SOFTWARE ENTREPRENEURSHIP

Framework for Software Startups

MSc – Software Development

Aalborg University, Denmark

Kasper Bisgaard

Master Thesis

06/06/2016

Department of Computer Science
Aalborg University
Selma Lagerlöfs Vej 300
http://www.cs.aau.dk

Title: Framework for Software Startups

Topic: Software Entrepreneurship

Project period: SWD10, Spring 2016

Project group: SWD10, is1013f16

Group members: Kasper Bisgaard

Supervisor: Ivan Aaen

Number of pages: 77

Finished on: 06/06/2016

Signature: _____

CONTENT

1 INTRODUCTION

Software entrepreneurship has the potential to be a very lucrative field these days due to the rising implementation of it-controlled devices both in the home and everywhere else. Today most people in the world has a smartphone, and with the latest uprising of wearables and smart-devices in the home and in the workplace, there is an increasing need for software products.

When producing software products, there are virtually no development costs besides the time that is put into developing these products. Therefore, practically everyone with an interest in IT and software, has the potential to be a software entrepreneur. But releasing a software product to the public does not make someone a software entrepreneur. The developer needs to be able to see how the software product can turn into a money earning business, and in order to do that, the entrepreneur needs to create a business model which takes the various aspects of business into consideration.

With that in mind, this thesis aims to combine a set of tools in the field of "Entrepreneurship", "Software Development" and "Innovation", that together form a solid foundation for starting a business in the field of software. The tools that I intent to use are "Configuration tables" from (Aaen 2016), "Business Model Canvas" from (Osterwalder, Pigneur 2013) and "SCRUM" as described by (Larman 2004). In order to test the method, I will apply it to a realistic case, in which I will develop a software product and a business model with the tools I have chosen.

The flow of this thesis will be as follows: First I will briefly present the theories and tools that I have chosen to work with. Then I will present a problem case, and my proposal to a software product that aims to solve the problem. Finally, I will start to develop the product and the business model using the tools and the approach that I have proposed.

1.1 PROBLEM STATEMENT

How can configuration tables from (Aaen 2016) be combined with the Business Model Canvas from (Osterwalder, Pigneur 2013) using an iterative software development process in order to create a sustainable software product and business model?

2 THEORY OVERVIEW

The purpose of this chapter is to provide an overview of the theory that I will use through this project. I will not go into details with the theory, but only provide a short description of each element.

2.1 ESSENCE CONFIGURATION TABLES

The configuration table from (Aaen 2016) aims to structure the way a problem is understood in relation to the product solution. Essence revolves around four views “Product”, “Paradigm”, “Project” and “Process”.

- Product describes the product proposal
- Paradigm describes the way the problem is understood
- Project describes how the proposed solution solves the observed problem
- Process describes the process of the project

Configuration tables also provide three levels of abstraction related to the four views “Rationale”, “Strategy” and “Tactics”.

- Rationale aims to describe **why** a product should be developed
- Strategy aims to describe **what** product should be developed
- Tactics aims to describe **how** the product should be developed

These four views and three levels of abstraction provides a 4x3 table that is easy to comprehend and makes it easy to get an overview of the project and product.

	Paradigm	**Product**	**Project**	**Process**
Rationale	*Challenge, Problem*	*Key technologies*	*Vision, Warrant*	*Rationale review*
Strategy	*Key elements*	*Key components*	*Justification*	Strategy review
Tactics	*Key scenarios*	*Key features*	*Key mapping*	Tactics review

2.2 System Design

In order to design the software system, I have chosen to include some theory from the "System Analysis and Design" field. I will not be using the processes of System Design in all of its aspects, but will only include the most important aspects in order to facilitate an effective design and development process (Mathiassen, Munk-Madsen et al. 2000).

I have decided to include "Use Cases" and "Object Models" as the only tools from the System Design field.

- Use cases are used to describe and understand a specific aspect of the application. There are various different ways of describing use cases, but I have chosen a very simple format that does not invite a lot of detailing. The use case format that I have chosen describes the "Actors" which is the person or system that is involved in the current use case. A "Summary" which provides a short description of the use case, and a "Scenario list" which describes a step-by-step overview of what happens in the use case.
- An object model is more technical than the use case. It provides an overview of the most important objects of the application in order to give the developer an idea of how the data is structured in the application.

2.3 Business Model Canvas

The Business Model Canvas from (Osterwalder, Pigneur 2013) provides an organized way to create and innovate a business model. It consists of nine building blocks, that together form the business model.

- **Key partners** describes the main partners that the company is associated with.
- **Key activities** describes the main activities that the company performs in order to create value for their customers.
- **Key resources** describes the main resources that the company possesses.
- **Value proposition** describes the value that the company creates for the customers.
- **Customer relationships** describes the relationship that the company has with the customers.
- **Channels** describes the distribution channels that the company uses to reach the customers.
- **Customer segments** describes the customer segments that the company aims to hit.

- **Cost structure** describes the costs that are related to producing the product.
- **Revenue streams** describes how the company earns money from selling their product.

2.4 SCRUM

SCRUM is a project management method specifically aimed at software development (Larman 2004). I will not be using all aspects of SCRUM as it is not relevant to this project. I will be using the elements “Product backlog”, “Sprint backlog” and the “iterative development process”, “Sprint review”.

- Product backlog is used for describing all remaining tasks of the current development project.
- Sprint backlog is used to describe the task that was planned for a specific sprint / iteration.
- “Iterative development process” means, that the development process occurs in loops. SCRUM recommends, that an iteration takes 30 days after which a sprint review should be performed, before a new iteration can begin.
- Sprint review is a process that helps the developers keep track of their progress and what remains to be done.

2.5 PARADIGMS – SOFTWARE ENTREPRENEURSHIP

In this section I will briefly describe two paradigms of software entrepreneurship that are logic opposites. I will briefly present the main characteristics of two paradigms “Causation” and “Effectuation” as defined by (Sarasvathy 2001), that I deem most relevant for this paper.

2.5.1 Causation

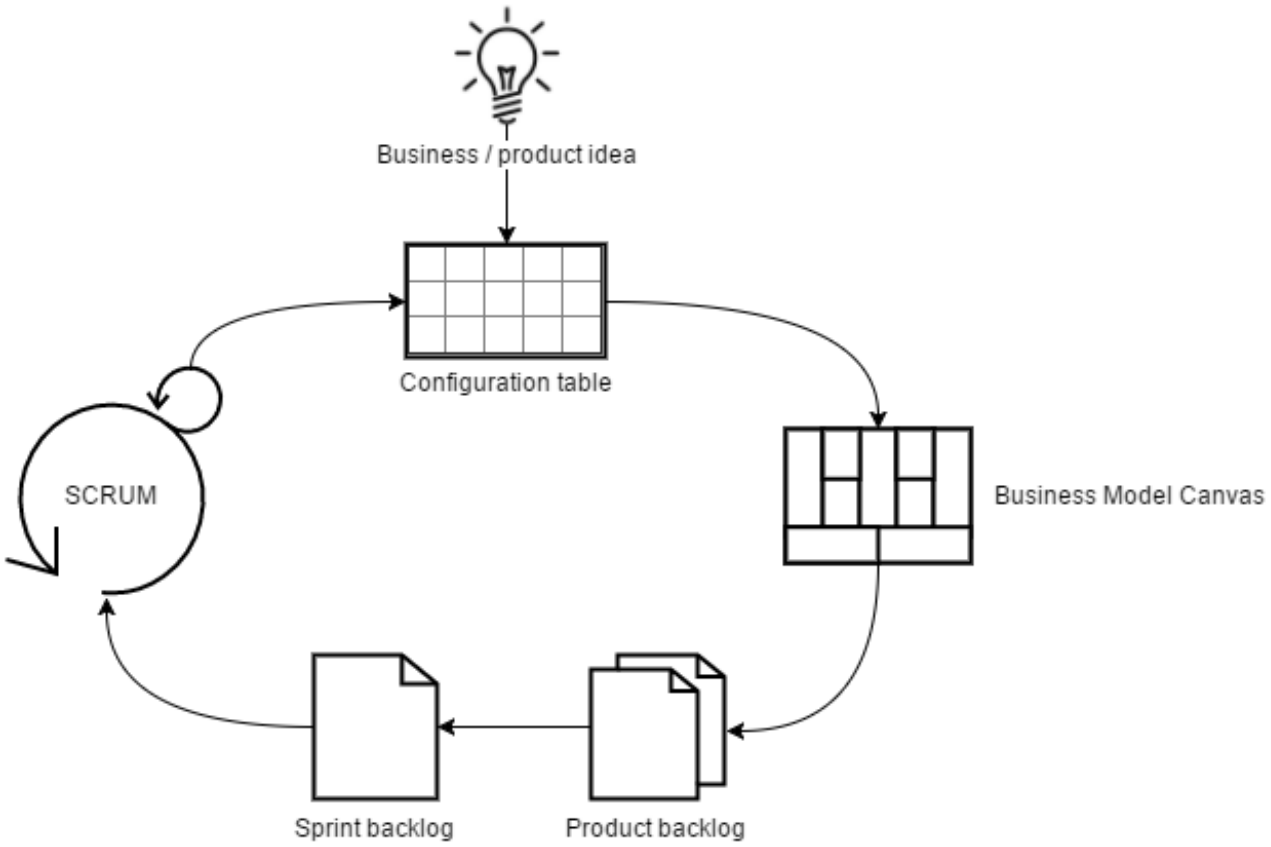
A causal process always starts with the entrepreneur having a goal in mind. The entrepreneur aims to create a specific product with a specific purpose for a specific customer segment, and there should be some sort of predictability as to what is needed. Every decision that the entrepreneur takes revolves around expected return of investment. When using causal thinking process, every other company are seen as competition. In causation, the production process is first planned in details, and then executed according to the plan, making it a linear process.

2.5.2 Effectuation

Effectuation starts with the entrepreneur, and the resources that they have at their disposal. Resources can be anything from personal skills of the entrepreneur to their personal and professional network. Rather than considering expected return, the effectual entrepreneur looks at affordable loss. The effectual entrepreneur sees other companies as an opportunity for corporation and creating strategic partnerships. Effectuation an iterative process as opposed to the linear process of causation.

Though these two paradigms are seen as opposites, (Kraaijenbrink 2008) claims, that an implementation of either of the two paradigms separately is not practical. In reality an entrepreneur would most likely use a combination of the two paradigms, mixing and matching the parameters depending on the personality of the entrepreneur.

3 NEW MODEL



The model above includes the theories and tools that I briefly described in chapter 0. The model does not take any idea generation into consideration. It assumes, that an idea already exists with an initial idea for a business model. The first step in the model is to fill in the configuration table according to the business/product idea. Alongside filling in the configuration table, a business model should begin to take form based on the Business Model Canvas. Once the configuration table and business model is in place, a product backlog can be defined and the first sprint can be planned with a sprint backlog. At this point, the development of the product should begin based on the sprint backlog. This process is repeated until the product and business model is finished. The iteration process indicates that I am using principles of the effectuation paradigm.

4 CASE

Today, more and more buildings are filled with electric installations that controls the building, and ensures that the indoor climate is always at an acceptable level of quality. Sometimes an electrical component needs to be switched with a newer model, or disconnected for maintenance. In order to do this, an electrician needs to know how the component is wired, so he does not mess up something else in the building.

In an ideal world, the electrician would have access to detailed documentation that describes exactly how everything is connected. But the process of documenting installations is too long and complicated, and most of the time documentation is not created or updated.

There are basically two main scenarios for an electrician when he starts a job:

- Installation in a new building
- Installation / updating an old building

In the first case, documentation is not an issue since a newly build building will always have detailed documentation in the form of a CAD document, that describes how everything should be wired. It is the electricians job to setup everything according to the documentation.

In the second case the electrician arrives at an older building. Either the documentation is up-to-date, or it is not. If the documentation is up-to-date, the electrician needs to make sure that whatever changes he makes, is properly updated in the documentation so that it stays up to date. If the documentation is not up-to-date, he first needs to figure out how everything is set up, before he can start installing.

Updating documentation is not an easy process since it requires knowledge in using CAD software, and it cannot be done on-site. Today updating documentation is a 3-step process:

1. Handwritten documentation on-site
2. Transcribing the handwritten documentation in the office
3. CAD expert updates the documentation

4.1 SOLUTION PROPOSAL

I propose a product which allows the electricians to document their work as they install it on-site. The product is a smartphone application based on a graph database. Graph databases are especially good for storing data which is highly connected.

The basic concept is that all electric components have a number of inputs and outputs, and any number of components can be connected via these inputs / outputs illustrated by the simple model below.

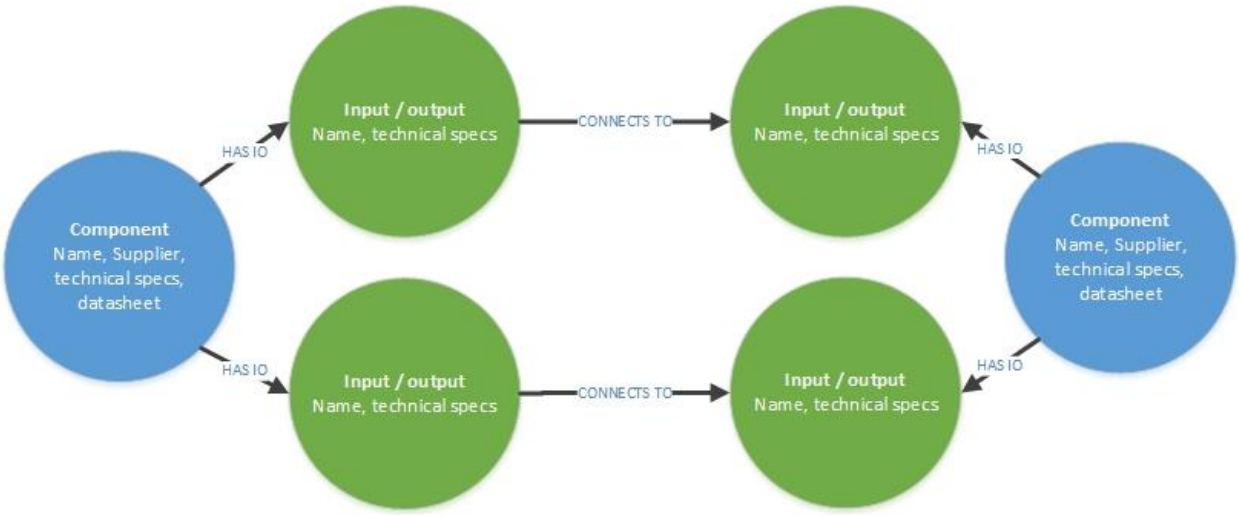


Figure 1 - Basic concept example

As it is shown in the illustration, there are two types of data “components” and “input /outputs”. The relations between the Input / outputs is what will eventually create foundation for the digital documentation.

5 PREFACE

The purpose of this section is to create the initial version of the product, configuration table and the business model canvas. The first step in this phase will be to define some use case descriptions in order to better describe my vision for the product. Based on the use cases I will define the first configuration table which will describe the problem and my solution proposal in more details. Once the configuration is defined, I will start creating the first business model canvas based on the knowledge I have of the problem domain and the product at this stage.

5.1 USE CASES

Use cases are very effective when planning the development of an application. Use cases provides an overview of the main functionality that the application must support. Use cases should be described in short and precise sentences, and should not contain irrelevant information in order to avoid any misunderstandings.

I have chosen to use a very simple use case format. It consists of an "Actor" which describes who or what is involved in the use case. A "Summary", which describes the purpose of the use case. A "Scenario" which is a numbered list that describes the flow of the use case as it progresses from the point of view of the actor.

In order to keep the use cases simple, I have chosen not to include "Preconditions" and "Postconditions" which are popular ways of describing what should be "True" in the beginning of a use case, and what should be "True" in the end of a use case.

The use case descriptions serve as the foundation for the further planning process, and are specifically useful when trying to identify objects and methods for the applications classes. Objects are picked out of the use case description by identifying all the nouns in the descriptions, though not all of the nouns will become objects in the application.

5.1.1 Login

Actor*:* User

Summary: Allow a user to log in to the system.

Scenario:

1. User enters email and password
2. User clicks “login”
3. If credentials are correct the user is redirected to the main page

5.1.2 Add project

Actor: User

Summary: Allows the user to add a new project to their list of projects.

Scenario:

1. The system displays a list of projects that the user is currently involved in
2. The user clicks the “Add project” button
3. The system displays a dialog with input fields for the name and geo location of the new project
4. The user enters a name and a geo location and clicks “Add project”
5. The new project is displayed on the main page

5.1.3 Add building

Actor: User

Summary: Allow the user to add a new building to an existing project.

Scenario:

1. The system displays a list of buildings for the chosen project
2. The user clicks the “Add building” button
3. The system displays a dialog with an input field for the name of the new building
4. The user enters a name and clicks “Add building”
5. The new building is displayed on the “Choose building” page

5.1.4 Add room

Actor: User

Summary: Allow the user to add a new room to an existing building.

Scenario:

1. The system displays a list of rooms for the chosen building
2. The user clicks the “Add room” button
3. The system displays a dialog with an input field for the name of the new room
4. The user enters a name and clicks “Add room”
5. The new room is displayed on the “Choose room” page

5.1.5 Add location

Actor: User

Summary: Allow the user to add a specific location to an existing room.

Scenario:

1. The system displays a list of locations for the chosen room
2. The user clicks the “Add location” button
3. The system displays a dialog with an input field for the name of the new location
4. The user enters a name and clicks “Add location”
5. The new location is displayed on the “Choose location” page

5.1.6 Connect components

Actor: User

Summary: Allow the user to connect various components inputs and outputs in order to create documentation.

Scenario:

1. User chooses two components to connect.
2. System displays two columns with each of the components’ inputs/outputs.
3. User chooses which inputs/outputs to connect.
4. User clicks the “connect” button.
5. System creates a new connection between the chosen inputs/outputs in the database.

5.2 CONFIGURATION PREFACE

The purpose of the configuration tables is to create an overview of problem domain alongside the solution and product proposal.

The configuration table is divided into four columns “Paradigm”, “Product”, “Project” and “Process”. The paradigm column aims to describe the current understanding of the problem domain. The product column describes the product proposal based on the understanding of the problem domain. The project column aims to argue how the proposed product solves the problem. The process column describes important expectations in the beginning of the iteration and findings in the end of the iteration.

View	**Paradigm**	**Product**	**Project**	**Process**
Value	*Reflection*	*Transaction*	*Reasoning*	*Appreciation*
Rationale: Why?	*Challenge* Allow electricians to document their installations from their smartphone *Problem* The process of documenting electric installations is too long and resource demanding	*Key technologies* - Neo4J Graph Database - Touch screen interface	*Vision* Connexion - easily create documentation for electrical installations *Warrant* Good documentation can save the electrician from spending hours on figuring out how a system is set up.	*Rationale review* *Expectations:* Component suppliers are willing to provide us with data about components and their IO's *Findings:* Cannot count on component suppliers to provide data about components and their IO's
Strategy: What?	*Key elements* Objects: User, Project, Building, Room, Location, Component, IO	*Key components* - On-site documentation of installations - Overview of existing installations	*Justification* *Backing:* A simple system makes it easy to document installations. *Qualifier:* Requires internet access *Rebuttal:* Most people have phones with internet access	*Strategy review* *Expectations:* The key components are necessary and sufficient to facilitate proper documentation *Findings:* A planning module should be included
Tactics: How?	*Key scenarios* - Setup of components inputs/outputs, and mapping them to specific locations in a room - Get an overview of what is connected and how.	*Key features* - Create documentation from smartphone	*Key mapping* - On-site module enables user to document installations while setting them up on the job site. - Overview module gives the user an idea of how the existing installations are set up.	*Tactics review* *Expectations:* Documentation is correct and sufficient *Findings:* Additional photo documentation should be implemented

Figure 2 - First configuration

5.2.1 Rationale

The “Rationale” row is concerned with the question “Why do we, spend time on trying to solve this problem?”.

Paradigm

The purpose of the “paradigm” column is to understand the challenge and the problem domain from an outside perspective.

The first cell in the “paradigm” column is meant for describing the “Challenge” which is the overall purpose of the project. The challenge is backed by the “Problem” which describes why the challenge is relevant.

The challenge of this project is to allow electricians to document their installations from their smartphone. This challenge comes from the fact that the current documenting process is too long and too resource demanding. The current process has three steps:

1. Document the installations with pen and paper in the field as they are installed
2. Rewrite the documentation on a computer into I/O Papers
3. Document the installations into a CAD program – This process is usually not done by the electrician themselves as they do not have training in this.

Product

The purpose of the “product” column is to describe and explore the solution proposal.

The first cell in the “product” column is meant for mapping and describing the technologies that will be a part of the proposed system solution. The key technologies should be focused on solving the “challenge” described in the “paradigm/rationale” cell.

In this configuration I have only narrowed the technologies down to the following:

- Neo4J Graph Database system – the reason why I have chosen graph databases are, that they are especially good for handling data which is tightly connected and this is the case for this application. For example, a component is connected to a location, which is connected to a room, which is connected to a building, which is connected to a project and a user is connected to a project.

- Touch Screen Interface – the strength of the application is that it works on a handheld device. And since most people have some sort of smartphone now, it makes sense to focus on touch screen based devices.

Project

The purpose of the "project" column is to reason for the choices made in the "Product" and "Paradigm" columns.

The first cell in the "project" column is used to describe the overall vision of the project whilst providing reasoning as to why the problem should be solved. This cell is strongly influenced, and backed by the "paradigm/rationale" cell.

For this configuration I have given the application a name "Connexion". The vision is to allow the electrician to easily document their work in order to save time. By providing an application that allows them to document their work on the job site, I prevent that they postpone the documentation work and thereby help to lower the amount of installations that are not documented.

Process

The purpose of the "process" column is to evaluate the process of the project.

The first cell in the "process" column is used to express outstanding questions in the beginning of the iteration related to the "rationale" row. These are expectations that are critical for the solution to be realized. In the end of the iteration, the answer for the expectation should be described under "Findings". The findings will be critical for the following iteration.

Expectations

In the beginning of this iteration, I had an expectation, that component suppliers are willing to share detailed information regarding their components in a format that is easy to add to a database. The database of components is critical in order to provide the users with a list of components that they can choose from when they are setting up components in a building. A strategic partnership like this indicates that I am using principles from the effectuation paradigm.

Findings

In the end of this iteration I have found, that it is not possible to get information regarding components from the supplier. This means, that I need to find an alternative way to provide that information to the customer. This problem will be passed on to the next iteration, and it will be critical for the further process.

5.2.2 Strategy

The “Strategy” row aims to answer the question “What should we develop in order to solve the problem?”.

Paradigm

The second cell in the “paradigm” column is meant for mapping the objects of the system solution.

In the first configuration I have defined the following objects:

- User – this object contains information belonging to the users (electricians) of the system like email, name and password
- Project – this object contains information regarding a current project that a user is working on like name of the project and location of the project
- Building – this object contains information about a number of building that belong to a certain project. The purpose of this object is to allow the electricians to locate components in an installation.
- Room – this object contains information regarding a number of rooms that belongs to a certain building in a certain project. The purpose of this object is to further narrow down the specific location of a certain component.
- Location – this object contains information about location inside a certain room. Again this object aims to further narrow down the location of a component.
- Component – this object is the backbone of the application. The component object contains detailed information belonging to the various types of components in a project. This information includes schematics, datasheets and information about the supplier of the component in the event of technical problems or need for support.

- IO – this object belongs to the component object. It provides information about the various Inputs/Outputs of a component. This information includes installation information, voltage level etc.

These objects do not necessarily reflect all objects that will eventually be in the application.

Product

The second cell in the "product" column describes the "key components" of the system.

- On-site documentation – this component allows the electrician to document their installations while they are on the job site. This component maps to the "Setup of components..." and the "Editing existing setup…" key scenarios.
- Overview of existing installations – this component provides a view where the electrician can get a structured overview of how the various components are connected. This can be in the form of I/O papers or some other graphical representation.

Project

The second cell in the "project" column is called "Justification" and is used for arguing how the proposed product solves the observed problem. It is tightly coupled with the "paradigm/rationale" cell. The justification cell is modelled after Toulmin's argumentation model and includes the "Backing", "Qualifier" and "Rebuttal" (Aaen 2016). The backing aims to provide further arguments in line with the "Warrant" from "Project/Rationale", explaining how and why the proposed solution solves the observed problem. The qualifier describes potential cases where the product does not solve the problem. The Rebuttal argues why the product solution is still viable despite the problems raised in the qualifier.

- Backing - "The system makes it easy to document installations." This adds to the warrant because it doesn't only point to the fact that the proposed product solution will not only help electricians document their work. It will help them to do so in a very easy and time saving manor.
- Qualifier - "Requires internet connection." This qualifier is important because the application cannot work without internet connection.

- Rebuttal - In most cases the qualifier is not in play since most people today have phones with internet access.

Process

The second row in the “process” column is used to express expectations and findings related to the “strategy” row.

Expectations

My expectations for this iteration is, that my understanding of the problem is good enough, and that the two components that I have defined are sufficient in order to create a product that the user is willing to buy.

Findings

In the end of this iteration I have found, that the two components that I have defined are not enough. I need to add an additional planning module, which will allow the user to define how a setup should be installed before they leave the office. This way, when they get to the job site, they just need to follow the instructions provided by the application.

5.2.3 Tactics

The “Tactics” row aims to answer the question “How do we develop the product that will solve the problem?”.

Paradigm

The third cell in the “paradigm” column is meant for describing the key scenarios that the system should support in order to completely understand the problem domain.

For this configuration I have defined three key scenarios:

Setup of components inputs/outputs, and mapping them to specific locations in a room. This scenario is the most important scenario of the application, and the one that directly solves the challenge described in the Paradigm/Rationale cell. This scenario allows the electrician to attach components to a specific location in a room. Once a component is mapped to a location, the components I/O’s can be connected and the electrician can write comments to that specific

connection if necessary. The information that is collected doing this process is what makes up the documentation in the end.

Get an overview of what is connected and how.

This scenario has multiple applications since it can both be used for figuring out what components the electrician needs to bring to the job site, but can also be applied when he need to figure out how the current installations are installed, which will save him a lot of time on investigating the existing setups.

Product

The third cell in the "product" column describes the "key features" of the system. The key features should reflect necessary features to solve the challenge described in "paradigm/rationale".

The key feature of the application is, that it allows the electrician to create documentation from their smartphone at the job site.

Project

The third cell in the "project" column is used for arguing how the various system components solves the observed problem. This cell is tightly coupled with the "product/strategy" cell and the "paradigm/rationale" cell.

The On-Site module solves the observed problem by providing a tool which allows the electrician to document their work while they are at the job site, hence making it a one-step process as opposed to the old three-step process.

The Overview module provides a convenient way for the electrician to get an overview of how an existing installation is set up, which saves them a lot of time figuring that out.

Process

The third cell in the "process" column is used to express expectations and findings related to the "tactics" row.

Expectations

In this iteration my expectations were, that my understanding of the problem domain was good enough, and that the elements for the documentation information was sufficient to provide proper documentation.

Findings

My findings for this iteration is, that in addition to the mapping of connections between components' inputs and outputs, an additional option to upload photo documentation should be implemented. The photo documentation serves as an extra checking mechanism in order to check that the components was actually setup according to the documentation.

5.3 BMC PREFACE

The purpose of the Business Model Canvas is, to create a business model which is easy to read and edit. It makes it very easy to get an overview of the most important concepts in the business model. The contents of the Business Model Canvas are strongly influenced by the configuration tables. The connections between the configuration table and the Business Model Canvas will be described later in the "Mappings" section.

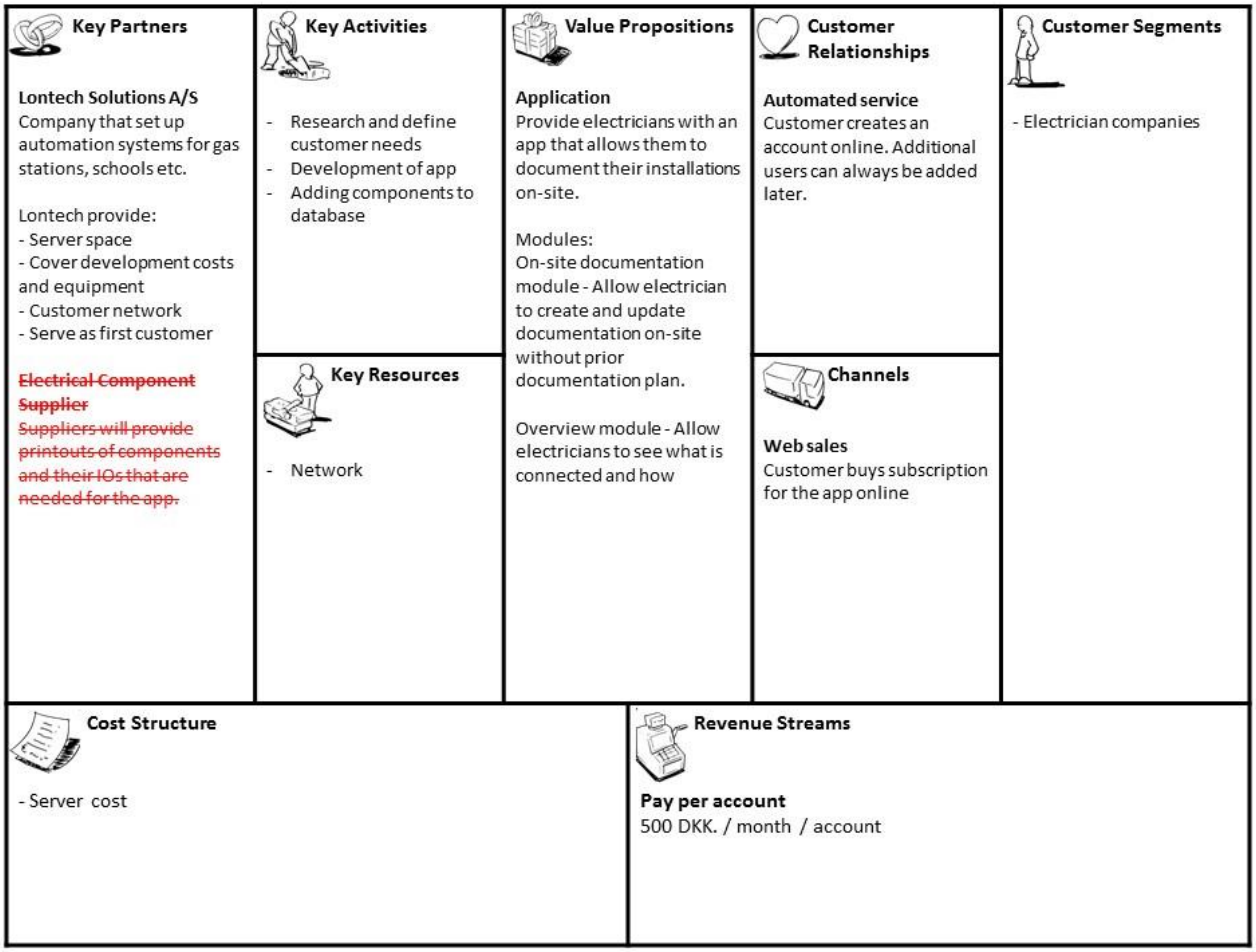


Figure 3 - First Business Model Canvas

5.3.1 Key Partners

The "Key Partners" cell of the business model canvas is used to describe important partners who have an impact on the business model. For example, this column can be used to describe a strategic partnership where two companies utilize each other's domain knowledge or technical knowledge.

Lontech Solutions A/S

Lontech Solutions A/S is a company located in Struer, Denmark. Their primary business evolves around setting up automation systems in schools, gas stations, companies etc. Lontech have been struggling with the task of documenting the setups that the do. At the moment they write them down on a piece of paper, and the rewrite them when they get back to the office. Lontech had the initial product idea, and will serve as the new company's (New Company) first customer. They will also provide server space and cover development costs.

Electrical Component Supplier

In order for the system to work, an extensive database of electric components and their specifications will be needed. Therefor electrical components suppliers like Schneider Electrics, Solar etc. will serve as partners to provide those data.

The continues use of strategic partners indicates that I am using principles from the effectuation paradigm.

5.3.2 Key Activities

The “Key Activities” cell is used to describe the most important activities that the company performs in order to create value for the customer.

New Company’s “Key Activities” will be to develop the documentation app. In order to do this, the customer’s needs should be established alongside requirements for proper documentation. Also the development and population of the components database will be a key activity.

5.3.3 Key Resources

The “Key Resources” cell is used to describe the most important resources that the company possesses. Resources can be anything from “customer network” to “technical knowledge” or “production setups”.

New Company’s “Key Resources” are primarily the network through Lontech Solutions A/S who have connections to electricians, suppliers and customers.

Also New Company’s skills in software development is a key resource.

5.3.4 Value Proposition

The “Value Proposition” cell describes how the company’s product creates value for the customer.

New Company’s “Value Proposition” is to provide electricians with a tool, that allows them to easily and fast document their work in the field. The system provides two modules that allow them to do that.

On-site document/update module

The on-site module allows the electrician to document their installations as they install them in the field. When using this module, no prior planning needs to be done. They can also use this module to update already existing documentation.

Overview module

The overview module provides a graphical presentation of the documentation for a particular project.

5.3.5 Customer Relationship

The “Customer Relationship” cell describe how the company’s relationship to their customers are. This includes the form of service that the company provides e.g. “automated service”, “personal assistance” or “self-service”.

The “Customer Relationship” is “Automated service” this means, that the customer can create an account, and review their subscription online.

5.3.6 Channels

The “Channels” cell is used to describe what distribution channels are used for selling the product. The channels can be anything from “web sales” to “wholesaler”.

The “Channels” are primarily “Web sales” this means that the customer buys the product through a webpage.

5.3.7 Customer Segments

The “Customer Segments” cell is used to describe the company’s main customer groups.

The “Customer Segments” are electricians and their companies.

5.3.8 Cost Structure

The “Cost Structure” cell is used to describe the various costs that the company has.

The “Cost Structure” is simple. The product is software, so the only costs related to the company is the server hosting of the application and database. Lontech Solutions A/S has provided us with server space, so the costs will be 0 DKK.

5.3.9 Revenue Streams

The “Revenue Streams” cell is used to describe the pricing model that the company uses to sell their products. For example, the revenue streams can be subscription based or a one-time payment.

The “Revenue Streams” are subscription based. Electrician companies need to buy a subscription for each employee they want to be able to document setups. I estimate that a price of 500 DKK / month is a fair price compared to the price of licenses for CAD software etc.

5.4 MAPPING PREFACE

The purpose of this section is to create a mapping between the configuration table and the business model canvas in order to explain how the changes made in one of them will affect the other.

In order to make the mapping easier to comprehend, I have included simplified versions of the configuration table and the business model canvas.

Configuration table

	Product	**Project**
Rationale		*Vision* Connexion - easily create and edit documentation for electrical installations
Strategy	*Key components* - On-site documentation of installations - Overview of existing installations	

Business Model Canvas

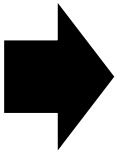
Value Propositions
Application Provide electricians with an app that allows them to document their installations on-site. Modules: On-site documentation module - Allow electrician to create and update documentation on-site without prior documentation plan. Overview module - Allow electricians to see what is connected and how

Figure 4 – Mapping 2 - Configuration 1 / BMC 1

The value proposition in the Business Model Canvas was strongly influenced by two cells in the configuration table. The “Vision” creates the foundation, and combined with the “Key components”, it adds up to a strong and detailed value proposition.

Configuration table **Business Model Canvas**

	Process
Rationale	*Expectations:* Component suppliers are willing to provide us with data about components and their IO's



Key Partners
~~**Electrical Component Supplier**~~ ~~Suppliers will provide printouts of components and their IOs that are needed for the app.~~

Figure 5 - Mapping 2 - Configuration 1 / BMC 1

My initial ideas were to add "Electrical Component Suppliers" as partners in order for them to provide me with information regarding their components. But during this phase I have realized, that it is not feasible to have the component suppliers as partners. Therefor I have remove them from the Business Model Canvas.

5.5 Product backlog

In order to start the development process, I need to define the product backlog as a list of tasks to be performed. I will prioritize the backlog items in order to know where to start.

Name	Description	Priority
APIs	Create/read/update/delete: users, projects, buildings, rooms, locations, components	1
Login function	Allow users to log in to the system	2
Create project	Allow users to create a new project	3
Add building to project	Allow users to add a building to an existing project	4
Add room to building	Allow users to add a room to an existing building	5
Add location to room	Allow users to add a location to an existing room	6
Setup component in room	Allow users to add a component to a location and connect inputs/outputs	7
Create user	Create user function	8

5.6 Technology Research

In order to ensure a fast an effective development process, I have to carefully select which technologies to use for the application. My primary programming skills lie within the field of web development, and also a web based application will allow users to use the application no matter which platform they are using, so I am going to develop a website optimized for smartphones. The research is divided into two categories namely "Frontend" and "Backend". I am using the "means" at my disposal. This indicates that I am using effectuation.

5.6.1 Frontend

There are various good technologies to choose from when deciding which frontend framework to use. For this application I have chosen to use a JavaScript library called AngularJS alongside an AngularJS extension called Angular Material. These technologies are great for rapid prototyping.

AngularJS

AngularJS is a JavaScript library developed by Google meant for building single-page applications. This means, that it is possible to run an entire app without reloading the page. This makes it great for mobile webpages where the user potentially has a slow internet connection (Hevery 2016).

AngularJS uses a development pattern called MVC (Model, View, Controller). This is a structure, that makes it possible for the developer to keep those three aspects of an application separated. "Model" represents the data structure of the application, this part will handle the connection to the database and the logic behind the data. The "View" handles everything that the user sees on the screen, when dealing with webpages this is the HTML part of the application. The "Controllers" handles the logic behind the application – an example is that the controller decides what to do when a user performs a specific action.

Another great feature of AngularJS is something called two-way data binding. The way it works is, that by binding a certain property in Angular to for instance a text field in HTML, angular will automatically update the property in Angular once it is updated in the text field. This saves the developer for writing a lot of code in order to fetch the data from the HTML elements.

Angular Material

Angular Material is an AngularJS extension library specifically designed for developing mobile webpages. It is basically a layout standard based on Google Material Design standards that ensures that all elements on the webpages appear as it would in a native android application.

Angular Material provides elements like grid layout systems, icons, menus, form elements, toolbars and lists.

5.6.2 Backend

Since I am dealing with highly connected data, I have chosen to focus on graph databases. A graph database stores data in "Nodes" which can have multiple "properties", Nodes also have something

called “Labels” that describes which type of node it is. Nodes are connected with lines or arrows called “Edges” which also have labels and properties.

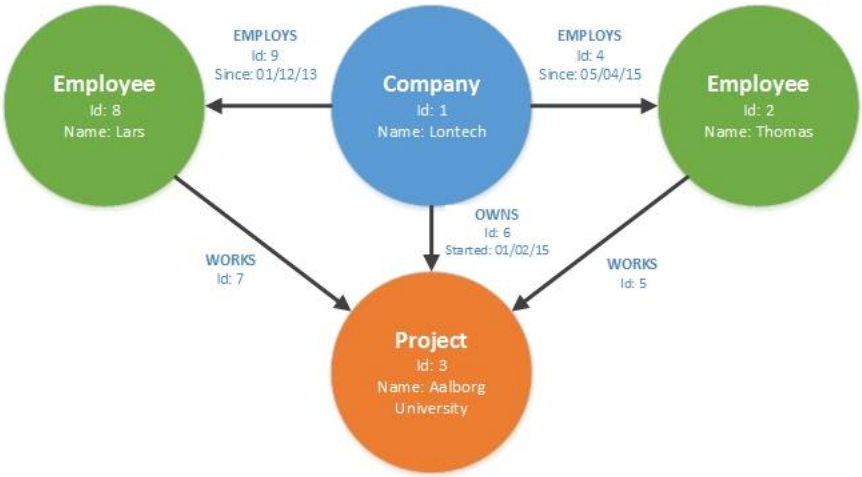


Figure 6 - Graph example

Figure 6 is an example of a very simple section of a graph database. The circles represent “Nodes” and the big text inside the node is the “Node label”. Underneath the labels are the “Node properties”. The arrows connecting the nodes are “Edges”, and again the big text is the “Edge label” and underneath it is the “Edge properties”.

Neo4J

For the application backend I have chosen to use Neo4J Graph Database system for data storage. The reason for why I have chosen Neo4J is, that it is currently the most widely used graph database system in the world. Other than that, Neo4J uses a very human readable query language called “Cypher” (Eifrem 2016).

```
MATCH (n) RETURN n
```

Figure 7 - Cypher query – match all

Figure 7 is an example of a Cypher query in its simplest form. This query matches all nodes in the database and returns them. The brackets represent a node, and whatever is inside the brackets becomes the reference to that node.

```
MATCH (thomas:Employee {name:"Thomas"})-[:WORKS]->(project:Project)
RETURN thomas, project
```

Figure 8 - Cypher query – match path

Figure 8 is an example of how Cypher allows you to match a path in a very easily understandable way. The first bracket matches a node with the label "Employee" and the reference to that node is "thomas". The curly brackets are used to restrict the match to employees who have the "name" attribute set to "Thomas". The dash after the closing bracket represents an outgoing edge, and the square brackets restricts it to an edge with the label "WORKS". The arrow after the closing square bracket means that it will match a directed relationship towards a node with label "Project". If we consider the example in Figure 6, the database will return the "Employee" node with name "Thomas", and the "Project" node with name "Aalborg University" (Figure 9).

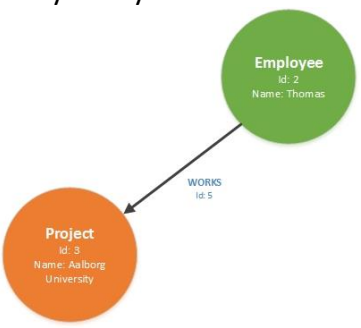


Figure 9 - Query result

Laravel PHP Framework

JavaScript and therefor AngularJS is a client side programming language. This means, that AngularJS cannot communicate with the server and therefore cannot communicate with the database. Therefore, I need to develop a series of CRUD (Create, Read, Update, Delete) APIs (Application Programming Interface) in order to communicate with the database. An API will allow me to manipulate the data in the database through HTTP requests.

For this purpose, I have chosen to use Laravel which is a popular and very powerful PHP framework for building MVC applications in PHP. I will be using Angular as frontend for my application, so I will only be utilizing the "model" and "controller" part of Laravel to build my APIs. What makes Laravel powerful is its "Eloquent" package. Eloquent makes it really easy to communicate with and manipulate database entries through PHP classes.

```
$user = new User();
$user->name = "Thomas";
$user->save();
```

Figure 10 - Laravel Eloquent example

The example in Figure 10 will create a new object of type "User", and when the "save" function is invoked, it will save it to the database.

Out of the box Laravel is fit to handle many different database types like “MySQL”, “Postgres”, “SQLite” and “SQL Server”, but Neo4J graph databases is not one of them. In order to use Neo4J with Laravel I need to install a new database driver called NeoEloquent. NeoEloquent is built specially to Laravel, and will allow me to use Neo4J exactly how I would MySQL from Figure 10.

```
$company = new Company();
$company->name = "Lontech";

$user = new User();
$user->name = "Thomas";

$company->users()->save($user);
```

Figure 11 - Laravel Eloquent example with relationship

Figure 11 is an example of how I would use Laravel and NeoEloquent to create the “Company” and “User” nodes from the example in Figure 6 and connect them with an edge. NeoEloquent will name the edge label according to what is specified in the “Company” class.

```
class Company{

    public function users(){
        return $this->hasMany('User', 'EMPLOYS');
    }

}
```

Figure 12 - Laravel Eloquent class definition

Figure 12 is an example of how a class should be set up according to the conventions of Laravel. The “hasMany” function is one of the three main relationship types that can be used. The other two are “hasOne” and “belongsTo”. The first parameter of the function is the name of the class that it should connect to – in this example “User” and the second parameter is the label of the edge.

```
$company = Company::find(1);
$users = $company->users();
```

Figure 13 - Laravel Eloquent retrieve users

Figure 13 is an example of how to find a Company node by its id and then retrieving a list of users associated with that company. The list of users will be in the form of an array of objects with the type “User”.

6 ITERATION 1 – ENVIRONMENT AND APIS

This iteration aims to partly develop the product described in the configuration table from section 5.2. But before the actual application can be developed, the backend system needs to be in place. Therefore, this iteration is dedicated to setting up the development environment described in section 5.6 (Technology Research), and develop the application backend.

6.1 OBJECT MODEL

From the use cases described in section 5.1, it is possible to identify the objects by picking out the nouns from the scenarios. The nouns that are going to become objects are marked with a blue underline. The nouns that are marked in green will most likely become attributes in some objects. Not all of the nouns are marked, this is because experience tells me that there will be no object called “system” and “credential ls” etc. The nouns that are going to become objects are “User”, “Project”, ”Building”, “Room” and ”Location”. The attributes are “Name”, “Email”, “Password” and “Geo location”.

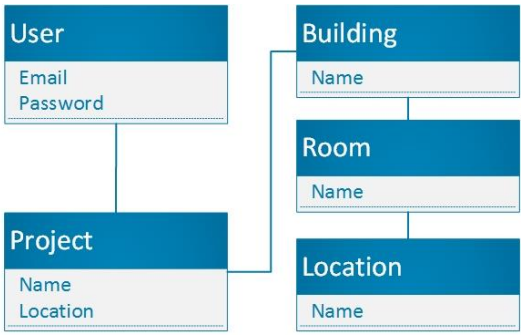


Figure 14 - Object model

Figure 14 shows the object model derived from picking out the nouns from the use cases. The attributes are also derived from the use cases.

6.2 SPRINT PLANNING

In order to be able to get and manipulate data from AngularJS, a variety of APIs must be developed. For this iteration I have decided to only cover the “GET” and “POST” variations of the APIs. “GET” and “POST” refers to the HTTP nouns used for getting and creating records in the database. The records are namely “User”, “Project”, “Building”, “Room” and “Location”. The system knows which

record is called on by looking at the URL that is requested. As an example if a request URL is "http://localhost/api/v1/users", and the HTTP noun is "GET", the API should return a list of all users. On the other hand, if the noun is "POST", it indicates that the user wants to add a new record to the database, and in that case, a data object should be passed in the request body containing the user data.

Noun	URL	Request body	Response body
		Users	
POST	/users	email, password, name	User object
POST	/users/login	email, password	User object, token
		Projects	
POST	/users/{user_id}/projects	name, location	Project object
GET	/users/{user_id}/projects		Array of project objects
GET	/projects/{project_id}		Project object
		Buildings	
POST	/projects/{project_id}/buildings	name	Building object
GET	/projects/{project_id}/buildings		Array of building objects
GET	/buildings/{building_id}		Building object
		Rooms	
POST	/buildings/{building_id}/rooms	Name	Room object
GET	/buildings/{building_id}/rooms		Array of room objects
GET	/rooms/{room_id}		Room object
		Locations	
POST	/rooms/{room_id}/locations	Name	Location object
GET	/rooms/{room_id}/locations		Array of location objects
GET	/locations/{location_id}		Location object

Figure 15 - RESTFUL Api's

When using the Laravel framework, it will probably not take that long time to develop the APIs. I estimate that it will take longer time to set up the Laravel environment, and modifying it to use Neo4J instead of using the standard database drivers. Therefor I have set off three days for this sprint, and plan to use two of the days on setting up the environment and just one day for creating the APIs.

Task number	Task	Estimated time (hours)
1	Setup development environment	14
2	Users API	2
3	Projects API	2
4	Buildings API	2
5	Rooms API	2
6	Locations API	2

Figure 16 - Product backlog

6.3 SPRINT REVIEW

Through this iteration, the database has been set up, and the API's described in Figure 15 have been developed.

I have updated the configuration table according to the findings of this iteration, and that configuration will form the foundation for the updated product backlog which in turn will facilitate the next iteration. Based on the updated configuration table, I have updated the business model canvas.

6.3.1 Database structure

The database is the outcome of this iteration. Underneath I have included a screenshot of the database as it looks in the Neo4J database management system with some "dummy data". Note that this screenshot only displays the name and type of the various nodes. Several of the nodes have properties which are not displayed. The Neo4J database management system does not allow unlimited colors to be used which is why the "Project" nodes and "Location" nodes are the same color.

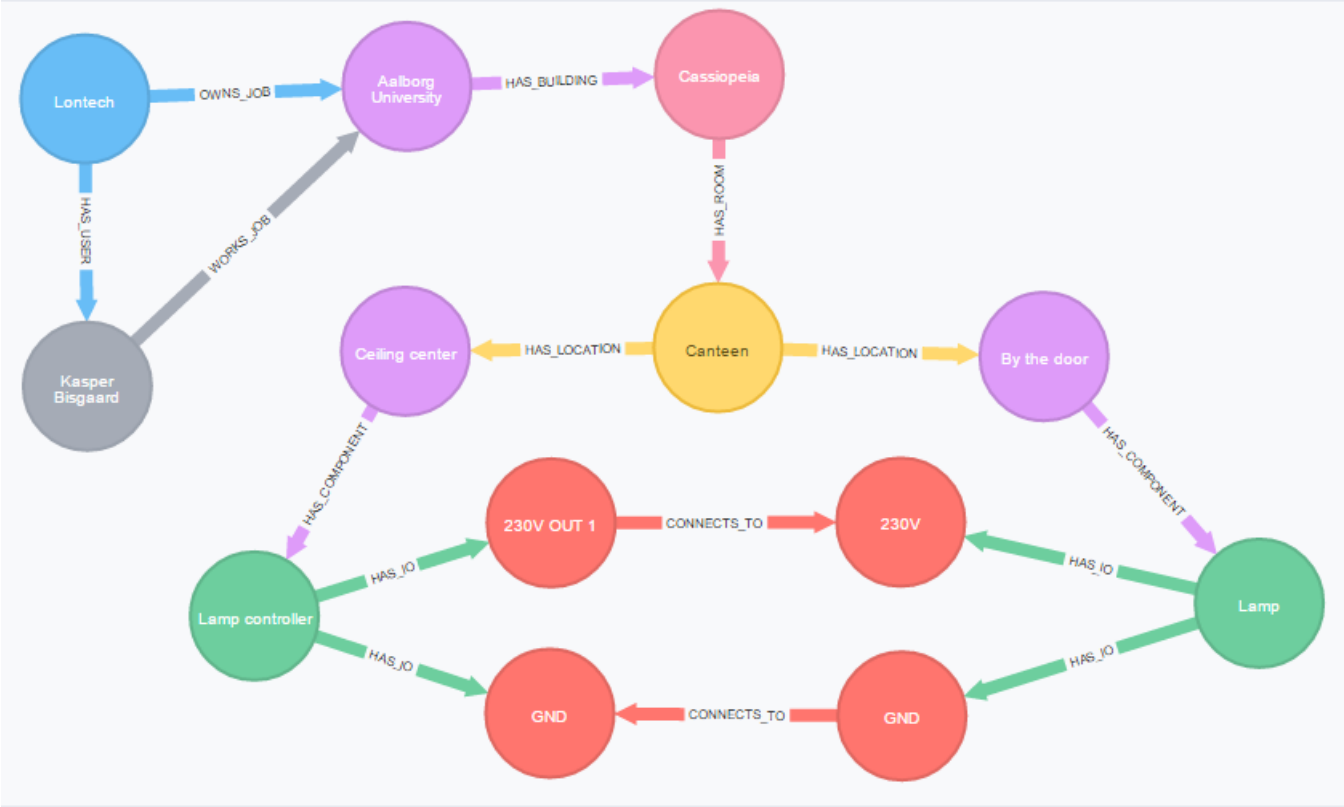


Figure 17 - Screenshot of database

The database displays a “Company” called “Lontech” (blue node) which leads to two other nodes: A “User” (grey node) and a “Project” node (purple) called “Aalborg University”. The project node has one “Building” associated with it named “Cassiopeia”. The building node has one “Room” node called “Canteen” (yellow) which have two “Location” nodes attached (also purple). Each of the two locations have a component attached to them (green) and each of those components have two “Input/outputs” (orange). The “Input/outputs” are connected according to the figure.

6.3.2 Configuration iteration 1

View	**Paradigm**	**Product**	**Project**	**Process**
Value	*Reflection*	*Transaction*	*Reasoning*	*Appreciation*
Rationale: Why?	*Challenge* Allow electricians to document their installations from their smartphone *Problem* The process of documenting electric installations is too long and resource demanding	*Key technologies* - Neo4J Graph Database - Touch screen interface - Laravel PHP Framework - AngularJS - Angular Material	*Vision* Connexion - easily create and edit documentation for electrical installations *Warrant* Good documentation can save the electrician from spending hours on figuring out how a system is set up.	*Rationale review* *Expectations:* It is possible to make standard templates for all kinds of components *Findings:* *It is possible to some degree, but slight variations can occur.*
Strategy: What?	*Key elements* Objects: User, Project, Building, Room, Location, Component, IO, Photo	*Key components* - Installation planning from office - On-site documentation of installations - Overview of existing installations	*Justification* *Backing:* Simple system makes it easy to document installations. *Qualifier:* Requires internet access *Rebuttal:* Most people have phones with internet access	*Strategy review* *Expectations:* It is possible to plan installations from the office. *Findings:* It is possible as long as up-to-date documentation exists already.
Tactics: How?	*Key scenarios* - Planning component setups from the office. - Setup of components inputs/outputs, and mapping them to specific locations in a room - Editing existing setup of connected components. - Get an overview of what is connected and how.	*Key features* - Create documentation from smartphone - Add photo to documentation - Edit existing setups	*Key mapping* - Planning module enables the user to plan installations from the office. - On-site module enables user to document installations while setting them up on the job site. - Overview module gives the user an idea of how the existing installations are set up.	*Tactics review* *Expectations:* Documentation is correct and sufficient *Findings:* Documentation is sufficient to substitute the IO papers that are currently used by Lontech Solutions A/S

In the following section, I will describe the considerations tied to the changes that I made to the configuration above. The changes are marked with yellow.

Product/Rationale

For this configuration I have added three key technologies to the application. I made this decision based on the technology research I made in the preface section. I will be using Laravel PHP Framework for creating the data API's. I will use AngularJS for the frontend development, and I will include the Angular Material design library in order to get a native look and feel to the application.

Project/Rationale

For this configuration I have added a word to the vision in order to describe that it should be possible to edit already existing documentation through the application.

Process/Rationale

Expectations

My expectations for this iteration derives from the findings of the previous iteration, that I cannot count on component suppliers to provide me with information regarding the various components. My solution for this is, to create a variety of component templates, that will fulfill the user's needs therefor my expectation is, that it is in fact possible to create standard templates that take all variations into account.

Findings

My findings are, that this is possible to some degree, but variations can occur. This means, that I cannot be completely sure, that it will fulfill the customers' needs as they may have components from china that vary in e.g. The placement of inputs/outputs or naming of these.

Process/Strategy

Expectations

In the previous iteration I found, that an additional function to plan installations from the office should be added to the application. Therefor my expectations for this iterations I, that this is in fact possible to do without actually being at the job site.

Findings

In this iteration I found, that it is in fact possible to plan installations from the office. But it requires, that proper documentation already exists. The reason for this is, that electricians cannot possibly figure out how to setup components if they do not know how the existing setups are installed.

Product/Strategy

For this configuration I have included a module which allows the electrician to plan their installations from the office. This way they will only use the application as a reference tool once they are at the job site. This will allow a senior electrician to plan the setups in a detailed way so that a junior electrician can setup the components at the job site and just use the application as a reference tool. Once the junior electrician has setup the component as described in the setup plan, he takes a picture for documentation and checks off that installation. Once an installation is completed the next installation plan will be shown.

Paradigm/Tactics

Since I have added a planning module as per “product/strategy” and “process/strategy”, this scenario has to be added to the key scenarios.

I have also added an “Editing existing setup of connected components” scenario. This scenario addresses the problem where the electrician needs to setup new components in a building where the documentation tool has already been used. It will allow the electrician to switch out components for new versions or make changes to the way the input/outputs are set up.

Process/Tactics

Expectations

My expectations for this iteration is, that the documentation that my application provides is in fact sufficient to substitute the documentation that is currently being created.

Findings

In this iteration I have found, that the documentation that my application provides is sufficient to substitute the documentation process that is being used by Lontech Solutions A/S.

Generel information		Placement of controller	Ventilations Skab Teknikrum	Controller name	2	Neuron ID	07001103EF00
Input	**Component**	**Signal type**	**Input state**	**Component ID**	**Placement of Unit**	**Comment**	**Test**
IP1 Gnd	VF Varmeføler Indblæs	PT1000					EJ/ RS

Figure 18 - IO table from Lontech A/S

The table in Figure 18 shows a sliced section of an Input/Output table that is currently being used by Lontech Solutions A/S. The table shows an example of a "Controller" with the name "2". A temperature sensor is connected to "IP1" and "GND" of the controller, and it also displays the "Signal type".

Product/Tactics

For this configuration I have added three new key features.

The first feature "Add photo to documentation" came from the findings in "process/tactics". The feature allows the user to take a picture of the setup they are working on. This serves as an addition to the digital mapping of input/outputs, and will help other electricians to understand the setups. An electrician can attach as many photos as are needed to a setup.

The second feature "Edit existing setups" is meant for electricians who are installing new components in a building where the documentation tool has already been used. This will allow the electrician to rearrange inputs/output connection or add additional information to the setup.

Project/Tactics

Since I have added a planning module, an additional key mapping has to be added. The key mapping describes how the planning module solves the problem observed in "process/strategy".

6.3.3 BMC iteration 1

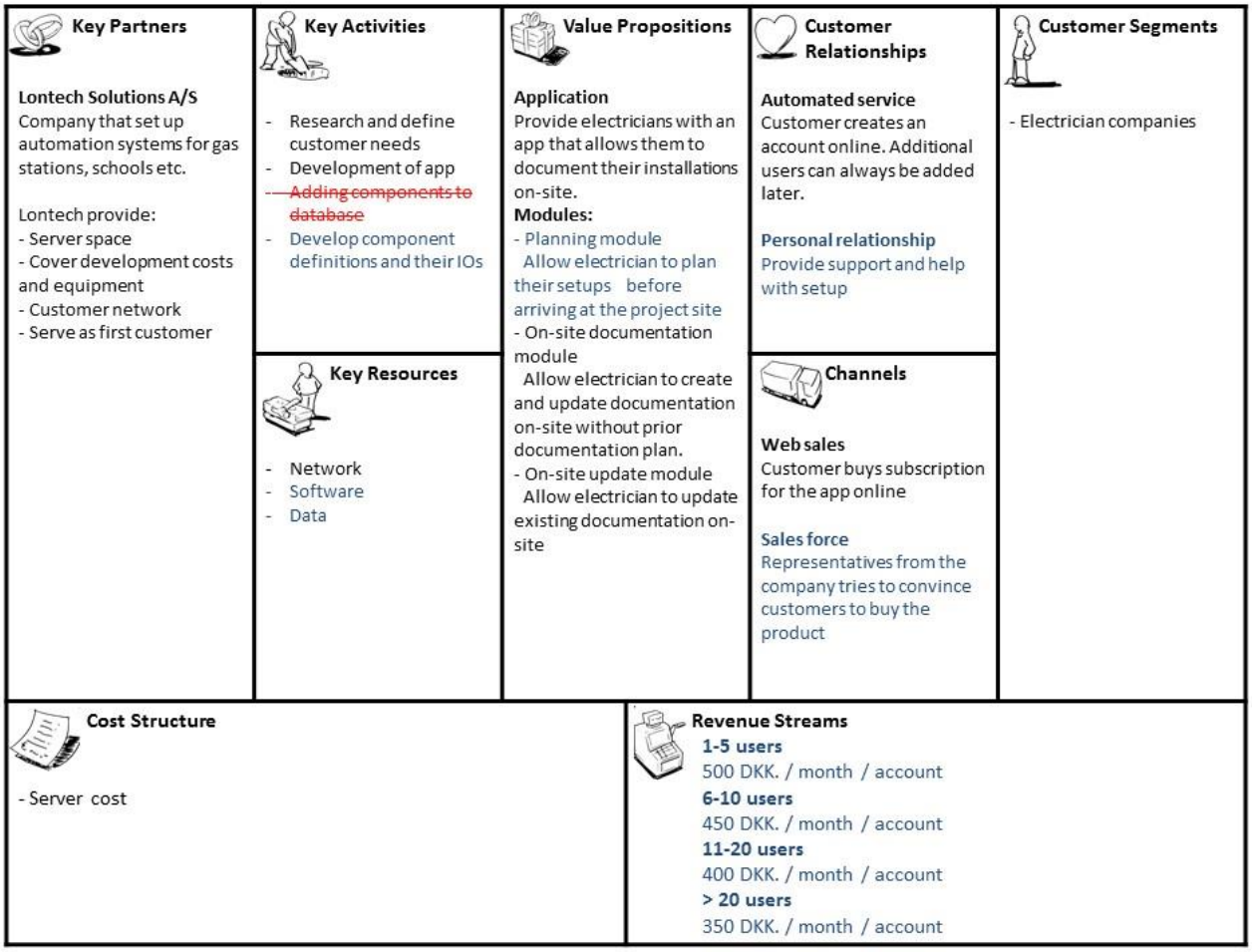


Figure 19 - BMC 2

Key Partners

As per the findings from configuration 2 ("process/rationale") I have removed Electrical Component Suppliers from the "key partners". Since I do no longer need information regarding electrical components, it makes no sense to include the suppliers as key partners.

Key Activities

Since I have decided to remove the suppliers from the business model, I no longer have access to their components. Therefor I have removed the "Adding components to database" activity in favor of a "Defining components and their IOs" activity. This will also include adding them to the database.

Key Resources

For this iteration I have added “Software” and “Data” as key resources. The software itself may not be that hard to duplicate for an experienced programmer, but the component data may be.

Value Proposition

As per the changes made in configuration 2, I have to add an additional module to the value proposition. The “Planning module” allows the electrician to plan their work from the office. The electrician can go through existing documentation for a project, and thereby exploring the best way to setup their products from the office. When they get to the project site, they can use the plan as a template for setting up the components.

Customer Relationships

I have decided to add an additional item to the customer relationships. Personal Relationship provides a closer connection to the customer, and allows me to further develop the application with the customer’s inputs.

Channels

As an addition to web sales I have decided to add “Sales force” to the channels. This allows me to contact the big electrician companies in order to convince them to buy my product.

Revenue Streams

In this iteration I have decided to revise the pricing mechanism. The reason for this is, that it should be more attractive for the companies to buy a big amount of accounts. Therefore, I have decided to take 500 dkk. for 1-5 accounts, 450 dkk. for 6-10 accounts, 400 dkk. for 10-20 accounts and 350 dkk. for anything above 20 accounts.

6.3.4 Mapping 1st iteration

Configuration table

	Process
Rationale	*Expectations:* It is possible to make standard templates for all kinds of components *Findings:* *It is possible to some degree, but slight variations can occur.*

Business Model Canvas

Key Activities
~~- Adding components to database~~ - Develop component definitions and their IOs

Figure 20 - Mapping Process/Key activities

Based on the findings in the "Process/Rationale" cell of the configuration table, I have deleted the "Adding components to database" activity from the "Key Activities" cell in the BMC. Instead I added an activity "Defining components and their IOs".

Configuration table

	Product
Strategy	- Installation planning from office

Business Model Canvas

Value Proposition
- Planning module Allow electrician to plan their setups before arriving at the project site

Figure 21 - Mapping Product/Value Proposition

In the configuration table I have added "Installation planning" to the list of "Key Components" therefor I have also added it to the Value Proposition in the Business Model Canvas.

6.3.5 Revised product backlog

Name	Description	Priority	Status
APIs	Create/read/update/delete: users, projects, buildings, rooms, locations, components	1	Done
Login function	Allow users to log in to the system	2	Not started
Create project	Allow users to create a new project	3	Not started
Add building to project	Allow users to add a building to an existing project	4	Not started
Add room to building	Allow users to add a room to an existing building	5	Not started
Add location to room	Allow users to add a location to an existing room	6	Not started
Setup component in room	Allow users to add a component to a location and connect inputs/outputs	7	Not started
Create user	Create user function	8	Not started
Planning module	Allow users to plan installations from the office	9	Not started
Photo documentation	Allow users to attach a photo for more detailed documentation	10	Not started

Figure 22 - Product backlog

At this point I have finished the first item of the product backlog. I have completed all of the API's which will allow me to easily create the frontends for the application without having to think about how the data is read and stored in the database.

I have added two additional items to the product backlog:

Planning module

The planning module will allow the user to plan their installations from the office. This way they will only be using the application as a reference tool once they are at the job site.

Photo documentation

By allowing the user to add photo documentation to the database, it ensures, that the documentation is more complete and detailed. As an example, the electrician can study the documentation photo in order to figure out which color wire they have used, or to check if the setups was actually installed as described in the documentation.

7 ITERATION 2 – GUI'S

For this iteration I will continue from the use cases described in section 5.1. Last iteration was focused on creating the backend APIs for those use cases, so this iteration will be focused on creating the GUIs (graphical user interfaces).

7.1 SPRINT PLANNING

The sprint planning for this iterations involves designing the various user interfaces. In order to do that I have created some simple mockups that shows the most important elements of the UI's. In this iteration I have only included the GUI's for logging in, viewing projects, viewing buildings for a specific project, viewing rooms in a specific building and displaying location for a specific room. I have not included the GUI's for creating and editing those data since I at this point is only trying to provide a proof of concept. Therefor I have added some "dummy data" to the application.

7.1.1 Mockups

The first step of this iteration is to create the mockups for each of the applications GUIs. I have chosen a very simple tool for creating the mockups since I do not need a lot of details. The mockups are shown below, with a short description of the purpose of each screen.

Figure 23 - Login view

This view is used for logging in to the system. It displays an “email” input, “password” input and one login button.

Figure 24 - Choose project view

Once the user is logged in, he/she is presented with a list of projects that their company is currently working on.
The list displays the name of the project, the location and when the current user last worked on this project. The project that was last worked will be displayed in the top.

Once the plus icon is clicked (new project), a popup will be displayed prompting the user for a name and a location for the new project.

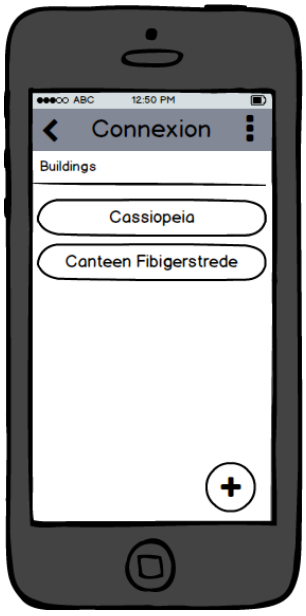


Figure 25 - Choose building view

Once a project is chosen, the user is presented with a list of buildings related to that project.

Once the plus icon is clicked (new building), a popup will be displayed prompting the user for a name for the new building.

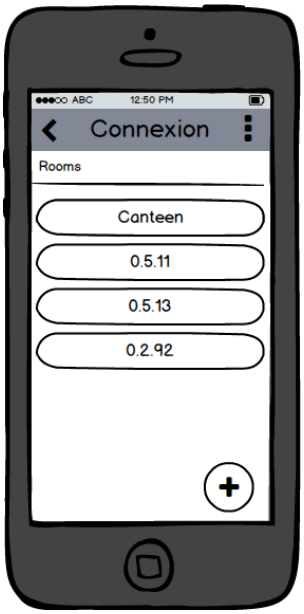


Figure 26 - Choose room view

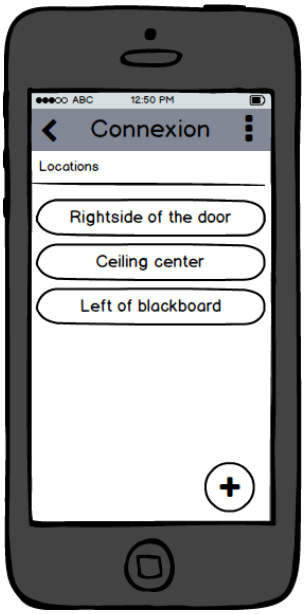


Figure 27 - Choose location view

Once a buiding is chosen, the user will be presented with at list of rooms related to that building.

Once the plus icon is clicked (new room), a popup will be displayed prompting the user for a name for the new room.

Once a room is chosen, the user will be presented with a list of locations related to that room.

Once the plus icon is clicked (new location), a popup will be displayed prompting the user for a name for the new location.

Since I already have the APIs setup and working, creating the GUIs should not take that long. Therefor I have decided to dedicate only two days for creating the five GUIs “Login”, “Project view”, “Building view”, “Room view”, “Location view”.

Task number	Task	Estimated time (hours)
1	Login view	5
2	Project list view	3
3	Building list view	3
4	Room list view	3
5	Location list view	3

Figure 28 - Sprint task list

I have chosen to set off five hours for the “Login view”. The reason for this is, that there is a bit more logic behind that functionality since I need to check if someone is trying to access a page without being logged in, and then redirecting them to the login page. Three hours for each of the remaining

views should be sufficient since I am using AngularJS which makes it very fast to create UIs if the APIs are already in place.

7.2 SPRINT REVIEW

Through this iteration I have created the GUIs described in section 7.1 (Sprint planning). I have revised the configuration table according to the findings of this iteration and some ideas for additional features. Based on the configuration table, I have revised the business model canvas, and I have added some points to the product backlog.

7.2.1 Configuration iteration 2

View	**Paradigm**	**Product**	**Project**	**Process**
Value	*Reflection*	*Transaction*	*Reasoning*	*Appreciation*
Rationale: Why?	*Challenge* Allow electricians to document their installations from their smartphone *Problem* The process of documenting electric installations is too long and resource demanding	*Key technologies* - Neo4J Graph Database - Touch screen interface - Laravel PHP Framework - AngularJS - Angular Material	*Vision* *Connexion - easily and quickly create and edit documentation for electrical installations* *Warrant* An app for documenting electrical setups can save the electrician from spending hours on figuring out how a system is set up.	*Rationale review* *Expectations:* *It is possible to allow the user to define component templates themselves if they do not find what they need.* *Findings:* It is possible because not that much information is needed to create sufficient documentation.
Strategy: What?	*Key elements* Objects: User, Project, Building, Room, Location, Component, IO, Photo	*Key components* ***Planning Module*** - Allow electricians to plan the setups before they leave the office. ***On-site module*** - Allow electricians to document their setups as they install them one at a time. ***Documentation View module*** - *Allow the electrician to see and print detailed documentation for the customer.*	*Justification* **Backing** - Web app allows everyone to view (and edit the documentation.) **Qualifier** - Requires internet access **Rebuttal** - Most people have phones with internet access	*Strategy review:* *Expectations:* The key components are necessary and sufficient to facilitate proper documentation *Findings:* A review module should be implemented to allow electricians to follow instructions based on a documentation plan.
Tactics: How?	*Key scenarios* - Planning component setups from the office. - Setup of components inputs/outputs, and mapping them to specific locations in a room - Editing existing setup of connected components. - Get an overview of what is connected and how.	*Key features* - Create documentation from smartphone - Attach documentation photo to certain documentation aspects *- Create component templates* *- Possibility for working offline*	*Key mapping* **Planning module -** enables the user to plan installations from the office. **On-site module** - enables user to document installations while setting them up on the job site. **Overview module** - gives the user an idea of how the existing installations are set up.	*Tactics review* *Expectations:* The documentation is sufficient to live up to legal legislation *Findings:* *?*

Figure 29 - Third Configuration

Project/Rationale

For this configuration I have changed the “warrant” so that it is focused around making a mobile application. This is not a change, but it makes the definition sharper.

Process/Rationale

Expectations

My expectations for this iteration derives from the findings of the previous, that it is only possible to define standard templates for components to some degree. Therefor I have decided to add a feature that allows the user to define templates themselves. They can also use an existing template as the foundation and make changes to that in order to define a new one. My expectation is, that it is possible to include functionality for this feature.

Findings

In this iteration I have found, that the expectation expressed above is a possibility. The information that is needed in order to define a component in order to create sufficient documentation is not that complex. Therefore, I have concluded, that this will in fact be possible for the user to do.

Product/Strategy

In this configuration I have given the key components descriptive names, and I have included a better description of the modules.

Project/Strategy

In this configuration, I have made the “backing” more explicit towards my choice of developing a web app. I have also included the possibility to edit documentation, to the description.

Process/Strategy

Expectations

In this iteration I assume, that the components that I have defined are sufficient in order to facilitate sufficient documentation functionality.

Findings

I have found, that in order to allow the users to setup components using a plan that was created from the office, I need to include a “review module”. The review module will enable the user to go

through all setups that needs to be installed one-by-one and checking them off and attach a photo once they are done.

Product/Tactics

For this configuration I have added the key feature “Create component templates” this decision derives from the findings in “process/rationale”. It should be possible for the user to define their own components and add the necessary inputs/outputs to the component. It should also be possible to attach documents like datasheets, and whatever else information is needed.
I have also added a feature “Possibility for working offline”. This feature will allow the users to use the application when they are in an area without internet connection.

Process/Tactics

My expectation from the previous configuration was that the documentation that the app can provide is sufficient to substitute the documentation process that is currently used. I have found, that the documentation provided by the app is sufficient to substitute the I/O papers that are currently used by Lontech Solutions A/S.

7.2.2 BMC iteration 2

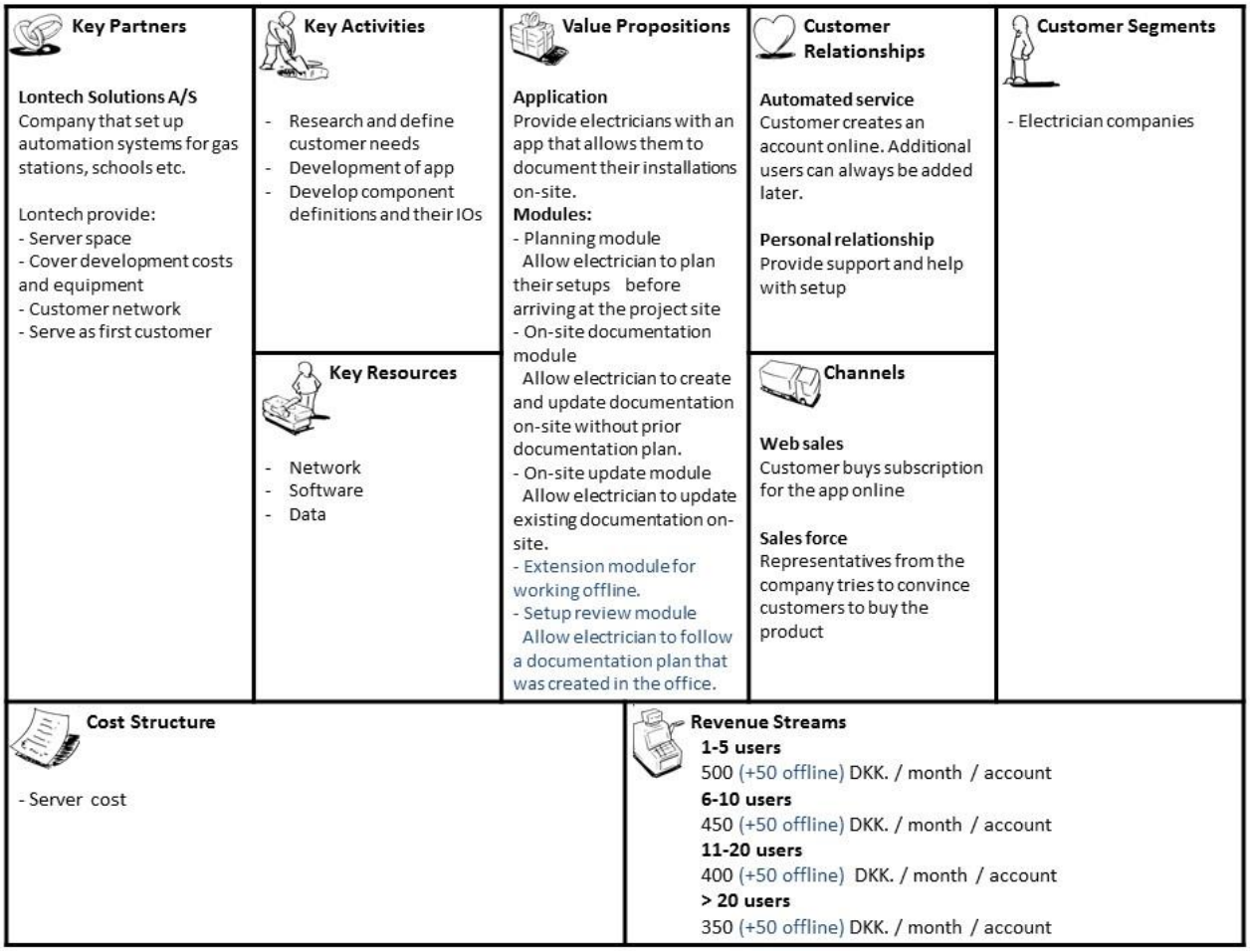


Figure 30 - Third Business Model Canvas

Value Proposition

In this iteration I have added a module to the value proposition. The "Offline extension module" will allow the users to work offline when they for example are in a basement where they don't have internet connection.

The "Setup review module" will allow the user to review a setup that was planned in the office.

Revenue Streams

I have chosen to add the "Offline module" as an optional extension module. This means, that the user will have to pay in order to gain access to that functionality. The module will cost 50 dkk. extra per user.

7.2.3 Mapping 2nd iteration

Configuration table

	Process
Strategy	*Expectations:* The key components are necessary and sufficient to facilitate proper documentation *Findings:* A review module should be implemented to allow electricians to follow instructions based on a documentation plan.

Business Model Canvas

Value Proposition
- Setup review module Allow electrician to follow a documentation plan that was created in the office.

Figure 31 - Mapping Process/Value Proposition

In the "Process/Strategy" cell of the configuration table, I found, that I need a review module that will help the electricians setup installations in a building based on already planned installations. This will create value for the customers, so therefor it needs to be included in the "Value Proposition" in the Business Model Canvas.

Configuration table

	Product
Tactics	*Key features* *- Possibility for working offline*

Business Model Canvas

Value Proposition
- Extension module for working offline.
Revenue Streams
1-5 users 500 (+50 offline) DKK. / month / account **6-10 users** 450 (+50 offline) DKK. / month / account **11-20 users** 400 (+50 offline) DKK. / month / account **> 20 users** 350 (+50 offline) DKK. / month / account

Figure 32 - Mapping Product/Value Proposition

In the configuration table I added a feature called "Possibility for working offline" which will allow

the users to use the application even when they do not have an internet connection. This added feature creates value for the user which means it needs to be added to the value proposition. Since I have decided to include the offline module as an extension module, it also needs to be represented in the "Revenue Streams" of the business model canvas. In this case it adds 50 dkk. For each user per month.

7.2.4 Revised product backlog

Name	Description	Priority	Status
APIs	Create/read/update/delete: users, projects, buildings, rooms, locations, components	1	Done
Login function	Allow users to log in to the system	2	Done
Create project	Allow users to create a new project	3	Done
Add building to project	Allow users to add a building to an existing project	4	Done
Add room to building	Allow users to add a room to an existing building	5	Done
Add location to room	Allow users to add a location to an existing room	6	Done
Setup component in room	Allow users to add a component to a location and connect inputs/outputs	7	Not started
Create user	Create user function	8	Not started
Planning module	Allow users to plan installations from the office	9	Not started
Photo documentation	Allow users to attach a photo for more detailed documentation	10	Not started
Review module	Allow users to follow a documentation plan that was created in the office	11	Not started
Offline functionality	Allow users to work offline	12	Not started

In this iteration I have added two items to the product backlog "Review module" and "Offline functionality". Both these items come from the configuration table.

I have completed the tasks of creating the GUI's for "Login", "Create project", "Add building to project", "Add room to building" and "Add location to room".

7.2.5 Screenshots of application

Below I have included the screenshots of the real application that I have created based on the mockups from section 7.1.1. The application utilizes the design principles of Googles Material Design specifications.

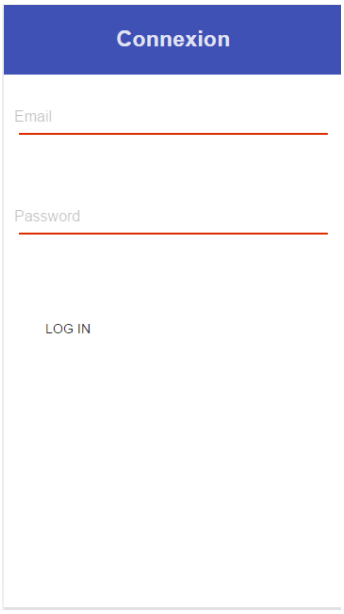


Figure 33 - Login view

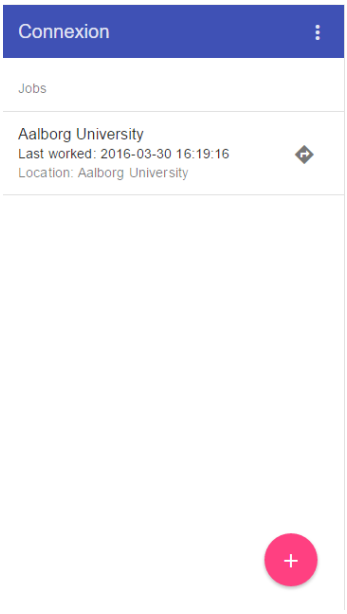


Figure 34 - Choose project view

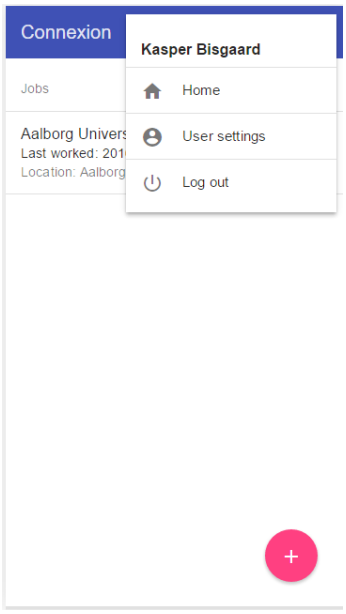


Figure 35 - Menu view

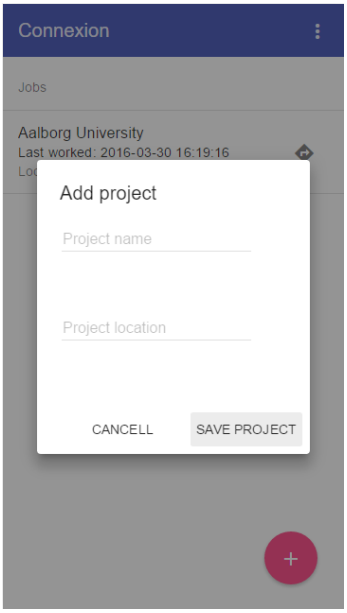


Figure 36 - Add project

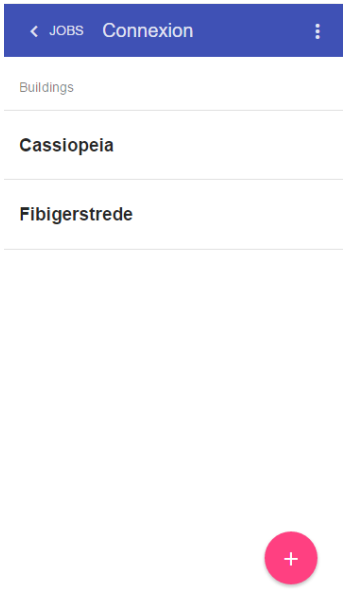


Figure 37 - Choose building view

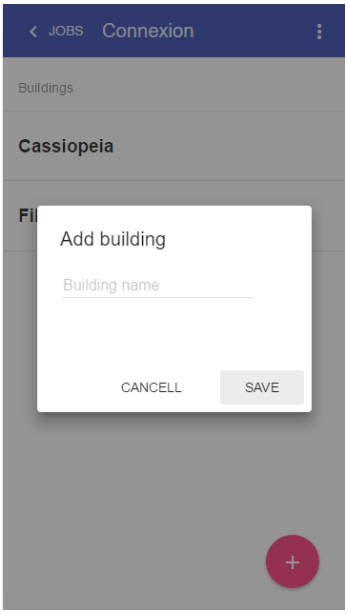


Figure 38 - Add building

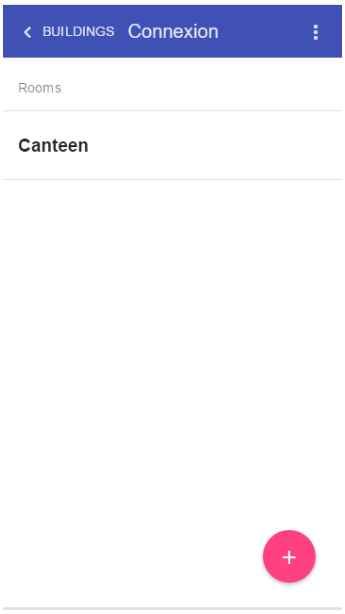


Figure 39 - Choose room view

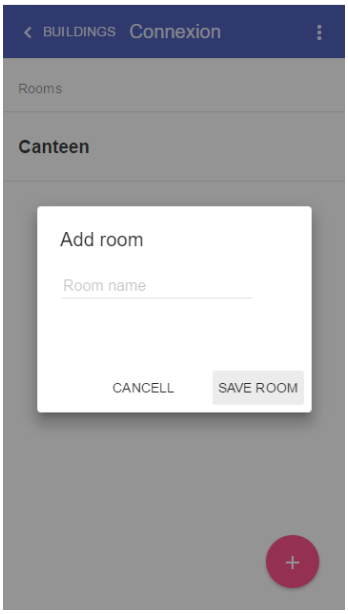


Figure 40 - Add room

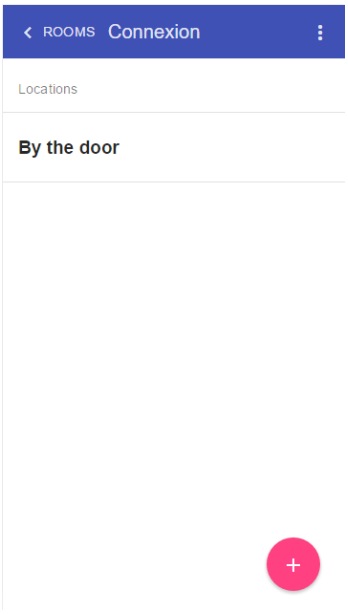


Figure 41 - Choose location view

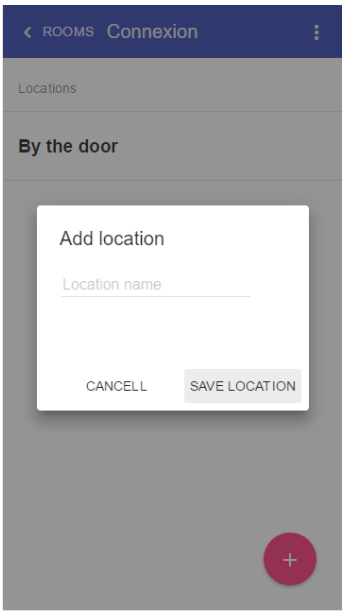


Figure 42 - Add location

8 ITERATION 3 – API + GUI + OFFLINE

In this iteration I have to completely redesign the application for several reasons:

1. I need to prepare the application for offline use. This means that I have to redesign the way the data is structured in the application in order to allow the users to add, edit and delete component in a project. Also they need to still be able to connect components when they are offline. All these changes should be added to the database once an internet connection is available. Offline capabilities cannot be added as long as the application is a web application. Therefor I have chosen to transition to a native application.
2. It should be possible to add components to a project without mapping them to a specific location. The reason for this is, that when the electrician is planning a setup, they might know which components should be used, but not the specific location or how they should be setup. Allowing the electrician to decide which components to use before they get to the job site will spare them a lot of time. This change means, that the structure of the database needs to be changed according to the illustration below.

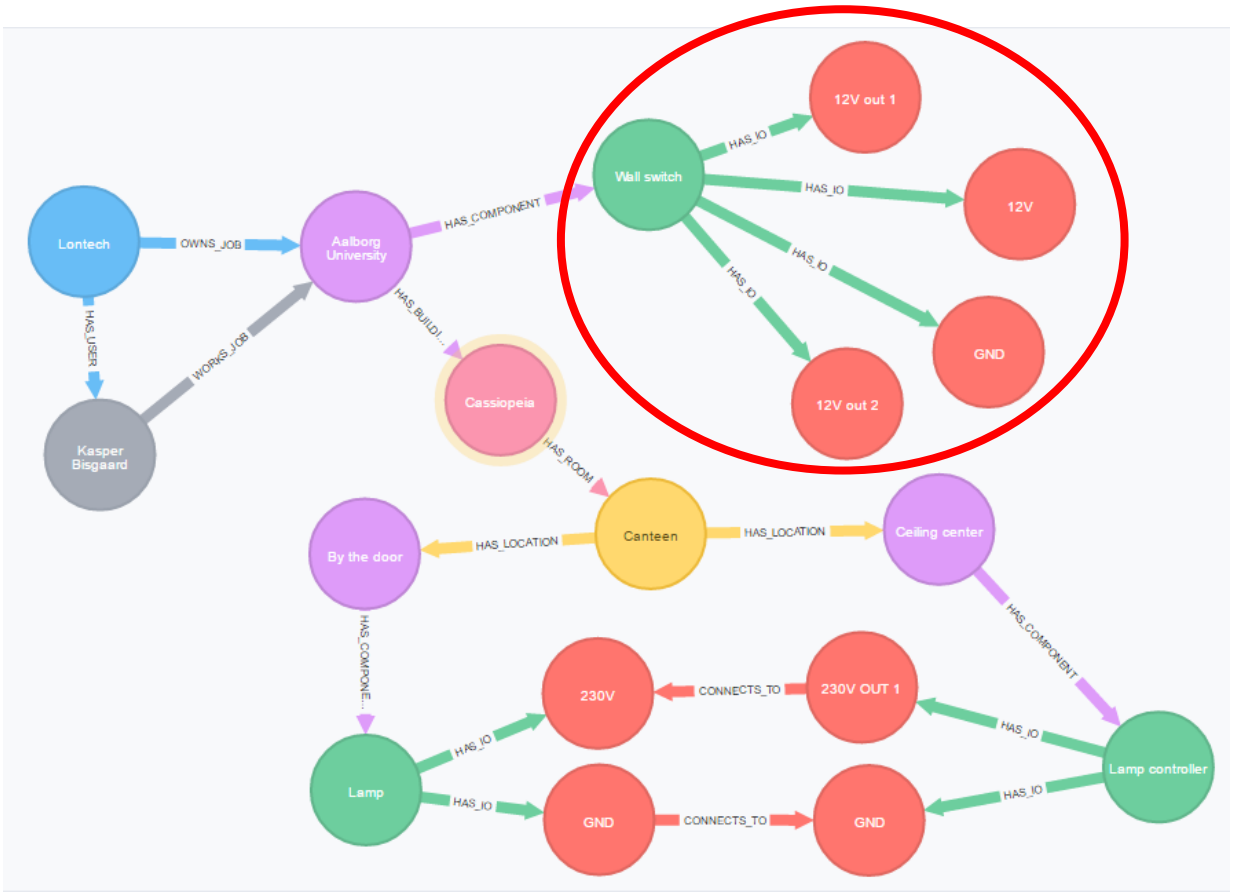


Figure 43- Revised database structure

The illustraion above shows how the new database structure will be. In stead of a component being attached to a specific location in a room in a building, it will also be possible to associate a component with a project alone, without mapping it to a location as it is shown in the red ring in Figure 43.

8.1 SPRINT PLANNING

Since I have redesigned the flow of the application, I have also revised the mockups in order to reflect that. The mockups show, that it is possible to add components to a project without mapping them to a location. I have designed the application flow in a way that allows the user to map the component to a location in the extend that they choose. If they want to just map it to a project, they can do that. If they want to map it to just a building, they can do that and so on.

8.1.1 Revised Mockups

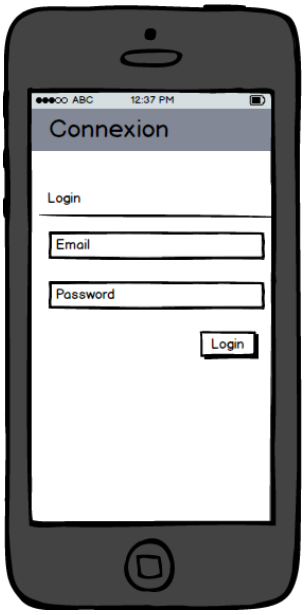


Figure 44 - Login view

This view is used for logging in to the system. It displays an "email" input, "password" input and one login button.

Figure 45 - Choose project

Choose project view stays the same.

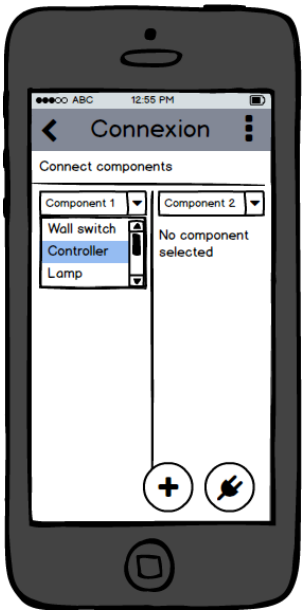


Figure 46 - Connect components

Once the user has chosen a project, they are presented with a split view screen. On each side there is a dropdown menu containing a list of all components associated to the chosen project.

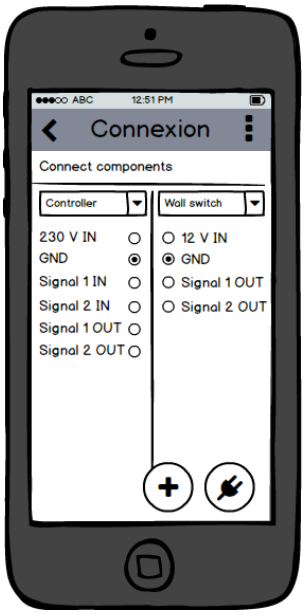


Figure 47 - Connect components (selected)

Once a component has been chosen, a list of inputs /

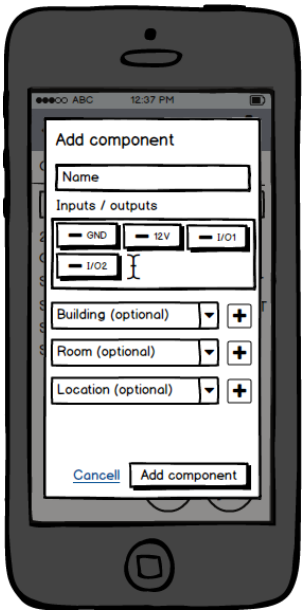


Figure 48 - Add component

Once the user clicks the "Add component" button, a popup

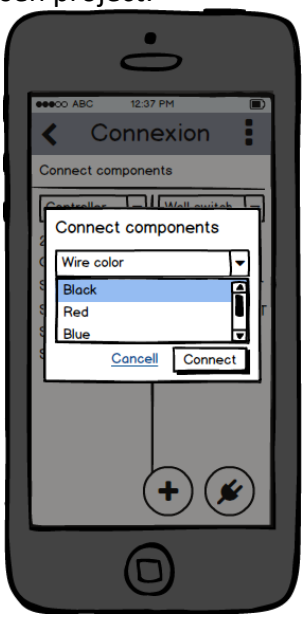


Figure 49 - Choose wire color

Once the user clicks "Connect components" button, a popup

outputs for that component will be displayed with radio select buttons associated to them.

is displayed. In this popup the user is prompted to enter a name and all I/O's for the new component. They also have the possibility to tie the component to a building, room and location. If they leave them blank, the will be prompted to select it upon the first setup of the component.

is displayed prompting them to select a color for the wire that connects the components.

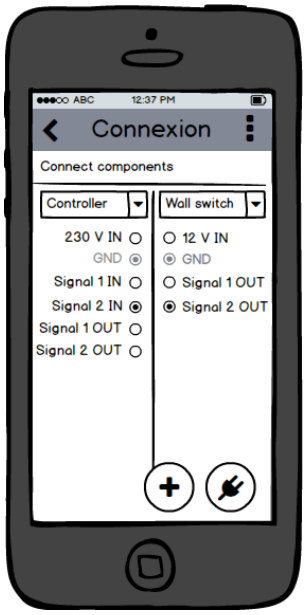


Figure 50 - Connected components

Once components are connected, the connected I/O's will be "disabled" and gray so the use can see that they are connected. Hereafter they can continue setting up the components.

8.2 SPRINT REVIEW

Through this iteration, some important findings have come to light. I have found, that an offline version of the application needs to be implemented for when the electrician is at a place that does not allow him internet connection. I have updated the configuration table according to the findings of this iteration. The revised configuration table led to changes in the business model canvas. For this thesis, I will not continue the development process due to time limitations.

8.2.1 Configuration iteration 3

View	**Paradigm**	**Product**	**Project**	**Process**
Value	*Reflection*	*Transaction*	*Reasoning*	*Appreciation*
Rationale: Why?	*Challenge* Allow electricians to document their installations from their smartphone *Problem* The process of documenting electric installations is too long and resource demanding	*Key technologies* - Neo4J Graph Database - Laravel PHP Framework - AngularJS - Angular Material - Ionic hybrid framework - Touch screen interface	*Vision* *Connexion - easily and quickly create and edit documentation for electrical installations* *Warrant* An app for documenting electrical setups can save the electrician from spending hours on figuring out how a system is set up.	*Rationale review* *Expectations:* It is possible to create an application that works offline *Findings:* It is possible, but it requires a restructure of the database and the app can no longer be a web based application.
Strategy: What?	*Key elements* Objects: User, Project, Building, Room, Location, Component, IO, Photo	*Key components* **Planning Module** - Allow electricians to plan the setups before they leave the office. **On-site module -** Allow electricians to document their setups as they install them one at a time. **Review module** Allow electricians to go through the installations they planned from home one-by-one. **Documentation View module -** Allow the electrician to see and print detailed documentation for the customer.	*Justification* ***Backing** - A native app allows everyone to view (and edit the documentation.) and facilitates use of the phone's sensors (camera, geolocation, filesystem for offline use etc.)* ***Qualifier -*** Requires a newer version of OS for android, iOS or Windows. Requires that the user has downloaded the entire project before going offline ***Rebuttal** - Most people have the newest version installed*	*Strategy review* *Expectations:* The key components are necessary and sufficient to facilitate proper documentation *Findings:* So far nothing has come to light, that suggests that the current set of functionalities are not sufficient.
Tactics: How?	*Key scenarios* - Planning component setups from the office. - Review a setup that was planned from the office - Setup of components inputs/outputs, and mapping them to specific locations in a room - Editing existing setup of connected components. - Get an overview of what is connected and how.	*Key features* - Create documentation from smartphone - Update documentation - View documentation - Attach documentation photo to certain documentation aspects - Possibility for working offline - Create component templates	*Key mapping* ***Planning module** - enables the user to plan installations from the office.* ***On-site module** - enables user to document installations while setting them up on the job site.* ***Overview module** - gives the user an idea of how the existing installations are set up.* ***Review module** - Allows the user to go through a documentation plan one component at a time*	*Tactics review* *Expectations:* The documentation is sufficient to live up to legal legislation *Findings:* Yes it is sufficient according to new legislation per 20th April 2016

Process/Rationale

In this iteration I aim to make the application work offline. Therefor my expectations are that this is in fact possible to implement.

Through the iteration I found that this is possible, but that I have to switch from a web based application to a native application in order to achieve this.

Product/Rationale

I have chosen to use the hybrid development framework “Ionic” to create the application from this point on. The reason to why I have chosen this specific framework is, that it enables the developer to develop native applications for both Android and iOS using Angular JS. It also allows the use of the various sensors in the smartphone like GPS, camera, filesystem etc.

Product/Strategy

In this iteration I have added a “Review module”. The review module enables the electrician to go through the installations that he has to set up one-by-one. The setups are based off an installation plan that was planned from the office.

Project/Strategy

In this iteration I have revised the backing, qualifier and rebuttal because I changed from a web based application to a native application.

Backing

In the backing, I argue that a native application will be better for the user since it allows the user to use the sensors in the smartphone. The interesting sensors are filesystem for offline usage and camera for photo documentation. Eventually more sensors may come in play.

Qualifier

The qualifier describes, that not all versions of Android, iOS will work with Ionic framework. It requires Android 4.1+ and iOS 7+.

Rebuttal

The newest version of iOS is 9.0 and the newest version of Android is 6.0, so I can safely assume that most users will have a newer version of their respective operating system.

Process/Strategy

My expectations for this iteration are, that the components that I have defined are sufficient to facilitate an effective documentation process.

So far I have not seen any scenarios that suggest that it is not sufficient.

Paradigm/Tactics

Since I have added a component called “Review module”, I have added a key scenario that utilizes that component “Review a setup that was planned from the office”.

Product/Tactics

For this iteration I have added two key features that was missing from the other configurations. “View documentation” and “Update documentation”.

Project/Tactics

Since I have added a new key component “Review module”, I also have to add it to the key mapping.

Process/Tactics

As per 20th April 2016 there is new legislation regarding electrical installation of low voltage below 1000 volts (Sikkerhedsstyrelsen 2016). They have published a list of requirements, that needs to be represented in the documentation of electrical installations:

1. *A general description of the electrical material.*
2. *A description of the applicable requirements, to the extent that this is relevant for the assessment of electrical equipment’s construction and use.*
3. *Design and manufacturing schematics of components and sub-assemblies.*
4. *The descriptions and explanations necessary for understanding these schematics and how the electrical equipment works.*
5. *A list of the fully or partially applied standards. If you have not applied standards you must prepare a description of the solutions chosen to meet the requirements of the Low Voltage Directive. In case of partial use of standards, the technical documentation shall specify the parts of the standards used.*
6. *Results of design calculations, examinations and test reports.*

Based on the combination of the possibility to provide detailed descriptions, schematics and documentation regarding the components alongside the detailed input/output mapping, physical placement of the components and photo documentation of the setup, I claim that my application live up to the legislation of 2016.

8.2.2 BMC iteration 3

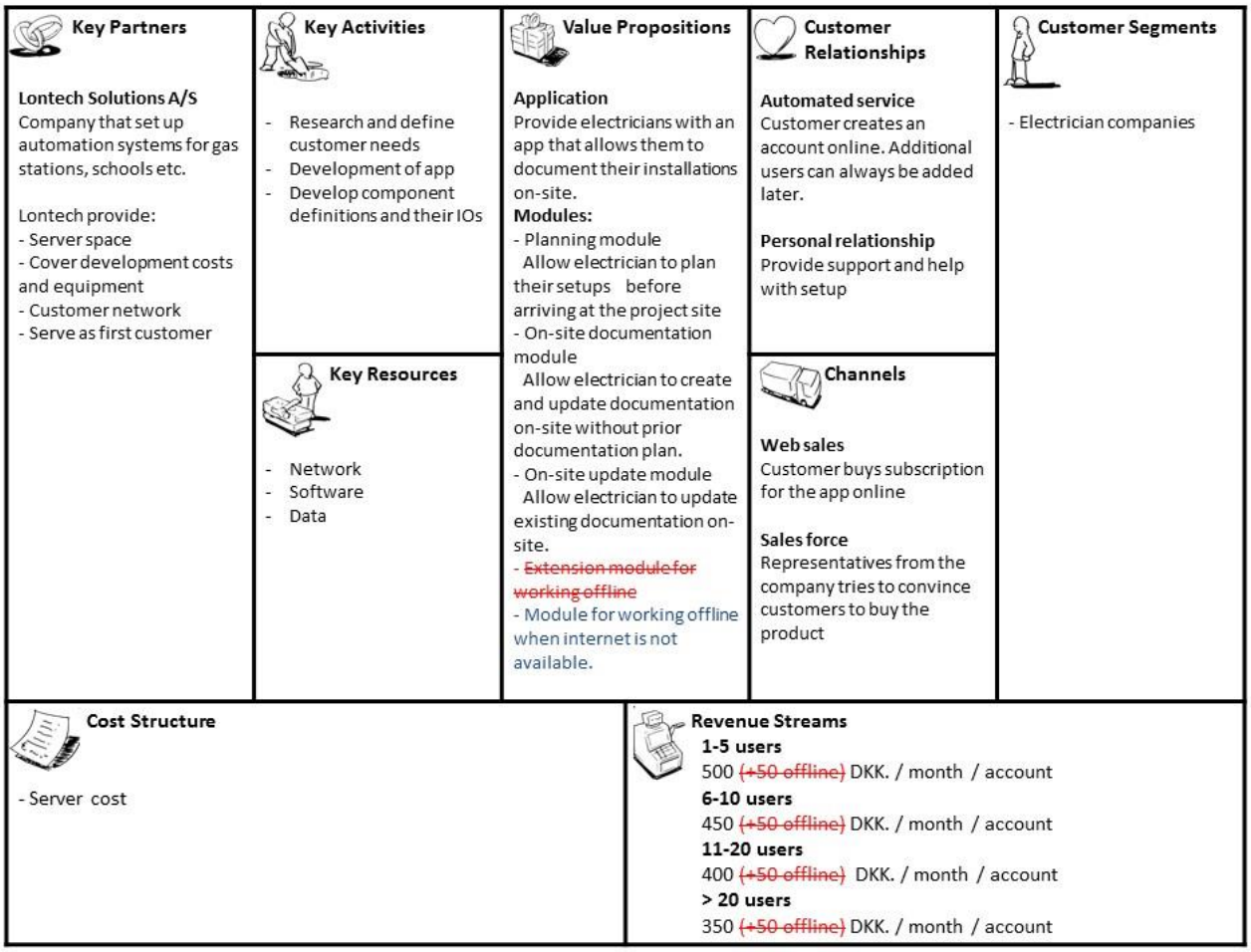


Figure 51 - Fourth Business Model Canvas

Value Proposition

In the value proposition I have deleted the “Extension module for working offline”. The reason for this is, that I have found that electricians more often than not, will be needing that module. Therefor there is no reason to make this an extension module. I have chosen to include the module as part of the standard application.

Revenue Streams

Since I have removed the extension module for working offline, I have also removed it from the revenue streams. The offline module will be included in the original pricing model.

8.2.3 Revised product backlog

Name	Description	Priority	Status
APIs	Create/read/update/delete: users, projects, buildings, rooms, locations, components	1	Done
Login function	Allow users to log in to the system	2	Done
Create project	Allow users to create a new project	3	Done
Add building to project	Allow users to add a building to an existing project	4	Done
Add room to building	Allow users to add a room to an existing building	5	Done
Add location to room	Allow users to add a location to an existing room	6	Done
Setup component in room	Allow users to add a component to a location and connect inputs/outputs	7	Done
Add components	Allow the user to create new components	8	Done
Planning module	Allow users to plan installations from the office	9	Not started
Photo documentation	Allow users to attach a photo for more detailed documentation	10	Not started
Review module	Allow users to follow a documentation plan that was created in the office	11	Not started
Offline functionality	Allow electrician to follow a documentation plan that was created in the office.	12	Not started

In this iteration I have added “Add components” to the product backlog. This addition will allow users to define their own components with inputs and outputs.

9 MODEL REVIEW

The purpose of this section is to create an overview of the strengths and the weaknesses of the model that I have proposed.

The model uses the tool “Configuration tables” in order to describe the product idea in a structured way. In my opinion this tool is very beneficial, as it helps the developer to keep track of the big picture, and it provides a way to reason for the choices that the they make during the design- and development process. However, Configuration Tables are not self-explanatory. The developer needs to understand the purpose of the four views, and how they differ. Furthermore, configuration tables are specifically designed for software projects, which in turn limits my model to software projects.

The Business Model Canvas provides an easily understandable way to describe and innovate a business model, but it does not invite to a lot of detailing and reasoning. Also the Business Model Canvas does not take factors of competition into account (Kraaijenbrink 2012). Kraaijenbrink proposes an alternative format of the business model canvas called “The Value Envelope” (Figure 52), that might be interesting to look at. It uses the components “Revenue model”, “Key partners”, “Key rivals”, “Key values”, “Key competences”, “Key customers” and “Value proposition” in the middle as the most important aspect (Kraaijenbrink 2012).

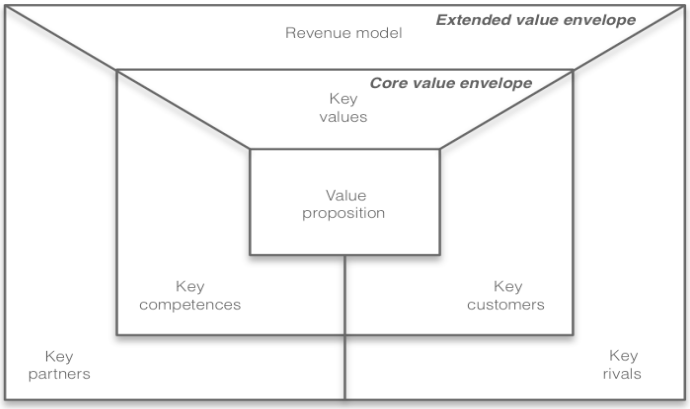


Figure 52 - The Value Envelope (Kraaijenbrink 2012)

With these critique points in mind, I still feel that the model has created value for me in the development process of the documentation application. Mainly the combination of the business model canvas and the configuration tables has been helpful in order for me to understand the problem area and the product proposal, and argue how the product solves the problem while keeping the business perspective in mind. The implementation of the product- and sprint backlog helped me keep track of what should be done, and the order in which it should be done. I believe, that most software startups could benefit from this approach, and it might be useful for development projects in established software companies as well.

10 CONCLUSION

The purpose of this thesis was to proposed an approach for developing a software product while keeping the business potential in mind. To that extend I have proposed an approach that utilizes the Configuration tables from (Aaen 2016), Business Model Canvas from (Osterwalder, Pigneur 2013) and elements from the well-known software project model "SCRUM". Using that approach, I have proposed an idea for a software product, and tested the approach by developing both the idea, the product and the business model to a point where, in my opinion, it has the potential to turn into a real business.

I have used the configuration tables from (Aaen 2016) to structure and develop the product idea, and in combination with the Business Model Canvas from (Osterwalder, Pigneur 2013) I have developed a business model. For each iteration, both the product and the business model has improved, but there is still a lot of work to be done before the product can be launched. Due to time limitations, I have only been able to do three iterations, but it is my intention to keep developing the product and business model until it is ready, and at that point I hope to make a business out of it.

It is my belief, that this approach can be applied to most software startups, and that it will help entrepreneurs to structure their development process whilst developing their business case into a successful business with a strong software product. However, since I have chosen to use configuration tables and SCRUM, the model is limited to software startups, and cannot be directly applied to other fields.

11 LITERATURE

AAEN, I., 2016. *Essence - Pragmatic Software Innovation.* Department of Computer Science, Aalborg University: .

EIFREM, E., 2016-last update, Neo4J. Available: http://www.neo4j.com [04/10, 2016].

HEVERY, M., 2016-last update, Angular JS. Available: http://www.angular.io [04/16, 2016].

KRAAIJENBRINK, J., 2012. Three Shortcomings of the Business Model Canvas. *Kraaijenbrink Training Advies Atom,* .

KRAAIJENBRINK, J., 2008. The nature of the entrepreneurial process: causation, effectuation, and pragmatism.

LARMAN, C., 2004. *Agile and iterative development: a manager's guide.* Addison-Wesley Professional.

MATHIASSEN, L., MUNK-MADSEN, A., NIELSEN, P.A. and STAGE, J., 2000. *Object-oriented analysis & design.* Citeseer.

OSTERWALDER, A. and PIGNEUR, Y., 2013. *Business model generation: a handbook for visionaries, game changers, and challengers.* John Wiley & Sons.

SARASVATHY, S.D., 2001. Causation and effectuation: Toward a theoretical shift from economic inevitability to entrepreneurial contingency. *Academy of management Review,* **26**(2), pp. 243-263.

SIKKERHEDSSTYRELSEN, 2016-last update, Lavspændingsdirektivet 2016. Available: https://www.sik.dk/Virksomhed/El-for-fagfolk/Love-og-regler-om-el/Lavspaendingsdirektivet-2016 [05/25, 2016].