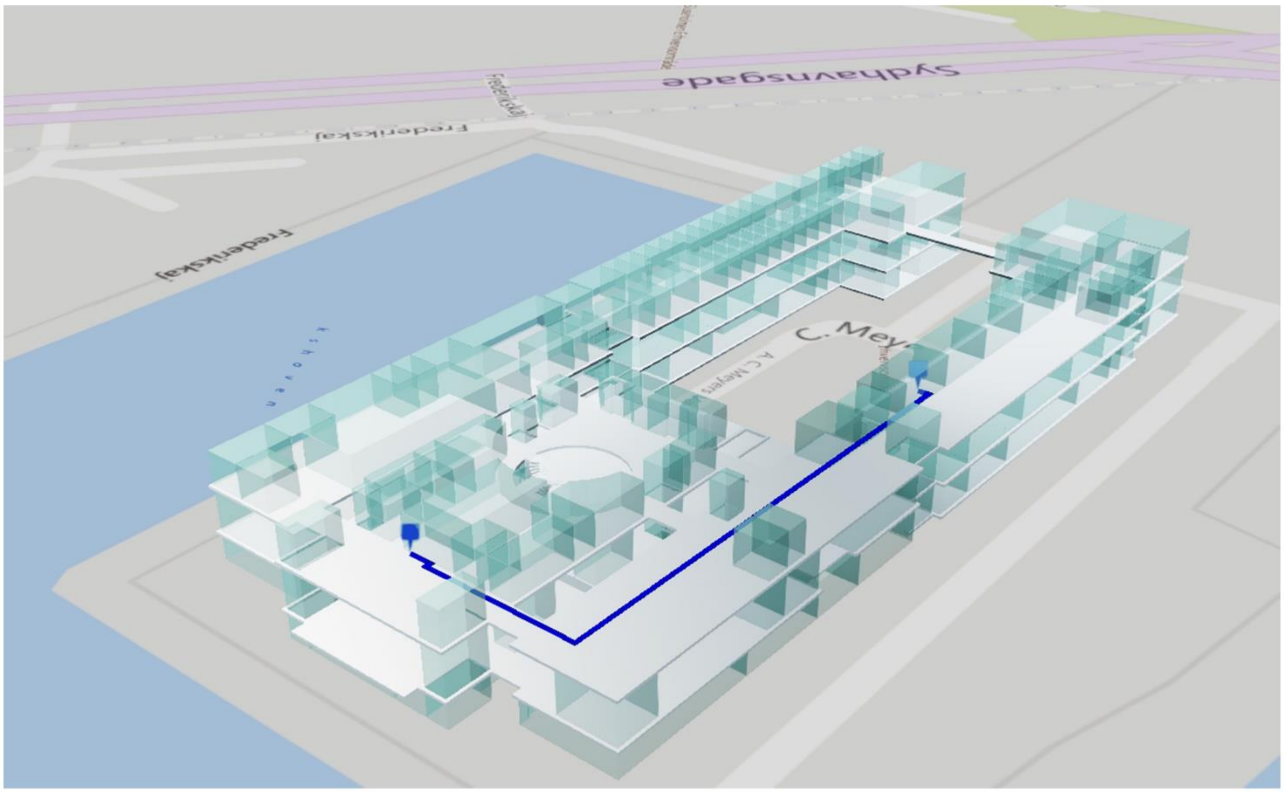
A 3D Routing Service for Indoor Environments

A service for the Aalborg University Campus
in Copenhagen



MSc in Geoinformatics

Master Thesis

Spring 2019

Marina Georgati

3D Photorealistic representation of the Malardalen University in Sweden by 3XN
source: https://www.designboom.com
Edit by the author

Acknowledgments

Undertaking this master thesis has been a truly life-changing experience for me and it would not have been possible to achieve it without the support and guidance from many respected persons, who deserve my deepest gratitude.

Initially, I would like to express my sincere gratitude to my thesis supervisor Carsten Kessler at the Aalborg University, Copenhagen, for his continuous support and patience and for providing me motivation and guidance when it was mostly necessary. His guidance helped me throughout the research, the implementation stage and the writing of this thesis.

Besides my advisor, I gratefully acknowledge the scholarship received towards my second year of studies from the Onassis Foundation. I would like to thank the Foundation not only for its financial support but also for its motive encouragement and for trusting my choices.

My thanks also to Branko Kuridza for his programming skills and his assistance during the development phase of the routing service and to Giorgos Karamitilios for our long focus discussions and his advices.

Finally, I would like to express a heartfelt thank you to my parents for their unfailing support and continuous encouragement throughout my years of studies and through the process of researching, developing and writing this thesis.

Abstract

The present study is motivated by the widespread need for applicable indoor navigation services in large and complex buildings. Fast and effective navigation is mostly desired in this kind of buildings that concentrate great masses of people and can contribute vastly in saving money and time while people avoid stressful situations. Until now, in contrast to their outdoor counterparts, indoor navigation services are limited in use due to their high implementation cost and their conceptual complexity that derives from the lack of navigational organization, the necessity of vertical connectivity and the existence of light objects.

This study is focused on the deep understanding of how an indoor navigation service works, what its major components are and how the above-mentioned challenges can be addressed in a short timeframe. Furthermore, the research examines methods for developing more flexible and reliable routing services that would provide coherent description of the examined places and would release the users from the insecurity of navigating in complex and crowdy indoor spaces. The aim of the research is the development of an indoor routing service that visualizes both the interior space of a real building and its routing paths supported in a three-dimensional virtual environment. This service is based on open source tools and has no installation requirements.

The selected case study of this project is the Building A at the University of Aalborg in Copenhagen and aims on the development of a desktop- based WebGIS that allows the three-dimensional navigation around the examined building and the calculation of the most suitable route between two points. Specifically, the service offers the possibility to the user to choose two points between which he/ she would like to move to and from, along with his/ her 3D virtual navigation around the building. The result of the routing calculation is the recommended path displayed in 3D mode among the interior places of the building, while the user gets the chance to watch a human model moving along the path. Moreover, the service provides written instructions on the side of the map highlighting the most significant places that the user has to pass by in order to reach his/ her destination.

The research concludes that 3D digital navigation offers great advantages on the deeper understanding of the explored environment, while it helps people improve their cognitive spatial maps when combined with the promotion of animated stimuli and landmarks.

Table of Contents

Table of Figures

Table of Codes

Table of Tables

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
API	Application Programming Interface
BIM	Building Information Modelling
BREP	Boundary Representation
CAD	Computer-aided Design
CityGML	City Geography Markup Language
CSS	Cascading Style Sheets
DIY	Do It Yourself
FME	Feature Manipulation Engine
FSS	Flexible Space Subdivision
GHz	Gigahertz
GIS	Geographic Information Systems
GML	Geography Markup Language
GNM	Geometric Network Model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HTML	Hyper Text Markup Language
ID	Identity document
IESM	Indoor Emergency Spatial Model
IFC	Industry Foundation Classes
IR	Infrared Radiation
JS	JavaScript
kb	kiloByte
LOD	Levels of detail
PHP	Hypertext Preprocessor
png	Portable Network Graphics
RFID	Radio-frequency identification
SQL	Structured Query Language
UWB	Ultra-wide band
WebGIS	Web Geographic Information System
XAMPP	Operating System Apache, MySQL, PHP, PERL
XML	eXtensible Markup Language

1. Introduction

Recently, the use of navigation systems has been rapidly growing with a great reduction on the wasted time and resources during the physical search of unfamiliar places. Position technologies and routing indicators not only facilitate every-day life, traveling and traffic flow but also improve emergency response and accelerate rescue operations providing advanced methods of decision making and control management. Considering the importance of the power that systems like that may have today, almost a decade ago Goodchild foresaw the possibility of monitoring positions, or '*knowing where everything is, at all times'*, through the use of satellite-based systems such as Global Positioning System and Galileo combined with ground-based systems such as Radio-frequency identification and mobile phone triangulation (Goodchild 2010).

Nowadays, outdoor navigation systems have reached this point that Goodchild predicted in 2010, gaining great popularity, covering a wide range of urban, suburban and rural areas, and presenting results of absolute accuracy. On the other hand, navigation in indoor environments, where GPS is inefficient, is still limited, with high cost of implementation and difficulties to be succeeded due to three basic reasons. Firstly, indoor environments lack of navigational organization. Pavements, street networks and bike networks that are the main elements of navigation organization outdoors cannot be found indoors, where free movement is allowed around obstacles without formal navigational classification. Additionally, vertical connectivity has to be also addressed along with horizontal mobility, contrary to outdoors navigation where it is not taken into account. The final reason why indoor navigation is considered hard to be achieved, is the often changes of light objects. These changes are either position changes, such as decoration modifications from time to time or simple relocations, such as these between a free and an occupied chair.

At the same time, regarding the necessity of developing a digital navigation system for indoor environments, it is well understood that people spend most of their daily life in indoor spaces, while it is proven that finding a way may be time consuming and requires a great amount of attention (Fallah et al. 2013). Based on Jamali et al. '*people spend almost 90% of their life in indoor building environment'* proving the essentiality of acceleration and facilitation that addressed navigation may offer (Jamali et al. 2017). The specific argument may be further explained considering the structural complexity and the big scale of most publicly accessible buildings, where advanced accessibility to all people and specially to disabled ones is of great significance.

Another factor that makes navigation systems necessary in indoor spaces is the quick and effective emergency response both for evacuation processes but also for reaching critical spots of the building or excluding hazardous areas such as chemical laboratories, high voltage panels and gas pipes. Availability of architectural and semantic information about the indoor structure of a building and its interaction with its surroundings may be critical in cases of indoor disasters providing first responders valuable knowledge to reach areas faster and with decreased routing uncertainty (Tashakkori, Rajabifard, and Kalantari 2015).

Therefore, despite the limitations in the implementation of such a system, this thesis notices the importance of an indoor navigation system and suggests its development in a three-dimensional virtual environment. Specifically, it focuses on the identification of the existing geospatial and technological infrastructure that is required for the proper function of an indoor navigation system, the research of the requirements of a case-based scenario for a desktop application that provides indoor routing instructions in a 3d virtual environment and the development of such an application covering the corresponding users' demands.

1.1.Motivation

The concept of this master thesis was developed out of the repetitive observation of the essentiality of an indoor navigation system in large and complex buildings, such as libraries, museums, malls or airports, where there is a great concentration of people and the majority of them wants to find its destination as fast and easily as possible. Apart from saving a lot of valuable time and money, an indoor navigation system may improve the routing experience of impaired people that gain the chance to explore new places without stress or asking for other people's help.

Three-dimensional navigation is the second major element of this thesis and is chosen firstly because of the strength of the 3D visualizations and then due to my former experience on 3D information models. During my internship at Atkins Denmark, I worked on the development of a service that provides the user the possibility to navigate around a 3D model with an evident geographic and elevational reference.

Most importantly though, three-dimensional navigation is selected in the specific case of routing as the most suitable form of representation offering considerable advantages compared to other means and improving orientation and guidance. In particular, Xu et al. indicate 3D models as more powerful than 2D maps in a complicate space because of the accuracy of their description (Xu et al. 2013). Along with the representation of the location of the objects, 3D models offer context information of these objects with adequacy and realism showing their color, shape and texture. Lastly, advanced possibilities to visualize vertical information (height of doors, irregular shapes) and mobility (stairs, elevators) are also available (Xu et al. 2013).

For the above-mentioned reasons, navigation in a 3D virtual environment is considered of great significance for this thesis and meets the demands of the contemporary digital society of high speeds. However, in order to apply an efficient routing system, various challenges have to be overcome, such as the suitability or availability of the 3D building model, the indoor navigation network, the vertical and horizontal connectivity, the data availability, the existing localization system and the approaches of path planning. These challenges are further explored in the following chapters.

1.2. Problem Statement

Navigation is one of the main activities of people that is being developed throughout life; from the first days of a child's life, the recognition of surrounding places progresses along with the rest of human senses. People walk, run, bike or drive daily finding their way from one place to another. It is a natural skill that is being learnt and developed by growing up and by using '*commonsense knowledge of geographic space'* (Raubal and Egenhofer 1998). However, spatial knowledge or spatial behavior relates to unique personal experiences and perception of space and differs among individuals. In other words, some people have great sense of orientation, while others face difficulties to find the correct way to their destinations. These difficulties may get even worse when navigation takes place in unfamiliar or crowdy places or when people are in hurry.

Digital navigation systems are available today to cover the need for easy and quick wayfinding, while particularly car navigation systems are now used in a worldwide level by the majority of the population. In spite of the great success and spread of the vehicle navigation systems, pedestrian ones in indoor spaces have not yet met the corresponding widespread application. This thesis recognizes the lack of applied instances of indoor navigation systems and the inefficiency of Global Positioning System to locate objects or spaces in a building and attempts to address this problem by exploring the existing technologies and the available methods for developing a routing service in indoor spaces supported in a three-dimensional virtual environment.

More particularly, the present research selects the case study of a building of the Aalborg university campus in Copenhagen and examines the possibilities and the requirements of an indoor navigation service displayed in a virtual 3D environment. Then, it proposes the development of such a service and proceeds to the evaluation of its functionalities by considering its advantages and disadvantages, the user's demands and the included elements.

1.3. Research Questions

The main research questions that can be addressed in the present study may include:

1. What are the fundamental requirements for developing a comprehensive routing service with 3D visualizations?
2. Which existing technologies are applied in indoor 3D navigation services? What is their range of implementation, the requirements and the components of a successful navigation service?
3. Which factors influence the technical performance of an indoor 3D navigation service?
4. How can a 3D indoor navigation service be developed using existing free and open source tools?

The research is focused on the theoretical background around indoor positioning and navigational systems and the technologies that are available for developing an indoor navigation service. The specific routing service is based on the 3D navigation of the user around the explored case study and the visualization of the routing path in a three-dimensional manner as well.

1.4.Methods

Similarly to any research, the present study's purpose is to discover answers to questions throughout the application of scientific procedures (Kothari 2004). Even though the purposes and the motivations of most researchers differ, research methods may usually fall into broad groupings. This section describes the research methodological techniques followed in this thesis aiming on answering the above-mentioned research questions.

The selected methods that are applied in the present thesis are: 1. the literature review, 2. the requirements analysis and 3. the prototype design. To begin with the literature review, that is one of the most popular research methods, a comprehensive study and interpretation of literature is conducted in chapter 2 Background for the survey and evaluation of the available technologies that are already applied in indoor 3D navigation services and their capabilities.

Furthermore, the review of a wide range of literature attempts to describe the basic components of indoor navigation services and the issues regarding their three- dimensional representations, while the factors that influence their technical performance are also explored. Moreover, existing tools that allow the development of 3D indoor navigation services are investigated.

In addition to the conducted literature review for the development of comprehensive routing services with 3D visualizations, a requirements analysis is also presented in section 4.3 Requirements analysis. A requirements analysis is a process for the definition of the expectations of the users of an application ("Requirements Analysis - Requirements Analysis Process,Techniques" n.d.) along with the challenges that are to be addressed in order to meet them. The specific section highlights the major requirements that building models should cover in order to accommodate comprehensive route planning.

After having examined the available technologies for the development of indoor navigation services and the building modelling requirements, the study proceeds to the development of a prototype service. A prototype is a model or a simulation that tests the feasibility of certain technical aspects of a service and usually remains incomplete, to be modified and supplemented by a later one (Carr and Verner 1997). The prototype design of the specific service is presented in chapter 5 Front End, where the implementation phase and the results of the final product are described.

2. Background

The following chapter examines various aspects around the development of a 3D routing service based on the conducted literature review. Initially, the components that are required for the proper function of a navigation system are explored, followed by a brief research on indoor space modelling and the prerequisites of 3D model development. The third section mentions instances of related work focusing on the frameworks of model development, application methods and tools, while the last section presents a wide range of technological tools that are available for the calculation of routing paths.

2. 1. Elements of indoor navigation systems

Navigation services consist of three major elements: the localization, the path determination and the guidance along the path (Becker, Nagel, and Kolbe 2009), which also define the form of its representation. Fallah et al. determine one additional issue to define the guidance along the path; the interaction with the user, which may be visual, audio or haptic depending on the audience that is meant to serve (Fallah et al. 2013). This section describes the three basic components of functional navigation services as they are defined throughout literature under the scope of exploring the available options and their evolution throughout time.

2.1.1. Localization

The determination of the user' s position and/or orientation is one the most important issues that define a navigation service. Since satellite based navigation systems such as GPS do not allow position sensing in indoor environments, technologies like WLAN, radio frequency identification (RFID), infrared and ultra-wide-band (UWB) provide sensing inside buildings, with differentiations on the application range of usage (Rüppel, Stübbe, and Zwinger 2010). Based on Fallah at al. there are four different groups of techniques for localization: dead reckoning, direct sensing, triangulation and pattern recognition, that are further explored in this section.

Dead reckoning, which estimates the position of the user based on an already known position is considered totally obsolete due to its inaccuracy and necessity of being combined with additional techniques. For this reason, it is not further examined in this section.

Direct sensing methods require the installation and use of identifiers or tags in the examined spaces and work either by storing the location information and the information on user's environment in the tag or by retrieving this information from a database using the tag's unique identifier. There are five technologies that can be used for the tags: the radio-frequency identifier description (RFID), the infrared (IR) transmitters, the ultrasonic identification, the Bluetooth beacons, the Ultra-wide band (UWB) and the barcodes (Fallah et al. 2013).

Initially, RFID is a form of wireless communication that uses the electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum and is able to uniquely identify an object. A RFID system includes three components: a scanning antenna, a transceiver and a transponder, the RFID tag (Rouse 2017). The RFID tags may be active or passive depending on their source of power, while a RFID system may have both active and passive tags. Active RFID tags store up to 128 kb, have longer read range, higher cost and require more frequent maintenance compared to passive RFID tags that can store only up to 128 bytes (Fallah et al. 2013).The range of a RFID system varies from a few centimeters to 10 meters based on its type (low frequency, high frequency, ultra-high frequency or microwave RFID) (Rouse 2017).

In the case of Infrared Radiation localization, IR transmitters are set in known positions and each one of them broadcasts unique ID to the user's IR receiver through infrared radiation. However due to the fact that IR may be affected by natural and artificial light, have high cost (Fallah et al. 2013), short-range and narrow- transmission- angle (H. Liu et al. 2007), it is not considered as a proper solution for an indoor navigation system.

Ultrasound identification emitters broadcast ultrasound waves with a short wavelength and are set in the infrastructure. The user has to carry receivers on both of his/her shoulders providing data about the orientation of the user as well. The drawback of the Ultrasound identification system is the necessary line of sight between the receiver and the emitters, while it may be also affected by the walls reducing its accuracy level (Fallah et al. 2013).

Regarding Bluetooth beacons, which might have been the most famous wireless technology until recently, uses 2.4GHz band as WIFI, with main disadvantage its measurement rate and its short range. The device might delay up to 10 seconds forcing the user to walk slower, while it is recommended to install a beacon in each room. Similar to RFID tags, a power source is required for Bluetooth beacons, but it is highly power efficient (Kárník and Streit 2016).

Likewise, UWB requires the installation of additional infrastructure in the area that needs to cover. UWB sends ultrasonic pulses and seems to transcend conventional RFID because of its shorter transmission over multiple bands of frequencies at the same time. UWB tags are less power consuming compared to the RFID tags and operate across a broad area of the radio spectrum. Furthermore, it is not affected by objects improving the performance and the speed of the system (H. Liu et al. 2007).

The last technology grouped in the direct sensing methods is the barcode. While walking around, the user can scan the barcode and locate himself/herself based on its unique ID. Right after that, the system may return information about the surrounding space and define the new routing path. Although the installation and maintenance is cheap and easy, the user has to look for the barcodes along his/her path, making the navigation slow or impossible for visually impaired people (Fallah et al. 2013). Passive RFID tags tend to replace the traditional barcode technology (H. Liu et al. 2007).

While considering the next localization technique, its seems that the three of the above-mentioned technologies (the RFID, the IR and the ultrasonic) can also be used in the triangulation method, but in this case the user' s position is determined by at least three points instead of one, which derives from the corresponding tags in the examined spaces (Fallah et al. 2013). Triangulation '*uses the geometric properties triangles to estimate the target's location'* and has two derivations: lateration and angulation (H. Liu et al. 2007). The difference between these two derivations that determine the user's position is that the former uses the distance between the user and three known points, while the latter uses the angular measurements from three points to the user (Zheng and Ni 2006). GPS is the most well-known technique that uses the triangulation method based on satellites' signals to calculate the user's location, but as it has been already mentioned it is not available for indoor environments.

However, cell-tower positioning and wireless local area networks (WLANs) are alternatives with applications in various places inside a building. Cell-tower positioning systems receive signals from cell towers from the surrounding area, while wireless local area networks (WLANs) require the installation of wireless base stations for transmitting signals and triangulate them. The accuracy of a typical WLAN positioning system is around 3-30 m. and the update rate in the range is just a few seconds (H. Liu et al. 2007).

The last method shown here regarding the determination of the indoor position of the user is the pattern recognition-based localization. This case '*uses data from one or more sensors carried or worn by the user and compare these perceived data with the set of prior collected raw sensor data that has been coupled with an environment map*' (Fallah et al. 2013). Pattern recognition-based localization method applies two major types of sensing techniques: the computer vision-based localization techniques and the fingerprint localization techniques.

In the former case of the computer vision-based localization techniques, the user has to carry a camera that records the user' s navigation and matches the images with the ones stored in a database. This way determines both the location and the orientation of the user but requires significant storage capacity and computing power. The latter techniques '*compare the unique signal data from one or more external sources sensed at a particular location with a map of prerecorded data*' (Fallah et al. 2013) with two stages of operation. The first one consists of the creation of the fingerprint database with sensors collecting data about the place and assigning them location values, while the second stage calculates the position of the user according to the stored values. The fingerprint localization techniques present low accuracy and many errors making it unreliable (Kárník and Streit 2016).

In a nutshell, this section gathers some of the most popular techniques and technologies for the detection of the user's position in the interior environment of a building, concluding that the triangulation method by using WLAN technologies may have sufficient and satisfactory results regarding both the range of application and the accuracy level.

2.1.2. Path planning

Path planning is the second significant element of an indoor navigation service that plays important role in the overall performance of the service. The path should cover the user's demands regarding the selection of the desired route and offer various options depending on the user's needs. Shortest path or shortest travel time may be suitable for the majority of the population most of the times, but there are also scenarios that should be evenly taken into account, such as wheelchair accessibility and existing landmarks that improve navigation.

This section presents two of the most famous algorithms in the computer sciences for path finding: Dijkstra' s and A*. Edsger Dijkstra developed the algorithm that was named after him in 1959, which '*works by storing the cost of the shortest path found so far between the start vertex, s, and each target vertex, t*' (Smith, Goodchild, and Longley 2018). The concept lies in two components. Firstly, each node stores its previous one and along with the total distance. Then, as far as there are nodes that are not yet included in the list of visited nodes, the path continues to the one with the smallest total distance *t*' (Smith, Goodchild, and Longley 2018).

A* is a more generic and improved version of the Dijkstra' s algorithm developed by Peter Hart, Nils Nilsson and Bertram Raphael in 1968 that uses heuristics, excluding the paths that do not lead to the destination (Reddy 2013). The selection of the next node to visit in A* includes the calculation of the Euclidean distance of each vertex from the target and the addition of the already recorded distance to it via the network to this vertex. A* algorithm usually results in fewer visited nodes increasing the speed of the results and improves the performance of the system (Smith, Goodchild, and Longley 2018).

Taking a further look in the literature regarding path finding in indoor spaces, another interesting algorithm is developed by Liu and Zlatanova with the notion of accessibility to be the main consideration, instead of distance. The door-to-door approach is '*an algorithm, which is applied to 2D floor plan of buildings with complex indoor structure. The algorithm consists of two-level routing: one is to get coarse route between rooms, and the other one is applied to single rooms to acquire the detailed route*' (L. Liu and Zlatanova 2011). The idea of this approach is better illustrated in Figure 1 in comparison to a traditional Dual Graph. The dashed line represents the door-to-door approach, while the solid line is the Dual Graph.

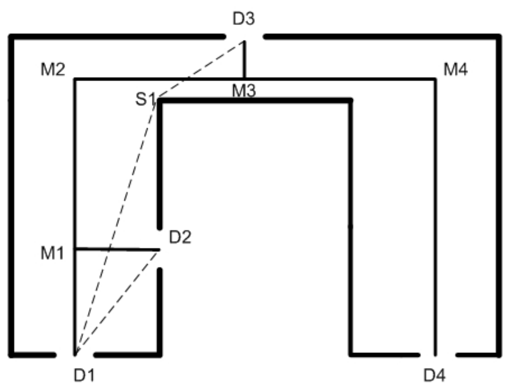


Figure 1 Comparison of DG and door-to-door route (L. Liu and Zlatanova 2011)

Even though the door-to-door approach is very interesting for the path planning in indoor environments, Dijkstra's algorithm is considered more suitable for the examined case due to its simplicity and universal recognition.

2.1.3. Representation

Except for the required processes for the detection of the user' s location and the proper calculation of the routing path, a successful navigation service should also be able to represent the produced information in a way understandable by the majority of the addressed audience. Many techniques of describing a route have been applied throughout the years varying based on the suitability of the target users and offering different kind of benefits. Text or written instructions, speech, 2D maps, route sketches, graphs and photographs are some of the most popular ways to represent a navigation path and are extensively put into practice, while their combinations enhance the perceptional ability of the user driving to a more efficient 'reading' of the recommended moves.

However, when considering navigation in the interior of buildings, floor maps combined with graphs are so far the only solution. For instance, most shopping malls with daily high congregation of visitors install tables displaying their floor maps at their entrance halls to improve the navigational experience of the people and guide them to their destination. These tables or digital screens although do not include any actual navigational information except for the plans of the building but provide the visitors the possibility to notice their destination's location and 'calculate' their own desired path. A method like that is widely spread providing information about the functionality of the rooms and the relationships among them. It is easily generated as it depends upon the buildings' blueprints only, but the provided amount of information is limited.

On the other hand, three dimensional models are regarded to be more appropriate for the representation of the interior of large, complex buildings due to the following reasons. Initially, they present a greater amount of detail and useful information that can be proven beneficial for indoor navigation in places that attract massively crowds without any language restrictions (Fallah et al. 2013). Furthermore, the simultaneous visualization of multiple floors is feasible by using a 3D model, while switching among floor plans may cause confusion and lack of orientation. Moreover, better understanding of an unfamiliar space through the 3D model leads to faster and more comfortable navigation saving time and stress and providing advanced navigation experience even to people with special needs (handicapped). Nonetheless, the higher cost for the creation of 3D models and the increased requirements for storage and processing time are drawbacks that should also be taken into consideration.

Because of the above mentioned reasons that prove the beneficial role of 3D visualizations in routing, this thesis focuses the attention on the development of a routing service that represents both the interior spaces of the building and the navigation path in a three-dimensional virtual space that enables digital navigation in 3D mode.

2. 2. Indoor space modeling

As mentioned in the previous section, the representation of the routing path in its background is of vital importance in a navigation service setting the building model and the subdivision of its indoor spaces in a significant place. As this thesis focuses on the three-dimensional visualization rather than any other mode of representation, the following section describes methods, standards and prerequisites for the generation of indoor 3D building models that host the navigation path.

3D spatial models are divided in three categories: the Boundary Representation (BREP) models, the 3D '*dump*' solid models and the 3D parametric solid models (Worboys and Walton 2008). The Boundary Representation models, which have topological and geometric parts, use limits for the representation of shapes making a solid from the collection of connected surface elements. Faces, edges and vertices are the main topological items of the BREP models, while CAD softwares use them extensively with Boolean operations and extrusions. On the contrary, the 3D '*dump*' solid models (cubes, spheres) use basic three-dimensional geometric forms as solid volumes that can be edited but the corresponding softwares have motion limitations. Finally, the 3D parametric solid models are created by adjustable features. Building Information Modelling (BIM) includes this type of models that expresses geometry, spatial relationships, geographic information and context properties of the building components. (Worboys and Walton 2008)

Apart from the necessity of selecting a drawing technique, another issue that plays significant role in the development of 3D spatial models for indoor environments is the classification and the hierarchical order of the spaces along with the desired level-of-detail (LOD). These issues arise due to the inherent complexity of the interior of most buildings that require a navigation service and is left on the developer's sensitivity to define the most appropriate subdivision and structural order dependent on the purposes and the unique characteristics of the examined building. UML diagrams are usually used for the presentation of the recommended classification method and the description of the relations among the building's components. The examples that are explored in the next section attempt to propose a suitable solution regarding these issues.

Moving another step further, the development of an accurate and broadly understandable model of complex indoor spaces requires a few standards and the cooperation of two fields; the Computer-aided design (CAD) or more specifically the Building Information Modelling (BIM) and the Geographic Information Systems (GIS). These two fields present the Industry Foundation Classes (IFC) and the City Geography Markup Language (CityGML) standards respectively that offer geographic, semantic and topological description of the built environment and the relationships among its components (Diakité and Zlatanova 2018a).

According to their definitions BIM is '*an intelligent 3D model-based process that gives architecture, engineering, and construction (AEC) professionals the insight and tools to more efficiently plan, design, construct, and manage buildings and infrastructure*' ("What Is BIM | Building Information Modeling | Autodesk" n.d.), while IFC is generally known as a data model developed by the buildingSMART to facilitate interoperability in the building industry ("Solibri | About BIM and IFC" n.d.). It is '*a data representation standard and file format used to define architectural and construction-related CAD graphic data as 3D real-world objects*' and uses a plain text file ("Solibri | About BIM and IFC" n.d.).

On the other hand CityGML is '*an open data model and XML-based format for the storage and exchange of virtual 3D city models*' ("CityGML | OGC" n.d.). The development of CityGML is based on the need of a universal definition of the basic entities, attributes and relations of a 3D city model ("CityGML | OGC" n.d.). Specifically, the building model is one of the most thematic concepts of CityGML and allows the representation of the building in 5 levels of detail, LOD0 to LOD4 (Gröger et al. 2012). Each level of detail includes different components of the building, while an object may be represented simultaneously in different LODs by providing distinct geometries for the corresponding LODs. Figure 2 shows the differences in the structural entities of the building among the various LODs.

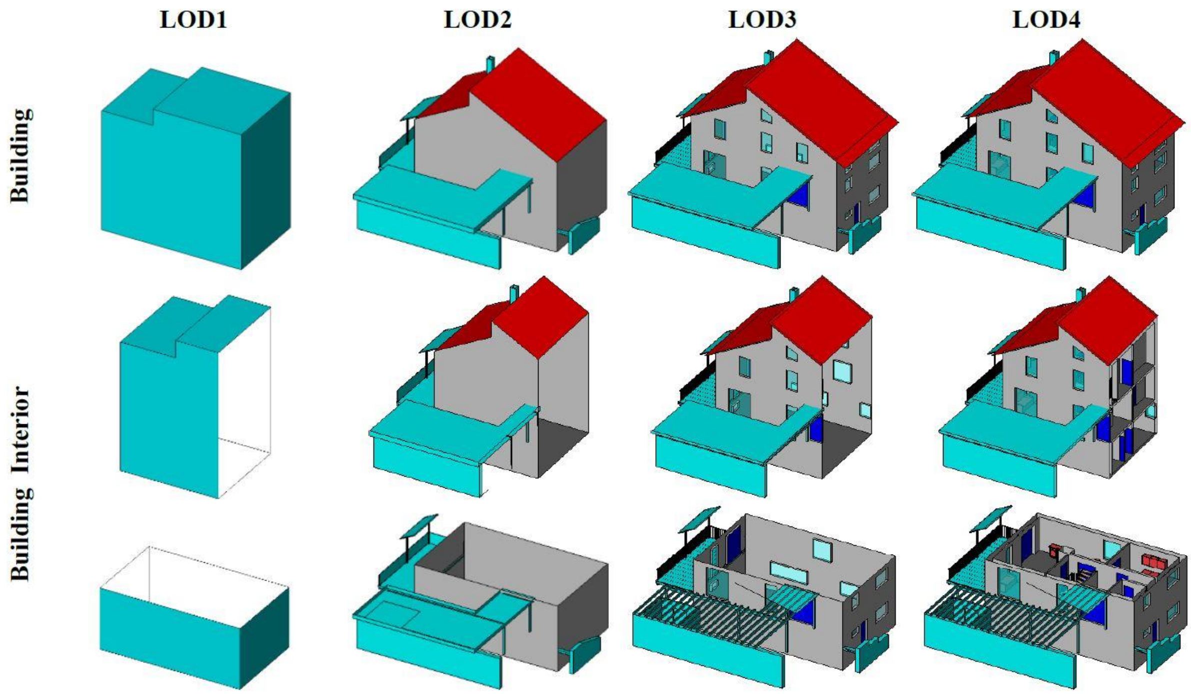


Figure 2 Building model in LOD1 – LOD4 (Gröger et al. 2012)

Both standards have the potential to support building smart indoor navigation tools and the topographic space information required for indoor navigation but both of them present drawbacks in various processing levels. More particularly, according to Brown et al. indoor obstacles or non-navigable spaces cannot be represented in CityGML, while specific building parts cannot include semantic information or be decomposed to sub-parts. Moreover, CityGML does not provide semantic information for indoor spaces and the connection among them. Similarly, IFC lacks semantic information for connected indoor spaces and does not support dynamic obstacles. In contrast to CityGML, IFC holds hundreds of classes but only a few of them are relevant for indoor navigation topographic space (Brown et al. 2013).

Even though BIM is a valuable tool containing geometric and semantic representation of the buildings elements, it is not used in the implementation phase of this project due to the simplicity of the model of the case study and time limitations. Another reason why BIM is not considered absolutely necessary for the creation of the building model in the present study is that CityGML and 3D Tiles that are used for loading the model on the website do not provide the level to detail that BIM can offer. Simple CAD processes are executed to generate geometrically the 3D model and during its conversion to CityGML, semantic information is added.

2. 3. Related work

During the last few years, a lot of ideas have been published promoting indoor navigation systems in 3D spaces with differentiations on their three basic elements and variations on the hierarchical order of their models. This section presents some of the most interesting examples of these routing systems found in the recent literature focusing on the aspects regarding model development, application methods and tools.

The most up-to-date approach comes from Diakité and Zlatanova and introduces the Flexible Space Subdivision (FSS). This framework takes into account the dynamic changes occurring in the scenes and allows the identification of free and non-free indoor spaces through the objects and the activities that take place in these spaces. In other words, the movement of the objects and the relationships among each other along with the assignment of roles in the navigation process provide valuable insight in the functional side of the resource, its availability or accessibility and results in Navigable or nonNavigable cell spaces. IndoorGML is the standard being used in this case based on a cellular model and CellSpace that defines the Navigable and NonNavigable cells (Figure 3) (Diakité and Zlatanova 2018a).

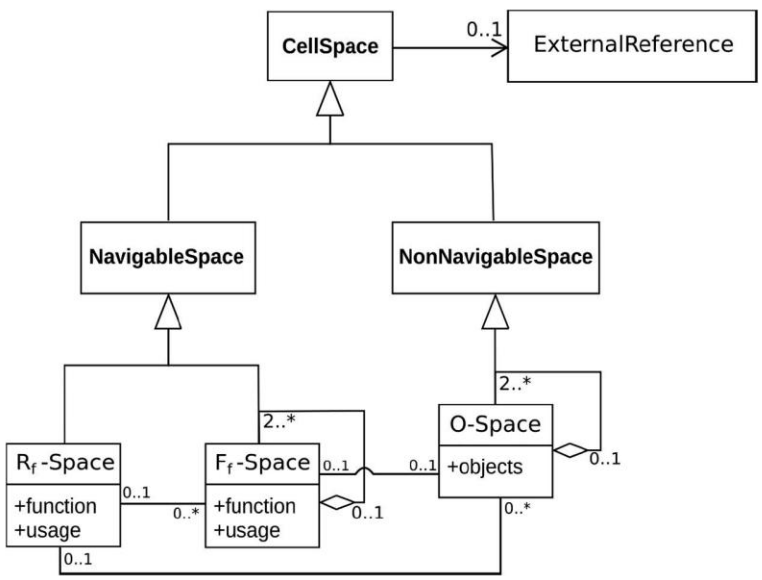


Figure 3 UML diagram of the FSS integrated to IndoorGML

Figure 4 illustrates in a more detailed way the spatial partitioning of the indoor environment assigning O-space to NonNavigable, F-space to Navigable under conditions and R-space to freely navigable. Additional notions that are being introduced under this framework characterizing the objects are the static (S-object) for objects that remained fixed, semi-mobile (SM-objects) that cannot move by themselves but can be moved and mobile (M-objects) that can move themselves.

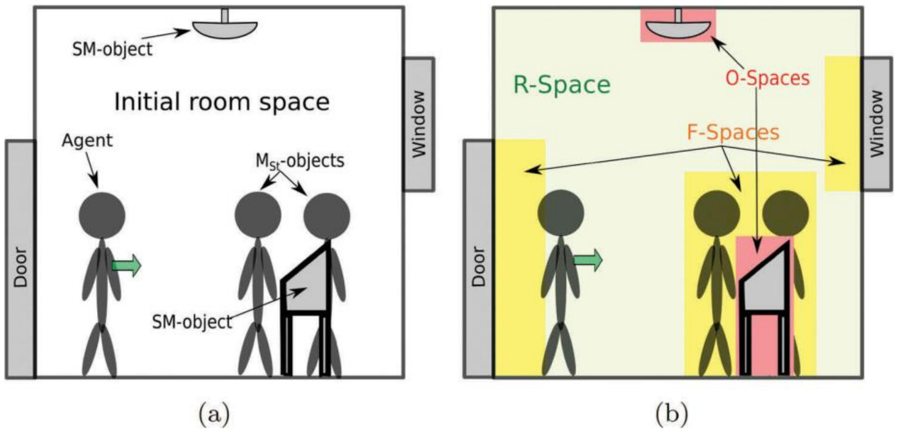


Figure 4 Different subspaces created during the FSS (Diakité and Zlatanova 2018a)

The main goal of the FSS framework is the creation of a realistic, non-abstractive 3D indoor space that reveals its complexity and '*suitable navigational areas can be automatically identified for a given user*' (Diakité and Zlatanova 2018a). The specific approach is applied to a BIM model where the interaction between the user and the system is of great weight. Even though the research does not examine any aspects of the localization methods or the path planning techniques, the promoted approach is innovative and highly elaborate that can lead the way to a new era in the booming industry of indoor navigation services.

The approach introduced by Jamali et al. shows an automated method for 3D modeling of indoor navigation. The 3D model is structured in cells as in the previous procedure, that leave no gaps among them, but this model is more abstract and relies only on geometrical and topological relationships, excluding the semantic aspect of the objects and the spaces. Particularly, the 3D model is considered as primal graph, while indoor navigation network is generated by connecting surveyed benchmarks- dual nodes through the Delaunay triangulation. Both benchmarks and the rooms of the 3D model have a specific ID as primary key and are connected in accordance with it. The routing path is calculated as the shortest path among the points using the Dijkstra algorithm, but there are no suggestions on a preferred localization system or representation method (Jamali et al. 2017).

Purpose- oriented, the Indoor Emergency Spatial Model (IESM) '*integrates 3D indoor architectural and semantic information required by first responders during indoor disasters with outdoor geographical information to improve situational awareness about both interiors of buildings as well as their interactions with outdoor components*' (Tashakkori, Rajabifard, and Kalantari 2015). The model is based on IFC and tested through the ESRI GIS platform, while the Dijkstra algorithm is used for the calculation of the quickest path. This approach elaborates on gathering only the required information for an emergency situation excluding any redundant information that might be included in other cases for visualization purposes. In this case apart from the indoor model and the route visualization, the integration with outdoor environment and the road network is also considered of great importance.

Figure 5 shows the framework for the development of the IESM. Firstly, Revit is used for the creation of the 3D building models from the original 2D indoor CAD files at the same time with the 3D indoor geometric network model (GNM). These two elements are then exported to the ESRI geodatabase format in ArcGIS proving a 3D indoor navigable model. The indoor GNM is getting connected with the road network enabling the use of the Network Analyst making the system work properly. Additional features such as the exclusion of restricted areas in case of emergency can also be activated.

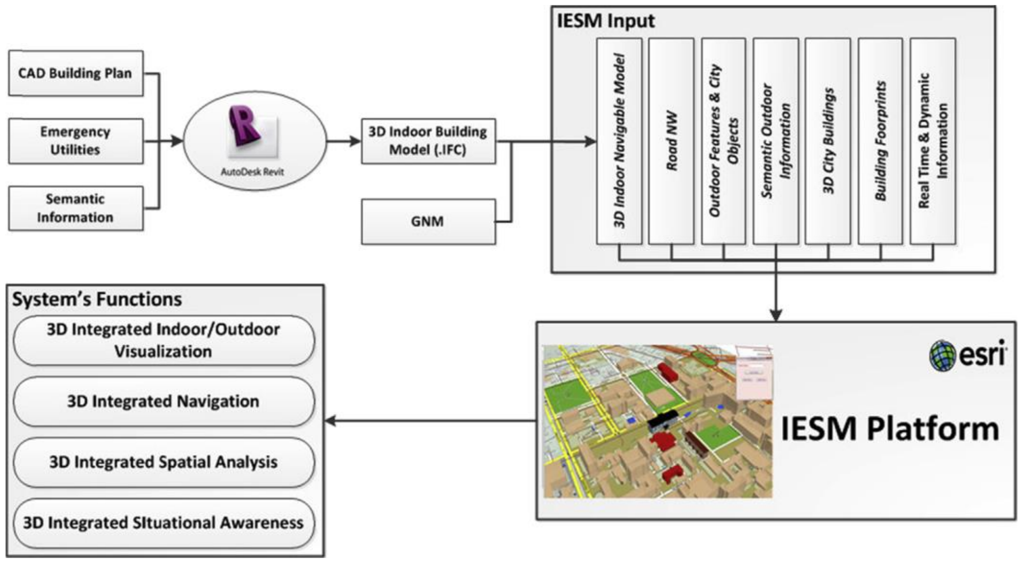


Figure 5 IESM implementation framework.

The Geometric Network Model (GNM), which is mentioned above, is a "*graph consisting of nodes and edges in which nodes represent position or location of an object such as a room while edges represent connection between nodes*" (Jamali et al. 2017) and it is a widely acceptable navigable network. Other approaches that have also looked at the GNM are provided by Li and Lee that integrated it with IndoorGML (Li and Lee 2010) and by Luo, Cao and Li that created it from 3D imaging and scanning technologies (Luo, Cao, and Li 2014). Specifically, the latest concept is implemented in a visual C++ and OpenGL development environment and tested by using real data from the East China Normal University Campus at Shanghai, China following three steps. Initially, the 3D building has to be decomposed to floors, then the geometric network elements have to be created interactively and finally, the geometric network models have to be generated automatically (Luo, Cao, and Li 2014). A contribution of this research regarding the GNM is its improvement by using doors of rooms as nodes instead of rooms. That fact makes sense as the door is the entrance of the room. The same approach is followed in this thesis as well.

A case study of a 3D network analysis in the interior of the Lampadarios building of the School of Rural and Surveying Engineering of the National Technical University of Athens is developed for the exploration of the potentials of the ESRI products in this field. The 3D network is created by using the 2D architectural plans of the floors in ArcScene, while the 3D model is generated in CityEngine for interactive visualization purposes. Having the model and the network stored in an ESRI geodatabase, a geoprocessing model is developed via Model Builder in ArcGIS that calculates the least-cost paths in 3D (Tsiliakou and Dimopoulou 2016). A system for the localization of the user is not necessary because starting and ending point are inserted manually every time the model runs.

The last part of this section describes an older scenario as it is the only example that results in a web service. The service is applied in the Hubei Provincial Museum and supports localization, navigation and visualization functions. The first step in this case includes the creation of a geometrical model with semantic which shows the navigation environment and displays context information. The next step uses the geometrical model and generates the network with topological values, while the final step connects the geometrical model with the network. The required information is stored in the PostgreSQL database with the PostGIS extension and the shortest path is calculated with the A* algorithm of the pgRouting extension (Xu et al. 2013).

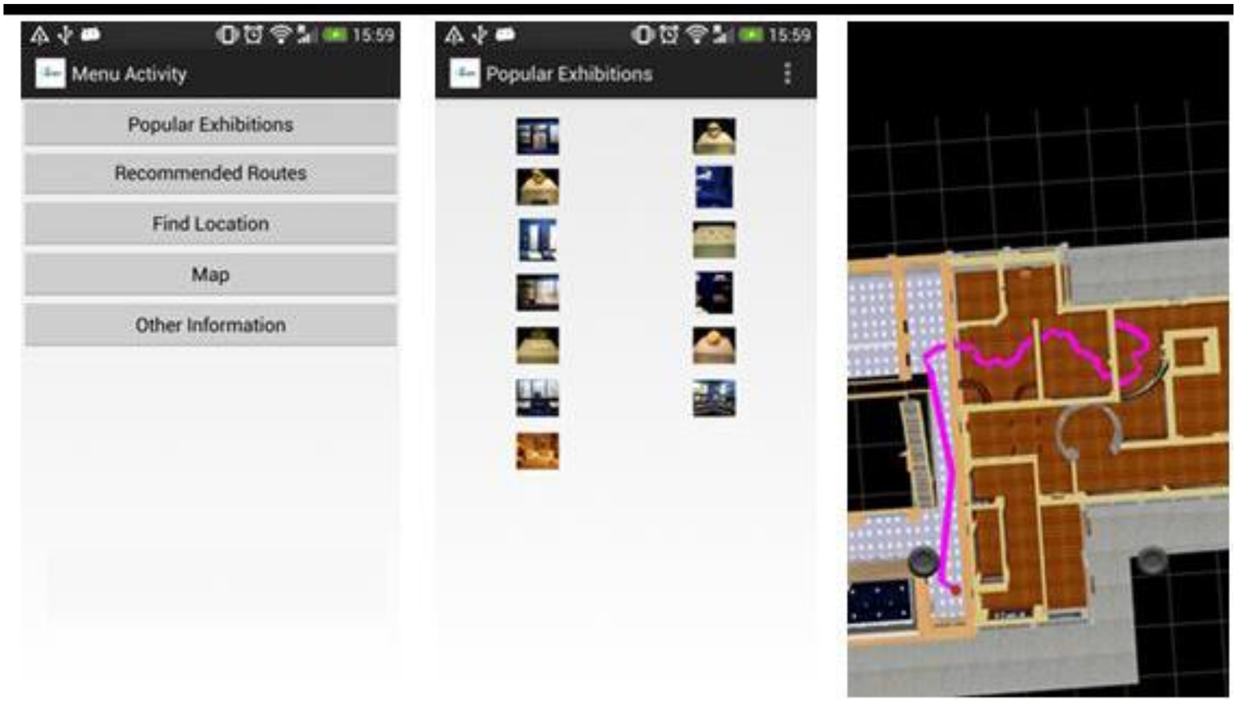


Figure 6 Left: main menu. Centre: popular exhibitions. Right: map view (Xu et al. 2013)

In conclusion, most of the examined cases are built on the three fundamental functional components presented in the previous section: localization, navigation and 3D model visualization, but follow different approaches. The most recent examples mainly focus on the semantic information provided by the model and the hierarchical structure of the models definitely plays a significant role in each of these cases.

2. 4. Technologies for routing calculations

This section shows the various existing technologies that are available in the market by evaluating their capabilities, their strengths and weaknesses and the offered possibilities in order to define the most proper tool for the development of an indoor navigation system. For this thesis's purposes, the selected technology should be free and open source, easy and fast to develop, while it should offer multiple options regarding the travel mode. Table 1 presents the selected technologies in correspondence to the priorities and standards set by the thesis's framework followed by a short description of the functionalities of the most interesting examples of the examined technologies.

Product	License	Cost	Language	Pedestrian	Wheelchair	Comments	Desktop/ mobile
Indoor Location	MIT License	Free ++	IOS Android JS	Yes	Yes	*'open-source framework for the connection of any indoor positioning technology to the Map wise Indoor Maps'*	Desktop App shared with key
OpenRoute service	CC-By 4.0	Free	Java	Yes	Yes	Further tools and libs for JavaScript, R, Python, JS QGIS etc.	Desktop
GraphHopper	Apache License 2.0	Free ++	Java	Yes	No	Fast and Memory efficient	Desktop + mobile
OpenTripPlanner	GNU Lesse General Public License v3	Free	Java	Yes	Yes	-	Desktop + mobile
Pyroute	GNU Lesse General Public License	Free	Python	-	-	Old, many items moved to OpenStreetMap	Desktop
OsmAnd	GPL v3 and artwork and design element	Free	C/Java	Yes	No	OSM Map, POI and Routing (car/bike/foot, online & offline) for	Desktop + mobile

	s CC-BY-ND-NC					Android and iOS	
Osm4routing		Free		Yes	No	Command-line tool for parsing OSM data into a routing graph.	Library/ Development tools
pgRouting	GPLv2	Free	SQL	Yes	Yes	PostGIS-based routing engine. Special tool osm2pgrouting for importing OSM data to internal graph structure. Works directly on top of SQL database tables.	Library/ Development tools
indrz	GNU General Public License v3.0	Free	_postgres user _virtualenv with python 3.4	Yes	No	-	-
Indoor Routing Xamarin ("Indoor Routing for Xamarin | ArcGIS for Developers" n.d.)	Copyright 2017 Esri Apache License, Version 2.0	Not Free		Yes	No	example app called Indoor Routing for iOS devices built in Xamarin with the ArcGIS Runtime SDK for .NET.	Mobile
Here Indoor Positioning and Navigation	HERE	Starter and Premium	Android and iOS SDKs	Yes	No	-	Mobile SDK
Navigine	Navigine	Free Demo	Android and iOS SDKs	Yes	No	Need of Beacons, Maybe similar to Indoor Location	Mobile SDK
Indoor Atlas	Indoor Atlas	Not Free	Android and iOS SDKs	Yes	No	-	-

Table 1 Available routing services ("Routing - OpenStreetMap Wiki" n.d.)

To begin with Indoor Location, it is an '*open-source framework for the connection of any indoor positioning technology to the Mapwise Indoor Maps*' under the MIT License ("Indoor Location - The Open-Source Framework for Indoor Mapping." n.d.). The various indoor positioning technologies are integrated, combined and shown on maps. This case uses Mapwise as base map application in iOS, Android and

JavaScript offering a tutorial for the creation of venues by using a png image of the floors on the map and drawing the paths and the rooms. However, the produced route and the background can be displayed only in 2D.

Openrouteservice provides a wide range of routing services, tools and libraries for JavaScript, R, Python and JS QGIS. It is a service based on user- generated free geographic data from OpenStreetMap with global coverage. GraphHopper is an equally impressive tool that allows routing planning through its API, which is also based on OpenStreetMap, but has no functionality for the use of wheelchair.

Additionally, Osm4routing provides an OpenStreetMap data parser to turn them into a nodes-edges adapted for routing applications. The input is an OpenStreetMap XML file and the output of this process may be either a csv file or a database theta consists of two files or tables: the nodes and the edges. The table of the nodes collects data about the coordinates of the decision points, while the table of the edges contain information about the foot accessibility (“Osm4routing,” n.d.). The specific tool cannot be used in the case of indoor spaces due to the exclusive outdoor coverage of OpenStreetMap’s data.

Another tool that provides geospatial routing functionalities is the pgRouting extension of the Postgres/PostGIS database. The database routing approach offers great benefits as data and attributes can be modified by many clients, either they use PCs or mobile devices. Furthermore, data changes can be reflected instantaneously through the routing engine without any precalculations, while the “cost” parameter can be dynamically calculated through SQL and its value can come from multiple fields or tables. PgRouting library contains several features, such the shortest path A* and the Dijkstra shortest path. It is open source and available under GPLv2 license (“PgRouting Project — Open Source Routing Library” n.d.).

Summing up, this section investigates the available technological options for the calculation of routing paths aiming on the selection of the most suitable tool for the purposes of this thesis. The tool should be cost-free and easy to integrate with the rest elements of the system (3D model and navigation). For these reasons, pgRouting is selected and being applied during the implementation phase of the present thesis.

3. Software and Technologies

The final product of this thesis is a prototype service that provides routing instructions and planning supported in a three-dimensional virtual environment. This chapter presents the selected tools and technologies that are being used for the development of this service and are divided in 3 sections based on their functionalities.

Specifically, the first section describes the selected programming languages and libraries along with the purposes of their selection in the process of developing a WebGIS service enabling 3D visualizations. The second section provides information about the web server and the use of a database for storing data about the network and the calculation of the routing path. Finally, the last section examines several softwares that are being used for the generation of the 3D building model and the network of the chosen case study.

3.1.Programming Languages and Libraries

Regarding the programming languages, HTLM, CSS and JavaScript are being used in this service for the creation of interactive webpages. HTML, which stands for Hyper Text Markup Language, is the core markup language of the World Wide Web ("HTML 5.2: 1. Introduction" n.d.) and creates static webpage content. HTML is important in this project because it provides the basic structure of the webpages and makes the websites accessible by anyone connected to the Internet ("What Is HTML? | HyperText Markup Language Explained" n.d.).

In addition to HTML, CSS is used for defining the style of the webpage. CSS, which stands for Cascading Style Sheets, is a programming language that describes the style of an HTML document and how HTML elements should be shown on the screen (Tutorial and Brookshier 2015). CSS is a means to style an HTML file by easily manipulating the fonts, the backgrounds and the positions of the elements on the screen. It is used in this project for styling purposes and for indicating the differences among the elements based on their usage.

Finally, a JavaScript library is used in this project for adding functionalities on the map. JavaScript is the third core pillars of web development along with CSS and HTML. It is a programming language complementary to HTML that runs within the HTML structure of the webpage and adds interaction in the webpage by performing dynamic tasks in it ("Introduction to JavaScript" n.d.).

Specifically, the library that is being used for enabling a 3D georeferenced virtual environment is called CesiumJS and is an open source JavaScript library for the creation of world-class virtual 3D globes and maps. It is a geospatial 3D mapping platform providing the background of a web-based globe for free and allows the visualization of dynamic data with ease, precision and visual quality ("CesiumJS - Geospatial 3D Mapping and Virtual Globe Platform" n.d.).

It is finally worth describing the term of Cesium 3D Tiles, as they are extensively used in this project. Cesium 3D Tiles are an '*open specification for streaming massive heterogeneous 3D geospatial datasets'* expanding the ability of Cesium to visualize terrain and imagery on the web ("Introducing 3D Tiles | Cesium.Com" n.d.). Any kind of 3D content, from buildings to people, and any kind of 3D file format, from point clouds and CAD/BIM to vector data, can be streamed on 3D Tiles and support interactive selection and styling.

One of the most important benefits gained by using 3D Tiles is the great improvement of streaming and rendering performance of the datasets due to its Hierarchical Level of Detail (HLOD) capability, that enables the streaming of only visible tiles. As a result, the rendering, which is achieved with WebGL, is fast and simple and minimizes the client-side processing, while the WebGL draw calls are reduced as the tiles are pre-batched or batched on-the-fly ("Introducing 3D Tiles | Cesium.Com" n.d.). In this particular study case, the 3D building model is converted to 3D tiles via FME, a data integration platform which is further described in the next section, and then called in the 3D WebGIS.

3.2.Database and Server

One of the main characteristics of the developed web service is the fact that greatly interacts with a database that stores information about the network and the places of interest or the decision points. The selected database is the PostgreSQL with the PostGIS extension because it supports spatial data and is open source. It also provides the possibility to query data in a time and computing power efficient way.

pgRouting is an extension of the PostGIS/ PostgreSQL geospatial database being chosen in this thesis to provide geospatial routing functionality ("PgRouting Project — Open Source Routing Library" n.d.) for three reasons. Firstly, there is no need for precalculations as data changes are reflected through the routing engine at once. Then the cost parameter can come from multiple fields, which is convenient in the case of various mobility options decreasing the size of stored data. Finally, pgRouting library contains features that calculate routing with several algorithms such as the Dijkstra' s algorithm and the A*. Further information about the database design and the relationship among the elements is described in the section 5.1.3 Network and Routing Path.

Apache is the selected web server that provides the link between the user and the service, while handling multiple requests from the client. It is one of the most popular web servers and is used in the current project as part of the XAMPP open source package.

The connection between the webpage and the database is achieved through the webserver, SQL, PHP and Fetch API. These three tools are being used to query the database, call the data and pass them to the JavaScript code respectively. A small PHP script containing a SQL query calls data from the database, such as the latitude, the longitude and the elevation of the starting and ending points and the Fetch API provides the possibility to fetch this information in the JavaScript code and use it for the visualization of the corresponding points and the routing path.

3.3.Software

As the examination of the specific case study of the Aalborg University in Copenhagen lacks of an existing 3D building model and network, the first phase of this thesis includes the generation of the required data. This section presents some additional tools that are used for the creation of data and models, their preprocessing or validation. The workflow of data conversions is illustrated in Figure 7 for both the building model development and the network while chapter 5 describes in detail the development of these two elements.

Initially, AutoCAD is a 2D and 3D computer- aided drafting software application provided by Autodesk and is widely used in the fields of architecture and construction for the preparation of technical plans ("What Is AutoCAD?" n.d.). AutoCAD is the primary tool used in this project for the generation of the editable digital floor plans of the examined building and its network.

Then, 3ds Max, which is a 3D modeling and rendering software that '*helps you create massive worlds in games, stunning scenes for design visualization, and engaging virtual reality experiences'* ("3ds Max | 3D Modeling, Animation & Rendering Software | Autodesk" n.d.), is used in this case study for the creation of the 3D building model.

ArcScene is '*a 3D visualization application that allows you to view your GIS data in three dimensions*', overlaying multiple layers of data in a 3D environment ("3D Analyst and ArcScene—Help | ArcGIS Desktop" n.d.). The use of ArcScene is limited to visualization purposes for minor processing and validation of the network.

AutoCAD Map 3D is a software that '*provides access to GIS and mapping data to support planning, design, and data management*' ("Free Software for Students & Educators | AutoCAD Map 3D | Autodesk" n.d.). This software is used for the creation of the network in CAD format and its exportation to GIS format.

Finally, FME is a data integration platform developed by Safe Software which simplifies and facilitates the transformation of data by using automate workflows ("Safe Software | FME | Data Integration Platform" n.d.) without requiring the access to many commercial softwares. It is not limited by versions and can handle heavy files conversions. The main reason why FME is used in this project is the conversion of the generated 3D building model to Cesium 3D Tiles, as shown in Figure 7.

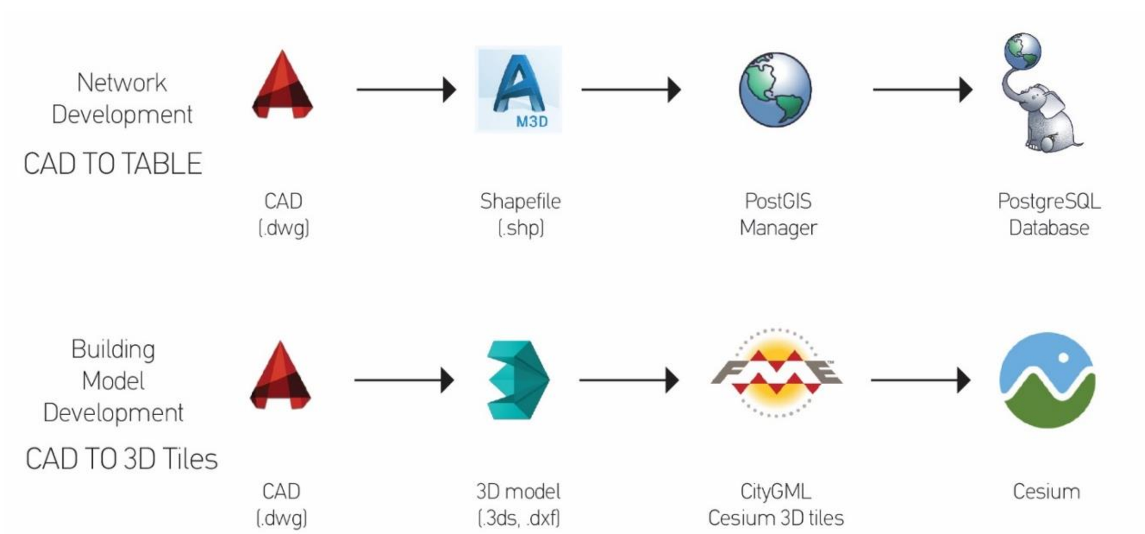


Figure 7 Conversions workflow for the network and the building model development

4. Conceptual Design

As it has already been mentioned in chapter 2 Background, a successfully operating system for route planning consists of three components: the detection of the user' s position in the examined area, the calculation of the routing path and the representation of the former elements in a comprehensive and coherent manner leading the most suitable way to the desired destination.

The present study proposes a navigation service in a 3D virtual environment for the calculation of routing paths between two points. The solution focuses the attention on the path planning and the representation techniques, leaving alone localization since it needs special equipment installations that exceed the specifications of such a project. The project's case study is the Building A at the University of Aalborg in Copenhagen, which lacks three-dimensional routing tools. The aim of this study is the development of a desktop- based WebGIS that allows the three-dimensional navigation around the examined building and the calculation of the most suitable route between two selected points.

The service should be cost-free, easily and quickly developed as a desktop application, providing technical robustness without installation requirements. The environment of the system should be friendly and easy to understand and to use by the user, while a balanced and comprehensive final visual result is considered of great significance due to the fact that few people are familiar to 3D navigation while it is proven that it requires some adjustment time before being used to it.

The conceptual structure of the service includes two stages. The first stage displays the examined building on its actual location on the map and offers the user the chance to navigate around it and explore it in 3D mode identifying the outer space of the case study. At this point, the user should select the desired start and destination point and the travel mode. The second stage presents the routing path between the chosen start and end points, the corresponding points and the indoor space of the building in an abstract form.

Apart from the technical- technological conditions of the service that should be set from the beginning and are presented in chapter 5, there are also some further reflections that should be taken into account regarding the impact of the service to the user, the interaction between the service and the user and basically the methods of enhancement of the navigational experience. To put the matter another way, since the routing service is desktop-based and does not include localization mechanisms, it should export easily understandable, meaningful and memorable results for the user. But how can this be achieved?

The following sections of the present chapter are devoted in the examination of techniques that may improve the performance of the service allowing the user to memorize conveniently the results and navigate independently in the corresponding spaces. The first section explores how navigation is influenced by memory or differently said how navigation is achieved without digital means, concluding to the key-components that enhance way-finding performance. The next section identifies the significance of landmarks and animations in 'absorbing' spatial information before starting the physical research of the host places. It also notices the most important features of landmarks that are distinguished by most of the people, while the last section describes the requirements that should be taken into consideration during the model development.

4.1.Spatial memory and Way-finding Performance

Navigation and orientation are proven to be influenced by biological and neurological mechanisms constituting a spatial positioning system in the brain (Brandt, Zwergal, and Glasauer 2016). Recently, the interest on the relations between the neurological systems that are involved in the topological memory has been growing rapidly revealing great progress on understanding how visual stimuluses affect navigation and how the brain processes them. This study although does not explore the functions of the neurological systems that affect spatial navigation but focuses on the perceptible cues that form spatial memory and define the instant position and the movement of an individual in a three-dimensional environment.

Specifically, spatial memory refers on the part of the memory system that is responsible for encoding, storing, recognizing and recalling spatial information about the environment and the orientation of the user in it (Madl et al. 2015). Its evolution varies due to both genetic and environmental factors and it may be triggered by a wide range of cues that provide enhanced effects regarding navigation depending on person's unique experiences and spatial perception. Already by 1960 Lynch defined way-finding as '*a consistent use and organization of definite sensory cues from the external environment*', while looking a little further about how people develop spatial knowledge and cognitive navigational abilities, Cousins, Siegel and Maxwell set three measures of way-finding: 1. the landmark knowledge that provides reference in relation to the environment, 2. the route knowledge that puts the landmarks into sequence and 3. the configurational knowledge that organizes the landmarks and the routes in coordinated representations (Cousins, Siegel, and Maxwell 1983).

Either they are called cues or measures of spatial knowledge, they are definitely essential elements of people's cognitive maps. Most individuals develop daily cognitive maps that are based on their environmental information through perception, natural language, and inferences (Raubal and Egenhofer 1998) and improve their navigational skills. But what are exactly these key-elements that improve the construction of a cognitive map and eventually a way- finding performance? And what are the key-elements for the construction of a cognitive map of an unfamiliar place before visiting it? These two questions are of fundamental significance to be addressed during the development of an abstract navigational service in order to properly satisfy its major purposes.

According to Lynch, imageability, which is the '*quality in a physical object which gives it a high probability of evoking a strong image in any given observer'* is the foundation for a proper navigational experience while for him the city image can be divided into paths, edges, regions, nodes and landmarks (Lynch 1960). Following a different approach, Weisman proposed 4 classes of environmental variables that affect the navigational performance. These variables are: the visual access, the degree of architectural differentiation, the use of signs and room numbers and the plan configuration and arise a great discussion about their degree of importance (Weisman 1981) (Raubal and Egenhofer 1998).

In a nutshell, for both authors, visual access, or otherwise imageability of an object or a space, gains the greater weight when considering the trigger point of spatial memory. It stands out from the rest variables, making it the most significant element to be taken into account when building a cognitive spatial map. Especially in the investigating case of a routing service that operates on the desktop clues, visibility and visual recognition of the various qualities of the spaces play significant role in stimulating spatial memory and in provoking better memory performances. But is visual access enough for enhancing advanced spatial memory performances? This study supports that the object or the space that gains a place in the 'visibility spectrum' of the user should be unique, worthy to be noticed and to be remembered, such as a benchmark or a landmark. The following section investigates the features that make an object or a space to be noticed as a landmark and the elements that raise its imageability along with its interaction with 3D models.

4.2.Animation and landmarks for enhancing spatial memory

Human navigation relies on the mobility and orientation skills of the individual; people use either path integration moving relatively to their starting point or landmarks based on visual stimulus or the combination of these two above mentioned elements (Fallah et al. 2013). As the previous section analyzes, visual access and imageability of a space are of great significance for describing a landmark, while a variety of landmarks is most commonly being provided for effective directions. 'Move across the bridge', 'Follow the path to the church' or 'Go left to the park' are some instances of given instructions that use landmarks to provide guidance to the asked destination. But what are the most important measures of saliency that 'convert' a simple space or an object to a landmark in most people's eyes?

According to Schroder, Mackaness, and Gittings name, size, age and color are revealed to be the most salient features to distinguish a building in outdoors spaces, followed by the emotions towards the space, the decoration, the relative location, the construction materials, its architecture, function, shape and condition. Considering that the present study works on the interior spaces of a building, some of these characteristics or their combinations may be of great help for providing memorable, written instructions along the routing path. Such instructions in a university building may refer basically to the function of the spaces and could guide the user move towards the canteen, take the spiral metal staircase or move through a blue wall. Table 2 shows the selected landmarks that are being used in the final product of this study.

Landmark	Features to salient	Written Instructions
Big Canteen	Function, Size	Move through the Big Canteen!
Spiral metal Staircase	Shape, Function	Take the spiral staircase!
Stairs	Function	Take the stairs!
Elevator	Function	Take the elevator!
Info Point	Function	Pass by the Info Point!
Atrium	Shape	Walk through the atrium! Walk around the atrium!
Blue wall	Color	Pass by the blue wall!
Entrance/Exit	Function	Use the Entrance/Exit!

Table 2 Identification of landmarks

More specifically, the proposed service may both visualize the text on the side of the map and the landmark on the 3D map for memory and vision initiations. Based on Sharkawi, Ujang, and Abdul-Rahman landmarks are easier recognizable in 3D models rather than in a symbolic 2D map making 3D visualization more relevant for 3D navigation. Furthermore, the high visual correspondence between map objects and real world objects increases the navigational value of the 3D map (Sharkawi, Ujang, and Abdul-Rahman 2008) and allows the easier recognition of the spaces (Kray et al. 2005), leading to a more intuitive navigation inside the map in comparison to a 2D-map (Haist, Reitz, and Coors 2006).

Apart from the visualization of landmarks on the 3D map and their indication on written guidelines, another factor that is believed to enhance user's spatial abilities is animation. Bederson and Boltman support that '*animation improves users' ability to reconstruct the information space, with no penalty on task performance time*', while Hays's study results in proving that animation makes significantly great gains especially to low-spatial-ability subjects compared to these that do not receive animation. Particularly, technologies for 3D visualization and interactive animation offer great advantages regarding management and access of large information spaces, providing even further profits when the structure of the information is visualized (Robertson, Mackinlay, and Card 1991). Understanding the significance of such a technique in the improvement of the user's way-finding performances, the system that is proposed in this study attempts to produce an animated visualization of the routing path in the three-dimensional virtual environment. The results are presented in the last part of chapter 5 Front End.

4.3.Requirements analysis

Topographic Space as referred in Brown et al. is crucial for indoor navigation and represents the interior spaces of the examined building semantically decomposing it into its elements. Based on the same source, building model is the means that provides topographic space information underlying although the risk of inconsistency or incompletion of its captured components due to '*the lack of understanding of the use cases for indoor navigation topographic space information and the corresponding requirements*' (Brown et al. 2013). In other words, in order to enhance the semantics of the model by defining its components and its hierarchic order a series of various requirements should be fulfilled. This project considers the exploration of these requirements as a valuable step for the understanding of the proper preparation of a suitable building model and the successful function of the produced service for the desired purposes.

Before making the list of requirements that a building model should cover, there are several challenges that a general- purpose building model is meant to meet. These challenges vary depending on the navigation tasks, the navigation environment, the users, the modes of locomotion and the scenario (Brown et al. 2013). For example, the model might represent a hospital, a library or an airport, address to pedestrians on foot or in a wheelchair with complete, limited or restricted access to specific areas and work in rush hours or in emergency situations. Undoubtedly, the requirements are greatly dependent on the use of topographic space information and its conceptual index, while the visualization mode may also play a significant role in their investigation. Consequently, the particular use cases are firstly explored and listed aiming in the latter abstraction of the corresponding requirements.

The selected case study is a university building which mainly consists of common areas, teaching rooms, laboratories, auditoriums and offices. The users might be either students, educational staff or visitors of the campus for meetings and gatherings moving on foot or in a wheelchair. This service should not provide exclusive information in emergency scenarios but may alert for the use of limited- access areas, such as rooms accessible only with student or guest card. The routing service that hosts the building model and calculates the navigation path is desktop- based giving rise to some additional requirements that are also examined in the last part of the section.

According to the observations of many authors (i.e. (Brown et al. 2013), (Isikdag, Zlatanova, and Underwood 2013), (Tsiliakou and Dimopoulou 2016)) and after having strictly defined the use case, the clarification of the requirements may be achieved progressively. General semantic information is the first and foremost requirement noticed in the literature review which identifies the building's elements (storeys, rooms, landmarks) and the relationships among them, while enhances orientation and guidance during navigation. Apart from semantically, the spaces and objects should also be represented geometrically assigning simultaneously context properties to the buildings' elements for improved navigational experience. For instance, the visualization of the shape, size, texture and color of a door may be determinant for reaching a destination point.

Then, categorization and decomposition of spaces may also be helpful for the determination of the appropriate direction. Furthermore, all indoor spaces should be geocoded and stored in accordance to the semantic information of their usage, function and capacity. Connecting or transfer spaces should be clear for the derivation of the navigation network, while 3D information of fixed building parts (stairs, columns, elevators) is significantly useful throughout navigation in the interior of a building. Moveable objects with further semantic information assigned should also be taken into account whilst the definition of their functional state considering various temporal changes is important for advanced navigation. Lastly, interaction with outdoor spaces should be ensured providing supplementary reference points which establish improved accessibility and guidance both inside and outside the building (Brown et al. 2013).

Focusing now the attention on the selected case study, the last part of this section attempts to answer the way how the above- mentioned requirements are fulfilled based on the various priorities and possibilities set throughout the generation and processing phase of the building model. To start with the implemented semantic information, the building is divided into storeys that consist of slabs, vertical built elements (walls), open and closed spaces (corridors and rooms) and vertical mobility parts (elevators and staircases). Vertical built parts in each room leave an opening but there is no indication of entrance in the volume of the closed spaces. The closed spaces may be further categorized based on their usage to auditoriums, seminar rooms, offices or laboratories. These objects are represented geometrically and variations in colors indicate their usage. Similarly, the vertical mobility components (staircases and elevators) are specifically colored volumes. The model is georeferenced in the WGS 84/ UTM zone 32, EPSG:32632 coordinate system.

The requirements explored in this section concentrate on the characteristics of a building model that accommodates route planning. Having covered the major issues regarding the building model in this section and the prerequisites for improving spatial memory performances in the previous ones, various requirements seem to be necessary to make the whole routing service trustable and satisfactory. Even though an efficient navigation system should always follow the user, in the present thesis the service does not include localization infrastructure and is desktop- based. This fact means that the user searches directions before starting his/her navigation and most likely does not have access to the service during navigation.

Recognizing this specific aspect of the service, the need of creating general guidelines emphasizing in the most critical points along with the routing path, instead of step-by-step instructions, is produced. Moreover, the use of animations that navigate through the calculated route aids memorizing the path and raises awareness about the surrounding spaces that the user should pass by. Last but not least, the significance of benchmarks is worth mentioning at this point explaining the need of locating them in the building model. Highlighting landmarks facilitates the perception of the passing through spaces, while the use of marks that are easily caught by the user’s eye and correspond to actual landmarks in the building is proven to be extremely efficient in providing route directions and leading the way to the final destination, as they are considered intermediate destinations.

.

5. Front End

The following chapter includes the prototype design. It describes the development of the proposed routing service and presents the results of the final product. The chosen study area is the Building A at the University of Aalborg in Copenhagen limited to its first three floors due to time limitations and repetitions of the processes that would not offer any additional benefits in this thesis. Chapter 5.1 Implementation focuses on the description of the workflow of the service and the generation of the required data for the thesis's purposes, while 5.2 Results shows the navigation WebGIS service, its structure and examines the most important parts of the applied script.

5.1.Implementation

This chapter details the work tasks performed in this study during the implementation phase. Building A at the University of Aalborg in Copenhagen is the selected case study of this thesis even though the unavailability of important data may be considered a drawback for the implementation of the routing service. The creation of the 3D spatial building model of the corresponding study area and its 3D network is considered of great significance in the beginning of the study followed by the formation of the database for storing useful data for the route finding and the presentation of the route on the web service. The first section briefly presents the workflow of the produced service, while the next sections show how the network and the 3D model are relatively generated.

More particularly, section 5.1.2 Model Development describes the process of creating editable CAD data of the floors of interest and the transformation of this data to CityGML and 3D Tiles formats. Regarding the network development and the route planning, section 2. 4 Technologies for routing calculations presents various technologies that provide the possibility of calculating routing paths between two or more points in indoor environments. However, none of them is able to calculate the route by using three-dimensional data. pgRouting which is the selected tool to be used in this service works directly on top of SQL database tables and allows the route planning in 2D. For this reason, the network is divided in floors and transitions; each floor and transition level corresponds to one table, that later are combined all together in one. Section 5.1.3 Network and Routing Path presents this process in detail.

5.1.1 Workflow and Data

After having chosen the case study, three successive steps are defined for the development of the indoor navigation service: 1. the creation of the 3D building model of the examined environment, 2. the formation of its network along with the route calculation and 3. the actual development of the service that displays the two previous steps. These three steps are described in detail in the following sections, while the present section makes a short introduction in the way the service works, the data that are required and the used technologies.

Figure 8 illustrates the conceptual workflow of the final routing service, which consists of two webpages. The first one should show the examined building and its surrounding area and offer the user the chance to select the points of interest among which wishes to navigate. The second webpage should indicate the route, the start and the destination points and provide helpful instructions for reaching the desired destination providing the possibility of 3D navigation around the path.

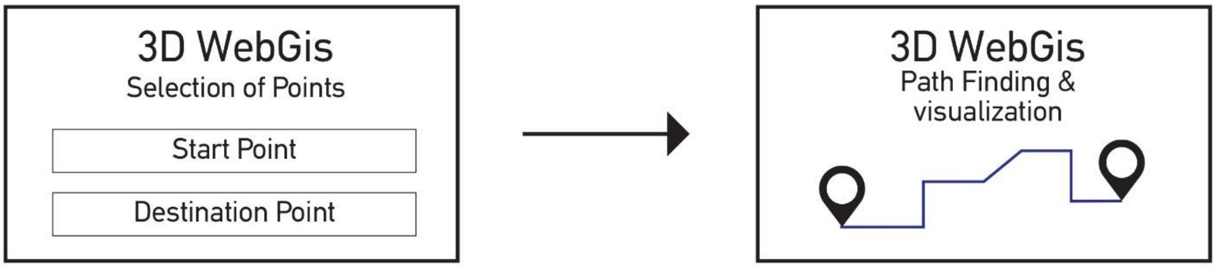


Figure 8 Conceptual Workflow of the service.

The applied technologies and their position in the workflow of the service are shown in the following image. The two webpages are developed by using CSS, HTML, JavaScript and the Cesium JS library and both of them load data about the model from the local server in the form of 3D Cesium Tiles. The difference between these two websites is the interaction of the second with the database from where it gets data about the network and the points of interest. pgRouting is used for finding the route path. The appropriate script runs in PHP calculating the route and storing information about it in the database, while another script calls the routing information for visualization on the 3D map. More details about it are found in 5.2 Results.

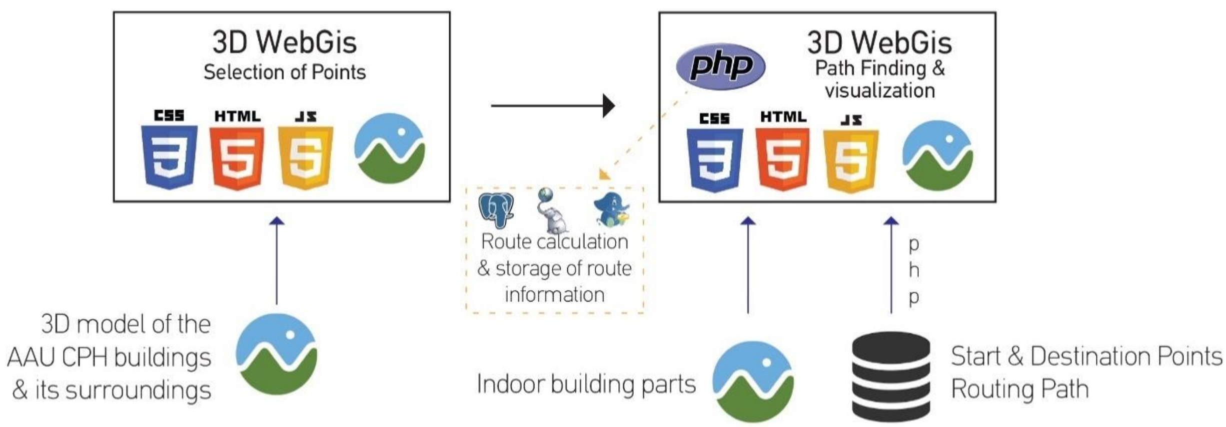


Figure 9 Workflow of the applied technologies

Considering the lack of an available 3D model of the interior of the building and the necessity of generating one, data about the blueprints of the building are required. Even though, finding an editable format of the floor plans of the building was not successful, their images are provided in a pdf format in the university's webpage ("Campus Map" n.d.) (Figure 10) and can be used for the extraction of the principal features.

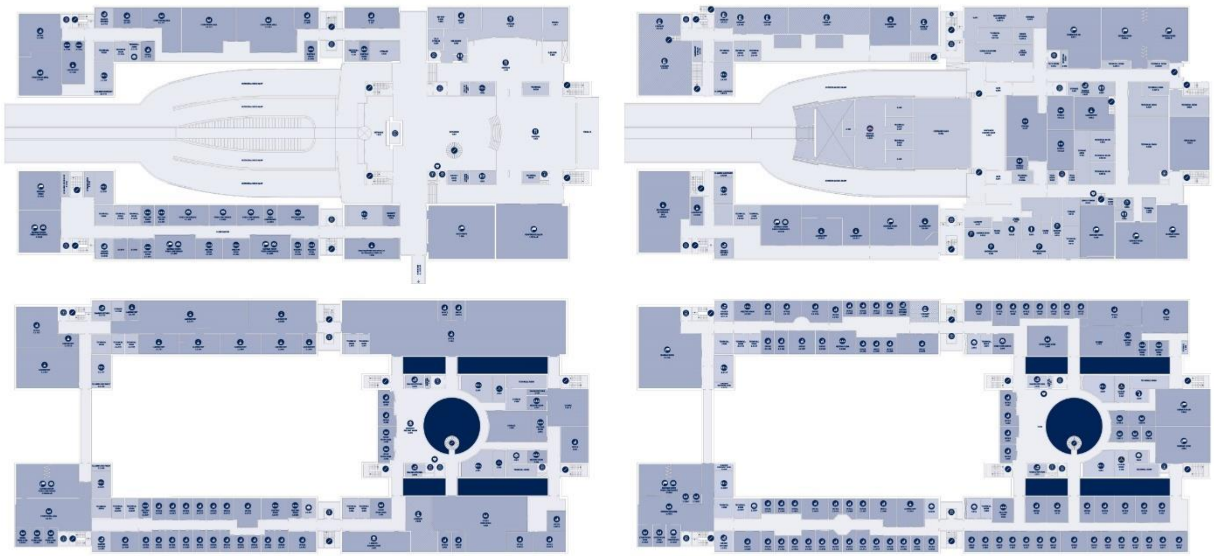


Figure 10 Floor plans of the Building A (groundfloor,1st, 2nd, 3rd floor)

5.1.2 Model Development

The 3D indoor building model is one of the two most important elements of a 3D indoor routing application enabling visualization of the spaces of the building. Section 2. 2 Indoor space modeling details the significance of the hierarchical structure of the interior environments and the various Level-of-Details that can be achieved based on the need of the user. The model in this study is chosen to be built in LOD 2.5 for improved visualization performances and due to time limitations as a 3D model needs long time to be built. The LOD 2.5 model is as simple as taking into account closed and open spaces, corridors and walls. No windows or doors are represented in this case.

The initial phase of the model generation is the extraction of the main elements of the floorplans in an editable CAD format. AutoCAD is used for the 2D drawing of the interior and exterior walls of the building, the slabs and the outline of the closed spaces divided by storey (Figure 11). Each element should be drawn as closed polyline and attached to specific layer for facilitating the categorization of the components and their automated extrusion to the third dimension for the creation of geometrical forms. Simple lines are also drawn to indicate the position of doors and are later used for the creation of the network.

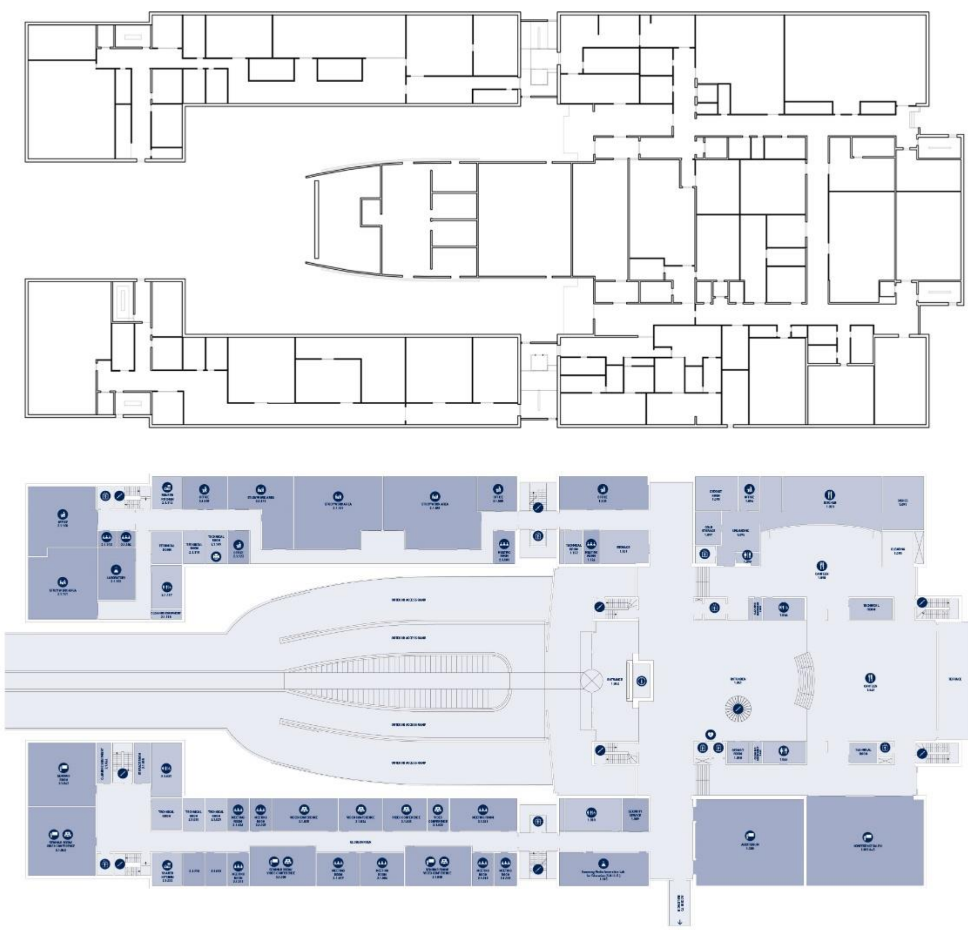


Figure 11 The original plan of the ground floor (down) and the produced drawing (phase 1)

The next phase includes the generation of the 3D model in 3DS max. Even though AutoCAD support 3D modelling, 3DS max is selected for building the model due to its flexibility and advanced properties with complex geometries. Each storey has 4.5m height and consists of the slabs, the volumes of the closed spaces (Figure 12) and the indoor walls, while there are also volumes for the staircases and the elevators (Figure 13).

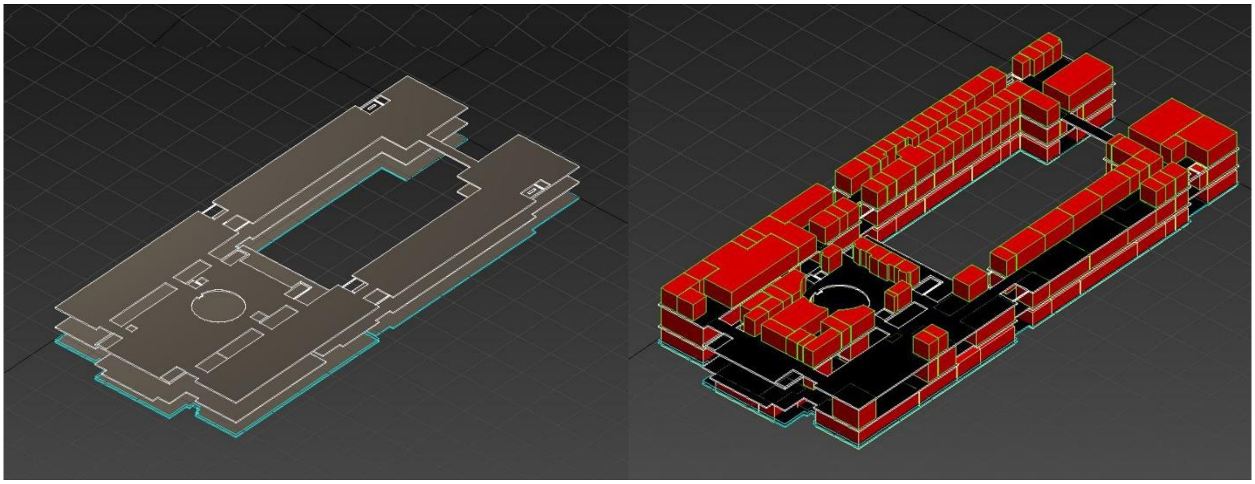


Figure 12 Slabs of the 3D model (LOD1: left) and interior volumes

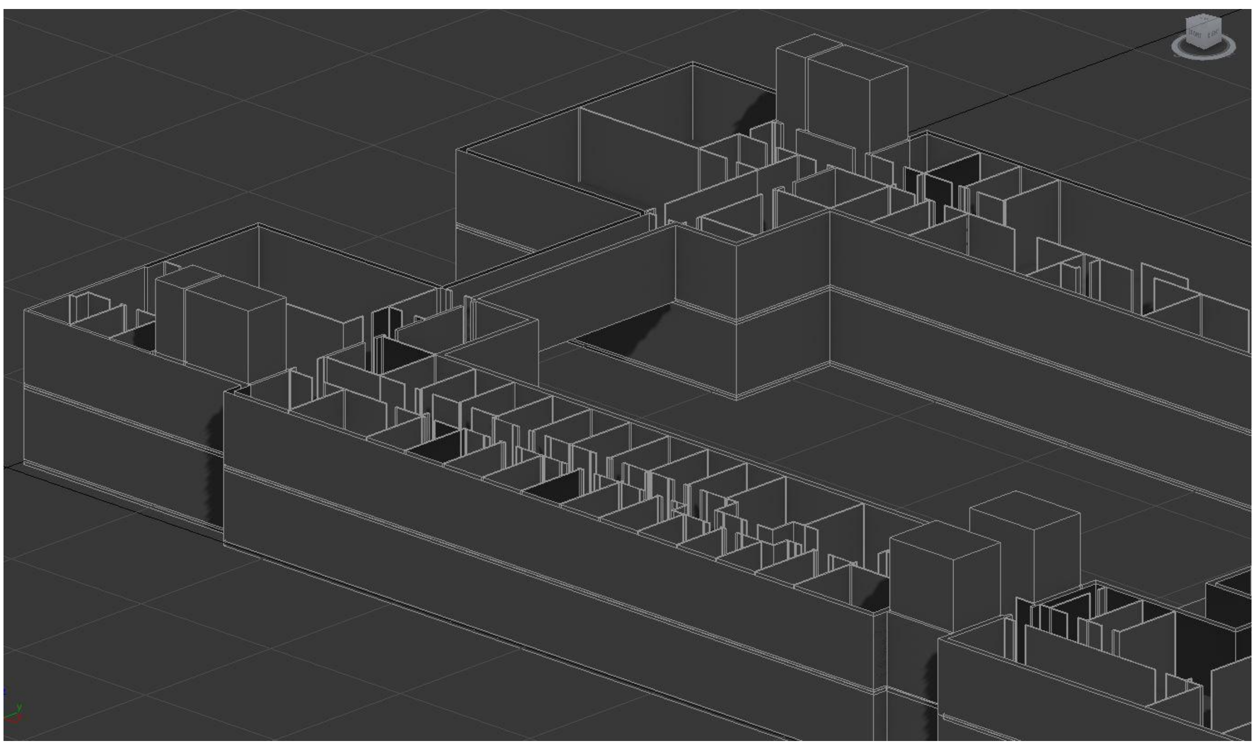


Figure 13 Interior and exterior walls of the building (LOD 3)

Then, the model is exported in .fbx format and has to be converted to .gml before the final transformation to 3D Cesium Tiles. These last two processes take place in two FME workspaces. The first workspace is presented in Figure 14 where the corresponding transformers assign coordinate system, direction and scale to the model. The last two transformers before the writer proceed for the preparation and validation of the data by creating specific attributes necessary for writing the CityGML file.

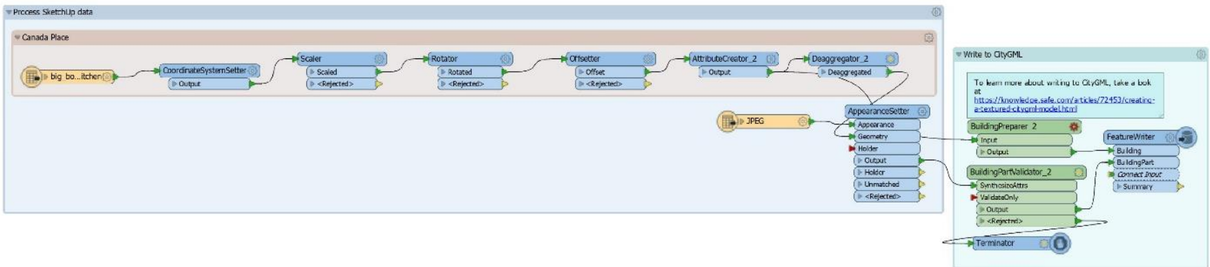


Figure 14 FBX to CityGML FME workspace

The final step converts the CityGML file to 3D Cesium Tiles in another FME workspace where only a reader and a writer are necessary. The produced files are later moved to the local server for loading in the webpage.

5.1.3 Network and Routing Path

The next essential component for the proper function of the navigation service is the network and the calculation of the navigation path. As it has already been mentioned in the introduction of this chapter, due to the unavailability of a suitable routing tool that could manipulate 3D information, pgRouting is selected to work directly on top of the SQL database tables and calculate the routing paths in 2D. To overcome the three-dimensional challenges, the network is originally generated by floor and vertical mobility; each floor and transition level corresponds to one graph/ table and is differentiated by layer.

Going into more details, the network is firstly drawn in AutoCAD Map by using the floor plans after having georeferenced them in the actual position of the building in the WGS 84/ UTM zone 32, EPSG:32632 coordinate system through the ArcGIS for AutoCAD tool. It is a graph model with connected edges among the decision points for each floor as can be seen in red in Figure 15. The graph reaches all various closed and open spaces but does not cover mobility in the rooms as moveable equipment is not taken into account in this thesis.

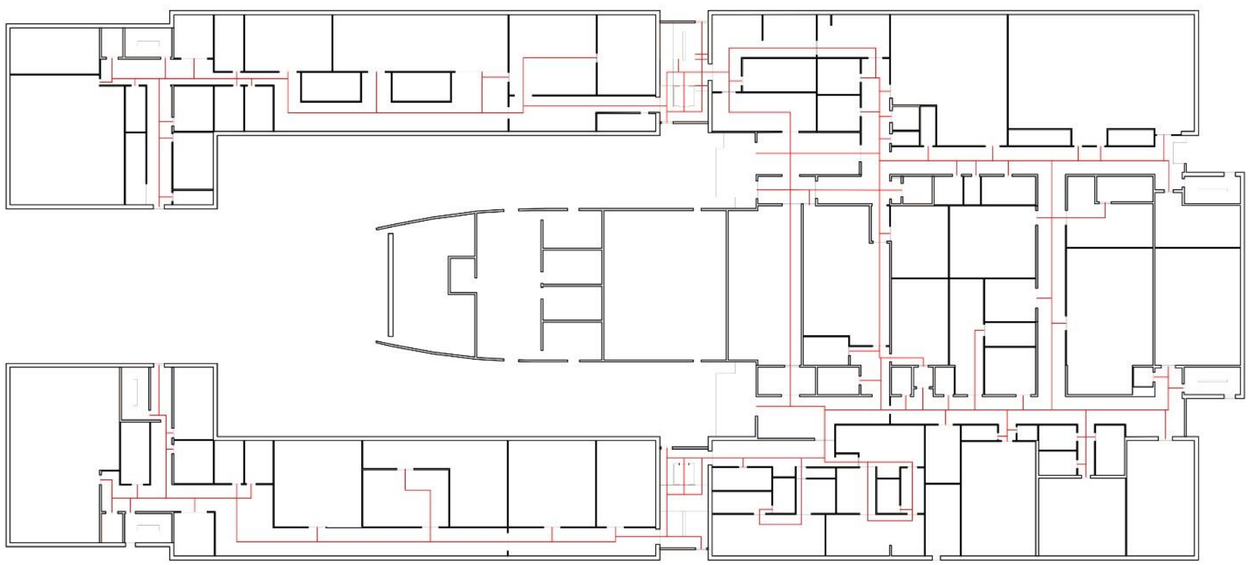


Figure 15 Ground floor network

Three different graphs are created for each of the examined floors, assigned to different layers and set in their corresponding elevations. Vertical movement is drawn in 3D representing the actual movement around the stairs and the elevator. The relationships among the produced graphs is shown in Figure 16.

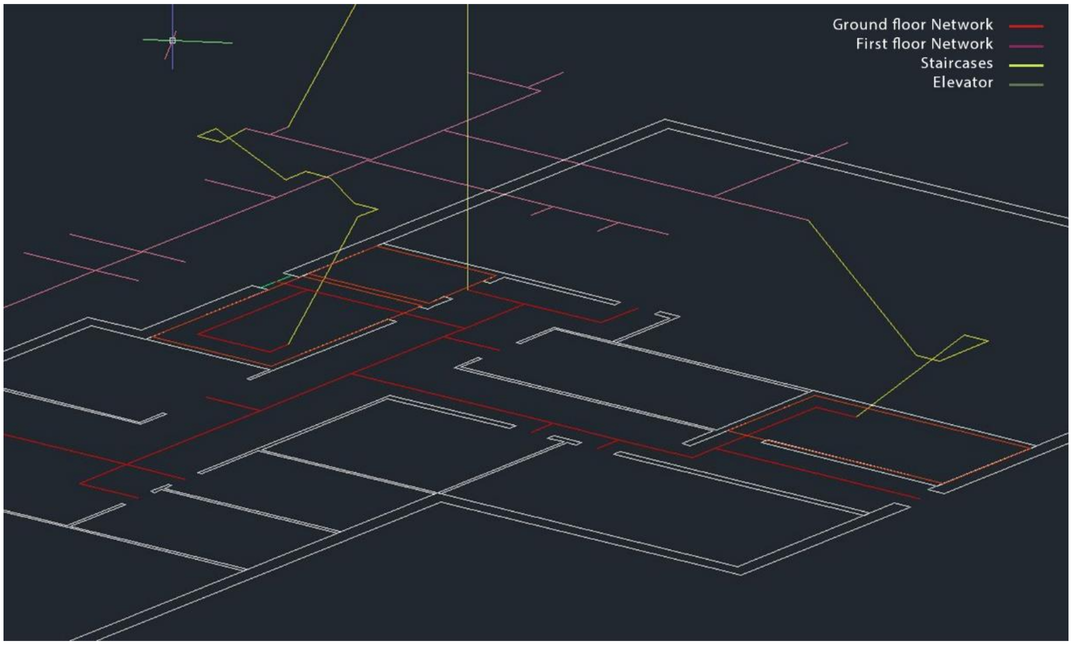


Figure 16 The network graphs and the relations among them in 3D representation

At this point the network is complete but it is only a CAD graph and does not have any attributes to allow routing calculations. The solution to that is given by starting using the PostgreSQL database with the PostGIS and the pgRouting extensions. Each of the graphs is exported in different shapefiles through the AutoCAD Map and then imported in the database via the PostGIS Shapefile Import/Export Manager. Exporting the graphs in different shapefiles is important for the facilitation of the next processes due to the fact that pgRouting is not able to calculate the shortest path in 3D.

Specifically, the standard steps followed for achieving routing calculations through pgRouting are:

1. Create a routing Database with PostGIS and pgRouting extensions
2. Load vector data about the network
3. Build a routing topology for the network data
4. Create a table with the decision points of the network
5. Compute the path

Accordingly, a database called routing is firstly created and the necessary extensions are loaded. Afterwards, the exported shapefiles are imported in the database by using the PostGIS manager. The crucial point that affects the usual procedures and determines variations in the processes in the specific case is the three-dimensional properties of the data and the inability of the corresponding tool to build the routing topology when the geometry of the network is stored as 3D linestrings (LineStringZM). The first phase of the routing database is shown in Figure 17.

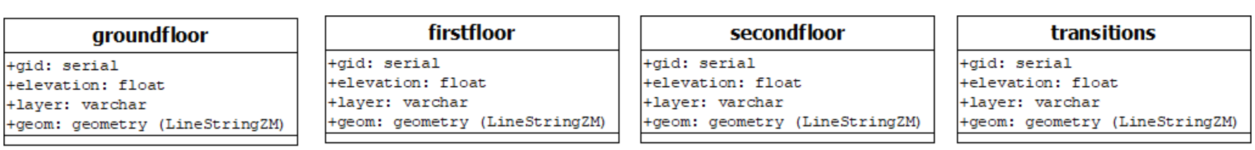


Figure 17 First phase of the routing database

To overcome these limitations each shapefile and respectively each table in the database stores information about only one floor regardless its elevation which is stored as a different attribute in the table along with the name of the layer. Before proceeding to the next step, three additional columns should be added in each of the tables called source, target and length, where the former two are required to be used during the following phases and the length is populated during the running phase.

Having forced the geometry column to keep a two-dimensional type linestring (LineString) and having created the required columns of source and target, the pgr_createTopology and the pgr_createVerticesTable queries can be now implemented successfully to the tables of all the floors and transitions and populate the additional columns, as displayed in the following image (Figure 18). The pgr_createVerticesTable constructs the vertices table based on the source and target information("Pgr_createVerticesTable — PgRouting Manual (2.0.0)" n.d.), while the pgr_createTopology builds the network topology based on the geometry information ("Pgr_createTopology — PgRouting Manual (2.4)" n.d.).

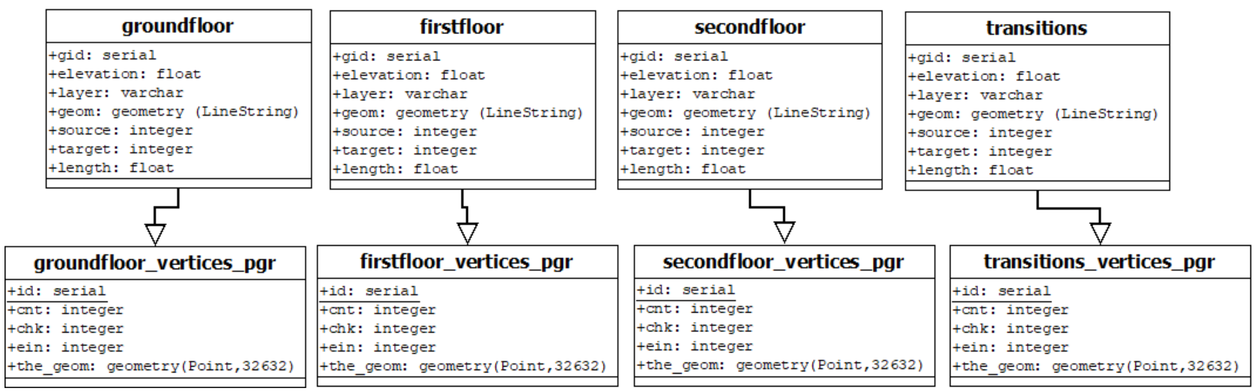


Figure 18 The second phase of the routing database

The next step requires the combination of the line tables to one single table called ways and the point tables to one called ways_vertices_pgr from where the calculation of the routing is executed. As the gid attribute of each of the lines and points tables are the same not allowing the immediate copy to the new table, unique identifiers are summed upon the gid, the source and the target columns in each of the floor tables, while the same number is also summed upon the id column in each of the tables that store the decision points. Then, the new tables are created storing the same attributes and all the original data are copied in them. Some further features such as the name and the instructions are added in order to store useful data for the next steps and for the facilitation of the database manipulation. The final phase of the routing database can be seen in the Figure 19.

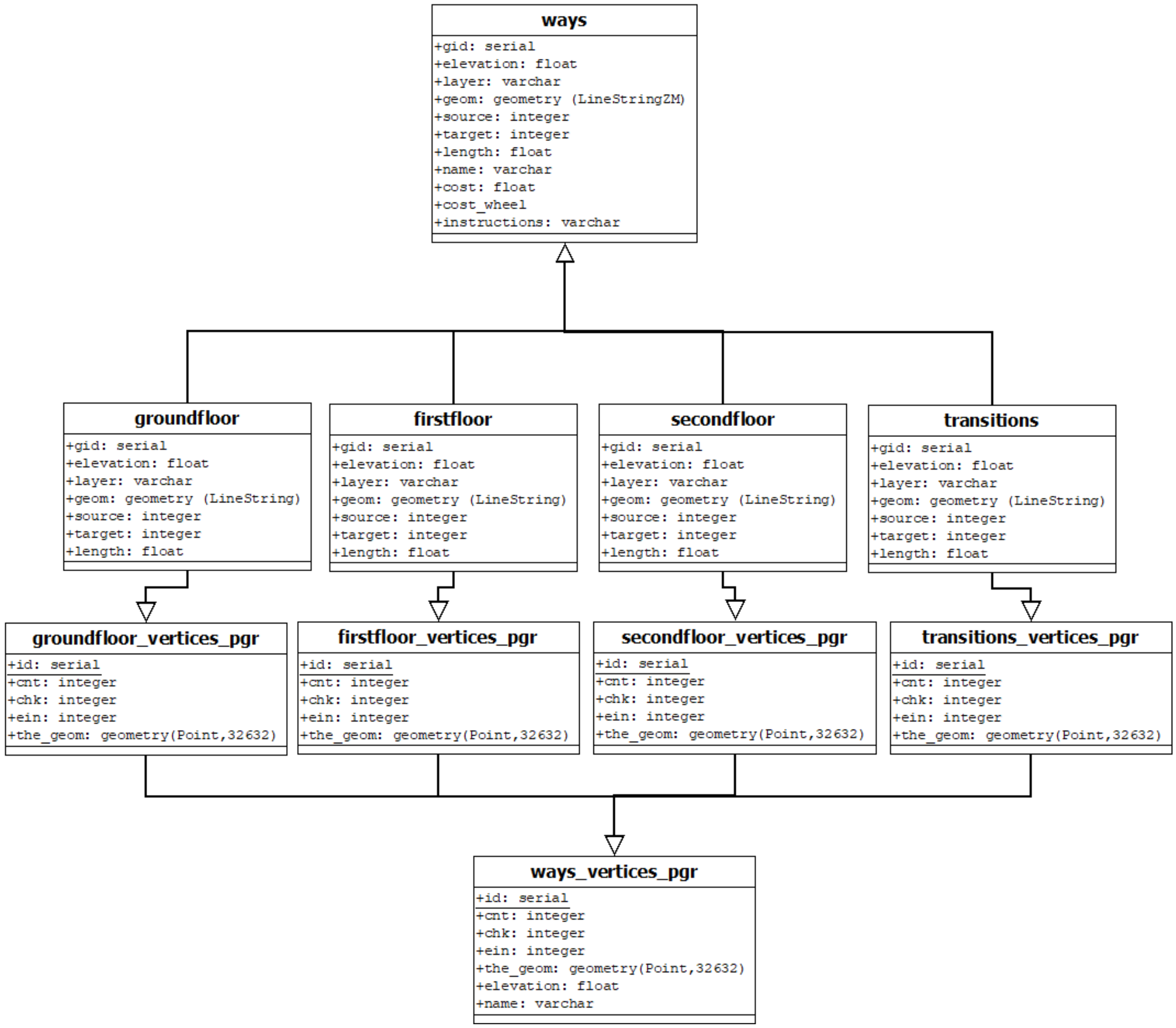


Figure 19 Final database form

Some manual changes should be made after the creation of the final tables for validating the connection among the points of the floors and the transitions as source and target points. This process takes place in ArcScene, while some trial queries are executed for the verification of the results. The trial query for the calculation of the routing path from the Info Point to the Terrace can be seen in Figure 20, the table with the results of the routing is shown in Figure 21, while Figure 22 illustrates the exported results in ArcScene.

```
"SELECT a.*, b.geom FROM pgr_dijkstra(
    'SELECT gid AS id,
        source,
        target,
        cost AS cost
        FROM ways',
    (SELECT id FROM ways_vertices_pgr WHERE name = 'Info Point'),
    (SELECT id FROM ways_vertices_pgr WHERE name = 'Terrace'),
     directed := false)as a
     left join ways as b
     on (edge=gid) order by seq;"
```

Figure 20 Trial query from the Info Point to the Terrace

	seq integer	path_seq integer	node bigint	edge bigint	cost double precision	agg_cost double precision	geom geometry(LineStringZM)
1	1	1	1055	324	4.0466321281	0	01020000C002000000180F1DD5E50F26414BCCB36B2B8C5741D827C3C5020008400000000
2	2	2	1054	325	7.2084122499	4.0466321281	01020000C0020000006C90770AED0F264133AC3BFB2C8C5741D827C3C5020008400000000
3	3	3	1052	322	8.903785764	11.255044378	01020000C0020000005CDE7B9EDD0F2641537C27182E8C5741D827C3C5020008400000000
4	4	4	1051	323	3.6903824995	20.158830142	01020000C0020000004858263AD70F26415D193F8E2E8C5741D827C3C5020008400000000
5	5	5	1053	870	2.5806975802	23.8492126415	01020000C0020000004858263AD70F26415D193F8E2E8C5741D827C3C5020008400000000
6	6	6	1076	346	0.9596175526	26.4299102217	01020000C00200000040B61E97D30F2641B3FC78D12E8C5741EC93E162010012400000000
7	7	7	1064	331	4.6749999998	27.3895277743	01020000C0020000003CCCB3D4C90F26412DC6C7852F8C5741EC93E162010012400000000
8	8	8	1061	330	5.2250000005	32.0645277741	01020000C0020000003CCCB3D4C90F26412DC6C7852F8C5741EC93E162010012400000000
9	9	9	1063	340	2.1999999992	37.2895277746	01020000C002000000BC42E9C7C00F264160F9FA2C308C5741EC93E162010012400000000
10	10	10	1072	342	4.2099905928	39.4895277738	01020000C002000000988F6BF8BC0F2641C55F6173308C5741EC93E162010012400000000
11	11	11	1073	-1	0	43.6995183666	

Figure 21 Results table of the routing calculations

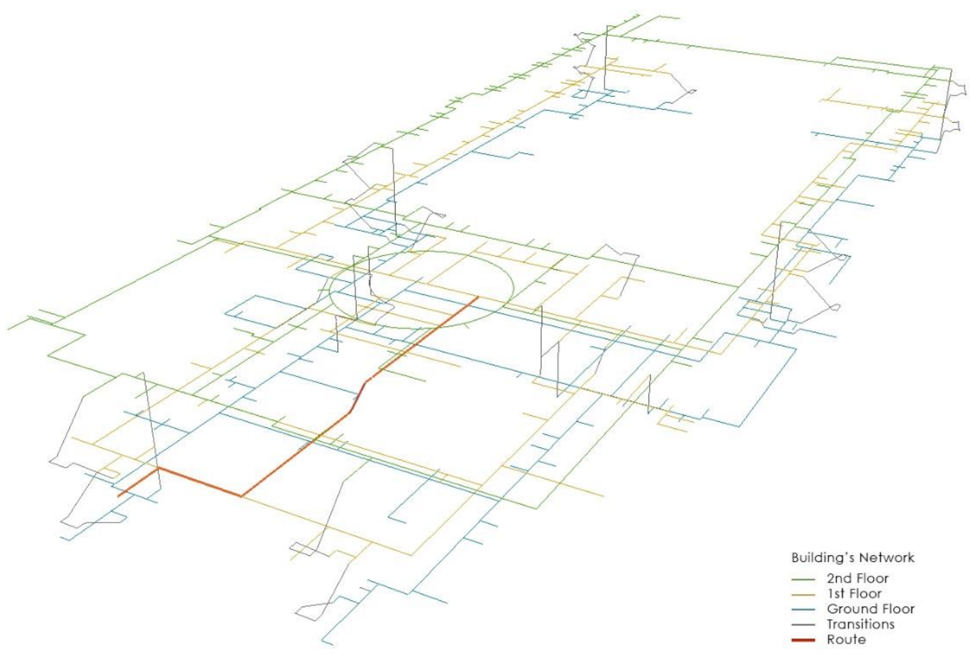


Figure 22 Visualization of the routing results in ArcScene

5.2.Results

The following chapter displays the results of the produced service and describes the provided functionalities in combination with the most important parts of the scripts, while snapshots of the application accompany the description. The whole scripts of the application are included in the Appendix, whereas the present section focuses initially on the presentation of the functionalities of the service, then on the way of calculating the path and finally on the main features of the Cesium library, which allows the three-dimensional virtual navigation. The service consists of two sequential webpages: the home page where the navigation options should get completed by the user, and the main page, where the routing path is displayed along with the animation.

To begin with the first page of the service, the webpage is divided in three divisions. The upper part presents the title of the project, while the main part, which is shown right under that, is divided in two components. The left part includes the map and the 3D building model, and the right part shows the options and the various functionalities of the service. More particularly, on the left part, the 3D model of the whole campus of the Aalborg university in Copenhagen is shown in a light blue color (Figure 23) and the user gets the chance to navigate around it in a 3D mode. It is a 3D volumetric model that derives from Open Data Copenhagen in EPSG:32632 in a CAD format and is converted to CityGML and 3DTiles formats through a process executed in a FME workspace similar to the one described in 5.1.2 Model Development.

Figure 23 The Aalborg University Campus in Copenhagen

In contrast to the above mentioned process, the model in this case is already geolocated and the workspace, which is presented in Figure 24, follows a different sequence of transformations based on conditional filters that distinguish the roof parts from the walls of the buildings. Details on the applied workspace and its functions can be found in “A WebGIS Service for 3D Visualizations” (Georgati 2018).

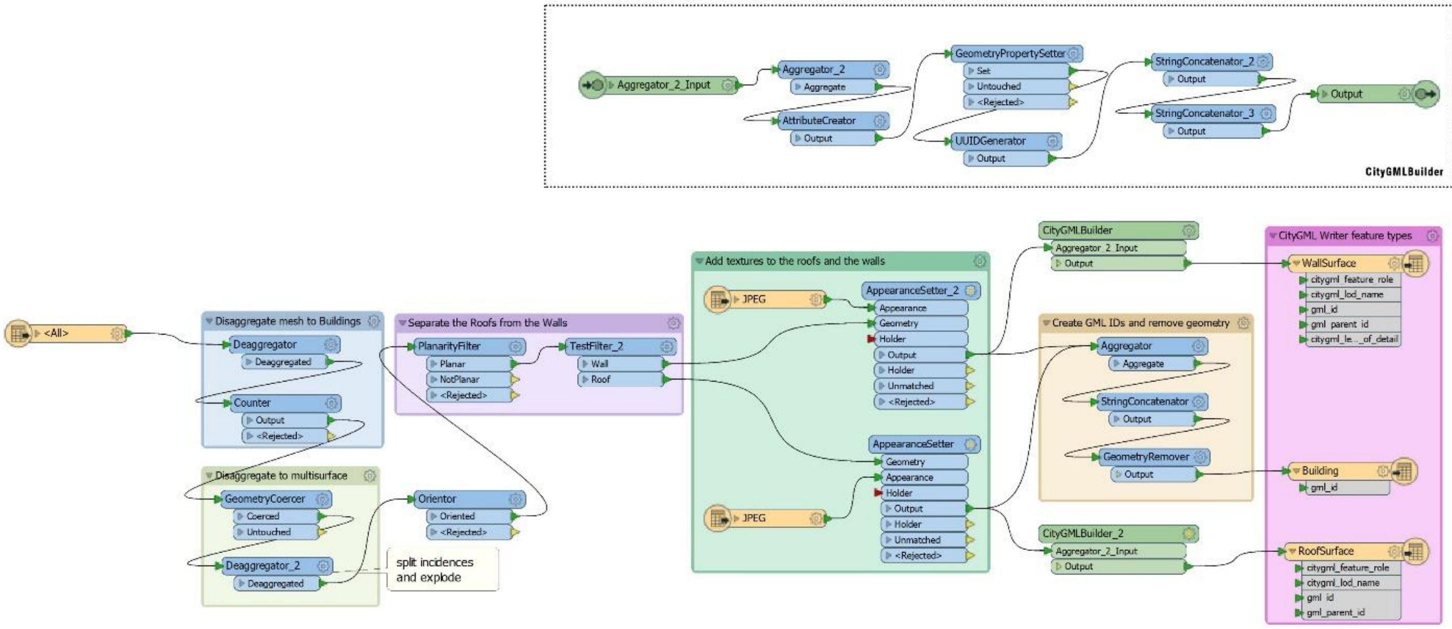


Figure 24 FME Workspace for geolocated 3D models

In the same way, the 3D model of the surrounding buildings is transformed in the required format and loaded in the webpage in gray- white color, as shown in Figure 25. The user controls their visibility by the corresponding buttons on the right column.

Figure 25 The Aalborg University Campus and its surrounding buildings at Sydhavn

Considering the routing process, the user has to select the desired travel mode between stairs and elevator. The selection is activated by the radio buttons in the upper part of the right column. Then, he/she has to insert the start and destination points in the corresponding boxes under the radio buttons. Code 1 controls the dynamic population of a drop-down menu under the text boxes for the easier selection of the desired points by recalling the keywords from the name column of the decision points tables in the routing database, while Code 2 defines the above mentioned functionalities along with the visibility of the surrounding buildings.

```
<?php

  //Connect to the database
  $db_connection = pg_connect("host=localhost port=5432 dbname=routing user=postgres password=1311");
    //Query the database for the name of the desired point
  $query2 = pg_query($db_connection,
      "SELECT name FROM ways_vertices_pgr where searchtext @@ to_tsquery('keyword');");
  //Return the values from the query above
  $rows = pg_fetch_all($query2);
  echo $rows;
?>
```

Code 1 Query for searching the name of the start/ destination point

```
  <div class="search_box">
      <form name="insert" action="calcrouteD.php" method="POST" >
        <input type="radio" id="cost" name="cost" value="cost" checked> <label for="stairs">Stairs</label>
        <input type="radio" id="length" name="cost" value="length"> <label for="elevator">Elevator</label>
        <input size="16%" type="float" name="start" id="place" placeholder="Start"/> <br>
        <input size="16%" type="float" name="destination" id="place"placeholder="Destination"/><br>
        <input type="submit" id="button" value="Confirm" />
      </form>
      <ul class="show_list">
        <li><id="button_show" ><a onclick="showBuildings()"><b>Show Buildings</b></a></li>
        <li><id="button_show" ><a onclick="hideBuildings()"><b>Hide Buildings</b></a></li>
      </ul>
```

Code 2 Radio buttons and text boxes

As the code above displays, the POST method is selected in the particular case for sending information about the start and destination points and the travel mode through the cost or length values of the routing from this form to the next page where the calculation of the routing takes place. These values are used in an SQL query in the next PHP file, where the route path is calculated by using the pgRouting extension and the data is stored in a new table. This process is shown in Code 3. The first part of this script consists of a query for the creation of a table that stores the results of the routing calculations and the second part executes the routing calculations and populates that table. The new table is called route and is later used for the visualization of the routing path.

```
<?php

  //Connect to the database
  $db_connection = pg_connect("host=localhost port=5432 dbname=routing user=postgres
password=1311");
  //Query the database for calculating the route path
  $query = "DROP TABLE IF EXISTS route;
        CREATE TABLE route (seq integer, path_seq integer, node bigint, edge bigint,
cost float, agg_cost float, geom geometry(LineStringZM));";
  $result = pg_query($query);

  $query1="INSERT INTO route SELECT a.*, b.geom FROM pgr_dijkstra(
          'SELECT gid AS id,
             source,
             target,
             $_POST[cost] as cost
            FROM ways00',
          (SELECT min(id) FROM ways_vertices_pgr WHERE '$_POST[start]'=
ways_vertices_pgr.name),
          (SELECT min(id) FROM ways_vertices_pgr WHERE '$_POST[destination]'=
ways_vertices_pgr.name),
          directed := false)as a
          left join ways00 as b
          on (edge=gid) order by seq;";
  $result1 = pg_query($query1);
?>
```

Code 3 Calculation of the routing path

Having executed this step, the routing path is successfully calculated by using the dijksta algorithm and the table ‘route’ stores information about the points that are visited and their sequence, the edges that consist the route and the cost or otherwise the length of the path.

Moving forward to the last part of this section, the way of visualizing this information in a three-dimensional WebGIS is now presented. Cesium JS library is the selected platform for the development of the 3D WebGIS service offering several benefits in regards to the requirements of the system and the provided functionalities. Specifically, Cesium allows the simultaneous visualization of the building, the routing path and the start and ending points. Moreover, it offers various possibilities of animation techniques with great performances and simple ways of development. It is easily and quickly developed, provides comfortable navigation to the user while the interface environment is very friendly.

Figure 26 shows the main page visualizing all these major features on the model and the map. The white column on the right of the page collects similar functionalities as in the previous page, but now controling the visibility of the building floors. The lower part of the column sums up the length of the routing path showing the whole walking distance. It also presents written instrunctions about the clue points that are crossed on the way, highlighting the visiting landmarks.

Figure 26 Main page of visualizing the routing path

Regarding the technical characteristics of the webpages, the creation of a Cesium application requires the HTML's head section to include the Cesium JS library, while an HTML elements called cesiumContainer has to hold the Cesium JS widget, the access token and a variable called viewer has to be created in the beginning ("Getting Started | Cesium.Com" n.d.). This is the basic widget for building the application and includes all the standard Cesium widgets, such as the terrain, the layer switcher or the full screen button. In this case, the OpenLayers map is used as the base map, the scene is set only to 3D saving GPU memory, as the geometry instances are rendered only in 3D ("Viewer - Cesium Documentation" n.d.), and a few buttons are deactivated for the simplification of the interface, as shown in Code 4. The lighting effects are also defined in this part of code.

```
//Define the viewer
var viewer = new Cesium.Viewer('cesiumContainer',{
  terrainProvider : Cesium.createWorldTerrain(),
  terrainExaggeration : 0.0,
  //Hide the base layer picker
  baseLayerPicker :false,
  fullscreenButton: false,
  navigationHelpButton:false,
  scene3DOnly:true,
  homeButton:false,
  geocoder:false
  });

viewer.scene.globe.depthTestAgainstTerrain = true;
viewer.scene.globe.enableLighting = true; // Lighting effects

//Define the scene
var scene = viewer.scene;
```

Code 4 Creation of the Cesium viewer widget

Tilesets are loaded in both webpages for visualizing the campus and the surrounding buildings, the slabs of the examined building and the volumes of its closed spaces in different colors and transparency levels. Code 5 calls the 3D Tiles tileset of the campus buildings and of its surroundings. It streams massive heterogeneous 3D geospatial datasets (“Introducing 3D Tiles | Cesium.Com” n.d.), while Code 6 and Code 7 indicate the styling control of the tileset and the application of the style on it relatively.

```
//Load the tilesets
var tileset = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_AAU/tileset.json',
  debugShowBoundingVolume : false
}));

var tileset02 = scene.primitives.add(new Cesium.Cesium3DTileset({
    url: 'AAUtiles/tileset_buildings/tileset.json',
    debugShowBoundingVolume : false,
    show : true
  }));
```

Code 5 Loading 3D geospatial datasets as tilesets and defining their styles

```
var transparentStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.9)",
  show : true
});

var transparentBoxStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.5)",
  show : true
});
```

Code 6 Creation of custom style

```
tileset02.style = transparentStyle;

```

Code 7 Application of the style on the tileset

Furthermore, the main page has some additional features worthy to be discussed, such as the polyline displaying the navigation path or the animation possibilities. The most important components of the appropriate visualization of a routing path are the start and destination points and the line that connects them following the shortest path based on the corresponding options. For this purpose, three different functions are created in Cesium JS that recall data from the database through other PHP files by using the fetch API; the first two create the start and the destination points (Code 8) and the third creates the polyline that represents the path and the animated motion of a selected model (Code 9).

Two blue pins represent the chosen points and a blue line connects them following one by one the points that are stored in the table ‘route’ of the database, as shown in Figure 27. The line follows the path either through the stairs or the elevator based on the desired selection. Instances of the variations among the travel modes are shown in Figure 28 and Figure 29, which represent the connection between the same two points through the elevator and the staircase relatively. The visibility conditions of the floors can be easily changed by the button on the right part of the screen.

```
function createstartPin() {
  fetch("callpointgeom.php",{
      method:'get',
      credintentials: 'include',
    })
    .then(function(reply) {return reply.json()})
    .then(function(info) {

    var pinBuilder = new Cesium.PinBuilder();
    for (elm in info) {
      record = info[elm];

      var lat_start = parseFloat(record['lat_start']);
      var lon_start = parseFloat(record['lon_start']);
      var z_start = parseFloat(record['z_start']);
      var hei_start=z_start+1.0
      var startPin = viewer.entities.add({
        name : 'Start pin',
        position : Cesium.Cartesian3.fromDegrees(lat_start,lon_start,hei_start),
        billboard : {
          image : pinBuilder.fromColor(Cesium.Color.ROYALBLUE, 28).toDataURL(),
          verticalOrigin : Cesium.VerticalOrigin.BOTTOM
        },
      });
    }
    });
};
```

Code 8 Function for the visualization of the start point

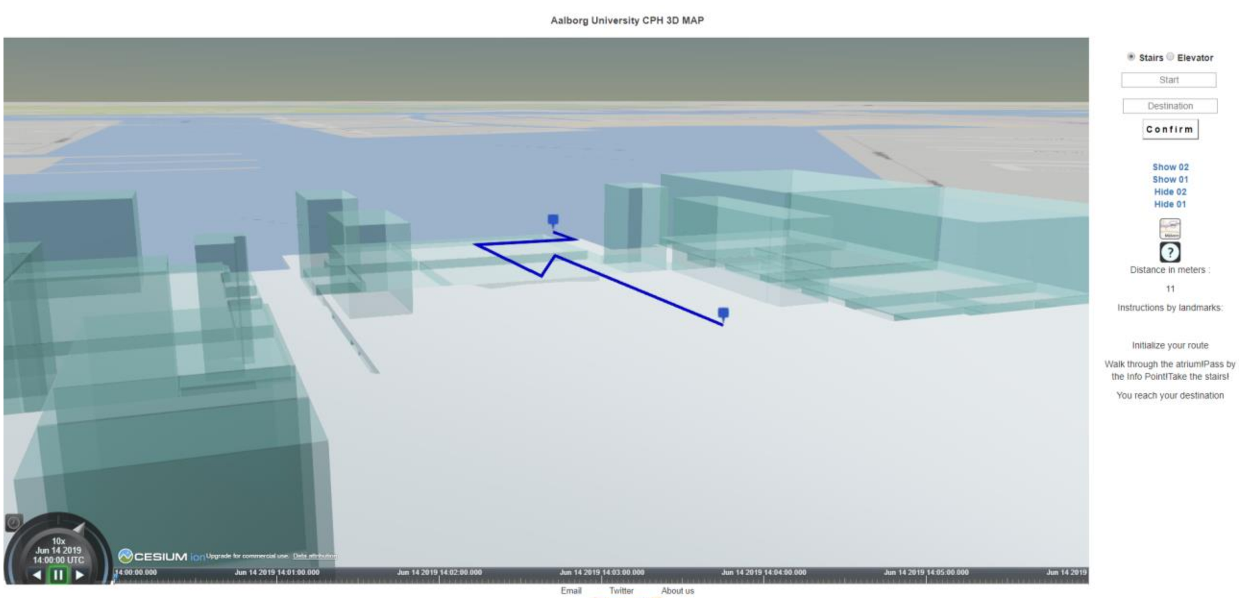


Figure 27 View of the routing path and the start and destination points

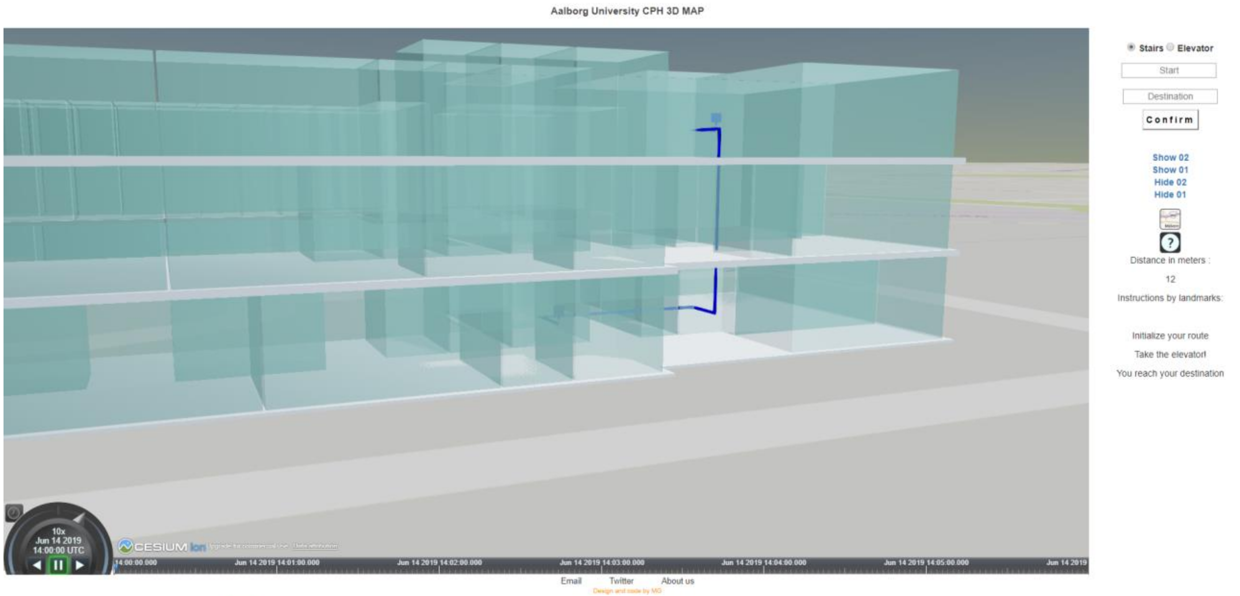


Figure 28 View of a routing path by using the elevator

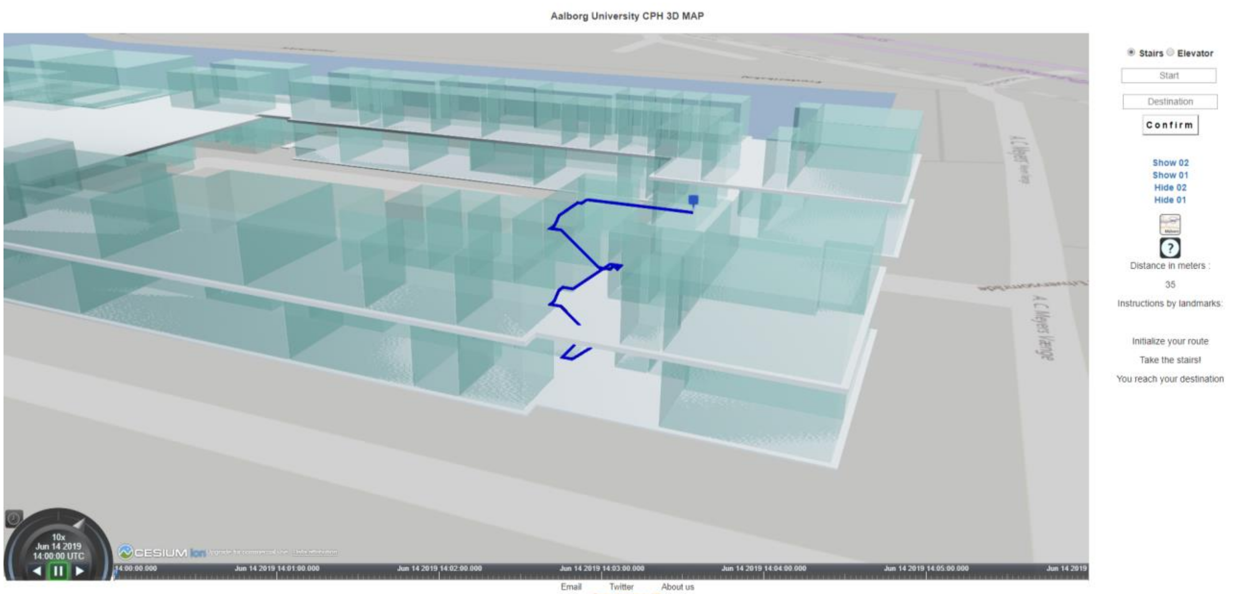


Figure 29 View of a routing path by using the stairs

```
function createLine() {
  uio = [];
  var u=[];
  fetch("callroutegeom.php",{
    method:'get',
    credintentials: 'include',
  })
  .then(function(response) {return response.json()})
  .then(function(data) {

    for (elm in data) {
      record = data[elm];
      var lat = parseFloat(record['lat']);
      var lon = parseFloat(record['lon']);
      var z = parseFloat(record['z']);
      var hei=z+2;
      u.push(lat,lon,hei);
      uio.push({lat,lon,hei});
      //console.log(uio[1]);
      var line=viewer.entities.add({
        name : 'Route Line',
        polyline : {
          positions : Cesium.Cartesian3.fromDegreesArrayHeights(u),
          width : 4,
          material : new Cesium.PolylineOutlineMaterialProperty({
            color : Cesium.Color.BLUE,
            outlineWidth : 0.5,
            outlineColor : Cesium.Color.BLACK
          })
        }
      });
      viewer.zoomTo(viewer.entities);
    }
  })
  .then(()=> {
          animation(uio);
          function animation(uis){
            let seConds = 0;
            var position = new Cesium.SampledPositionProperty();
            console.log(uis.length);
              for(i = 0; i< uis.length;i++){
                seConds = seConds + 10;
                position.addSample(Cesium.JulianDate.addSeconds(start, seConds, new Cesium.JulianDate()), Cesium.Cartesian3.fromDegrees(uis[i].lat, uis[i].lon, uis[i].hei));
              }
              var entity = viewer.entities.add({
                  availability : new Cesium.TimeIntervalCollection([new Cesium.TimeInterval({
                      start : start,
                      stop : stop
                  })]),
                  position : position,
                  orientation : new Cesium.VelocityOrientationProperty(position),
                    model : {
                      uri : 'AAUtiles/man/NewFeatureType.gltf',
                      minimumPixelSize : 64,
                      scale:0.6
                    },
                    path : {
                      resolution : 1,
                      material : new Cesium.PolylineGlowMaterialProperty({
                        glowPower : 0.1,
                        color : Cesium.Color.YELLOW
                      }),
                        width : 10
                    }
              });
```

Code 9 Function for the visualization of the route path and the animation

Apart from these features, another interesting option that is also available in this service is the animated motion of a model, as can be seen in Figure 30. The model 'walks' along the path activated by the play button on the timeline widget at the lowest part of the map. The creation of the animated motion requires setting up the Cesium clock for keeping track of simulated time. Code 10 defines the start and stop time of clock while a multiplier is also added for determining how much time advances ("Clock - Cesium Documentation" n.d.).

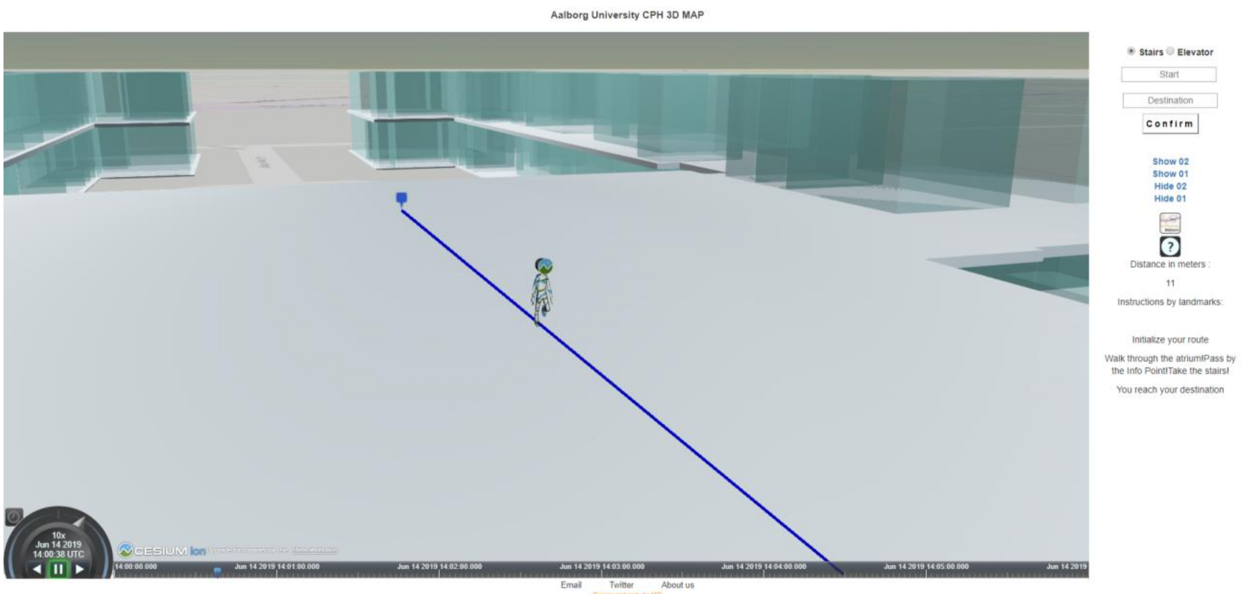


Figure 30 Visualization of the animated model on the routing path

```
var start = Cesium.JulianDate.fromDate(new Date(2019, 5, 14, 16));
var stop = Cesium.JulianDate.addSeconds(start, 360, new Cesium.JulianDate());

viewer.clock.startTime = start.clone();
viewer.clock.stopTime = stop.clone();
viewer.clock.currentTime = start.clone();
viewer.clock.clockRange = Cesium.ClockRange.LOOP_STOP;
viewer.clock.multiplier = 10;
```

Code 10 Setting up the clock

Eventually, Code 11 is considered of great importance for the proper function of the application, as It consists of a promise about adding the tileset, zooming to it and setting a bounding box to it that is resolved when the tileset's root tile is loaded and the tileset is ready to render. This part of code also defines the zooming level and angle of the camera to the model.

```
  //Zoom to the tileset
  tileset.readyPromise.then(function(tileset) {
    viewer.scene.primitives.add(tileset);
    viewer.zoomTo(tileset, new Cesium.HeadingPitchRange(-1.5, -0.35,
 tileset.boundingSphere.radius * 1.0));
  }).otherwise(function(error) {
    console.log(error);
  });
```

Code 11 Promise function for zooming to the tileset when it is ready to render

6. Discussion

This chapter provides insight thoughts about the study and evaluates potential strengths and weaknesses on the followed implementation methods. The present study is motivated by the widespread need of an applicable indoor navigation system in large and complex buildings with great concentrations of people where fast and effective navigation is mostly desired for saving money and time and for avoiding stressful situations.

The research is focused firstly in the deep understanding of how an indoor navigation system works, what its major components are and how it may be developed in a timeframe of 3 months, with open source tools and without any installation requirements. Based on the literature review, the most suitable representation method for efficient route planning is shown to be by using three-dimensional tools and 3D information models, while the investigation of methods for 3D model development is one of the most important issues examined in this thesis.

Then, the study attempts to address the posed challenges by structuring the conceptual design of the research and by imposing it in a selected case scenario of a familiar building at the Aalborg University Campus in Copenhagen, aiming to the development of a desktop- based WebGIS that allows the three-dimensional navigation around the examined building and the calculation of the most suitable route between two selected points. Starting by the configuration of the conceptual design, various ways for enhancing the performance of both the service and its interaction with the user are examined defining several requirements that the service should cover for accomplishing its purposes in the most profitable manner.

Specifically, the service offers firstly the possibility to the user to choose two points between which he/ she would like to move to and from, along with his/ her 3D virtual navigation around the building. After having selected the desired starting and destination points, the service displays mainly the interior places of the building and the calculated routing path, while it provides the chance to the user to watch the animated movement of a human model along the path. Moreover, the service provides written instructions on the side of the map highlighting the most significant places that the user has to pass by in order to reach his/ her destination.

Summing up, the evaluation of the service may be divided into its elements and should be conducted separately. The elements of the navigation system that can be assessed include the technical robustness of the service itself, the preparation of the exposed material and the effectiveness of the final product. More particularly, the service is easily developed in a short period of time offering technical robustness without installation requirements. It applies existing technologies that have continually been used in various cases with successful results and without cost.

Then, the preparation of the 3D building model and the network is considered to be adequate enough for the specific case study, even though it presents inaccuracies. The digitalization of editable drawings from pdf files is never accurate due to changes in scalability and width of printed lines. For this reason, both the model and the network may present relevant erroneousness up to around 30cm. Regarding the extrusion of the 3D model, the selection of the 4.5m height of each floor is a safe hypothesis made by the author but does not correspond to real measurements, increasing the erroneousness rate a little bit more. However, due to the fact that the specific service does not interact with localization systems and is mainly addressed to precalculated routings that occur before the initiation of the actual navigation, this degree of inaccuracy is not regarded as a great disadvantage.

On the other hand, the environment of the system is friendly, easy to understand and to be used by the user, while the final visual product is generally balanced and comprehensive without irrelevant information. The exported messages are easily understandable, meaningful and memorable for the user, while the three-dimensional navigation in indoor environments usually attracts the attention of people gaining great popularity lately.

Considering further improvement of the usability of the service, the enhancement of the accuracy of the 3D building model could be of great importance especially when taking into account that construction usually differs from the blueprints. To overcome the challenges of differentiations among planned and constructed elements or inaccuracies due to outdated or non-editable blueprints, the 3D scanning of the building could be of great help both for surveying the whole actual building but also for contributing to its semantic identification with greater details. The three-dimensional survey of the building could also extent its level-of-detail and its functionalities by even including moveable components and furniture and enhancing navigation in the interior of the building.

Based on a more accurate 3D building model, a more complete network could also be accomplished both in two- and three- dimensions, attempting routing around the moveable objects, while enrichment of the provided information about the accessibility of spaces could notify users about the accessibility status of some areas or their opening hours. The network could also be extended to outdoors and be connected to the city’s transportation network.

Additional improvements could also be achieved by providing more animation possibilities. For instance, the presentation of the actual path in a 3D mode where the user may change the speed level and navigate on the path among the interior spaces would offer great benefits for the better understanding of the surrounding spaces, while it would also raise the spatial perception of the user. Such a possibility could also be extended to include virtual reality features and navigation through special VR headsets. Another instance could include the presentation of photographs taken from various points of interest displaying significant spots of the building in a row and guiding the user accordingly.

7. Conclusion

The study investigates methods for developing more flexible and reliable routing systems that would provide a more coherent description of the examined place and would release the users from the insecurity of navigating in complex and crowdy indoor spaces. The research concludes that 3D digital navigation offers great advantages on the deeper understanding of the explored environment, while it helps people improve their cognitive spatial maps when combined with the promotion of animated stimuli and landmarks.

It also attempts to answer questions about the availability of existing technologies for the generation of an indoor navigation system and their range of implementation, while indoor 3D modelling concentrates a great part of the thesis's attention, concluding that the level of detail required in a navigation service is defined by the researcher's purposes. The specific model does not include great amounts of details as the thesis's purposes aim to comprehensive and memorable results that should not be confused with additional, irrelevant issues, such as the functionality differentiations among the rooms or moveable objects.

Finally, the present study proves that the implementation of an online indoor navigation system can be easily achieved in an either new or old building by using its footprints or its 3D model created in a CAD format by the responsible architectural and design team. Storing data about the network and the decision points in the database also provides various benefits regarding changes on the selected client-server communication model.

7.1.Future Development

Summing up, the final routing service offers both 3D indoor and outdoor navigation around the selected part of the building A at the Aalborg University in Copenhagen and visualizes the volumetric perception of its outdoor and indoor spaces. These 3D online visualizations provide context information and a good overview of the visited spaces. The applied framework provides proper routing for two types of locomotion: by walking or by using a wheelchair, and calculates the most suitable way between two points in the building, while written instructions are also shown listing the necessary moves to be followed after accessing the building.

However, there are still several aspects of the application that could be improved in the future. Initially, the service should also be applied in a mobile device or it should provide the chance to export and send the results to a mobile device. The semantic information of the 3D model could be enriched and visualized in more visually attractive ways. Further context information could also be included providing insight on conditionally accessible spaces, locked up spaces or opening hours of specific rooms.

In addition to that, the completion of the whole building model and its network could be accomplished, while the precision of the model could also be improved. 3D laser scanning survey of the examined case could further improve the accuracy of the 3D model, while the automation of the processes for the creation of the 3D model and its network could be considered one of the biggest challenges.

Further visual digital stimuluses could also be added for the enhancement of their imageability in the real space, while an investigation on the effectiveness of the selected landmarks and their way of representation should be conducted for the evaluation of the selection.

Last but not least, testing the service in real life and conducting field experiments in the university campus is a key- element for its proper evaluation that should be taken closely into consideration before further development. Students, staff or visitors should participate into this testing, while their observation and interviewing after the experimentation may reveal significant results regarding the effectiveness and the technical performance of the service.

7.2.Other use cases of the produced service

The final product of this thesis includes a desktop-based navigation application of an indoor space that visualizes both the interior environment of the building and the routing path in 3D mode and allows the user to move around it without any installation limitations or cost. Such an application may find ground of further development and implementation in various fields and markets depending on the level of detail and the spatial representation of the model. For example, a corresponding application may be used not only for university buildings or teaching rooms, but also for airports, shopping malls, hospitals and community buildings.

The physical search of gates in big scale and complex airports may be easily facilitated by the use of 3D navigation services, while the needless walking around shopping malls looking for the desired shop could be eliminated, and the gained time could be beneficially offered to other purposes. Furthermore, nowadays that self-service and DIY commerce gain more and more popularity, customers could address to the storages and get straight forward access to goods by using 3D navigation tools. IKEA, for instance, could provide this service in its buildings along with the barcode of the chosen product leading the user the way in the correct order of shelfs for collecting the selected items. A similar service could also be used in community buildings where people often address by taking time off their work and tour around from office to office.

As shown in relate past works, the interest of a 3D navigation system has already been focused on museums or university campuses, but raising the level of detail up to moveable light objects, the application could also work in smaller scale searches or buildings, such as the search of books in libraries or shops, and allow e-commerce as well. Moreover, 3D routing services in that scale could also allow addressed renovations and repairs to special building parts or engineering places, such as the vessels or factories engines.

8. Bibliography

"3D Analyst and ArcScene—Help | ArcGIS Desktop." n.d. Accessed April 24, 2019. http://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/3d-analyst-and-arcscene.htm.

"3ds Max | 3D Modeling, Animation & Rendering Software | Autodesk." n.d. Accessed November 19, 2018. https://www.autodesk.eu/products/3ds-max/overview.

Becker, Thomas, Claus Nagel, and Thomas H. Kolbe. 2009. "A Multilayered Space-Event Model for Navigation in Indoor Spaces." In *3D Geo-Information Sciences Lecture Notes in Geoinformation and Geography*, edited by Jiyeong Lee and Sisi Zlatanova, 61–78. Springer. https://doi.org/10.1007/978-3-540-88183-4.

Bederson, Benjamin B., and Angela Boltman. 1999. "Does Animation Help Users Build Mental Maps of Spatial Information?" Edited by IEEE. *The Craft of Information Visualization*. https://doi.org/10.1016/b978-155860915-0/50015-9.

Brandt, Thomas, Andreas Zwergal, and Stefan Glasauer. 2016. "3-D Spatial Memory and Navigation." *Current Opinion in Neurology* 30 (1): 90–97. https://doi.org/10.1097/wco.0000000000000415.

Brown, Gavin, Claus Nagel, Sisi Zlatanova, and Thomas H. Kolbe. 2013. "Modelling 3D Topographic Space against Indoor Navigation Requirements." *Lecture Notes in Geoinformation and Cartography*, 1–22. https://doi.org/10.1007/978-3-642-29793-9-1.

"Campus Map." n.d. Accessed April 27, 2019. https://www.en.cph.aau.dk/campusintra-students/practical-information/campus-map/.

Carr, Mahil, and June Verner. 1997. "Prototyping and Software Development Approaches." *Prototyping and Software Development Approaches* 2004 (3): 1–16.

"CesiumJS - Geospatial 3D Mapping and Virtual Globe Platform." n.d. Accessed November 23, 2018. https://cesiumjs.org/.

"CityGML | OGC." n.d. Accessed April 25, 2019. https://www.opengeospatial.org/standards/citygml.

"Clock - Cesium Documentation." n.d. Accessed May 14, 2019. https://cesiumjs.org/Cesium/Build/Documentation/Clock.html.

Cousins, Jennifer H., Alexander W. Siegel, and Scott E. Maxwell. 1983. "Way Finding and Cognitive Mapping in Large-Scale Environments: A Test of a Developmental Model." *Journal of Experimental Child Psychology* 35 (1): 1–20. https://doi.org/10.1016/0022-0965(83)90066-8.

Diakité, Abdoulaye A., and Sisi Zlatanova. 2018a. "Spatial Subdivision of Complex Indoor Environments for 3D Indoor Navigation." *International Journal of Geographical Information Science* 32 (2): 213–35. https://doi.org/10.1080/13658816.2017.1376066.

Diakité, Abdoulaye A, and Sisi Zlatanova. 2018b. "Spatial Subdivision of Complex Indoor Environments for 3D Indoor Navigation." *International Journal of Geographical Information Science* 32 (2): 213–35. https://doi.org/10.1080/13658816.2017.1376066.

Fallah, Navid, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. 2013. "Indoor Human Navigation Systems: A Survey." *Interacting with Computers* 25 (1): 21–33. https://doi.org/10.1093/iwc/iws010.

"Free Software for Students & Educators | AutoCAD Map 3D | Autodesk." n.d. Accessed April 24, 2019. https://www.autodesk.com/education/free-software/autocad-map-3d.

Georgati, Marina. 2018. "A WebGIS Service for 3D Visualizations."

"Getting Started | Cesium.Com." n.d. Accessed December 3, 2018. https://cesium.com/docs/tutorials/getting-started/.

Goodchild, Michael F. 2010. "Twenty Years of Progress : GIScience in 2010" 1 (1): 3–20. https://doi.org/10.5311/JOSIS.2010.1.2.

Gröger, Gerhard, Thomas Kolbe, Claus Nagel, and Karl-Heinz Häfele. 2012. "OGC City Geography Markup Language (CityGML) Encoding Standard." *Open Geospatial Consortium*, 1–344. https://doi.org/OGC 12-019.

Haist, Jorg, Thorsten Reitz, and Volker Coors. 2006. "Innovations in 3D Geo Information Systems." In . https://doi.org/10.1007/978-3-540-36998-1.

"HTML 5.2: 1. Introduction." n.d. Accessed November 18, 2018. https://www.w3.org/TR/html/introduction.html#introduction.

"Indoor Location - The Open-Source Framework for Indoor Mapping." n.d. Accessed February 17, 2019. https://www.indoorlocation.io/.

"Indoor Routing for Xamarin | ArcGIS for Developers." n.d. Accessed February 17, 2019. https://developers.arcgis.com/example-apps/indoor-routing-xamarin/.

"Introducing 3D Tiles | Cesium.Com." n.d. Accessed November 23, 2018. https://cesium.com/blog/2015/08/10/introducing-3d-tiles/.

"Introduction to JavaScript." n.d. Accessed November 18, 2018. https://www.thoughtco.com/what-is-javascript-2037921.

Isikdag, Umit, Sisi Zlatanova, and Jason Underwood. 2013. "A BIM-Oriented Model for Supporting Indoor Navigation Requirements." *Computers, Environment and Urban Systems* 41: 112–23. https://doi.org/10.1016/j.compenvurbsys.2013.05.001.

Jamali, Ali, Alias Abdul, Rahman Pawel, Pankaj Kumar, and Christopher M Gold. 2017. "An Automated 3D Modeling of Topological Indoor Navigation Network." *GeoJournal* 82 (1): 157–70. https://doi.org/10.1007/s10708-015-9675-x.

Kárník, Jiří, and Jakub Streit. 2016. "Summary of Available Indoor Location Techniques." *IFAC-PapersOnLine* 49 (25): 311–17. https://doi.org/10.1016/j.ifacol.2016.12.055.

Kothari, C. R. 2004. *Research Methodology Methods and Techniques*. New Age International (P) Ltd., Publishers.

Kray, Christian, Christian Elting, Katri Laakso, and Volker Coors. 2005. "Presenting Route Instructions on Mobile Devices." In *Exploring Geovisualization.*, Elseviers, 529–50. Amsterdam.

Li, Ki-joune, and Jiyeong Lee. 2010. "Indoor Spatial Awareness 2010-Indoor-Standard.Pdf."

Liu, Hui, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. "Survey of Wireless Indoor Positioning Techniques and Systems." *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. https://doi.org/10.1109/TSMCC.2007.905750.

Liu, Liu, and Sisi Zlatanova. 2011. “A ‘Door-To-Door’ Path-Finding Approach For Indoor Navigation.” *Proceedings Gi4DM 2011: GeoInformation for Disaster Management, Antalya, Turkey, 3-8 May 2011*, 3–8. http://repository.tudelft.nl/assets/uuid:de84a1c2-6596-4888-a0dd-898886e1f48f/270048.pdf.

Luo, F., G. Cao, and X. Li. 2014. “An Interactive Approach for Deriving Geometric Network Models in 3D Indoor Environments.” *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems* 2014-Janua: 9–16. https://doi.org/10.1145/2676528.2676531.

Lynch, Kevin. 1960. *The Image of the City*. *The M.I.T. Press*. Vol. 21. https://doi.org/10.2307/427643.

Madl, Tamas, Ke Chen, Daniela Montaldi, and Robert Trappi. 2015. “Computational Cognitive Models of Spatial Memory in Navigation Space: A Review.” *Neural Networks* 65: 18–43. https://doi.org/.1037//0033-2909.I26.1.78.

“Osm4routing.” n.d. https://github.com/Tristramg/osm4routing.

“Pgr_createTopology — PgRouting Manual (2.4).” n.d. Accessed May 3, 2019. https://docs.pgrouting.org/2.4/en/pgr_createTopology.html#pgr-create-topology.

“Pgr_createVerticesTable — PgRouting Manual (2.0.0).” n.d. Accessed May 3, 2019. https://docs.pgrouting.org/2.0/en/src/common/doc/functions/create_vert_table.html.

“PgRouting Project — Open Source Routing Library.” n.d. Accessed May 15, 2019a. https://pgrouting.org/index.html.

———. “PgRouting Project — Open Source Routing Library.” n.d. Accessed April 24, 2019b. https://pgrouting.org/.

Raubal, M., and M. J. Egenhofer. 1998. “Comparing the Complexity of Wayfinding Tasks in Built Environments.” *Environment and Planning B: Planning and Design* 25 (6): 895–913. https://doi.org/10.1068/b250895.

Reddy, Harika. 2013. “PATH FINDING - Dijkstra ’ s and A * Algorithm ’ s,” 1–15.

“Requirements Analysis - Requirements Analysis Process,Techniques.” n.d. Accessed May 27, 2019. https://reqtest.com/requirements-blog/requirements-analysis/.

Robertson, G G, J D Mackinlay, and S K Card. 1991. “Cone Trees: Animated 3D Visualizations of Hierarchical Information.” *1991 SIGCHI Conference on Human Factors in Computing Systems, CHI 1991*, 189–94. https://doi.org/10.1145/108844.108883.

Rouse, Margaret. 2017. “What Is RFID (Radio Frequency Identification)? - Definition from WhatIs.Com.” Tech Target Internet of Things Agenda. 2017. https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification.

“Routing - OpenStreetMap Wiki.” n.d. Accessed February 17, 2019. https://wiki.openstreetmap.org/wiki/Routing.

Rüppel, Uwe, Kai Marcus Stübbe, and Uwe Zwinger. 2010. “Indoor Navigation Integration Platform for Firefighting Purposes.” *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, no. September: 15–17. https://doi.org/10.1109/IPIN.2010.5647401.

"Safe Software | FME | Data Integration Platform." n.d. Accessed November 23, 2018. https://www.safe.com/.

Schroder, Catherine J., William A. Mackaness, and Bruce M. Gittings. 2011. "Giving the 'Right' Route Directions: The Requirements for Pedestrian Navigation Systems." *Transactions in GIS* 15 (3): 419–38. https://doi.org/10.1111/j.1467-9671.2011.01266.x.

Sharkawi, K H, M.U. Ujang, and A. Abdul-Rahman. 2008. "3D Navigation System for Virtual Reality Based on 3D Game Engine." In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 513–18. http://www.isprs.org/proceedings/XXXVII/congress/4_pdf/91.pdf.

Smith, Michael J de, Michael F. Goodchild, and Paul A. Longley. 2018. *Geospatial Analysis 6th Edition, 2018 - de Smith, Goodchild, Longley and Colleagues*. www.spatialanalysisonline.com.

"Solibri | About BIM and IFC." n.d. Accessed April 26, 2019. https://www.solibri.com/bim-ifc.

Tashakkori, Hosna, Abbas Rajabifard, and Mohsen Kalantari. 2015. "A New 3D Indoor / Outdoor Spatial Model for Indoor Emergency Response Facilitation." *Building and Environment* 89: 170–82. https://doi.org/10.1016/j.buildenv.2015.02.036.

Tsiliakou, E., and E. Dimopoulou. 2016. "3D Network Analysis for Indoor Space Applications." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 42 (2W2): 147–54. https://doi.org/10.5194/isprs-archives-XLII-2-W2-147-2016.

Tutorial, Bpmn, and Daniel Brookshier. 2015. "CSS Tutorial," 183. https://www.w3schools.com/css/.

"Viewer - Cesium Documentation." n.d. Accessed December 3, 2018. https://cesiumjs.org/Cesium/Build/Documentation/Viewer.html.

Weisman, Jerry. 1981. "Evaluating Architectural Legibility: Way-Finding in the Built Environment." *Environment and Behavior*.

"What Is AutoCAD?" n.d. Accessed November 19, 2018. https://study.com/what_is_auto_cad.html.

"What Is BIM | Building Information Modeling | Autodesk." n.d. Accessed April 25, 2019. https://www.autodesk.com/solutions/bim.

"What Is HTML? | HyperText Markup Language Explained." n.d. Accessed November 18, 2018. https://www.yourhtmlsource.com/starthere/whatishtml.html.

Worboys, Michael, and Lisa Walton. 2008. "Towards a Model of Indoor Space." *ISA Project Technical Report*. http://www.spatial.maine.edu/ISAmodel/documents/ISAreport2008.pdf.

Xu, W, M Kruminaite, B Onrust, H Liu, Q Xiong, S Zlatanova, and Remote Sensing. 2013. "A 3D MODEL BASED INDOOR NAVIGATION SYSTEM FOR HUBEI PROVINCIAL MUSEUM" XL (December): 11–13. https://doi.org/10.5194/isprsarchives-XL-4-W4-51-2013.

Zheng, Pei, and Lionel Ni. 2006. *Smart Phone and Next Generation Mobile Computing*. Morgan Kaufmann Publishers.

9. Appendix

9.1 Database manipulation

Create routing database and load required extensions	create extension postgis; create extension pgrouting; create extension postgis_sfcgal;
After loading data in the database, force Linestrings to 2D	ALTER TABLE groundfloor ALTER COLUMN geom TYPE geometry(LineString) USING ST_Force2D(geom); ALTER TABLE firstfloor ALTER COLUMN geom TYPE geometry(LineString) USING ST_Force2D(geom); ALTER TABLE secondfloor ALTER COLUMN geom TYPE geometry(LineString) USING ST_Force2D(geom); ALTER TABLE transitions01 ALTER COLUMN geom TYPE geometry(LineString) USING ST_Force2D(geom); ALTER TABLE transitions12 ALTER COLUMN geom TYPE geometry(LineString) USING ST_Force2D(geom);
Add further columns for source, target and length, Populate length	ALTER TABLE groundfloor add column source integer; ALTER TABLE groundfloor add column target integer; ALTER TABLE groundfloor ADD COLUMN length FLOAT8; UPDATE groundfloor SET length = ST_Length(geom); ALTER TABLE firstfloor add column source integer; ALTER TABLE firstfloor add column target integer; ALTER TABLE firstfloor ADD COLUMN length FLOAT8; UPDATE firstfloor SET length = ST_Length(geom); ALTER TABLE secondfloor add column source integer; ALTER TABLE secondfloor add column target integer; ALTER TABLE secondfloor ADD COLUMN length FLOAT8; UPDATE secondfloor SET length = ST_Length(geom); ALTER TABLE transitions01 add column source integer; ALTER TABLE transitions01 add column target integer; ALTER TABLE transitions01 ADD COLUMN length FLOAT8; UPDATE transitions01 SET length = ST_3DLength(geom); ALTER TABLE transitions12 add column source integer; ALTER TABLE transitions12 add column target integer; ALTER TABLE transitions12 ADD COLUMN length FLOAT8; UPDATE transitions12 SET length = ST_3DLength(geom);
Create table for decision points and populate it	SELECT pgr_createVerticesTable('groundfloor','geom','source','target'); SELECT pgr_createVerticesTable('firstfloor','geom','source','target'); SELECT pgr_createVerticesTable('secondfloor','geom','source','target'); SELECT pgr_createVerticesTable('transitions01','geom','source','target'); SELECT pgr_createVerticesTable('transitions12','geom','source','target');
Create the topology, populate the source and target columns accordingly to the decision points tables	SELECT pgr_createTopology('groundfloor', 0.0001, 'geom', 'gid'); SELECT pgr_createTopology('firstfloor', 0.0001, 'geom', 'gid'); SELECT pgr_createTopology('secondfloor', 0.0001, 'geom', 'gid'); SELECT pgr_createTopology('transitions01', 0.0001, 'geom', 'gid'); SELECT pgr_createTopology('transitions12', 0.0001, 'geom', 'gid');
Update the unique identifier of each floor and points table and the source and target columns	update firstfloor set gid=gid+1000; update firstfloor set source=source+1000; update firstfloor set target=target+1000; update firstfloor_vertices_pgr set id=id+1000; update secondfloor set gid=gid+2000;

	update secondfloor set source=source+2000;
	update secondfloor set target=target+2000;
	update secondfloor_vertices_pgr set id=id+2000;
	
	update transitions01 set gid=gid+10000;
	update transitions01 set source=source+10000;
	update transitions01 set target=target+10000;
	update transitions01_vertices_pgr set id=id+10000;
	
	update transitions12 set gid=gid+20000;
	update transitions12 set source=source+20000;
	update transitions12 set target=target+20000;
	update transitions12_vertices_pgr set id=id+20000;

Create table that collects all data about the networks	CREATE TABLE public.ways (gid integer NOT NULL, elevation numeric, layer varchar, geom geometry(LineString), source integer, target integer, lenght float, name varchar, CONSTRAINT ways_pkey PRIMARY KEY (gid)) WITH (OIDS = FALSE) TABLESPACE pg_default; ALTER TABLE public.ways OWNER to postgres; -- Index: ways_geom_idx -- DROP INDEX public.ways_geom_idx; CREATE INDEX ways_geom_idx ON public.ways USING gist (geom) TABLESPACE pg_default; -- Index: ways_source_idx -- DROP INDEX public.ways_source_idx; CREATE INDEX ways_source_idx ON public.ways USING btree (source) TABLESPACE pg_default; -- Index: ways_target_idx -- DROP INDEX public.ways_target_idx; CREATE INDEX ways_target_idx ON public.ways USING btree (target) TABLESPACE pg_default;
Populate the table ways that collects all data from all floor tables	insert into ways select * from groundfloor; insert into ways select * from firstfloor; insert into ways select * from secondfloor; insert into ways select * from transitions01; insert into ways select * from transitions12;
Create table that collects all data about the decision points	-- Table: public.ways_vertices_pgr -- DROP TABLE public.ways_vertices_pgr; CREATE TABLE public.ways_vertices_pgr (

	id bigint NOT NULL, cnt integer, chk integer, ein integer, eout integer, the_geom geometry(Point,32632), elevation numeric, name character varying, searchtext tsvector, CONSTRAINT ways_vertices_pgr_pkey PRIMARY KEY (id)) WITH (OIDS=FALSE); ALTER TABLE public.ways_vertices_pgr OWNER TO postgres; -- Index: public.searchtext_gin -- DROP INDEX public.searchtext_gin; CREATE INDEX searchtext_gin ON public.ways_vertices_pgr USING gin (searchtext); -- Index: public.ways_vertices_pgr_the_geom_idx -- DROP INDEX public.ways_vertices_pgr_the_geom_idx; CREATE INDEX ways_vertices_pgr_the_geom_idx ON public.ways_vertices_pgr USING gist (the_geom);
Populate the table ways_vertices-pgr that collects all data from all decision points tables	insert into ways_vertices_pgr select * from groundfloor_vertices_pgr; insert into ways_vertices_pgr select * from firstfloor_vertices_pgr; insert into ways_vertices_pgr select * from secondfloor_vertices_pgr; insert into ways_vertices_pgr select * from transitions01_vertices_pgr; insert into ways_vertices_pgr select * from transitions12_vertices_pgr;
Trial query for route planning with Dijsktra's algorithm	SELECT * FROM pgr_dijkstra('SELECT gid AS id, source, target, length AS cost FROM ways', (SELECT gid FROM ways WHERE gid = 1040), (SELECT gid FROM ways WHERE gid = 10014), directed := false);
Make the column name searchable	SELECT ST_AsEWKT(ST_Translate(ST_Transform(geom ,4326), 0, 0,0)) FROM ways00 ; ALTER TABLE ways_vertices_pgr ADD COLUMN searchtext TSVECTOR; UPDATE ways_vertices_pgr SET searchtext = to_tsvector('english',name) UPDATE ways_vertices_pgr SET searchtext = to_tsvector('english', title || '' || content || '' || tags) CREATE INDEX searchtext_gin ON ways_vertices_pgr USING GIN(searchtext); CREATE TRIGGER ts_searchtext BEFORE INSERT OR UPDATE ON ways_vertices_pgr FOR EACH ROW EXECUTE PROCEDURE tsvector_update_trigger('searchtext', 'pg_catalog.english', 'name')

9.2 Main Page

```
<?php

  //Connect to the database
  $db_connection = pg_connect("host=localhost port=5432 dbname=routing user=postgres password=1311");
    //Query the database for the name of the desired point
  $query2 = pg_query($db_connection,
      "SELECT name FROM ways_vertices_pgr where searchtext @@ to_tsquery('keyword');");
  //Return the values from the query above
  $rows = pg_fetch_all($query2);
  echo $rows;
?>
<!doctype html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/widgets.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/CesiumWidget/Cesiumwidget.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/lighter.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/BaseLayerPicker/BaseLayerPicker.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/NavigationHelpButton/NavigationHelpButton.css">

  <script src="https://code.jquery.com/jquery-2.2.3.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
  <script src="cesium/Build/Cesium/Cesium.js"></script>

  <script src="links/PopFromTopMessage/js/modernizr.custom.80028.js"></script>
  <link href="https://fonts.googleapis.com/css?family=PTSans" rel="stylesheet">

  <link rel='stylesheet' href='links/PopFromTopMessage/css/style.css'>
  <!-- Use correct character set. -->
  <meta charset="utf-8">
  <!-- Tell IE to use the latest, best version. -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- Make the application on mobile take up the full browser screen and disable user scaling. -->
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no">

  <style>

/* split page so we can have the upper division, the middle part with the map and the lower part with the footer */
  topbar{
    text-family: 'PT Sans',  Arial, sans-serif;
  }
  section {
    width: 87%;
    top:0%;
    height: 30%; !important;
```

```
    text-family: {'PT Sans',  Arial, sans-serif};
  }

  .cesiumContainer {
    height: 100%;
    width: 100%;
    }

  #bt {
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.1em;
    text-align:center;
    font-weight:bold;
    color:#414141;
    margin:1em;
  }
  .t {
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.3em;
    text-align:center;
    font-weight:bold;
    color:#414141;
    margin:1em;
  }

/*Create the search box */
  .search_box {
    background-color: rgb(255, 255, 255);
    position: absolute;
    top:5.5%;
    left:87%;
    width: 13%;
    font-family: 'PT Sans',  Arial, sans-serif;
    text-align:center;
    padding:1%;
    }
  .search_box img{
    width:100%;
  }
  #button{
    padding:2%;
    background-color: white;
    color: black;
    font-weight: 700 !important;
    letter-spacing: 3px;
    transition: all 0.3s ease 0s;
  }
  #button:hover {
    color: white!important;
    font-weight: 700 !important;
    letter-spacing: 3px;
    background: #00a2db;
    -webkit-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);
    -moz-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);
    transition: all 0.3s ease 0s;
  }
  #place {
    background-color: white;
    font-family: 'PT Sans',  Arial, sans-serif;
    text-align:center;
```

```
    margin:4%;
  }


/* Set the style for the timeline, the animation container, the fullscreen button and the toolbar of cesium */
  .cesium-viewer-timelineContainer{
    position:absolute;
    bottom:0;
    left:169px;
    right:29px;
    height:27px;
    padding:0;
    margin:0;
    overflow:hidden;font-size:14px;
  }
  .cesium-viewer-animationContainer{
    position:absolute;
    bottom:0;
    left:0;
    padding:0;
    width:100px;
    height:51px;
  }
  .cesium-viewer-fullscreenContainer{
    position:absolute;
    bottom:0px;
    right:0;
    padding:0;
    width:29px;
    height:29px;
    overflow:hidden;
  }
  .cesium-viewer-toolbar{
    display:block;
    position:absolute;
    top:15px;
    right:10px;
    background-color:#fff4e2;
  }

/* Set the style for the footer */
  #footer {
    background-color: white;
  }

  #footer footer {
    background-color: white;
    color: #ffa100;
    height: 10%;
    text-align: center;
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:0.7em;
    padding: 1px;
  }
  #footer footer a {
    color: #373737;
    padding: 5px 20px;
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.3em;
```

```
  }
  #footer footer a:hover {
    color: #F4F4F4;
  }
    ul.show_list {
      margin-top:1em;
      list-style-type: none;
    }
    </style>
    <title>Aalborg University CPH in 3D </title>
  </head>
  <body>

  <topbar>
   <!-- The navigation bar with links to how this application works and contact information  -->
    <div id="bt">Aalborg University CPH 3D MAP</div>
  </topbar>
  <div class="search_box">
      <form name="insert" action="calcrouteD.php" method="POST" >
        <input type="radio" id="cost" name="cost" value="cost" checked> <label for="stairs">Stairs</label>
        <input type="radio" id="length" name="cost" value="length"> <label for="elevator">Elevator</label>
        <input size="16%" type="float" name="start" id="place" placeholder="Start"/>
    <br>
        <input size="16%" type="float" name="destination" id="place"placeholder="Destination"/><br>
        <input type="submit" id="button" value="Confirm" />
      </form>
      <ul class="show_list">
        <li><id="button_show" ><a onclick="showBuildings()"><b>Show Buildings</b></a></li>
        <li><id="button_show" ><a onclick="hideBuildings()"><b>Hide Buildings</b></a></li>
      </ul>
      <!-- Base Layer switcher   -->
      <div id="baseLayerPickerContainer"></div>
      <!-- Button opening navigational instructions window  -->
      <div id="navigationHelpButtonContainer"></div>
  </div>

  <section>
    <!-- The Map  -->
    <div id="cesiumContainer" class="cesiumContainer"></div>
    <script type="text/javascript">

  // Grant your CesiumJS app access to your ion assets
  // This is your actual default access token, you can copy/paste this directly into your app code
  Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJjYWQ2NDhkNy04MjNiLTQ4MjctOGQ4MC1mNjU1M2I1Y2U5Y2EiLCJpZCI6MzQ4MywiaWF0IjoxNTM3Nzc2Nzg0fQ.rYtvQGW7DzVmhsOokZ9J5dIxI93GXz0r1GCHFssVw-E'

  ////Create the list of available providers we would like the user to select from.
    var imageryViewModels = [];

      imageryViewModels.push(new Cesium.ProviderViewModel({
       name : 'Bing Maps Road',
```

```
        iconUrl :
Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingRoads.png'),
        tooltip : 'Bing Maps Road',
        creationFunction : function() {
          return new Cesium.IonImageryProvider({ assetId: 4 });
        }
      }));

      imageryViewModels.push(new Cesium.ProviderViewModel({
        name : 'Bing Maps Aerial',
        iconUrl :
Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingAerial.png'),
        tooltip : 'Bing Maps',
        creationFunction : function() {
          return new Cesium.IonImageryProvider({ assetId: 2 });
        }
      }));

      imageryViewModels.push(new Cesium.ProviderViewModel({
        name : 'Earth at Night',
        iconUrl :
Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/earthAtNight.png'),
        tooltip : 'The lights of cities and villages trace the outlines of
civilization \
    in this global view of the Earth at night as seen by NASA/NOAA\'s Suomi NPP
satellite.',
        creationFunction : function() {
          return new Cesium.IonImageryProvider({ assetId: 3812 });
        }
      }));

      imageryViewModels.push(new Cesium.ProviderViewModel({
        name : 'Bing Maps Road',
        iconUrl :
Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingRoads.png'),
        tooltip : 'Bing Maps Road',
        creationFunction : function() {
          return new Cesium.IonImageryProvider({ assetId: 4 });
        }
      }));

      imageryViewModels.push(new Cesium.ProviderViewModel({
        name : 'Map of Denmark',
        iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/sentinel-
2.png'),
        tooltip : 'Bing Maps Road',
        creationFunction : function() {
          return new Cesium.IonImageryProvider({ assetId: 10257 });
        }
      }));

  //Define the viewer
  var viewer = new Cesium.Viewer('cesiumContainer',{
    terrainProvider : Cesium.createWorldTerrain(),
    terrainExaggeration : 0.0,
    //Hide the base layer picker
    baseLayerPicker :false,
    fullscreenButton: false,
    navigationHelpButton:false,
    scene3DOnly:true,
```

```
  homeButton:false,
  geocoder:false
  });

viewer.scene.globe.depthTestAgainstTerrain = true;
viewer.scene.globe.enableLighting = true; // Lighting effects

//Define the scene
var scene = viewer.scene;

//Create the baseLayerPicker widget using the view models.
var layers = viewer.imageryLayers;
var baseLayerPicker = new Cesium.BaseLayerPicker('baseLayerPickerContainer', {
  globe : viewer.scene.globe,
  imageryProviderViewModels : imageryViewModels
});
var navigationHelpButton = new Cesium.NavigationHelpButton({
  container : 'navigationHelpButtonContainer'
});

var transparentStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.9)",
  show : true
});

var transparentBoxStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.5)",
  show : true
});

//Load the tilesets
var tileset = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_AAU/tileset.json',
  debugShowBoundingVolume : false
}));

var tileset02 = scene.primitives.add(new Cesium.Cesium3DTileset({
    url: 'AAUtiles/tileset_buildings/tileset.json',
    debugShowBoundingVolume : false,
    show : true
  }));

var transparentStyle = new Cesium.Cesium3DTileStyle({
    color : "color('white', 0.0)",
    show : true
  });

var nontransparentStyle = new Cesium.Cesium3DTileStyle({
    color : "color('white', 1.0)",
    show : true
  });

tileset02.style = transparentStyle;

function showBuildings() {
  tileset02.style = nontransparentStyle;
};

function hideBuildings() {
  tileset02.style = transparentStyle;
```

```
  };

  //Zoom to the tileset
  tileset.readyPromise.then(function(tileset) {
    viewer.scene.primitives.add(tileset);
    viewer.zoomTo(tileset, new Cesium.HeadingPitchRange(-1.5, -0.35,
tileset.boundingSphere.radius * 1.0));
  }).otherwise(function(error) {
    console.log(error);
  });

  </script>
  </section>
  <div id="footer">
  <footer>
        <div class="contacts">
          <a href="">Email</a>
          <a href="">Twitter</a>
          <a href="">About us</a>
        </div>
        <p>Design and code by MG</p>
      </footer>
    </div>
  </body>
</html>
```

9.3 Routing Calculations

```
<?php

  //Connect to the database
  $db_connection = pg_connect("host=localhost port=5432 dbname=routing user=postgres
password=1311");
  //Query the database for calculating the route path
  $query = "DROP TABLE IF EXISTS route;
        CREATE TABLE route (seq integer, path_seq integer, node bigint, edge bigint,
cost float, agg_cost float, geom geometry(LineStringZM));";
  $result = pg_query($query);

  $query1="INSERT INTO route SELECT a.*, b.geom FROM pgr_dijkstra(
          'SELECT gid AS id,
             source,
             target,
             $_POST[cost] as cost
            FROM ways00',
          (SELECT min(id) FROM ways_vertices_pgr WHERE '$_POST[start]'=
ways_vertices_pgr.name),
          (SELECT min(id) FROM ways_vertices_pgr WHERE '$_POST[destination]'=
ways_vertices_pgr.name),
          directed := false)as a
          left join ways00 as b
          on (edge=gid) order by seq;";
  $result1 = pg_query($query1);
?>
<!doctype html>
<html lang="en">
  <head>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/widgets.css">
  <link rel="stylesheet"
href="cesium/Build/Cesium/Widgets/CesiumWidget/Cesiumwidget.css">
  <link rel="stylesheet" href="cesium/Build/Cesium/Widgets/lighter.css">
  <link rel="stylesheet"
href="cesium/Build/Cesium/Widgets/BaseLayerPicker/BaseLayerPicker.css">
  <link rel="stylesheet"
href="cesium/Build/Cesium/Widgets/NavigationHelpButton/NavigationHelpButton.css">

  <script src="https://code.jquery.com/jquery-2.2.3.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js">
</script>
  <script src="cesium/Build/Cesium/Cesium.js"></script>

  <script src="links/PopFromTopMessage/js/modernizr.custom.80028.js"></script>
  <link href="https://fonts.googleapis.com/css?family=PTSans" rel="stylesheet">

  <link rel='stylesheet' href='links/PopFromTopMessage/css/style.css'>
  <!-- Use correct character set. -->
  <meta charset="utf-8">
  <!-- Tell IE to use the latest, best version. -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- Make the application on mobile take up the full browser screen and disable
user scaling. -->
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no">

  <style>

/* split page so we can have the upper division, the middle part with the map and the lower part with the footer */
topbar{
    text-family: 'PT Sans',  Arial, sans-serif;
  }
  section {
    width: 87%;
    top:0%;
    height: 30%; !important;
    text-family: {'PT Sans',  Arial, sans-serif};
  }

  .cesiumContainer {
    height: 100%;
    width: 100%;
    }

  #bt{
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.1em;
    text-align:center;
    font-weight:bold;
    color:#414141;
    margin:1em;
  }
  .t {
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.3em;
    text-align:center;
    font-weight:bold;
    color:#414141;
    margin:1em;
  }

/*Create the search box */
  .search_box {
    background-color: rgb(255, 255, 255);
    position: absolute;
    top:5.5%;
    left:87%;
    width: 13%;
    font-family: 'PT Sans',  Arial, sans-serif;
    text-align:center;
    padding:1%;
    }
  .search_box img{
    width:100%;
  }
  #button{
    padding:2%;
    background-color: white;
    color: black;
    font-weight: 700 !important;
    letter-spacing: 3px;
    transition: all 0.3s ease 0s;
```

```
  }
  #button:hover {
    color: white!important;
    font-weight: 700 !important;
    letter-spacing: 3px;
    background: #00a2db;
    -webkit-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);
    -moz-box-shadow: 0px 5px 40px -10px rgba(0,0,0,0.57);
    transition: all 0.3s ease 0s;
  }
  #place {
    background-color: white;
    font-family: 'PT Sans',  Arial, sans-serif;
    text-align:center;
    margin:4%;
  }

/* Set the style for the timeline, the animation container, the fullscreen button and the toolbar of cesium */
  .cesium-viewer-timelineContainer{
    position:absolute;
    bottom:0;
    left:169px;
    right:29px;
    height:27px;
    padding:0;
    margin:0;
    overflow:hidden;font-size:14px;
  }
  .cesium-viewer-animationContainer{
    position:absolute;
    bottom:0;
    left:0;
    padding:0;
    width:100px;
    height:51px;
  }
  .cesium-viewer-fullscreenContainer{
    position:absolute;
    bottom:0px;
    right:0;
    padding:0;
    width:29px;
    height:29px;
    overflow:hidden;
  }
  .cesium-viewer-toolbar{
    display:block;
    position:absolute;
    top:15px;
    right:10px;
    background-color:#fff4e2;
  }

/* Set the style for the footer */
  #footer {
    background-color: white;
  }

  #footer footer {
```

```
    background-color: white;
    color: #ffa100;
    height: 10%;
    text-align: center;
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:0.7em;
    padding: 1px;
  }
  #footer footer a {
    color: #373737;
    padding: 5px 20px;
    font-family:'PT Sans',  Arial, sans-serif;
    font-size:1.3em;
  }
  #footer footer a:hover {
    color: #F4F4F4;
  }
    ul.show_list {
      margin-top:1em;
      list-style-type: none;
    }

   .floor_buttons {
            display: block;
            position: absolute;
            top: 75%;
            right: 8%;
        }

    </style>
    <title>Aalborg University CPH in 3D </title>
  </head>
  <body>

  <topbar>
   <!-- The navigation bar with links to how this application works and contact information  -->
    <div id="bt">Aalborg University CPH 3D MAP</div>
  </topbar>

  <div class="search_box">

      <form name="insert" action="calcrouteD.php" method="POST" >
        <input type="radio" id="cost" name="cost" value="cost" checked> <label for="stairs">Stairs</label>
        <input type="radio" id="length" name="cost" value="length"> <label for="elevator">Elevator</label>
        <input size="16%" type="float" name="start" id="place" placeholder="Start"/> <br>
        <input size="16%" type="float" name="destination" id="place"placeholder="Destination"/><br>
        <input type="submit" id="button" value="Confirm"/>
      </form><br>
      <ul class="show_list">
        <li><id="button_show"><a onclick="showFloor2()"><b>Show 02</b> </a></li>
        <li><id="button_show><a onclick="showFloor1()"><b>Show 01</b> </a></li>
        <li><id="button_show"><a onclick="hideFloor2()"><b>Hide 02</b> </a></li>
        <li><id="button_show"><a onclick="hideFloor1()"><b>Hide 01</b> </a></li>
      </ul>
              <!-- Base Layer switcher   -->
```

```
    <div id="baseLayerPickerContainer"></div>

    <!-- Button opening navigational instructions window  -->
    <div id="navigationHelpButtonContainer"></div>

      <p size="3%" face="PT Sans" color="#352d2f">Distance in meters : <p size="3%" face="PT Sans" color="#0000ff">
        <!-- Get information on how many people are evacuated in the specified area
          by retrieving data from the database -->
        <?php
          $db_connection = pg_connect("host=localhost dbname=routing user=postgres password=1311");
          $query1 = pg_query($db_connection, 'SELECT COUNT(cost) as count FROM route');
          $count = pg_fetch_result($query1,0,'count');
          echo $count;
        ?>
        <p size="3%" face="PT Sans" color="#352d2f">Instructions by landmarks: <p size="3%" face="PT Sans" color="#0000ff"> <br>
        <p size="3%" face="PT Sans" color="#352d2f">Initialize your route<p size="3%" face="PT Sans" color="#0000ff">
        <?php
          $query2 = pg_query($db_connection, 'Select distinct instructions as instru from ways00,route where ST_3DIntersects(route.geom,ways00.geom) ;');
          $numrows=pg_num_rows($query2);
          for ($i=0; $i< $numrows; $i++)
          {
            $instr1 = pg_fetch_result($query2,$i,'instru');
            echo $instr1 ;
          }
        ?>
        <p size="1em" face="PT Sans" color="#352d2f">You reach your destination<p size="1em" face="PT Sans" color="#0000ff">

    </div>

  <section>

  <!-- The Map  -->
  <div id="cesiumContainer" class="cesiumContainer"></div>

  <script type="text/javascript">
  // Grant your CesiumJS app access to your ion assets
  // This is your actual default access token, you can copy/paste this directly into your app code
  Cesium.Ion.defaultAccessToken = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJjYWQ2NDhkNy04MjNiLTQ4MjctOGQ4MC1mNjU1M2I1Y2U5Y2EiLCJpZCI6MzQ4MywiaWF0IjoxNTM3Nzc2Nzg0fQ.rYtvQGW7DzVmhsOokZ9J5dIxI93GXz0r1GCHFssVw-E'

  ////Create the list of available providers we would like the user to select from.
    var imageryViewModels = [];

      imageryViewModels.push(new Cesium.ProviderViewModel({
       name : 'Bing Maps Road',
       iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingRoads.png'),
       tooltip : 'Bing Maps Road',
       creationFunction : function() {
         return new Cesium.IonImageryProvider({ assetId: 4 });
```

```
      }
    }));

    imageryViewModels.push(new Cesium.ProviderViewModel({
      name : 'Bing Maps Aerial',
      iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingAerial.png'),
      tooltip : 'Bing Maps',
      creationFunction : function() {
        return new Cesium.IonImageryProvider({ assetId: 2 });
      }
    }));

    imageryViewModels.push(new Cesium.ProviderViewModel({
      name : 'Earth at Night',
      iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/earthAtNight.png'),
      tooltip : 'The lights of cities and villages trace the outlines of civilization \
   in this global view of the Earth at night as seen by NASA/NOAA\'s Suomi NPP satellite.',
      creationFunction : function() {
        return new Cesium.IonImageryProvider({ assetId: 3812 });
      }
    }));

    imageryViewModels.push(new Cesium.ProviderViewModel({
      name : 'Bing Maps Road',
      iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/bingRoads.png'),
      tooltip : 'Bing Maps Road',
      creationFunction : function() {
        return new Cesium.IonImageryProvider({ assetId: 4 });
      }
    }));

    imageryViewModels.push(new Cesium.ProviderViewModel({
      name : 'Map of Denmark',
      iconUrl : Cesium.buildModuleUrl('Widgets/Images/ImageryProviders/sentinel-2.png'),
      tooltip : 'Bing Maps Road',
      creationFunction : function() {
        return new Cesium.IonImageryProvider({ assetId: 10257 });
      }
    }));

  //Define the viewer
  var viewer = new Cesium.Viewer('cesiumContainer',{
    terrainProvider : Cesium.createWorldTerrain(),
    terrainExaggeration : 0.0,
    //Hide the base layer picker
    baseLayerPicker : false,
    fullscreenButton: false,
    navigationHelpButton:false,
    scene3DOnly:true,
    homeButton:false,
    animation:true,
    geocoder:false
  });

```

```
viewer.scene.globe.depthTestAgainstTerrain = true;
viewer.scene.globe.enableLighting = true; // Lighting effects

//Define the scene
var scene = viewer.scene;
var clock = viewer.clock;
var referenceFramePrimitive;
//Create the baseLayerPicker widget using the view models.
var layers = viewer.imageryLayers;

var baseLayerPicker = new Cesium.BaseLayerPicker('baseLayerPickerContainer', {
  globe : viewer.scene.globe,
  imageryProviderViewModels : imageryViewModels
});

var navigationHelpButton = new Cesium.NavigationHelpButton({
  container : 'navigationHelpButtonContainer'
});

//Create styles for the tilesets
var transparentStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.0)",
  show : true
});
var semitransparentStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 0.5)",
  show : true
});
var nontransparentStyle = new Cesium.Cesium3DTileStyle({
  color : "color('white', 1.0)",
  show : true
});

//Load the tilesets
var tileset = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_slab00/tileset.json',
  debugShowBoundingVolume : false,
  show : true
}));
var tileset01 = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_slab01/tileset.json',
  debugShowBoundingVolume : false,
  show : true
}));
var tileset02 = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_slab02/tileset.json',
  debugShowBoundingVolume : false,
  show : true
}));
var tileset_00 = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_poly00/tileset.json',
  debugShowBoundingVolume : false,
  show : true
}));
var tileset_01 = scene.primitives.add(new Cesium.Cesium3DTileset({
  url: 'AAUtiles/tileset_poly01/tileset.json',
  debugShowBoundingVolume : false,
  show : true
}));
var tileset_02 = scene.primitives.add(new Cesium.Cesium3DTileset({
```

```
    url: 'AAUtiles/tileset_poly02/tileset.json',
    debugShowBoundingVolume : false,
    show : true
  }));

  tileset.style = nontransparentStyle;
  tileset01.style = nontransparentStyle;

  tileset_00.style = semitransparentStyle;
  tileset_01.style = semitransparentStyle;

  tileset_02.style = transparentStyle;
  tileset02.style = transparentStyle;

  function showFloor2() {
    tileset_02.style = semitransparentStyle;
    tileset02.style = nontransparentStyle;
  };

  function showFloor1() {
    tileset_01.style = semitransparentStyle;
    tileset01.style = nontransparentStyle;
  };

  function hideFloor2() {
    tileset_02.style = transparentStyle;
    tileset02.style = transparentStyle;
  };

  function hideFloor1() {
    tileset_01.style = transparentStyle;
    tileset01.style = transparentStyle;
  };

  //Zoom to the tileset
  tileset.readyPromise.then(function(tileset) {
    viewer.scene.primitives.add(tileset);
    viewer.zoomTo(tileset, new Cesium.HeadingPitchRange(-1.5, -1.25, tileset.boundingSphere.radius * 1.8));
  }).otherwise(function(error) {
    console.log(error);
  });

  function createstartPin() {
    fetch("callpointgeom.php",{
        method:'get',
        credintentials: 'include',
      })
      .then(function(reply) {return reply.json()})
      .then(function(info) {

      var pinBuilder = new Cesium.PinBuilder();
      for (elm in info) {
        record = info[elm];

        var lat_start = parseFloat(record['lat_start']);
        var lon_start = parseFloat(record['lon_start']);
        var z_start = parseFloat(record['z_start']);
        var hei_start=z_start+1.0

```

```
    var startPin = viewer.entities.add({
     name : 'Start pin',
     position : Cesium.Cartesian3.fromDegrees(lat_start,lon_start,hei_start),
     billboard : {
      image : pinBuilder.fromColor(Cesium.Color.ROYALBLUE, 28).toDataURL(),
      verticalOrigin : Cesium.VerticalOrigin.BOTTOM
     },
    });
   }
   });
 };

 function createdesPin() {
  fetch("callpointgeomD.php",{
    method:'get',
    credintentials: 'include',
   })
   .then(function(reply) {return reply.json()})
   .then(function(info) {

   var pinBuilder = new Cesium.PinBuilder();
   for (elm in info) {
    record = info[elm];

    var lat_des = parseFloat(record['lat_des']);
    var lon_des = parseFloat(record['lon_des']);
    var z_des = parseFloat(record['z_des']);
    var hei_des =z_des+1.5

    var desPin = viewer.entities.add({
     name : 'Destination pin',
     position : Cesium.Cartesian3.fromDegrees(lat_des,lon_des,hei_des),
     billboard : {
      image : pinBuilder.fromColor(Cesium.Color.ROYALBLUE, 28).toDataURL(),
      verticalOrigin : Cesium.VerticalOrigin.BOTTOM
     }
    });
   }
   });
 };

var start = Cesium.JulianDate.fromDate(new Date(2019, 5, 14, 16));
var stop = Cesium.JulianDate.addSeconds(start, 360, new Cesium.JulianDate());

viewer.clock.startTime = start.clone();
viewer.clock.stopTime = stop.clone();
viewer.clock.currentTime = start.clone();
viewer.clock.clockRange = Cesium.ClockRange.LOOP_STOP;
viewer.clock.multiplier = 10;

viewer.timeline.zoomTo(start, stop);

 function createLine() {
  uio = [];
  var u=[];
  fetch("callroutegeom.php",{
   method:'get',
   credintentials: 'include',
  })
  .then(function(response) {return response.json()})
```

```
    .then(function(data) {

      for (elm in data) {
        record = data[elm];
        var lat = parseFloat(record['lat']);
        var lon = parseFloat(record['lon']);
        var z = parseFloat(record['z']);
        var hei=z+2;
        u.push(lat,lon,hei);
        uio.push({lat,lon,hei});
        //console.log(uio[1]);
        var line=viewer.entities.add({
          name : 'Route Line',
          polyline : {
            positions : Cesium.Cartesian3.fromDegreesArrayHeights(u),
            width : 4,
            material : new Cesium.PolylineOutlineMaterialProperty({
              color : Cesium.Color.BLUE,
              outlineWidth : 0.5,
              outlineColor : Cesium.Color.BLACK
            })
          }
        });
        viewer.zoomTo(viewer.entities);
      }
    })
    .then(()=> {
            animation(uio);
            function animation(uis){
              let seConds = 0;
              var position = new Cesium.SampledPositionProperty();
              console.log(uis.length);
                for(i = 0; i< uis.length;i++){
                  seConds = seConds + 10;
                  position.addSample(Cesium.JulianDate.addSeconds(start, seConds, new Cesium.JulianDate()), Cesium.Cartesian3.fromDegrees(uis[i].lat, uis[i].lon, uis[i].hei));
                }
                var entity = viewer.entities.add({
                    availability : new Cesium.TimeIntervalCollection([new Cesium.TimeInterval({
                        start : start,
                        stop : stop
                    })]),
                    position : position,
                    orientation : new Cesium.VelocityOrientationProperty(position),
                      model : {
                        uri : 'AAUtiles/man/NewFeatureType.gltf',
                        minimumPixelSize : 64,
                        scale:0.6
                      },
                      path : {
                        resolution : 1,
                        material : new Cesium.PolylineGlowMaterialProperty({
                          glowPower : 0.1,
                          color : Cesium.Color.YELLOW
                        }),
                          width : 10
                      }
                });
```

```
              }
    });
  };

  createLine();
  createstartPin();
  createdesPin();

  </script>
  </section>
  <div id="footer">
  <footer>
        <div class="contacts">
          <a href="">Email</a>
          <a href="">Twitter</a>
          <a href="">About us</a>
        </div>
        <p>Design and code by MG</p>
      </footer>
    </div>
  </body>
</html>
```