

Technical University of Denmark



DTU COMPUTE
MASTER THESIS

iBeacons application in the registration and processing of user data

Author	Magnus Bjarke Bang Nielsen, s071853 s071853@student.dtu.dk
University	Technical University of Denmark
Institute	DTU Compute
University Supervisor	Stig Høgh
Project Type	Master Thesis
Company	Siemens A/S Mobility
Company Supervisor	Jesper Lauge Kilde
Hand-in date	29/08-2016
Danish Title	iBeacons anvendelser indenfor registrering og behandling af bruger-data

This report is written as a cooperation with:

The logo for Siemens, featuring the word 'SIEMENS' in a bold, teal, sans-serif font.

DTU Compute
Department of Applied Mathematics and Computer Science

29/08-2016

Preface

This report is the result of a master thesis done in the field of Master of Science in Computer Science and Engineering at the Technical University of Denmark (DTU). The master thesis is a cooperation with Siemens A/S - Industry Sector - Mobility located in Ballerup. The project is advanced by Siemens as a pilot-project and is nominated for 35 ECTS points. The project was done in the period from 29-02-2016 to 29-08-2016.

In the original project registration, this thesis was to be written in Danish, but due to regulation for master thesis, the project have to be written in English instead. A copy of the original project registration can be found in appendix A.

This project consists of a report and a project product. The report is a documentation for the product and the product itself is the solution build to solve the purpose of the project.

The project products are programmed in C# and the mobile application was created with the Xamarin-project for Visual Studio [33]. For the Windows client was used a third-party library called Winbeacon [25]. The back-end of the system is spaced on a Mssql database. There will not be handed in any source code for this project product, the only reference to the source code is screen-shots and models within the report.

A digital version of this report is handed in via Campusnet and will thereby not be physically signed by the author.

In the project is created a variance of terms to describe the system and the process. In the end of this report there are therefore a glossary, describing the meaning of the terms.

Finally I have to thank Stig Høgh, university supervisor, and Jesper Lauge Kilde, company supervisor from Siemens A/S Mobility, for good guidance, solid support and feedback throughout the entire project.

Further more I would like to thank Jan Helebrant from Siemens A/S Germany for help with technical questions and for providing the Beaconinside BEACON model.

I would also like to thank my fellow student workers at Siemens A/S Mobility for input, ideas and for borrowing their smartphones for testing.

Magnus B. Bang Nielsen, s071853
s071853@student.dtu.dk

Abstract

It has become more common for companies to have hotdesk areas. This means that many employees shares their office desks and other equipment. To make sure, that there are enough desks, sufficient air ventilation and heating, the company needs to know when the desks are occupied and how many persons there are in a given area.

Here the beacon technology is chosen together with an application to be installed either on the users smartphone or laptop. The client application is made both for Android and PC and can later be compiled to IOS.

When a person arrives to an area / location, the mobile device (smartphone or laptop) receives a signal from a beacon and the mobile device sends this data with user identification and timestamp to a database via a REST web-service, for later use. The data is anonymized and contains a unique user ID, a timestamp, a unique registration ID and location data from the beacons.

The data is presented in a web-interface, where statistics are displayed.

Results of the implementation shows that it was possible to locate a person from the implemented solution within a given area. The test results showed an error in precision of 50-60%, so it is not possible to register a person at a desk as initially intended, but only in an area. The major problem seems to be the signal interference in the office environment.

The solution is usable, but not as precise as intended. If higher precision is required, the technology has to be changed.

The project is composed by Magnus Nielsen in cooperation with Siemens A/S Mobility.

Abstract in danish:

Det er blevet mere normalt for virksomheder at have områder med flyvende arbejdspladser, som er hvor flere ansatte deles om de samme skriveborde og andet udstyr. For at være sikker på at der er nok skriveborde og tilstrækkelig ventilation og varme, har virksomheden behov for, at vide hvornår de forskellige skriveborde er optagede og hvor mange personer der er i et givent område.

Her er det valgt et benytte beacon teknologien sammen med en applikation der skal installeres enten på brugerens smart-telefon og / eller bærbare computer. Klientapplikationen er udviklet både til Android og PC og kan senere kompileres til IOS.

Når en person ankommer til et område, vil mobilenheden (smart-telefonen eller bærbara) modtage et signal fra en beacon enhed, hvorefter mobilenheden sender dette data videre, med brugeridentifikation og tidsstempel tilføjet, via en REST web-service til senere brug. Data er anonymiseret og indeholder unik brugerID, et tidsstempel, unik registreringsID og lokations-data fra beacon enheden.

Data præsenteres i et web-interface, med grafisk præsentation af statistikker.

Resultaterne fra implementeringen viser at det var muligt at lokalisere en person i et givent område ud fra den implementerede løsning. Resultaterne fra testene viser at afstanden fås til at være 50-60% kortere og det er dermed ikke muligt at registrere en person ved et skrivebord, som det oprindeligt var tænkt, men kun at registrere personen i et givent område. Det primære problem virker til at være støj på signalerne i kontormiljøet.

Løsningen er brugbar, men ikke så præcis som tiltænkt. Hvis en større præcision ønskes, skal teknologien ændres.

Contents

Preface	1
Abstract	2
1 Introduction	5
2 Project development process	6
3 Analysis	8
3.1 Stakeholders - Who interacts with the system	8
3.2 User stories	9
3.3 Functionalities based on user stories	10
3.4 Use Cases	12
3.4.1 Use Case descriptions	13
3.5 Requirements	17
4 Design	19
4.1 Comparing beacon with other technologies	19
4.1.1 GSM triangulation	19
4.1.2 iBeacon	20
4.1.3 Standard GPS	21
4.1.4 Triangulating with Wifi	21
4.1.5 Technology summary	21
4.2 System design	22
4.2.1 Register person	22
4.2.2 Assign person	23
4.2.3 Configuration of area	25
4.2.4 Show data	26
4.3 Interface design	27
4.4 Technical Details	28
4.4.1 Frequency of collecting beacon-data	29
4.5 Domain model	30
5 Database	31
6 Implementation	32
6.1 Beacon	32
6.2 Bluetooth client	33
6.2.1 PC Bluetooth Client	35
6.2.2 Andriod Bluetooth Client	36
6.3 REST web-service to database	38
6.4 Show data - web-interface	39
7 Test	44
7.1 Test Area	45
7.2 Acceptance test	47
7.2.1 Simulation of the Acceptance test	47
7.2.2 Acceptance test result	48
7.3 System test	51
7.3.1 Precision of beacon registrations	52
7.3.2 Battery test	53

8 Discussion	54
8.1 Test results	54
8.1.1 Precision test results	54
8.1.2 Speed test - incomplete	55
8.2 Competitor analysis	56
8.3 Privacy of the user	57
8.4 Requirements fulfilled	58
9 Future Work	60
10 Conclusion	61
11 Glossary	62
12 Figure reference	65
References	69
A Appendix - Project Registration	73
B Appendix - Calculation of area based on two beacons	74
C Appendix - Database creation script	79
D Appendix - Final Database layout	81
E Appendix - Raspberry Pi and PC Install guide to use as a Beacon	82
F Appendix - Acceptance test of main success scenario	85
G Appendix - Precision test of mobile devices clients	87
H Appendix - Battery test of the Android bluetooth module	93
I Appendix - Battery test android application readout	94

1 Introduction

Today it is more common to share resources or a desk with co-workers. This concept is called hotdesk areas or free seating and is done to save resources and to be more cost efficient. Earlier open offices came but after the laptops have become normal work-computers, it has become possible to make standardized desks with a screen, mouse, keyboard and a docking station to the laptop. That way, the employees do not need their own private desk, so employees working at different hours, can use the same desk.

But how can it be measured or controlled how many that uses these areas to find out weather there are enough room for everybody, optimal air-conditioning and heating?

Originally the scope was to count the numbers of employees in a given area and to register employees at a specific object, such as a desk, that they would normally share. But during the process, it got changed so the focus of this project is now only to measure how many people there are in a given area

The solution is made with the beacon technology, started by Apple in 2013. The beacon technology is based on bluetooth signals, that is broadcasted from the beacon unit and can be picked up by anyone nearby, range up to 70m. The beacon technology requires that the bluetooth in the receiving and transmitting units are turned on constantly and is at least a version 4.0.

The system is designed to collect user data from the real world and put the data on to a scaled map of a building or a floor plan. From these data collections, statistics and a visual overview can be made to show where employees are located.

The collection of data is done through the everyday devices used by the users, such as a smartphone or a laptop. In this report these devices will be referred to as mobile devices. To protect the employees, the data are anonymized when it is collected to the system.

In order to test the implemented solution a test of the system was created with 4 Android smartphones and a laptop using 2 different bluetooth chips. The results shows that all the registered data was 50-60% to short in the measuring of the distances. All tests were done in a normal hotdesk area containing walls of plaster, many wifi hotspots and other devices using the 2.4GHz, all these things can interfere the beacon signals and are all expected to be found in a normal hotdesk area.

Half way through the project it was tried to get an accurate measuring of distance using a self-created beacon transmitter based on a Raspberry Pi and a laptop. The results from this showed that the precision could not be within more than a meter in best case scenario. Later the same was tested with a manufactured beacon unit with better results, but still not accurate enough. It was then decided to focusing on registering persons in an area instead of at an object.

2 Project development process

This project is developed according to the developing model of Siemens Processhouse, see figure 1. The working methods is structured around the V-model. The model handles hardware and software projects or a deviation of those. This is a software pilot-project, so only selected milestones from the model in figure 1 are used.

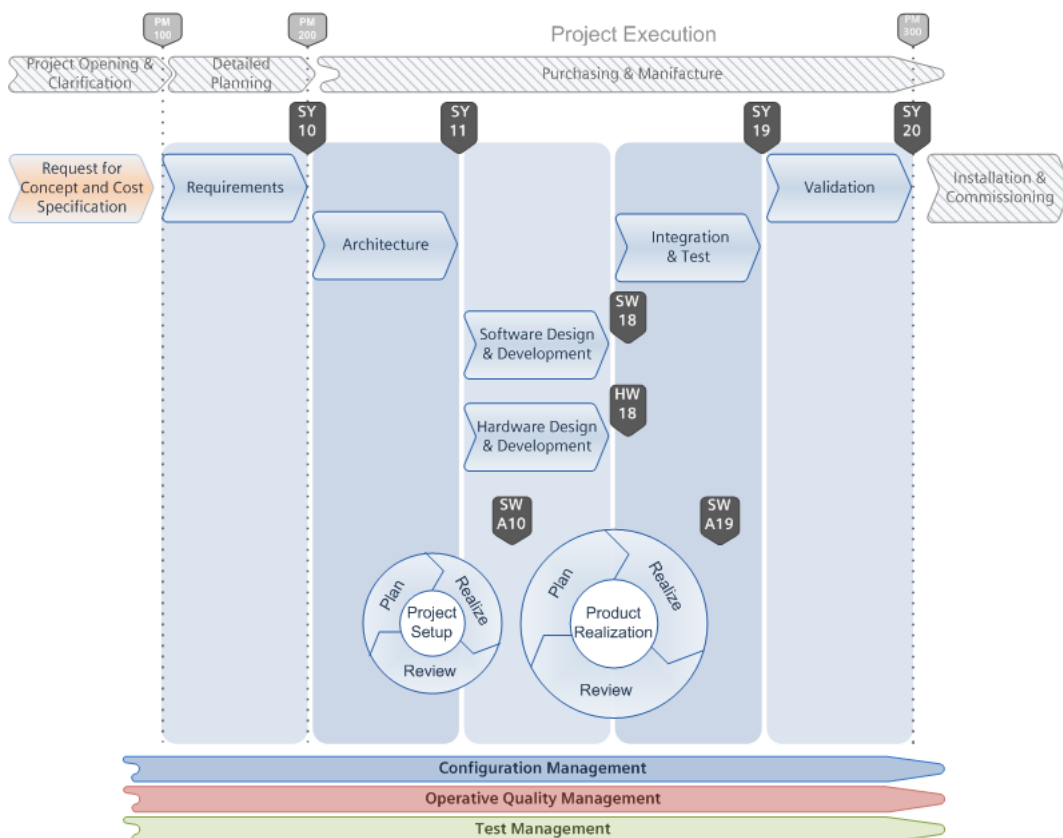


Figure 1: Developing model for projects at Siemens.

The chosen milestones are listed below including a description.

SY10 - Requirement specification

This milestone contains that all requirements, both functional and non-functional, to the project are defined. The definition of these requirements ends up being the requirement specification for the project. This milestone also includes designing of the tests.

SY11 - Architecture.

It consists of a specification of the architecture for the system, the interface between the different components and the specifications of the tests.

SW18 - Software design and development.

Here the requirement specification and the specification of the architecture are put together to defining the design of the software. From the software design the project can be programmed, and together with the specifications of the tests, several iterations are made, until the software are completed. The development process to reach this milestone is displayed in figure 2.

SY19 - Implementation and test.

The project is implemented and tested against the requirement specifications.

After SY19 are completed, the project is ready to be presented for potential clients / customers.

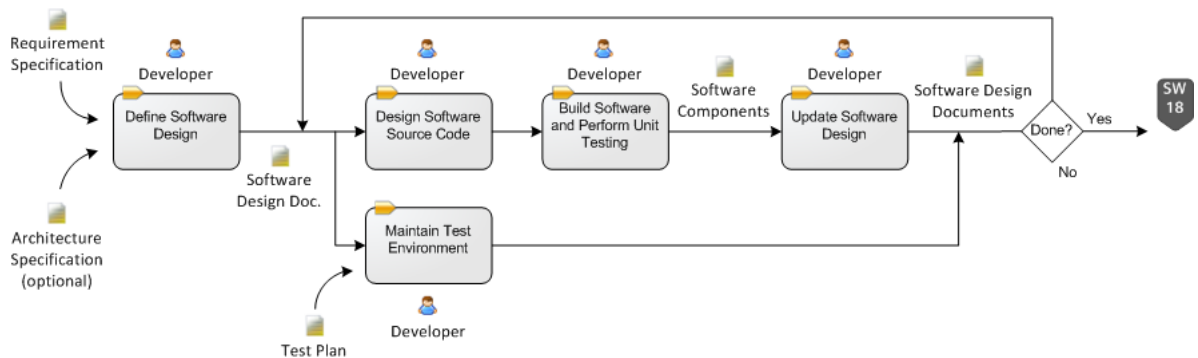


Figure 2: The developing process for the milestone SW18, with the possibility of several iterations.

3 Analysis

To define how the system is going to work, the surroundings of the system have to be taken into account and be analysed. Which persons are going to use it, what it can be used for and how it is going to be used.

First the system stakeholders are being identified. Now knowing who has interest in the environment, different user scenarios and wishes can be set up for each stakeholder to fulfil their wishes and needs. This is done by creating user stories. Later these user stories can be collected and merged into common use cases for the system to be build. The use cases are then detailed by creating a use case description and in the end functional and non-functional requirements can be extracted from the use cases.

3.1 Stakeholders - Who interacts with the system

The project description describe the environment where the solution may solve everyday problems. The environment is a hotdesk area, where people come and go as they wish. Due to this the solution have different stakeholders. All these stakeholders have daily contact or interaction with this environment. The stakeholders are defined below in table 1.

Person Also called employee. The person is one that have his every day work in a hotdesk area. Everyday challenges might be to find a free seat, an area with a low amount of people (quiet) or ensure that the person have the resources he / she needs for the work.

Manager Also called team leader or employer. It is in the interest of the manager that there are enough free seats for the team to work together in the same area and have a good working environment (ventilation and heating). Possible administrator of the system.

Building personnel Also called SRE¹employees. It is their job to maintain, repair and install new features within the building. They are also responsible for the leasing of the buildings and therefore have interest in knowing weather or not, there are enough room / seats to fulfil contracts.

Cleaning staff Is responsible for the daily cleaning of the buildings. If the cleaning staff were to know when an area would be empty or full, it would make it easier for them to plan their cleaning to be more efficient and less disturbing for the employees.

Table 1: The different stakeholders who would have contact to the environment.

¹Siemens Real Estate

3.2 User stories

To determine and limit what the system should be able to do, user stories have been constructed within the environment the system belongs to, see table 2. A user story is based on what a stakeholder may gain from using the system. A user story can e.g. follow the following simple template, which is used in this project:

As a <type of user>, I want <some goal> so that <some reason>

- Cited from Mountain Goat Software [46]

In some user stories, the goal and the reason might imply itself or be the same. The structure of user stories set-up is only used as guidelines.

Please notice, that the following user stories only are suggestions for possible use cases. They may not all be implemented in the final solution.

1. As an **Employee**, I want to **get a desk** that is **free**.
2. As **Cleaning staff**, I want to **know when an area is not occupied**, so I can **clean more efficient**.
3. As an **SRE Employee**, I want to make sure, **that there is enough space by (a - c)** for my customer to **keep them happy**.
 - (a) **See how many there are, in a given time period**
 - (b) **Find where there are seats for more**
 - (c) **Knowing which team is located where**
4. As a **Manager**, I want to make sure, **that there is enough ventilation** so my employees have **optimal work environment**.
5. As a **Manager**, I want to **know which parts of the building is used** the most to **optimize seating conditions in my team**.
6. As an **Employee**, I want to **login to the system**, to **find free seats**.
7. As a **Manager**, I want **login to the system** to see **statistics** for a given area and a given time period.
8. As a **Cleaning staff**, I want to **login to the system**, to see **free areas**.
9. As an **Employee**, I want to **see all free desk** to **choose a good spot**.
10. As an **Employee**, I want to **reserve a seat** for:
 - (a) **An entire day**
 - (b) **Part of a day**
 - (c) **Another person**
11. As an **Employee**, I want to **look up another employee** to **find where they are seated today**.
12. As an **Employee**, I want to **get guidings** to **find a person**, by **sending a location request**.

13. As an **Employee**, I want to **get guidings to find the nearest free seat**.
14. As an **Employee**, I want to **notify when I am going home so another can use my seat when I am gone**.
15. As an **Employee**, I want to **book an area for other users so we can find an area to work together**.
16. As a **Manager**, I want to **book an area for my entire team**.
17. As an **Employee**, I want to **choose one or more seats to book for later use**.
18. As an **Administrator**, I want to **create new seating areas**.
19. As an **Employee**, I want to **edit my own reservations**.
20. As a **Table / Seat** I want to **have a status as either occupied, free or reserved**.
21. As an **Employee**, I want to **have my mobile device turned on to notify the system that my seat is occupied**.

Table 2: User stories to define the system.

3.3 Functionalities based on user stories

From the user stories above, the functional requirements are determined by contracting those user stories with similarities, in a single functionality. Some user stories may be in more than one functionality, if it has different similarities with different user stories. The list below in table 3, shows the different functionalities and which user stories they cover.

- Register and locate persons within the building.
(User stories: 4, 21)
- Locate persons within range of a certain object or area.
(User stories: 20, 21).
- Extract statistics for registrations in the building.
(User stories: 2, 3(a-c), 5)
- View free seats.
(User stories: 1, 9, 13)
- Login authorization (differentiated access).
(User stories: 6, 7, 8)
- Reserve seats or areas or edit existing reservations.
(User stories: 10(a-c), 14, 15, 16, 17, 19)
- Get guidings to areas or persons.
(User stories: 11, 12, 13)
- Create new locations or edit existing.
(User stories: 18)

Table 3: User stories that fulfils the same purpose in the system are collected in groups.

The functionalities are prioritised in three categories according to their importance for the implementation of the system. The three categories are:

- 1 Main functionality. The system cannot work without it.
- 2 Main features. Supports main functionality.
- 3 Extra features. Nice to have, but the system can work without it.

In table 4 the functionalities are described in detail and are given a use case name. The name of the use cases will reference to the same use cases throughout the project.

Functionality	Priority	Purpose	Use Case
Register and locate persons within the building.	1	When a person enters an area, it has to be registered that the person has entered the area and where in the area the person is located.	<ul style="list-style-type: none"> • Register person
Locate persons within range of a certain object or area.	1	The registered data from a person are, if possible, linked together with an object or an area.	<ul style="list-style-type: none"> • Assign person
Create new locations or edit existing.	1	If the decor or functionality of an area is changed, it is necessary to adapt those changes in the system, by editing the area or create a new.	<ul style="list-style-type: none"> • Configuration of area
View free seats.	1	Show representation of all areas in real-time, so a user can see free, occupied and reserved seats and thereby find a seat for the day.	<ul style="list-style-type: none"> • Show data
Extract statistics for registrations in the building.	2	Get daily statistics for areas. This use case is a sub-use case for Show data .	<ul style="list-style-type: none"> • Show statistics
Login authorization (differentiated access).	2	The access to the system is hierarchically build so only the administrator can edit and create new areas. A manager can reserve an area for his / her team. A person / user can reserve a seat for them self and see the current amount of persons in the areas.	<ul style="list-style-type: none"> • Login
Reserve seats or areas or edit existing reservations.	3	A user can reserve a seat, a manager can reserve one or more seats or an entire area. A manager can also edit or cancel an existing reservation. Since areas and seats are different subjects, this functionality is split in to two use cases. These use cases are sub-use-cases for Login .	<ul style="list-style-type: none"> • Reserve a seat • Reserve area
Get guidings to areas or persons.	3	Help a person with guidings to another person or a certain seat. This functionality is split in to two use cases.	<ul style="list-style-type: none"> • Find seat • Find person

Table 4: Converting functionalities in to respective use cases, assigning priorities of the functionalities for the system and describing the purpose.

3.4 Use Cases

The use cases which were found in section 3.3 - Functionalities based on user stories are used for describing the functionalities of the system. The use cases with the priority of 1 are collected with their main actor of the system and put in to a use case overview for the entire system. This system overview can be found in figure 3.

Please notice that the role of the user can also be completed by the administrator, since the system is based on hierarchically access.

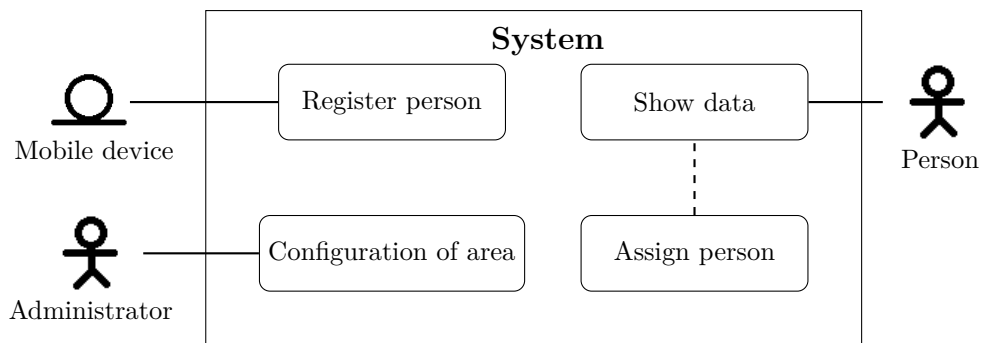



Figure 3: Use cases with priority 1 in a system overview.

The  symbol is the actor icon for another system or outside unit. This is a non-human actor that is associated to the system. In this use cases it is providing data to the system.

3.4.1 Use Case descriptions

To know how the system is going to work in detail, the found use cases are being systematically described. For that purpose the “Fully-Dressed” Use Case Template is used [44].

Compared to the original use case description model template, The Owner category of the use case has been removed, since there are only one employee / worker / student on this project. Furthermore is the Priority category also removed from the template, since the implemented use cases only covers the uses cases with priority 1 in table 4.

The use cases are Register person in table 5, Configuration of area in table 6, Show data in table 7 and Assign person in table 8.

ID:	UC 1.1
Title:	Register person
Description:	When a person enters within reach of a beacon, the application starts (headless-mode) on the persons mobile device (e.g. smart-phone) and there are searched for more beacons, in order to increase the chance to get a more accurate position. After a time period the collected data is stabilized (pick the median distance for each beacon) and stored in the system. Data stored is, the distance to each beacon and their respective beacon ID. In addition, a unique ID for the mobile device (hash MAC address for the bluetooth chip of the devices).
Primary Actor:	Mobile device
Preconditions:	The mobile device have found one or more beacons.
Post conditions:	A person registration (Location, timestamp and unique ID) is stored in the system database.
Main success Scenario:	<ol style="list-style-type: none"> 1. The mobile device registers a beacon nearby. (It is possible to determine a position for the mobile device). 2. Waiting for a time period to try to find more beacons and have a higher precision of those all ready found. (mobile device) 3. The mobile device sends the beacon-data to the system database.
Extensions:	<ol style="list-style-type: none"> 3.a It is not possible to connect to the system database. - 3.a1 Beacon data is stored in the mobile device and when the connection are reconnected, it tries to resend them.
Frequency of Use:	<p>Every 350ms (see section 4.4.1)</p> <p>Every time a mobile device registers a beacon.</p>
Status:	Implemented

Table 5: Use Case description of how a person is registered in an area and provides data to the system.

ID:	UC 2.1
Title:	Configuration of area
Description:	To configure an area (new or old), an area must be covered by at least one beacon. There can be one to several beacons per area. The number of beacons determines the precision of the area definition.
Primary Actor:	Administrator
Preconditions:	Beacons are set up and are able to covering the area to be configured.
Post conditions:	Can register users and cover them in the new area.
Main success Scenario:	<ol style="list-style-type: none"> 1. First select a map which the area is placed in. 2. Select an area name. 3. Then select the area by adding the area beacons (beacon ID) Beacon readings can now be linked to a given area. 4. Select an area by reading distance for all beacons within reach. 5. The position is determined to a sub-area (as a table or desk).
Extensions:	<p>3.a There is only one beacon in the area.</p> <ul style="list-style-type: none"> - 3.a1 If there is only one physically beacon in the area, the administrator is advised to attach an extra beacon to the area to extend the precision of the position in the system. - 3.a2 The administrator may choose to continue the set-up with only one beacon, but can only register the users in an area and not where in the area due to lack of precision. Proceed directly to step 4. <p>4.a An area should cover a larger location.</p> <ul style="list-style-type: none"> - 4.a1 First select a corner of the area. - 4.a2 Read distance to all area beacons for the first corner. - 4.a3 Step 4.a2 repeated to all corners are read. <p>4.b If the category is not square but eg. a circle.</p> <ul style="list-style-type: none"> - 4.b1 There is formed a square area within the original framework (eg. a circle) and continue as in Step 4.a. <p>5.a A sub-area should be within a known area.</p> <ul style="list-style-type: none"> - 5.a1 Select the area where the sub-area is located. - 5.a2 Use the extensions for 4.a and follow the instruction for the sub-area.
Frequency of Use:	When the system must get a new area or it is necessary to update an old.
Status:	Partially implemented. The main success scenario no. 5 and the extensions are not implemented due to precision issues described in section 8.1.1 - Precision test results.

Table 6: Use Case description of how to configure a new area with sub-categories such as tables in the system.

ID:	UC 3.1
Title:	Show data
Description:	Displaying data for different scenarios e.g. how many people there are in an area in a given time period or how many seats are occupied, by displaying it graphically on a drawing of the floor plan of the building.
Primary Actor:	Person
Preconditions:	There are registered beacon-data in the database.
Post conditions:	A map with the found areas is displayed with the number of people registered in the given period for each area.
Main success Scenario:	<ol style="list-style-type: none"> 1. Choose a location. 2. Show data for today, for the selected location. 3. The user change the parameters and the screen is updated. 4. The user is presented with a map with an overview of where the selected location is displayed. 5. Screen automatically updates with new data (see frequency).
Extensions:	<p>1.a No location is selected.</p> <p>- 1.a1 There appears a default screen to the user, with further instructions to the user.</p> <p>5.a It is not possible to update with new data, due to the connection of the database is interrupted.</p> <p>- 5.a1 The newest status stays on screen and there will be a message on top of the text: "Can not update data, waiting for connection."</p>
Frequency of Use:	Update the current data for the user (proposal once per minute) or whenever there is a change in the parameters of the screen.
Status:	<p>Partially implemented.</p> <p>At main success scenario no. 1, the user have to select the parameter and update before any change is made.</p> <p>At main success scenario no. 3, the automated update is replaced with a button "Hent Data" that the user have to press to update the data every time.</p>

Table 7: Use Case description of how a user views data on a web-interface in the system.

ID:	UC 3.2
Title:	Assign person
Description:	Searched through data for binding beacon-data to one or more areas.
Primary Actor:	Administrator (through use case UC 3.1)
Preconditions:	The database of the system holds Register person data
Post conditions:	The system place the found persons on a map of a building or floor plan.
Main success Scenario:	<ol style="list-style-type: none"> 1. The system finds all person registrations. 2. The person registrations are linked to areas based on which beacons they have encountered and the distance to them. 3. The user registrations are linked to a sub-area such as a desk or another object.
Extensions:	<ol style="list-style-type: none"> 1.a No user registrations found. <ul style="list-style-type: none"> - 1.a1 A message is displayed on the screen, saying: "No data found. Please Try again." 2.a Person registrations exist, but not an area to link them at. <ul style="list-style-type: none"> - 2.a1 Data is stored in a log-file for detecting errors. 3.a Person registrations can not be located for a sub-area. <ul style="list-style-type: none"> - 3.a1 The data is linked to the area only.
Frequency of Use:	Everytime the use case US 3.1 Show data is executed.
Status:	Partially implemented. Main success scenario no. 3 only works for the areas. The extensions are not implemented.

Table 8: Use Case description of how person registrations are located by linking them to an area and / or a sub-area.

3.5 Requirements

The requirements for the system are found from the analyses done on the user stories and use cases. The list of functional and non-functional requirements can be seen below.

Functional requirements(FR):

- **Register person**
The system should be able to register movement of a person and store it for later use.
- **Assign person**
The movement of a person can be mount to a location or area. This is made with data from the Register person.
- **Configuration of area**
The administrator can add more areas to the system and link these areas together with a floor or a building.
- **Show data**
An interface or representation should be provided in order to link the data from Register person together with an area in the building, e.g. a room.
- **Show statistic**
The statistics of the registrations in the system can be found for an area within a certain time period.
- **Login**
The accessibility of the system is controlled by a login and is hierarchically structured. This controls which parts of the system the user / person can access. An administrator can edit the full system but the normal user can only see the current statistics and reserve seats.
- **Reserve a seat**
A user / person should be able to reserve a seat in an area for a day or a part of a day. It is not possible to reserve a seat for more than 1 day.
- **Reserve area**
A leader / manager should be able to reserve a defined area for his / her employees for a day or part of a day, not more than 1 day.
- **Assign seat**
An area may contain several different objects. These can be assigned to an area and can hold their own individual person-count.
- **Assign person**
When a person is registered, it should be possible to assign the data for the person to one or more areas based on the data itself. This process should be automatically within the system.

Non-functional requirements(NFR):

- **The system have to benefit the user.**
The users needs to have some features in return for them providing data to the system. A feature can be that the users can reserve seats. That way the users want to use the system because they benefits from it.

- **Non / low user interaction with the system.**
The user should not have to do any thing when the system is collecting data. This way the user deliver data without being annoyed or disturbed in the everyday life.
- **Cross-Platform.**
The system should work independent of what platform the users have. Therefore the software, should work on an Iphone, Andriod or a PC(using the build in bluetooth in the laptops).
- **Low battery usages on clients mobile devices**
The solution must use a minimum of power on the mobile devices for the users to accept the system to be turned on constantly on their devices.
- **Power-supply or long battery lifetime for transmitters.**
The most transmitter or beacons models is battery driven either by a coin battery or a standard battery(AA / AAA).

4 Design

The most critically element of the system based on the analysis section is how to register a person with high precision within a building. The precision of the technology chosen for the system, is therefore a crucial factor for the system.

The first step in the design, is to compare the already chosen beacon technology with alternative technologies, to see weather the beacon is the optimal solution for this system.

Based on technology and the found use cases the system design can be put together. This is done by designing each use case and design the interfaces between them. Further on, the domain model and data model for the system can be created.

4.1 Comparing beacon with other technologies

There is a range of technologies for locating a person or an object, both indoor and out. To make sure, that the beacon technology is the right for this project it is compared to other technologies from the list below.

Please notice, that the precisions for the different listed technologies only are guidelines, because the precision depends on the receiving unit(e.g. smartphone or laptop), the conditions in the building and where the test has been done.

- A-GPS (gsm-triangulation)
- iBeacon
- Standard GPS
- Triangulating with Wifi

4.1.1 GSM triangulation

A-GPS(Assisted-GPS) is the term used when a mobile device such as a smartphone uses the gsm network to get the global position of the device itself. This is normally used for tracking of the device or guiding as in a car navigation. Outdoor it works very well, but for indoor positioning it can make disturbances in the precision, since the signal have to pas through building materials such as concrete, granite or windows with metalized window film.

Study [53, in table 3] shows that when trying to locating a device using A-GPS within a building, the average error on the tested devices was 15-16meter(motorola) and 8,78meter (Sanyo).

Because this technology is based on out side antennas, it can not detect on which floor the mobile device is located.

Pros:

- + No additional hardware has to be set up within the building / area.

Cons:

- The signal is disturbed by going through the building(reduced precision).
- The unit to be located needs to have a gsm-antenna.
- May not work in a building with more than one level.

4.1.2 iBeacon

The iBeacon technology is based on bluetooth communication between devices. From the beacon a signal is broadcasted and can be picked up by everybody. The data the beacon broadcasts is (for the Standard beacon) only the distance to where the beacon signal is picked up. This requires that the beacon module is set up within 70 meters of where it is going to be used. Like the other technologies, this signal can be disturbed by building materials and other units broadcasting on the 2.4GHz band.

It is hard to get a clean study on this technology. In one topic in a Q&A on accuracy of iBeacon, a replier writes that:

For example at low power you could detect a threshold 30cm at +/- 5cm accuracy after 10 seconds, and the signal could not be detected at 10m. If you go to full power you could probably detect a 10m treshold +/- 50cm after about 10 seconds. This signal could be detected at 50-70 but with huge fluctuations.

- Cited from Tim Baker [4]

Another source [9] writes that in a non-controlled environment (signal interference items in the area, e.g. other 2.4GHz devices and radiation from screens) it is impossible to get the distance error below 5 meters.

The third source is an article from the beacon producer, where they write:

On average, the **measurement error can be 20-30% of the actual distance**. You can increase signal reliability by **increasing the Broadcasting Power** and **lowering the Advertising Interval** of a beacon.

- Cited from Wojtek Borowicz, Estimote, Inc. [7]

Pros:

- + Cost efficient hardware. The price begins at kr. 20 [12] for the printed circuit board and a out of the box model cost around 89 Euro for a three pack [20].

Cons:

- Additional hardware needs to be installed in the building / area.
- Requires that the bluetooth is turned on, but that only uses a small amount of power (BLE v.4+)

4.1.3 Standard GPS

A GPS(standard GPS) as known from car navigation, is based on satellite positioning. For outdoor use or within a car, it is doing its job, but inside a building, the GPS may have problems due to construction of the building and other obstacles for reading the signal from the satellites.

Study [53, in table 3] shows that when trying to locating a device using standard GPS² within a building, the average error on the tested devices was 9-11 meter(Garmin) and 5-10 meters (Juno).

Because this technology is based on satellites, it can not detect on which floor the mobile device is located.

Pros:

- + No additional hardware needs to be added to the building / area

Cons:

- Lowered precision when used indoor. [53, comparing in table 2 and 3 af the study.]
- The receiving device needs to have a GPS-antenna installed. Not common in smartphones or laptops.
- May not work in a building with more than one level.

4.1.4 Triangulating with Wifi

Another method to locate devices indoor. Can be used with already existing wifi-equipment. The technology is based on the strength of the signals to the client device. So like the beacon technology, a distance is received, given in signal strength to the wifi transmitter. It is possible that the wifi transmitter is not installed in the same room, as the person receiving the signal. Building materials and other obstacles can reduce and disturb the signals.

The precision of the triangulating with Wifi, is "*relatively inaccurate (5-15m) compared to BLE³/RFID*" as an article [17] from infsoft blog describes.

Another source [45] writes that wifi triangulation can get an accuracy of less than half a meter and there have been work done in the area to improve it even further.

Pros:

- + Most places has already installed several wifi transmitters so it is not necessary to install additional hardware.

Cons:

- This method requires that the device has the wifi turned on, which may have a great use of power

4.1.5 Technology summary

The technology chosen for this project is the beacon technology. From the comparison between the other technologies, it was confirmed, that it was the best choice because it is the technology with the highest precision indoor and uses approximately the least amount af power.

The other technologies have an approximately lower precision, higher use of power or additional modules or antennas needs to be installed in the devices intended as clients, e.g. laptops and smartphones.

²In the study the standard GPS is called Autonomous.

³BLE includes beacon technology

4.2 System design

In order to describe, and thereby design the full system, all the use cases are explained with the components / technologies, they are supposed to work with. This design will be the core in the implementation.

4.2.1 Register person

As previously decided, the technology for locating a person indoor, is selected to be beacons using the Apple native data format (description found in later section 4.4 - Technical Details). The beacon technology is based on one-way communication. This means, that the client can only receive the beacon-data and then have to send it to the system. Since the client application is supposed to run on a number of platforms, due to the non-functional requirements, the interface is standardized by using a REST web-service which interfaces with the database of the system.

In order for the mobile device to send the collected beacon-data the mobile device needs an application. This application is designed to be headless since no user interaction is required.

The set up for the components of the system for registering a person, can be seen in relation to each other in figure 4.

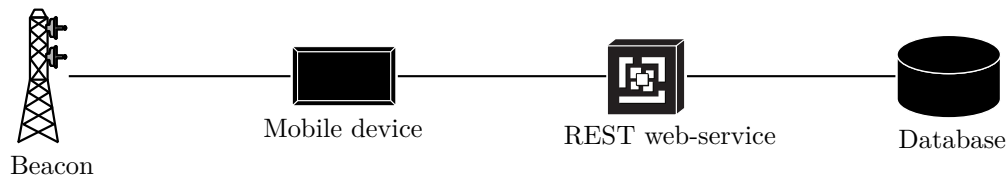


Figure 4: The components of the system, used to register a person

4.2.2 Assign person

An area is defined as a square with four corners (points). To know if a person is registered for the area, it is necessary to know if the person is outside or within the square describing the area.

In order to place the location of a person one needs to know the location of the person relatively from the surroundings. This is typically done by having a distance and a direction from a location of an already known item.

With the beacon technology there are only a distance read out without any specific direction. So if you e.g. got 30 meters to a beacon, you do not know if you are north, east, south or west of it. If one beacon is used to determine a registration-area (square) the system will get false positives from areas not within the square, see figure 5. The only description that can be made of the area is a minimum distance and a maximum distance, so a positive area would be registered between those two distances.

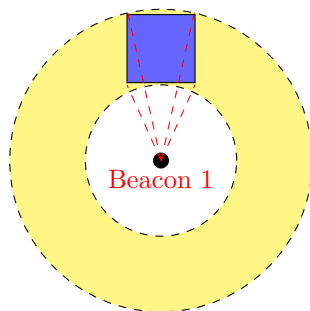


Figure 5: One beacon defining an area(blue) with false-positives area detection(yellow).

By using more beacons it is possible to get more information to find a more precise location of the person, reducing the area for false positives.

With 2 or more beacons there are provided enough information for calculating the relation of the points in the square to one another by using trigonometry. The beacon 1 and 2 (see figure 6) creates a triangle between the two beacons(line **S**) and one of the points in the square (line **B11** and line **B21**). This makes a triangle with 3 informations in form of the three distances. Now all angles in the triangle can be found using the cosine relation. The angel between **B11** and **S** is calculated.

The next step is to calculate the angel between **B12** and **S** using the three distances in the square that is formed by line **S**, line **B12** and line **B22**.

Then the angel between **B11** and **B12** is found, by subtracting the two previously found angels. This angel is used to find the distance between P1 and P2 by using the sinus relation. The rest of the distances between the corners in the area can be found likewise.

A challenge using this method, is that depending on were the beacons are placed in relation to the area, it affects how to calculate the angel to use in the sinus relation to calculate the distance. By looking at figure 6 it is seen, that the found angle between **B12** and **S** have to be subtracted from the angle between **B11** and **S** in order to get the angle between **B11** and **B12**. But in other cases, the two angles have to be added together, an example of this is shown in figure 7.

However it is possible to get a false positive area by mirroring the area on the **S** line. To remove the falls postives, 3 or more beacons are needed.

A full set of calculations for one square with two beacons can be found in appendix B.

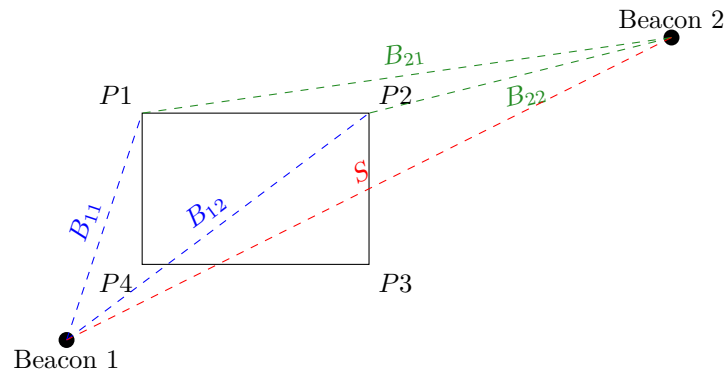


Figure 6: Two beacons defining an area. P1 to P4 is the corner points. The lines are named as Baecon<Beacon number><Point number>, so B12 is the first beacon and it goes to the second point.

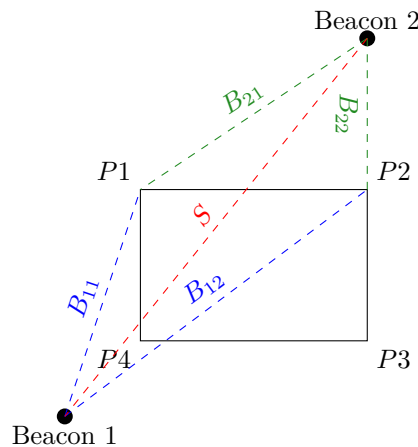


Figure 7: An area defined by two beacons where the location of Beacon 2 is changed compared to figure 6. This means that the two angels with Beacon 1 has to be added instead of subtracted as for figure 6.

Why not triangulation? Triangulation as it is known from the car navigation system can not be used for this project since no angels of the triangles are known. In this project only the distances for part of the triangles are known and without the angles it is distance without direction. The triangulation requires one known distance and two known angles for two corners in a triangle and can therefore not be used. But the calculation method described above requires 4 known directions and a coherence between those 4 points, so by also measuring the distance between the beacons, this method could work.

The method for calculating the size of the area is chosen to be the system based on using one beacon, see figure 5, but the interface for the implemented solution is set up for the two beacon model, see figure 6, so when the challenge with the calculations are solved, it is simple to implement the solution in the final product.

4.2.3 Configuration of area

When the persons are to be located by the system, it is done in different areas of the building. To display these areas, maps of the buildings and floor plans are used. Since the data from the beacons are in meters, and the maps are in pixels, a scale has to be made for the map, before an area can be created.

To present data for the user, the data must be in relation to the real world. Therefore the map, which is put in to the system, have to be put in to scale to get a meters pr. pixel count, in order to place real world data on the map. This is done by having a fixed measuring stick for scaling of the map itself, selecting a start and afterwards a finish point of the line, put in the number of meters, this distance represent, and save it by storing it in the database. Now the map is put in to scale. A mock-up screen for the map-calibration can be seen in figure 9. Once the map is calibrated, this step is not necessary in order to put a new area on the same map.

To create an area a map is selected. The map has to have a meter pr. pixels scale attached. To create the new area, the start-coordinates of the area is selected based on a user-click on the map and after that, all the beacons that is defining the area are selected. The area can then get assigned a name.

The beacons has to reach the entire area for this to work. From each selected beacon, four boxes, one for each corner appears and the distance readings of the beacons can be entered in their respectively box. If there are more than one beacon, the distance between the different beacons are put in as well. When all data is put in and saved, the area can be displayed in the Show data use case. A mock-up screen for the area configuration is shown in figure 8.

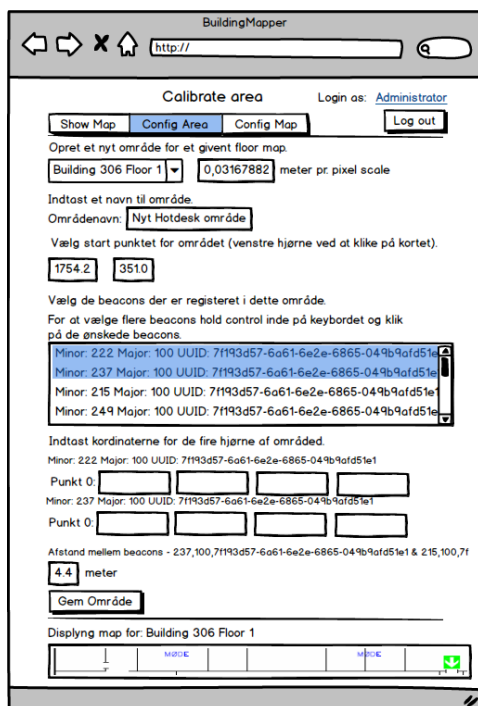


Figure 8: A mock-up screen for the web page used to configure an area on a map.

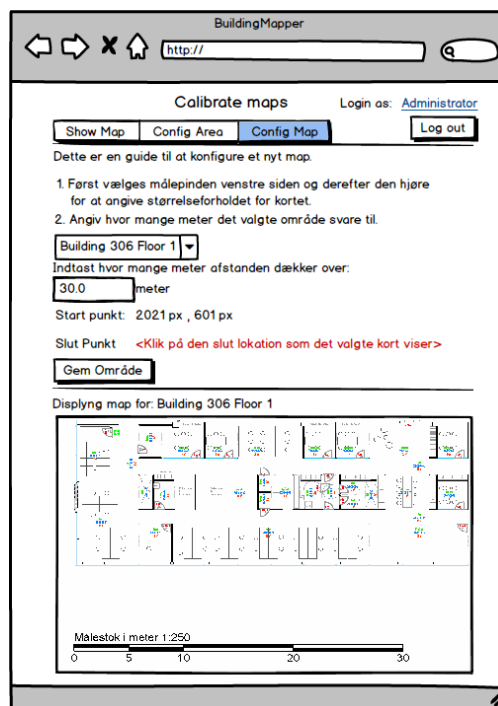


Figure 9: A mock-up screen for the configuration page of the map.

4.2.4 Show data

When the users have been registered the data have to be represented in a realistic context so the user can benefit from it. To fulfil that purpose, the user can go to a website, choose a time period and a location map and get the data by pressing the Get Data button.

The website interface consist of three zones, each with their own purpose. These zones are the main structure for all the websites in the system.

The first zone is the navigation. It contains menus to switch around in different parts of the system and a login form that can give an administrator access to more features.

The second zone is the individual website configurations. It is here the timespan and map selection is done, in other words, all user inputs needed to show the wished data, is inserted by the user here.

The third and last zone is used for displaying statistics and floor maps based on the users input. A representation of the three zones can be found in figure 10.

In figure 11, a mock-up of the main screen with the three zones is shown. In zone 3 there are also added a statistical graph of person registration in order to have the total person count over a selected period.

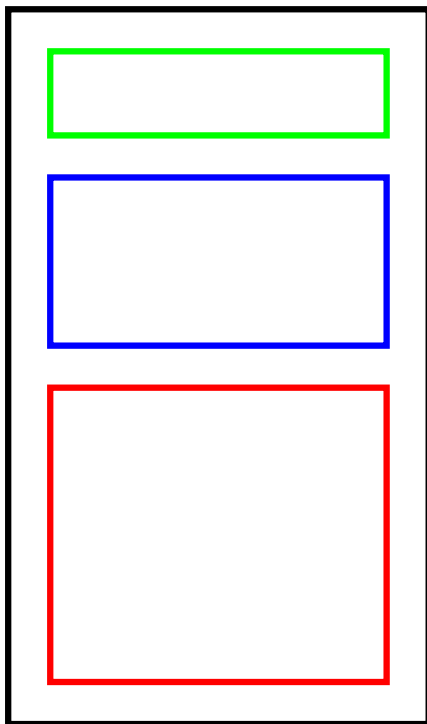


Figure 10: The web-page model (black frame) split into 3 zones. First is navigation zone (green), second is configuration zone (blue) and last the output zone (red)

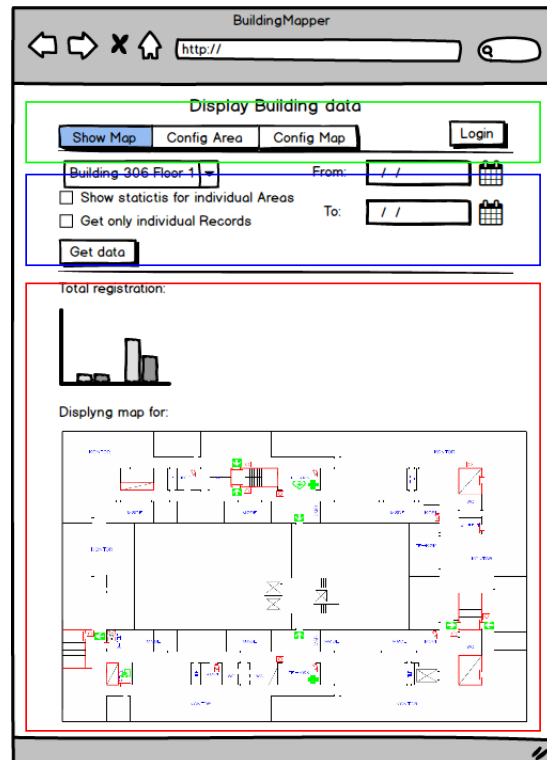


Figure 11: The Main screen the user will see.

4.3 Interface design

To know how the system components should communicate with one another the interface between them is defined. The full system run through with interface labels can be seen in figure 12.

In the first of the use cases (Register person), the beacon sends a signal to the mobile device with its bluetooth. This signal follow the patterns from the technology description in section 4.4 - Technical Details and the client have to receive that information. This depends on the client hardware model and OS.

From the mobile device the data is passed along to a web-service, but the technology of the web-service have to handle the data fast due to the amount of potential users contacting the web-service at the same time.

Studies shows [34] [11] that if comparing the two big web-service technologies REST and SOAP on performance, the REST technology on Android is better than SOAP, when talking about speed. But the REST uses a little more memory on the Android device, around 10% approximated 18KB more than SOAP when handling text. For audio and images memory usages increases [34, figure 2].

By adding JSON to REST web-service instead of the XML(REST Standard), the speed can be increased even further. By taking a look in table 9 at the results from the study [11] it shows that an Android device using REST-JSON is 9 times faster than SOAP and more than 6 times faster than REST-XML. For the Iphone 4S the REST-JSON is 30 times faster than SOAP and more than 15 times faster than REST-XML.

The results shows that the speed depends on specific device model and OS. The first study [34, figure 5 and 7] shows that when using a Windows Phone, the speed and memory usages is the opposite compared to Android and Iphone, so SOAP is fastest but uses more memory on af Windows Phone.

Due to speed, it is selected to use the REST-JSON for the web-service.

Device	REST-JSON (ms)	REST-XML (ms)	SOAP (ms)	REST-JSON Faster than SOAP
iPad 2	187	3301	5109	27x
iPhone 4S	320	4973	9694	30x
Samsung Galaxy S4	158	1019	1465	9x
iOS Simulator	151	585	462	3x

Table 9: The table shows the time in ms used for the three different technologies. The Simulator is not commented since it is not a "real" mobile device. Table data copied from [11].

The communication from the web-service to the database is based on a native ODBC SQL driver [36]. The system database is created in Mssql.

When a person uses a web browser looking for e.g. a free seat in the website(Use case Show data), this is done on a web-server using the HTTP or HTTPS protocol. When collection data for the website from the database, the ODBC SQL driver is used again.



Figure 12: Overview of system components with their selected interfaces attached.

4.4 Technical Details

When a user receives data from a beacon (see table 10), it is in the iBeacon-format as Apple inc.TM has defined. Another beacon formate is the version that Google TM has made, called Eddystone. The Eddystone formate [3] [6] offers four different modes, one of them is Eddystone-TLM. Appart from the iBeacon data, the Eddystone-TLM broadcast information about itself like battery voltages, room temperature and number of broadcast it made. For the project is chosen to use the native iBeacon-format, since it gives the possibility to select beacons from a wider Range of beacon products.

Data field Name	Description	Data format
Address	The MAC adress of the bluetooth unit inside the beacon.	String
CalibratedTxPower	The power level of the signal used to calculate the distance between the beacon and the client device.	Integer
ApproximateIos-Range	Range calculated for an IOS device Provided by the client library used for the client.	Double
CompanyId	Company identifier code. For Apple company the code is 0x004C.	Integer
IsAppleIbeacon	This datafield defines if the beacon signal comes from an Apple product or not.	Boolean
Major	The Major version of the beacon. Typically used to define a group of beacons used for one system or area.	Integer
Minor	The Minor version of the beacon. This is the identification number of the unit which makes it unique.	Integer
Range	The distance from the beacon to the client. Represented by decimal in meters between the units.	Double
Rssi	An abbreviation of Received Signal Strength Indicator [39].	Integer
Uuid	The highest version number of the broadcast number from the beacon. This is normally used as an identifier for the company where the beacon is set up.	String

Table 10: Data a beacon can deliver to the system [8] [40].

In addition, the client have to provide an identification in order to identify the users apart and a timestamp for when the register has been created. This is defined in table 11.

Data field Name	Description	Data format
TimeStamp	The date and time for, when the user has sent his registration of beacons.	DateTime
ClientId	In order to identify the users from each other, a hash of the MAC address of the bluetooth, is used as an identifier, in order to anonymize user data.	String

Table 11: Additional data added to a set of beacon data from table 10 to create a full person registration.

4.4.1 Frequency of collecting beacon-data

When collecting data from the beacon in order to get the user location, more than one beacon signals are needed. The more signals the higher precision it is possible to get.

Since beacons does not broadcast there signals at the same time, the system needs a timespan to register all possible beacons for one registration.

In that timespan the user has the possibility to move, and thereby move between the receiving beacon signals which creates uncertainty. Position-data is calculated from all signals as if they all where registered simultaneously, so if a person moves during the timespan, some of the data will place the person in one spot, and some in another spot which together will give some uncertainty to the spot registered in the system. Therefore a maximum distance the person can move within the timespan, so the uncertainty does not become too great, is defined.

Studies [22] shows that the standard value for walking speed of a person is 1 meter/sec. The beacons [19] used for this system can broadcast / advertise as fast at 20 times per second. The default advertising is ones every 350ms so this is used as the worst-case scenario.

$$\frac{1 \text{ time}}{350 \text{ pr. ms}} * 1000\text{ms/sec} = 2.85714285714 \text{ times pr. sec.} \approx 2.86 \text{ times pr. sec.} \quad (1)$$

$$\frac{1 \text{ meter pr. sec.}}{2.86 \text{ times pr. sec.}} = 0.34965034965 \text{ meter} \approx 35.0 \text{ cm} \quad (2)$$

These calculations shows that in the worst-case scenario, a beacon can broadcast 2.86 times pr. second (equation 1), and that means that the user can move a maximum of 35.0cm between beacon signals received(equation 2).

The best-case scenario for the Beaconinside BEACON model is broadcasting every 50ms(it will decrease battery lifetime) and that means it can broadcast 20 times a sec. This gives a maximum moving distance between broadcast to be 5cm(equation 3).

$$\frac{1 \text{ meter pr. sec.}}{20 \text{ times pr. sec.}} = 0,05 \text{ meter} \approx 5.0 \text{ cm} \quad (3)$$

The results shows that the system will have higher precision by increasing with a broadcast frequency to once pr. 50ms, but that will reduce the beacon battery life time.

In real world setup it is recommended [35] that the broadcast time is 100ms for scenarios where people are tracked walking around stores with beacon. This results in a maximum move distance approximately 10cm.

In the end, due to battery life time and because it is used for people sitting, it is chosen to keep the broadcast rate to the default at 350ms.

4.5 Domain model

In order to define the terms of data and their relationship, the domain model is drawn for the system. Within the domain model is also defined the datatypes of the terms, which is used for the final project. The domain model is based of the technical details of beacon and the system design. The domain model can be found in figure 13.

The beacon-data consists of four identifiers (Uuid, Major, Minor and Address), the transmission setting (CalibratedTxPower, Rssi), the manufacture data (CompanyId, IsAppleIBeacon) and the distance calculated for Apple and non-Apple devices (ApproximateIosRange, Range). A detailed description can be found in table 10.

A Registration is an unique identifier of the client (ClientId) and a timestamp for the registration (TimeStamp). A Registration can contain one or more beacon-data based on how many they have encountered in a given time interval.

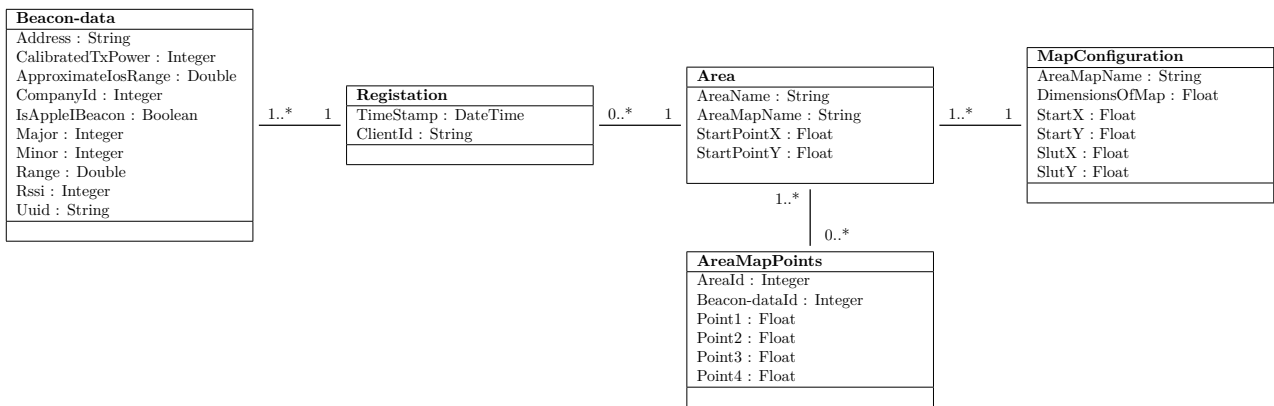


Figure 13: The domain model contains the terms of the project and the relationship between them in the system context.

An area have a name (AreaName) and the name of the map it belongs to (AreaMapName). Further more a set of start coordinates given in pixels, which is the start point of the area. An area can hold from 0 to many registrations. Points (Point1 - Point4) defining the area are added together with the area it belongs to (areaID) and which of the beacons the point reading are provided from.

A map(MapConfiguration) can hold none to many areas. A map consist of a name(AreaMapName) and a calibration of the map, which is a start(StartX, StartY), stop point (SlutX, SlutY) and a real distance between the two points (DimensionsOfMap).

5 Database

With the domain model, figure 13 and the technical details the database model, see figure 14, can be created. When a user is registered a record containing timestamp and unique ID is stored with a list of beacon-data. This gives a relation between "beaconregistration" and "beaconfound" as one beacon registration can have one to many "beaconfound".

The full database creation script with database-user and user rights automatic set-up can be found in appendix C.

The table "beaconfound" in figure 14 should originally have been split up into two tables due to normalization, but for keeping the figure more simple, it is chosen to have it in one.

A representation of the database fulfilling the normalization and a more consistent naming can be found in appendix D.

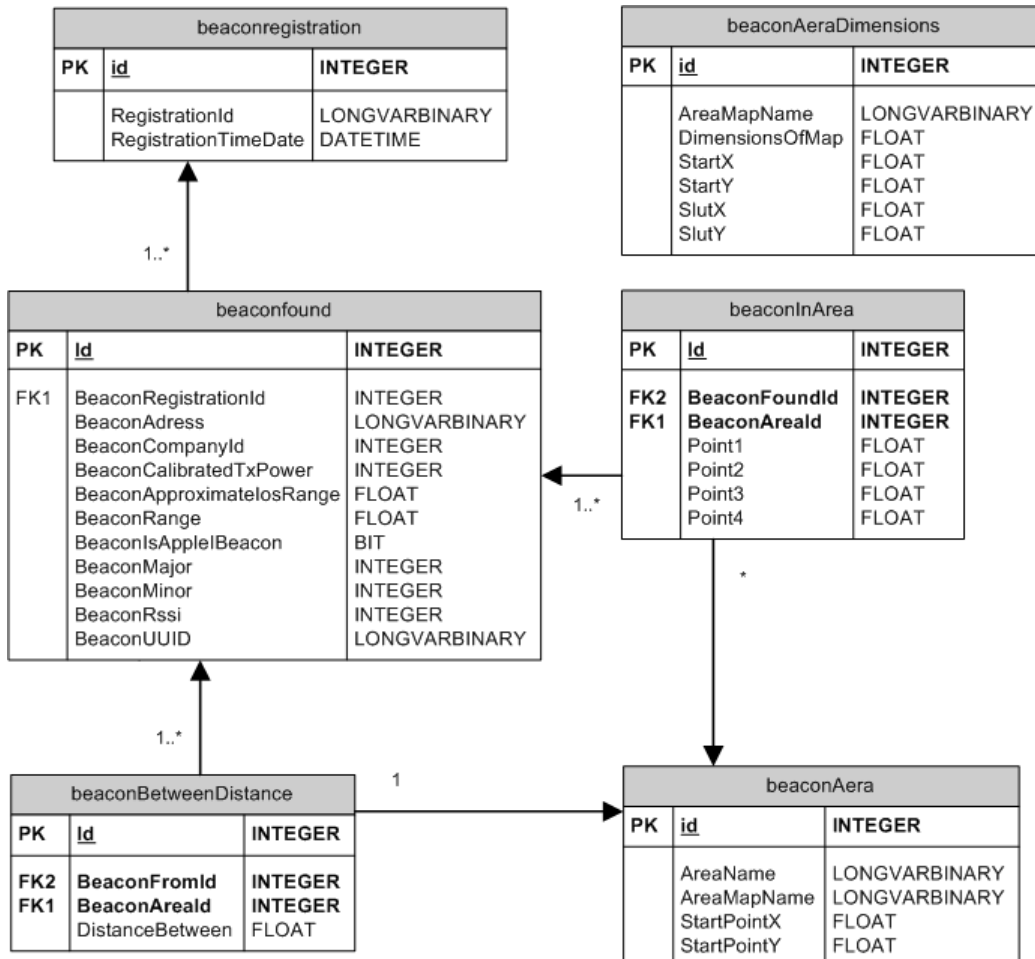


Figure 14: The database model with datatypes nullalbe. Not-nullable data fields is marked with **bold text**. Cardinality are 1 = 1..1, * = 0..1.

6 Implementation

The implementation of the system components is where the design of the system is put into reality. The goal is to explain and show methods library and implementation which have a special influence for this project.

The following section is split up into sub-sections of each of the system components.

6.1 Beacon

In order for the user to be registered at a location they need to receive a beacon signal. The beacons used in this project comes from the company Beaconinside GmbH [19] and they satisfies all the requirements for the project, see section 3.5 - Requirements. Especially the power source since it both can run on batteries and by power supply. Further more this type of beacon also supports the beacon standard format created by Apple and the Google framework for beacons called Eddystone. A picture of the beacon model can be seen in figure 15.

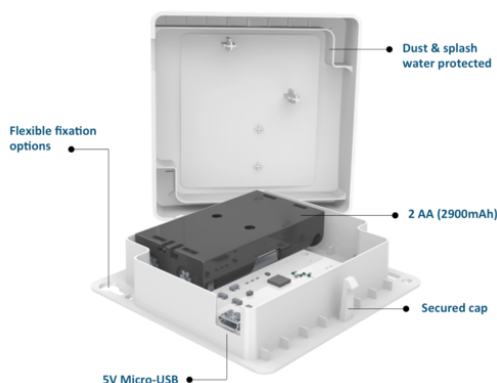


Figure 15: The Beaconinside BEACON model used in the project. It is open to show the battery pack (grey) and the white printed circuit board which is the beacon transmitter itself.

Since beacons are not locked to specific platforms, this project started out by having emulating a beacon transmitter running Linux and having a bluetooth dongle version 4.0 inserted. Later this was also implemented on a Raspberry Pi and another one-board computer called Intel Edison. Installation guides for these set-ups can be found in appendix E.

6.2 Bluetooth client

For the users to store their registration-data in the system they need an application on their mobile devices, used in their normal day of work. These devices can be either a smartphone or a laptop. A requirement is that the mobile device must have a bluetooth hardware device in the class of version 4.0 or higher.

This implementation is done for two device-types, an Android smartphone and a laptop running Windows 7. The smartphone OS version is to be a minimum of Android version 4.3 Jelly Bean (API level 18) [18]. Both implementations are done in C# with the additional help from third-party libraries. The first is used to serialize the data at the client to be send to the web-service where it is de-serialized. It is done with the Json.NET library from Newtonsoft [37] and is used in both clients, both Android and PC client. The rest of the libraries are described in the sections about the client where they are used.

The client software of the system works in a certain pattern. The main functionality of the client remains the same independently of platform(OS) or device. A full activity diagram [47] of the main functionality is found in figure 16 and described below.

When starting the client, the client starts by creating an identifier for the user by getting the MAC-address of Network card(PC client) or the bluetooth chip(Android client). This MAC-address is hashed in order to anonymize the user. The hashed MAC-address is now used as a unique identifier for this mobile device. All data send from this device will be tacked with this unique ID.

The software now starts listening for beacons and at the same time create a time event and the following processes happens simultaneously. When the software receives a beacon signal, it stores it in a list and keeps receiving incoming beacon signals. The time event is triggered one time every 350ms where it runs a sending procedure. The sending procedure takes all the found beacon-data from the list, serialize them and tries sending them to the REST web-service. If the sending succeeds, the list of beacons is reset and the system starts searching for beacons again. This is a successful run-through of the main scenario of the client.

If it fails to send the data, the list is kept and the system starts to search for new beacons and just adding them to the same list. This way all the data would still be gathered and resend when the connection is re-established.

An activity diagram of the full process for collecting and sending beacon data to the REST web-service can be found in figure 16.

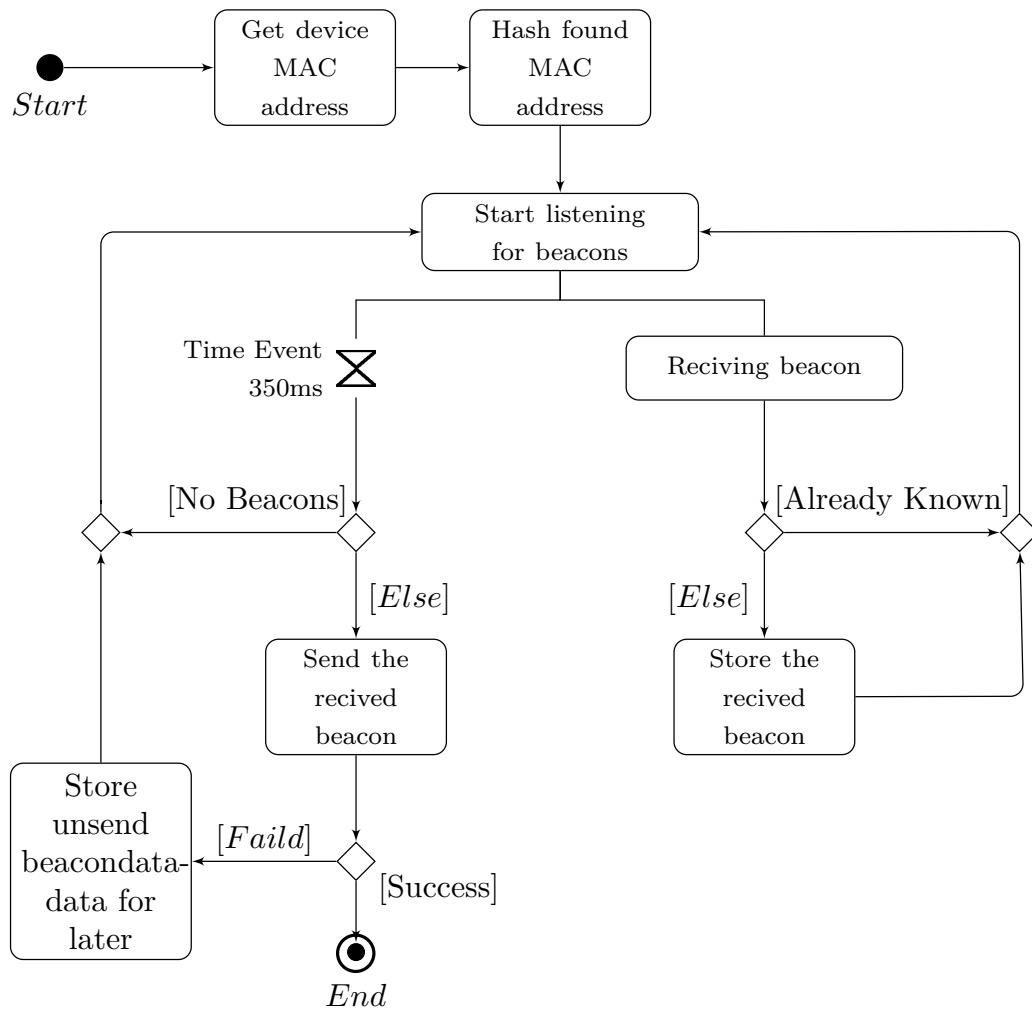


Figure 16: The activity diagram describes how a client receives beacon signals repeatedly and after a time trigger every 350ms sending data to a web-service. A successful passes through of the activity diagram goes from start to finish, but the End-parameter runs back to the "Start listening for beacons" in real scenario.

6.2.1 PC Bluetooth Client

One type of client is created for a laptop in order to receive a beacon signal and afterwards past these data to a REST web-service. The laptop client is programmed in C# and gives the possibility to locate the position of the laptop, and thereby an approximately position of the person using the laptop.

The application is supposed to be working on a laptop running Windows 7, which require third party software to make it work. In Windows 10 the possibility of receiving beacon signals comes native within the software with the library BluetoothLEAdvertisement-Watcher [32].

To receive beacon signals it is necessary to have two third party software components. The first component is a C# library called WinBeacon [25].

To make the WinBeacon library work on Windows 7, the original bluetooth driver is replaced with a generic USB driver(WinUSB driver) in order for the bluetooth device in the laptop to register the incoming signals. This is done with the Zadig tool [5]. This way the interface for the bluetooth module is standardized so the WinBeacon library can handle the incoming bluetooth signals.

Since the client application is supposed to be headless, it is only supposed to show up as an icon in the windows taskbar just left of the clock. But to show how it is working a console screen shot from the implementation is shown in figure 17.

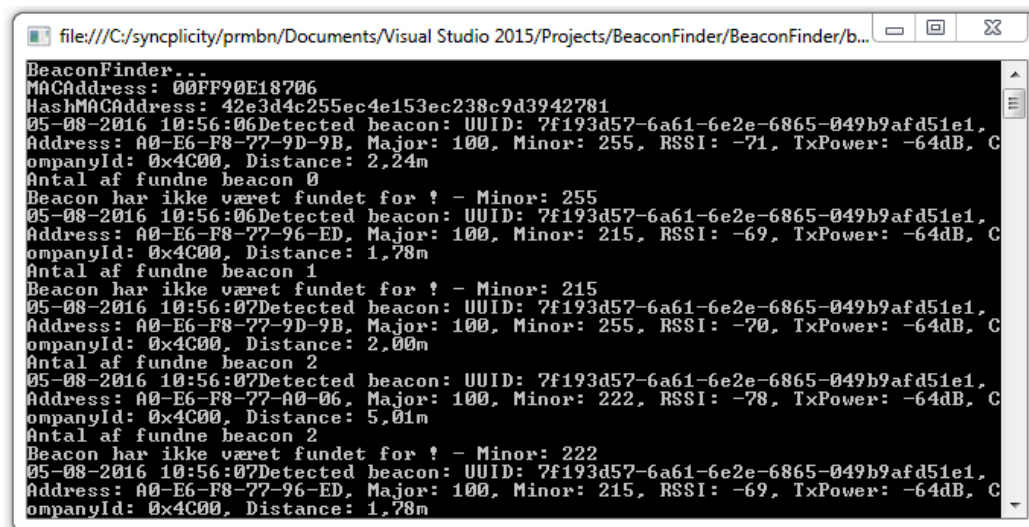


Figure 17: A screen-shot of the console running the PC client software.

A disadvantage for this set-up is that the originally bluetooth driver for the computer have to be overwritten with a standardized USB-driver in order to fetch the incoming beacon signals. This means that the user can not use the internal bluetooth module as normal, such as wireless headphones or a computer mouse using wireless bluetooth communication. This can either be solved by adding a bluetooth USB dongle such as the Asus Bluetooth 4.0 USB Adapter [27] or use a Windows 10 OS where this functionality can be accessed in the already existing bluetooth driver.

6.2.2 Android Bluetooth Client

To create an application users have on their phone, the Xamarin Project [33] from Xamarin Inc. is used. This gives the benefit of using the same C# code for the main functionality since Xamarin writes its Android applications in C#. It should also give the possibility of creating iPhone apps from the same C# code if that is needed.

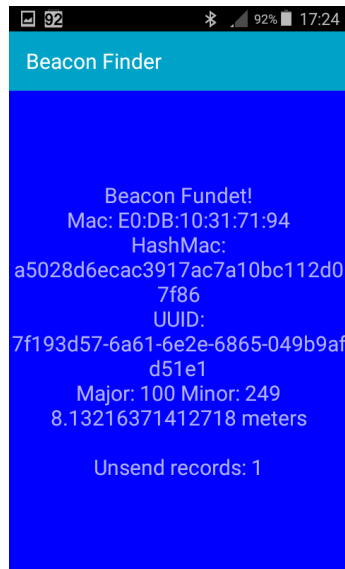
The project is based on the Android iBeacon App - FindTheMonkey [23] for setting up the bluetooth beacon receiver and the screen layout part of the project. The 4 main screens of the Android client application can be found below in figure 18, but is only for explaining how the application works. It will not be in a real world application, since it is supposed to be headless. The 4 screens show the main step of the client states as shown in figure 18. The first screen displays the found beacon data and if the device has any un-send records. The second screen (green) displays when the mobile device is sending data to the REST web-service. If there is any un-send beacon-data the third screen (purple) is displayed when it is trying to resend the un-send data. If an error occurs a red screen will show the error to the user.

In order for the Android app to work with the main functionality as intended, the user has to accept the app getting access to different parts of the smartphone. The list and reasons can be found in table 12.

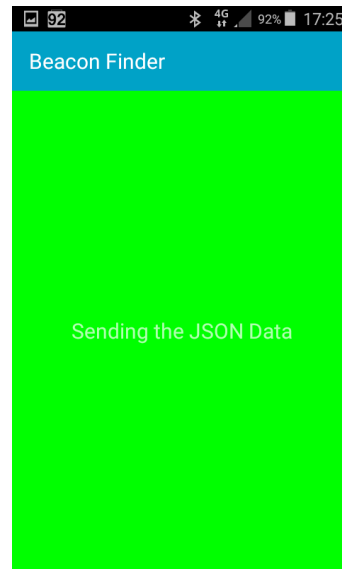
Android Permission name	Description
BLUETOOTH	This permission is required to access and use the functionality of the bluetooth module on the Android phone.
BLUETOOTH_ADMIN	Needed for the application to discover devices (find beacon) and modify the bluetooth settings from within the application.
INTERNET	Let the application gain access to the internet. This is needed in order to send beacon-data to the web-service from the client.
READ_PHONE_STATE	Make it possible to read the states of the Android phone.

Table 12: In the table is listed the Android permission needed to have the Android client application working as intended.

A disadvantage may be that the Android smartphone needs to have turned on bluetooth all the time and have an application running constantly in the background too. This may drain the battery more than with normal use. How great the disadvantage might be, is tested in the section 7.3.2 - Battery test.



(a) Screen display found beacon and smartphone data. (from top to button) Smartphone MAC, Smartphone Hash MAC, BeaconUUID, Beacon Major no. and Beacon Minor no.



(b) Screen display when the collectet beacon-data is send to the REST web-service to store data. Screen text saying "Sending the JSON Data".



(c) Screen display when there are unsend beacon-data and the application is trying to re-send them to the REST web-service for storage. Screen text saying "Sending unsend JSONData 0 of 1".



(d) When the system get an error on client-side, it is displayed on a red screen with the error. In this case the connection is lost to the server(REST web-service).

Figure 18: Screen-shots of the client application running on Android phone.

6.3 REST web-service to database

The REST web-service is, as mentioned earlier, the universal interface to the database for the client. The web-service is based on the visual studio, WCF⁴ framework and used to build, configure and deploy network applications and services. [29]

A full circle run for the web-service starts with a mobile device wants to store its beacon-data. The client wraps the data in to JSON and then serializes it using the library Json.NET from Newtonsoft for higher performance [38]. The client then send the serialized JSON via a HTTP-post request where it is send to the REST web-service interface called "mobilereg-isteration".

When arriving at the REST web-service the data is de-serialized and from the REST web-service the data is stored in the database. When a user wants a representation of the data via a web-browser, the web page gets the data directly from the database. All database handling is done with the native SQL library for visual studio.

A figure with the full circle of data from the client to the REST web-service can be seen in figure 19.

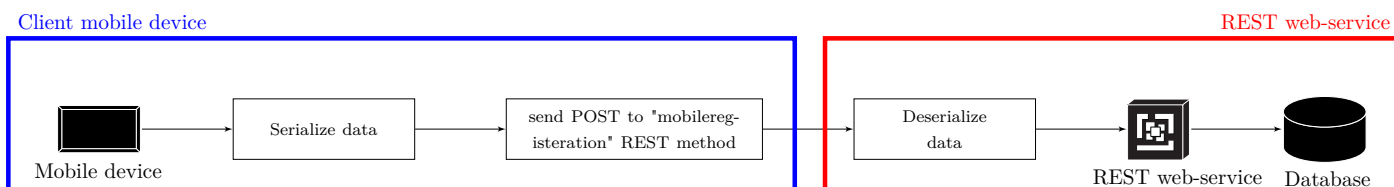


Figure 19: Transaction of beacon-data between the client(mobile device) and REST web-service with serialization and de-serialization of data.

⁴WCF is short for Windows Communication Foundation.

6.4 Show data - web-interface

The collected data is presented to the user in a web-browser in order to have universal access. This means that no additional software has to be installed. The web-interface is based on Visual Studio MVC model.

The MVC model provides three layers of data handling for each page as listed below. A representation of this can be found in figure 20.

- The first layer is the website itself based on HTML, CSS and JavaScript. It is these technologies that allow for validation of the content of the page and other things such as generating additional text-boxes on the page.
- The second part is the control layer where C#-functions are enabled to handle and manipulate code. The C# code is called either when the page is loaded or when a submit is called from the page and enable the C# code behind.
- The third layer called *.aspx.designer.cs is a visual representation generated based on the default.aspx.

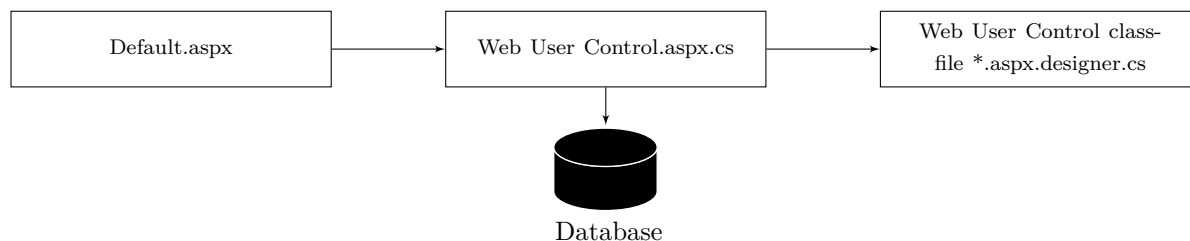


Figure 20: A graphical representation of the cohering between the files, that altogether generates the look and functionality of the webpage using the MVC-model.

But how does a full run-through of the implemented system actually works? The run-through is divided into the three subsections the web interface consists of, Configure Map, Configure Area and Display Data.

Configure Map

As mentioned in section 4.2.3 - Configuration of area, a map needs to be put in to scale by creating a relation between meters and pixels. This is done by selecting the map, click at the beginning of the scale meter, typically marked with a zero, and then click at the end of the scale meter. Both times a message box is displayed asking the user if the selected point is correct. Now the real distance in meters has to be written in the text-box and the configuration is completed by clicking the button "Gem Område". The implementation is seen in figure 21.

Configure Area

When the scaling of the map is completed, it is possible to configure an area on that map. This is done by first selecting the map and if the scaling exist, it will appear on the right side of the map selection. Now the user click on the map in the bottom of the screen where the top left corner of the area, approximately should be. The result is added as X and Y coordinates in pixels in the interface. The user now select one or more beacons and inserts the distances for each corner of the area. The distances are measured with a client application made for administrators which is not designed

nor implemented. Alternatively there are some already existing Android apps which can give distance readings to beacons [15]. Finally to create the area with the set configuration, the button "Gem Område" is clicked. The webpage set-up can be seen in figure 22.

Display Data

When the user want to display data e.g. show statistics for an area, the Display Data page is selected. Here the user have to put in, in which period of time he / she want to se data from. These periods are given in hole days. Then the user have 2 choices, either to see all collected registrations(one user can have hundreds) or only the individual records. At last the user have to select if he / she want to se registration statistics for all areas or just for the map chosen. To get the data the user have to press the "Hent Data" button. The output of the data request is now showed on a floor plan in the lower part of the webpage. The areas are marked with green and holding a name and afterwords a count of registrations or individual persons in the area for the given day. This can be seen in figure 23.

Please notice that in the three figures 21, 22 and 23 of the web-interface there can be seen a login form. The login functionality is not implemented in the system.

One of the challenges that have not been resolved is that the scaling of the various floor-plan maps are used in the interface as it can be seen in figure 23. This means that the user must scroll around in the web browser and have a hard time navigation the interface.

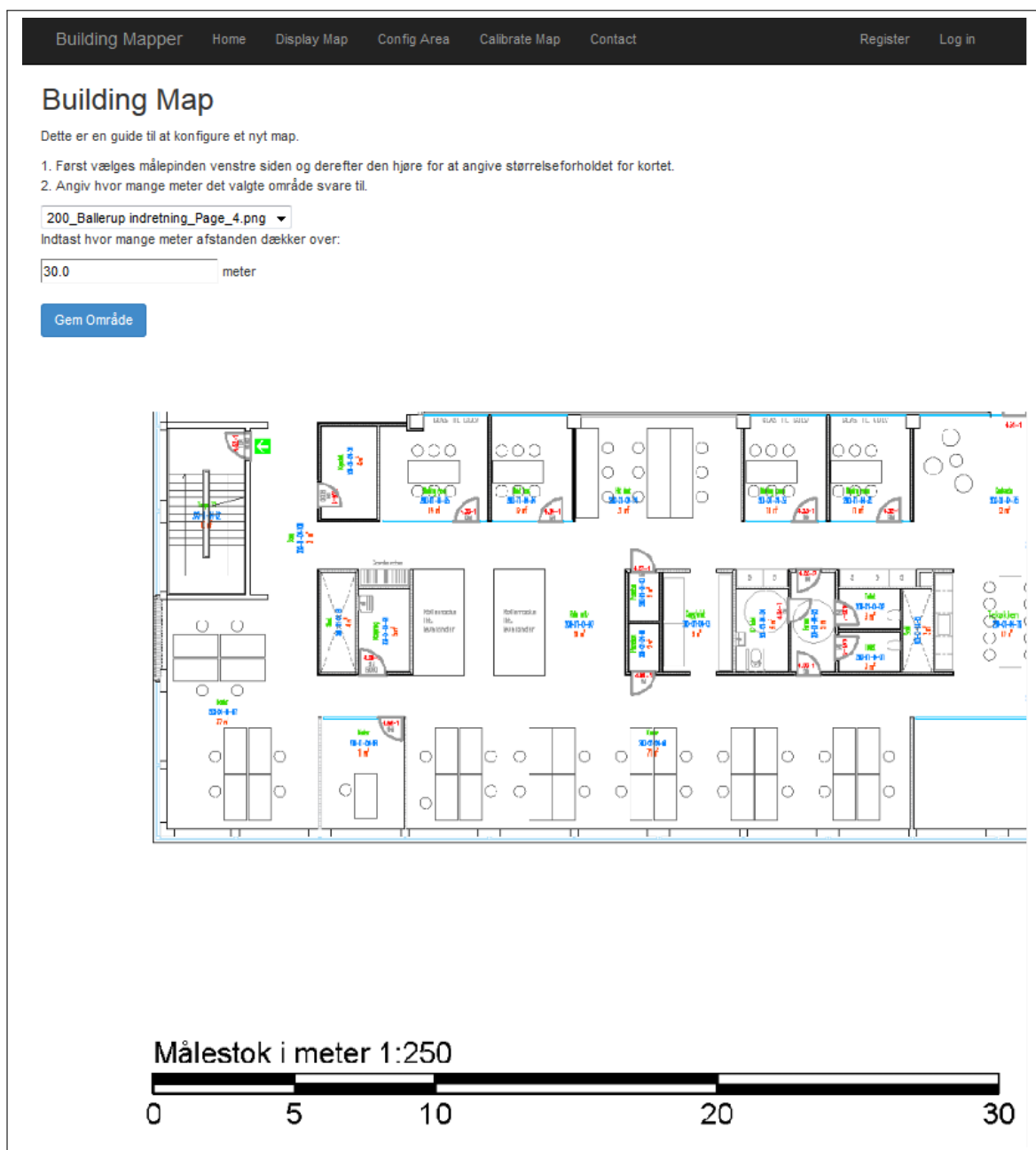


Figure 21: A screen shot of the webpage where a map defined in pixels are put in to scale with the metric system of the real world. The figure fulfils the zone set-up as shown in figure 9.

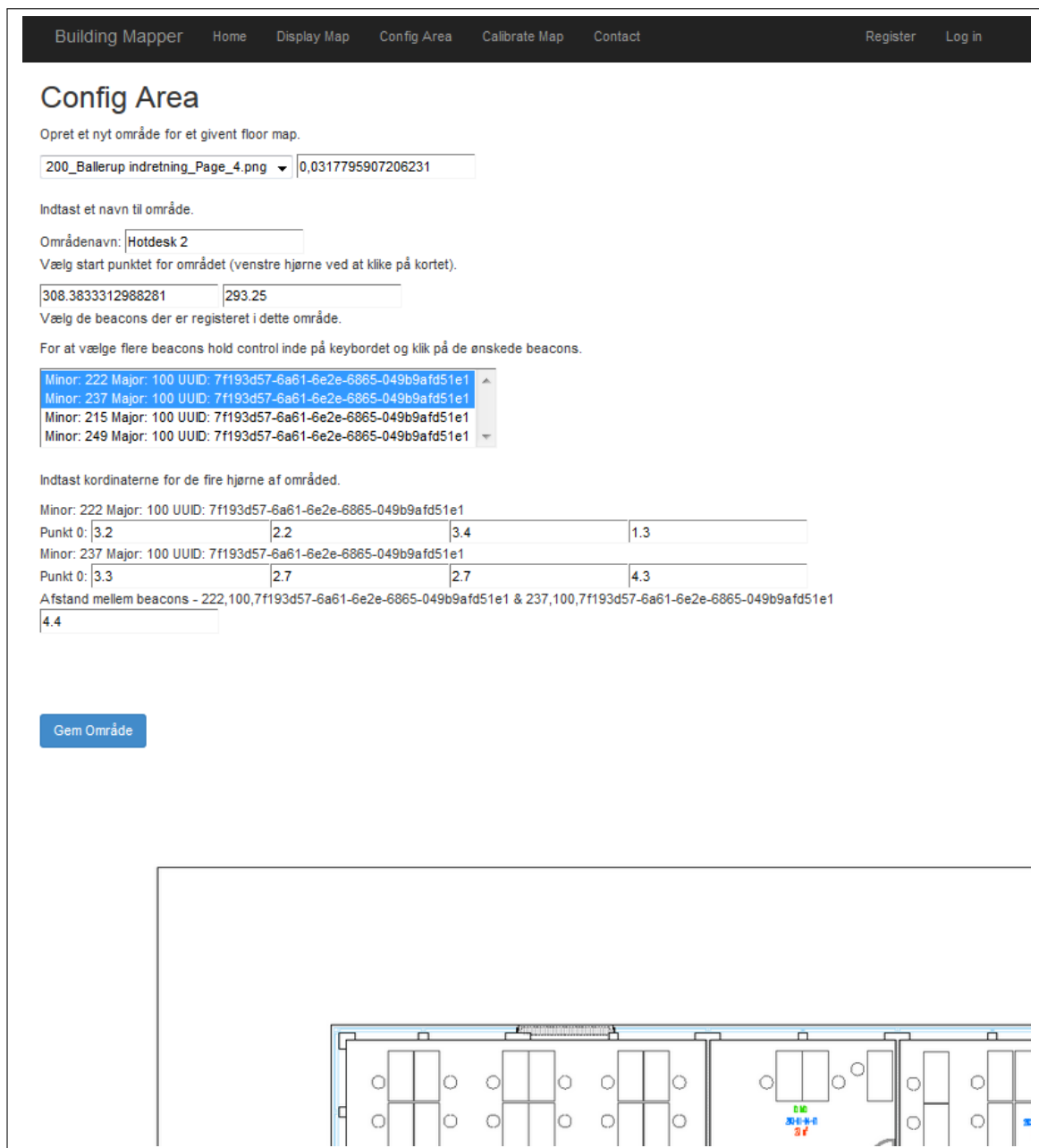


Figure 22: A screenshot of webpage called display data from the Configuration of area use case. In the web-interface is displayed how to select a floor plan from the drop down menu and afterwards add all the beacons which can be reached in the area. The figure fulfils the zone set-up as shown in figure 8.



Figure 23: A screenshot of webpage called display data from the show data use case. It is possible to insert the data for the period in which data to display, say if one will see all records or only the individual users and whether to display individual statistics for each area. The magnification shows an area named Hotdesk 2 where there have been a maximum of two persons for the displayed time. The statistics for the area is shown in the light blue diagram above the floor plan, to the right. The figure fulfils the zone set-up as shown in figure 11.

7 Test

In order to confirm that the system works as designed and fulfil the functional and non-functional requirements, a test plan is needed. All the functional and non-functional requirements are listed in section 3.5 - Requirements. Some of the requirements has to be technical tested in order to verify if they are fulfilled. This section is about these technical tests. The list of requirements to be tested and a description can be found in table 13.

Functional and Non-functional requirements	Test Description	Test name
All implemented functional requirements	To make sure that the product is fully implemented an acceptance test is made.	Acceptance test
Assign Person (FR)	One of the core functionalities of the system is that the precision of the registered persons is high enough to place a person within an area.	Precision of beacon registrations (System test)
Low battery usages on client mobile devices (NFR)	For the system to be a success the mobile device of the user should have a low battery usage and thereby not disturb the normal behaviour of the user. The test is split up into a test of the client application itself and the battery usage of the device having the bluetooth module turned in constantly.	Battery test (System test)

Table 13: Requirements to be tested with a test name and type.

7.1 Test Area

The tests are made in an office area, which contains some of the objects that might interfere with the system and objects that one could expect to find in an office, where the system are planned to be used. It could e.g. be large plants, computer monitors, and passing co-workers.

In the test area three beacons are set up, that allows the user to get information of the distance between the mobile device and each beacon. To emulate a person using a mobile device, the testing mobile device is placed on a stool at 0,79 meter which are approximately the same high as the pockets on a person.

A floor plan with the location of the mobile device tested and the distance to each beacon from the device can be seen in figure 24. The distances are measured with a laser for precision referencing,

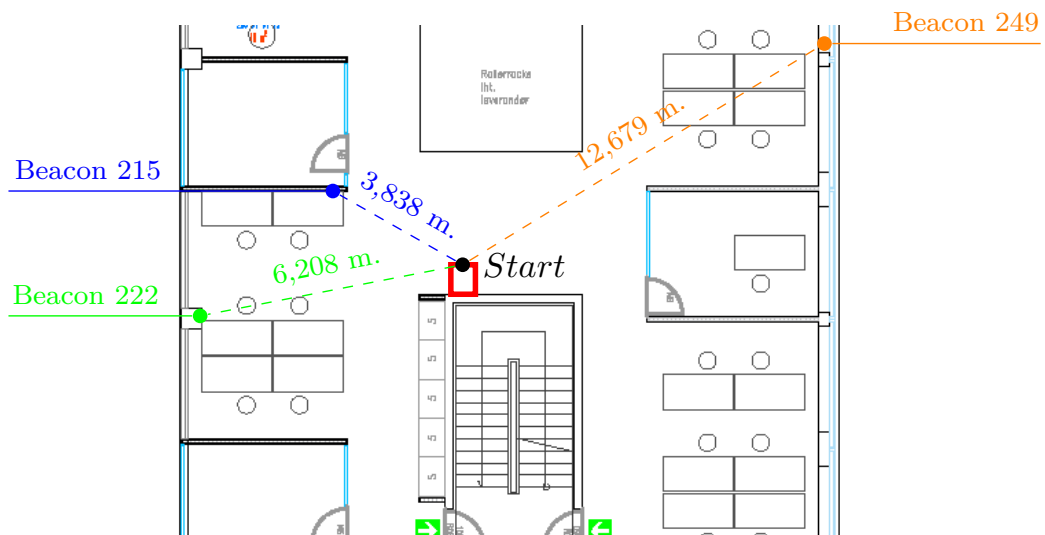


Figure 24: Floor plan overview with marking of the position of the mobile device and the distances to the three beacons. The mobile device is marked with a black spot in the red rectangle. The three beacons are marked with a coloured spot and their beacon minor number.

There are provided two pictures of the area which shows all the office equipment within the test area. The beacon model, see figure 15, is white and can be difficult to see in the photographs. The pictures of the test area can be found in figure 25 and 26.

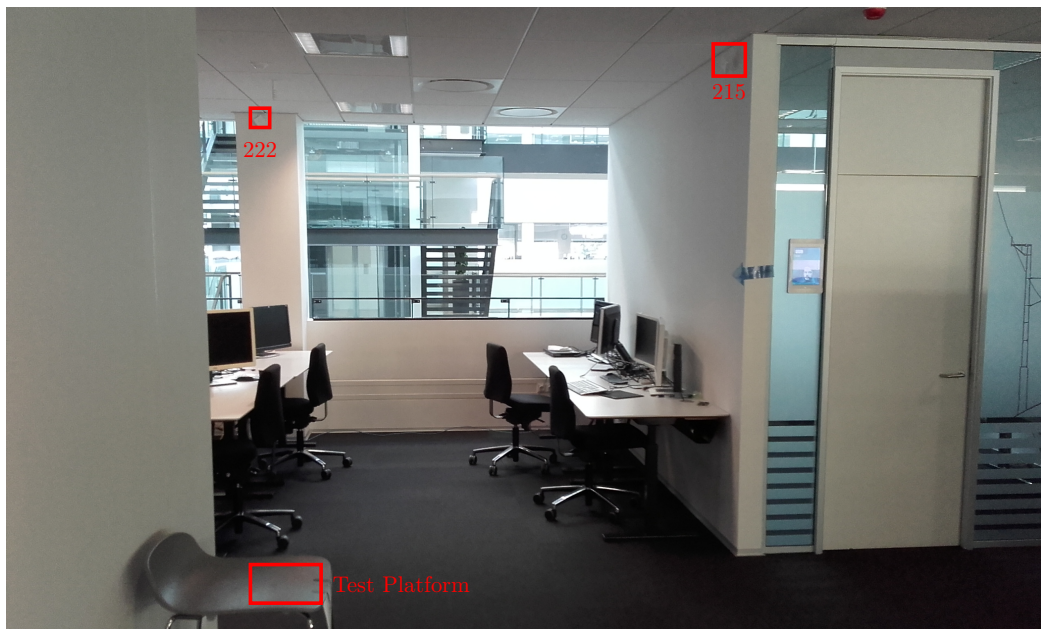


Figure 25: Picture of part of the test area showing the position of the stool holding the mobile device and the positions of the two beacons 215 and 222.

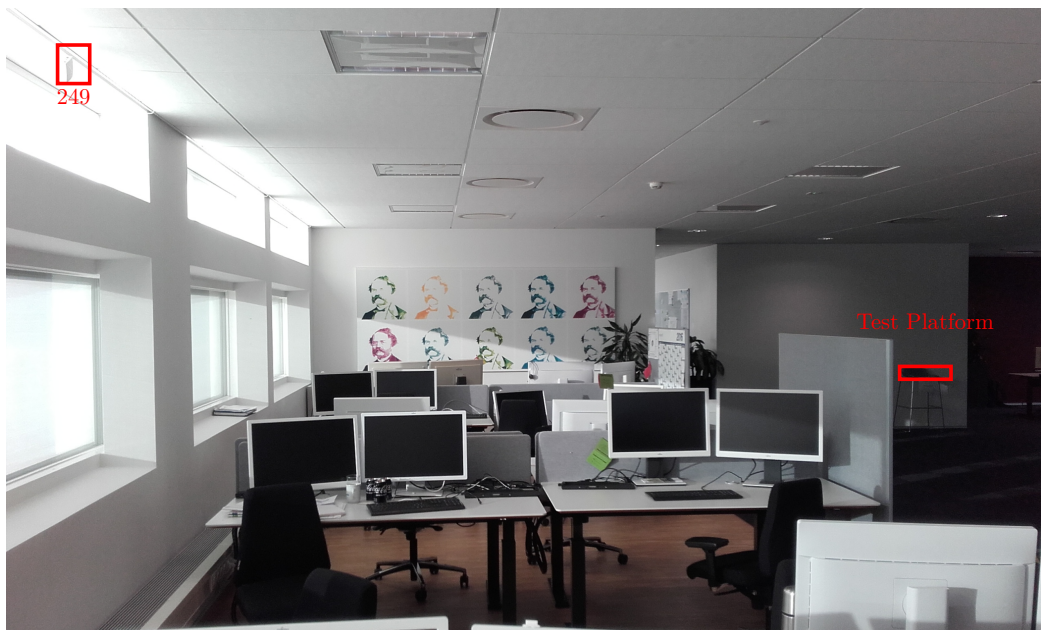


Figure 26: Picture of part of the test area showing the location of beacon 249. In the background the stool holding the mobile device can be seen.

7.2 Acceptance test

This test have the purpose of verifying that the system is correctly implemented according to the project use case descriptions. The acceptance tests are created as black-box tests and the expected outcome of these tests are based on the individual use case description status. E.g. if the status state that extension 1.a. is not implemented, this is the expected outcome of the test.

7.2.1 Simulation of the Acceptance test

In order to make this acceptance test, a test simulation program was programmed to remove the possible uncertainties, which can come when collecting beacon-data. In the simulation can be listed several persons, each person consists of name(used to make a unique user ID by hashing), if the person is going to lunch and which work area they stay in when working. Each person consists of an arrival time (random time between 07:00 to 10:59) and a leave time (random time between 14:00 to 18:59) for the work day. They have also marked if they are going to lunch for approximately half an hour or not. The work area consists of a pre-set of beacons defining the area.

The element of random time for the arrival, the leave and the lunch time is chosen to give a more realistic readout in the web-interface.

When running the simulation, all the simulated persons are generated and send to the Rest web-service as they where real incoming beacon registrations from mobile devices. This way a day of data can be generated for several persons in a couple of minutes and not a full days times of work. A graphical presentation of the simulation can be seen in figure 27.

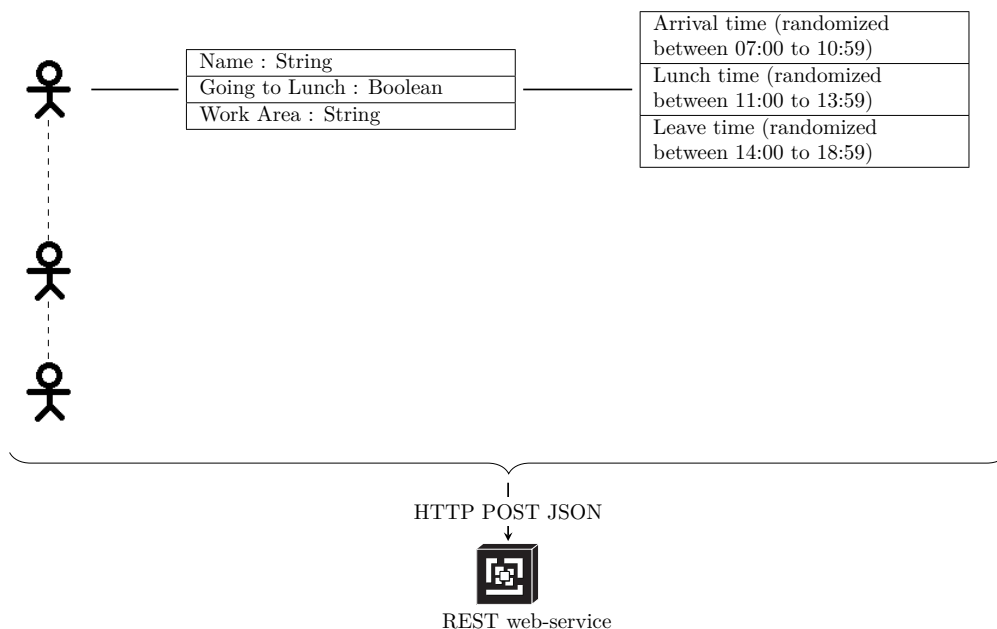


Figure 27: An overview of how a simulated person is structured data-wise and afterwards all the person data are serialized and send to the REST web-service to put them in to the system.

7.2.2 Acceptance test result

The main success scenario of all the use case descriptions has been tested with the described black-box method, they are listed in appendix F, and do all work as expected. Hereafter are all the extensions for each use case tested and listed below.

Further more the black-box test for UC1.1 is split up into two separate black-box tests, one for each implemented type of client. The extension black-box tests of the clients can be found in table 14 and 15. Note that it was found that the PC client was not stable through this test and may need to have a bug fix.

The black-box test for UC2.1 - Configuration of area is found in table 16, the black-box test for UC3.1 - Show Data is found in table 17 and the black-box test for UC3.2 - Assign Person is found in table 18.

Black-box for UC1.1 - Android implementation		
Extensions	Expected outcome	Real outcome
3.a It is not possible to connect to the system database.	- 3.a1 Beacon-data is stored in the mobile device and when the connection are reconnected, it tries to resend them.	As Expected.

Table 14: Black-box test for UC1.1 extension - Android client.

Black-box for UC1.1 - PC implementation		
Extensions	Expected outcome	Real outcome
3.a It is not possible to connect to the system database.	- 3.a1 Beacon-data is stored in the mobile device and when the connection are reconnected, it tries to resend them.	As Expected, but not fully stable.

Table 15: Black-box test for UC1.1 extension - PC client.

Black-box for UC2.1		
Extensions	Expected outcome	Real outcome
3.a There is only one beacon in the area.	- 3.a1 If there is only one physically beacon in the area, the administrator is advised to attach an extra beacon to the area to extend the precision of the position in the system.	As Expected
	- 3.a2 The user may choose to continue the set-up with only one beacon, but can only register the users in an area and not where in the area due to lack of precision. Proceed directly to step 4.	As Expected
4.a An area should cover a larger location.	- 4.a1 First select a corner of the area.	As Expected
	- 4.a2 Read distance to all area beacon for the first corner.	As Expected
	- 4.a3 Step 4.a2 is repeated to all corners are read.	As Expected
4.b If the area is not square but eg. a circle.	- 4.b1 There is formed a square area within the original framework (eg. a circle)	As Expected
5.a A sub-area should be within a known area.	- 5.a1 Select the area where the sub-area is located.	As Expected. Not implemented
	- 5.a2 Use the extensions for 4.a and follow the instruction for the sub-area.	As Expected. Not implemented

Table 16: Black-box test for UC2.1 extension - Configuration of area.

Black-box for UC3.1		
Extensions	Expected outcome	Real outcome
1.a There is no location selected.	- 1.a1 There appears a default screen to the user, with further instructions to the user.	As Expected. Not implemented
5.a If it is not possible to update with new data, due to the connection of the database is interrupted.	- 5.a1 The newest status stays on screen and there will be a message on top of the text: "Can not update data, waiting for connection."	As Expected. Not implemented

Table 17: Black-box test for UC3.1 extension - Show data.

Black-box for UC3.2		
Extensions	Expected outcome	Real outcome
1.a No user registrations found.	- 1.a1 A message is displayed on the screen, saying: "No data found. Please Try again."	As Expected. Not implemented
2.a Person registrations exist, but not an area to link them at.	- 2.a1 Data is stored in a log-file for detecting errors.	As Expected. Not implemented
3.a Person registrations can not be located for a sub-area.	- 3.a1 The data is linked to the area only.	As Expected. Not implemented

Table 18: Black-box test for UC3.2 extension - Assign person.

7.3 System test

To test the precision of the beacon registrations and the power usages of the battery from the application and bluetooth module, a range of mobile devices was set-up in different parts of the office environment. The full descriptions of the test area / environment can be found in section 7.1 - Test Area.

There are used four different mobile devices and a laptop with two different bluetooth chips, to test if there is any difference between models such as speed or precision.

With each of the mobile devices there are also provided a battery consumption test, to see how much power it took from the mobile device when having bluetooth turned on all the time, together with the client application that sends the beacon-data from the client to the back-end of the system.

The test smartphone devices are:

- Samsung Galaxy Xcover 3 [43]
- Samsung Galaxy S5 [41]
- Samsung Galaxy S6 Edge [42]
- Huawei P8 Lite [24]

The test Laptop devices are:

- Fujitsu FJNB23F [16] with the on-board bluetooth module
- Fujitsu FJNB23F [16] with the ASUS USB-BT400 [27]

7.3.1 Precision of beacon registrations

To test the precision for distance between the mobile device and the beacons, the mobile device is fixed to a location and have three beacons within range as described in section 7.1 - Test Area.

The mobile device starts registering the incoming beacon-data and store them. This goes on in a fixed period of 5 minutes but in order to have the full 5 minutes records, the recording of data is up to 6 minutes. Afterwards the distance from the mobile device to each beacon is measured with a laser-ruler called Bosch PLR 15 [21].

An example of the test results for one of the devices(the Samsung Xcover 3) can be seen in figure 28. In table 19 the minimum, average and maximum measured distances to the beacons are listed with the percentage error of the deviation to the reference distance.

Test results for all the devices and the calculations of errors to the minimum, average and maximum distance, can be seen in appendix G.

The test showed that nearly all registrations shows the devices to be closer to the beacons than they actually were. The results are further discussed in section 8.1 - Test results.

Samsung Xcover 3 - Precision test

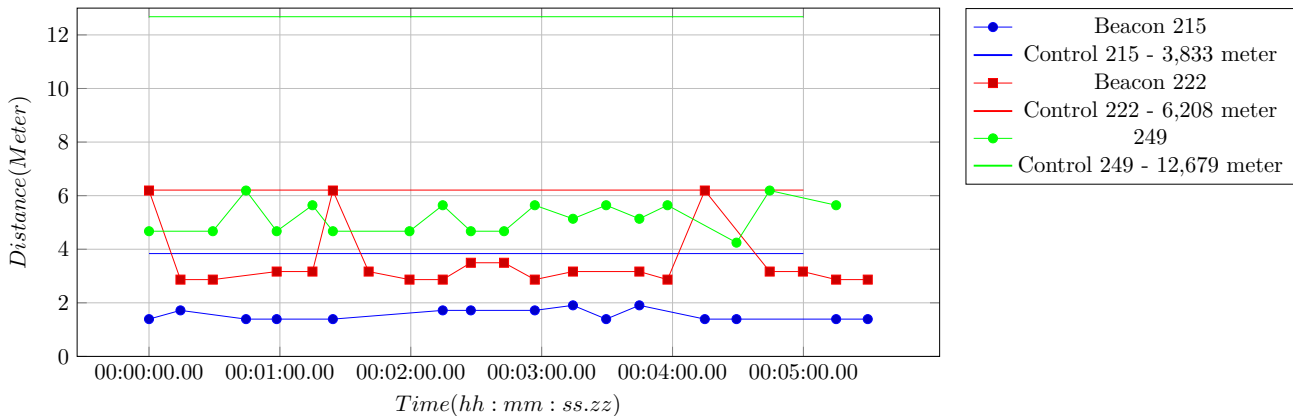


Figure 28: Show the beacon readings over time done for the Samsung Xcover 3 smartphone. The data-line and an actual distance-line(control) are showed for each beacon

Error statistis for Samsung Xcover 3				
Beacon Number	min distance %-error	avg distance %-error	max distance %-error	References Distance
215	1,39m -63,73%	1,55m -59,68%	1,91m -50,33%	3,83m
222	2,87m -53,82%	3,53m -43,08%	6,19m -0,27%	6,21m
249	4,25m -66,51%	5,22m -58,80	6,19m -51,17%	12,68m

Table 19: Table overview showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shorten to a two decimal number.

7.3.2 Battery test

To test the battery usages of the client mobile Android application, was used a battery testing and preserving app from GSAM Labs, called GSAM Battery Monitor [28].

The purpose of the test of the battery is to determine if one of the non-functional requirements withstands its goal. To do that, the power-usages of battery by the application must be at a minimum in order not to give the users frustrations by having to charge their mobile devices more than normally or otherwise change their behaviour.

The test of the client application itself can be seen in appendix I. There seems to be some coding errors since it was not possible to make a battery test of the app itself due to error messages from the OS itself stating that something is wrong with the application (Unusually large amount of battery power - 2,36% battery power). The client application was not further tested. Instead the battery usage of the bluetooth was tested because the bluetooth needs to be turned on for the application to work.

The test method is not scientifically accurate but does give guidelines for the battery usage of the bluetooth module, if it is turned on constantly. Since the battery usage is represented in percentage of the total battery usage for the devices, the comparison is done between other components essential for the devices usage and for how long it is used. A good comparison with the bluetooth module could be the phone radio (antenna module) because it is turned on constantly for the device to work.

The test are done for several devices since there have been problems between the different OS of the devices. Test results can be seen in table 20.

Mobile device	Andriod Version	Blue-tooth version	Screen	Phone Radio	Held Awake	Blue-tooth
Samsung Galaxy Xcover 3	5.1.1	4.0	28 % (00:27:21)	15 % (05:28:34)	3 % (03:24:20)	1 % (05:28:34)
Samsung Galaxy S5	6.0.1	4.0	47 % (00:20:21)	13 % (03:33:32)	0 % (00:02:21)	3 % (03:33:24)
Samsung Galaxy S6 Edge	6.0.1	4.1	44 % (00:20:28)	18 % (01:48:27)	1 % (00:17:46)	1 % (01:00:54)
Huawei P8 Lite	5.0.1	4.0	95 % (01:17:20)	3 % (01:17:20)	0 % (00:00:00)	0 % (01:17:20)

Table 20: Battery usage for the client beacon finder application for Android.
 The screen-shots of the test data shown in this table, can be found in appendix H.

Looking at the results in table 20, for the Xcover 3 mobile device, it took 28% of the total battery usage to have the screen turned on for 27 minutes, but only 1% to have the bluetooth on for 5,5 hours. Looking at modules running full time on the device, the Phone Radio takes 15% for the same time period where the bluetooth module only uses 1%. For the Xcover 3 device, it would not affect the battery usage enough for the user to notice, if the bluetooth module were turned on constantly. The same pattern can be seen for the other tested mobile devices.

8 Discussion

In the following section there will be discussed different aspects of the project. It is e.g. the privacy of the users of the system, how the system is analysed to be an how it was implemented and if the test results fulfil the system requirements.

8.1 Test results

There were made two tests. The first acceptance test was made to see if all the use cases were implemented correctly. The acceptance test showed that all the use cases implemented worked as intended, but the use cases were not fully implemented. This means that the system can be used for demonstration of the project but requires additional work to function in production.

The second test is a system test. Here is the goal to test the precision of the distance between the mobile device and the beacons and the battery usage of the client mobile device. The Battery test itself is discussed previously in section 7.3.2 - Battery test.

8.1.1 Precision test results

When the precision was tested the results showed(see appendix G) in nearly all the records that the mobile device calculated itself to be closer to the beacons than it actually where. Nearly all the results, except a few spikes, is approximately 50-60% off e.g. if the beacon is 10 meters away, the mobile device will state that there is 4-5 meters to the beacon.

A reason for this could be that the system set-up was made in an area with great possibility of signal interference. The signal interference comes from other electronics there is transmitting at 2.4GHz at the same time such as wireless mice, keyboards and headsets. The interfering may also come from the building materials used in the structure of the building, such as concrete, plaster and metal [26].

An interesting point to notice, is that the interference would delay the transmitting time and thereby making the beacon appear further away, but the tests showed the opposite. Maybe it is due to false positives from other transmitting devices or the client registering two beacon signals at the same time which will corrupt the data.

To look at another source of interference, a wifi scan was made and the results can be seen in figure 29.

The wifi scan shows that there is a great amount of wireless access points within the test area, all equally spaced over the wifi channels. This means that some of the interference have a valid possibility of coming from the wireless networks placed around the test area.

8.2 Competitor analysis

Does this system already exist? By looking at other companies it was found that they offered parts of the same functionalities, but not an of the shelf product or with easy implementation possibilities.

There was found a product that uses the same technology and which came close to having the same functionalities as this project product. The possible competitor product is made by the company called Estimote [13] where the functionality is incorporated in their own SDK.

In order to get the equal functionality between this project product and the one from Estimote there are still need for some programming with their SDK, but the accuracy of the user location come preconfigured.

A full configuration can be seen on a video [52]. Pictures from the configuration video can be seen in figure 30 and 31.



Figure 30: The configuration starts by placing a mobile device next to one of the beacons in the area to confirm the location.



Figure 31: Photo of the Estimote test area where the configuration person walks along the walls of the room. When meeting a setup beacon it is registered as described in figure 30.

The major difference between this project and Estimote is the way the configuration of an area is handled. Estimote want the user(or developer) to move a device along all the walls / borders of the area, in order to define an area. This can be difficult due to equipment or furnitures on the lines. After the configuration a code snip is generated and can afterwards be put in to the code where the area data is defined to register the area in a system.

In this project an area is defined by only 4 points which makes the configuration seem more simple. Further more the Estimote SDK is limited to only receive and use beacon signals from Estimote products, where in this project product all beacon types can be used.

To sum up, the Estimote product requires a programmer to register areas and is only compatible with other Estimote products.

In this project product, it requires only a minimum of technical knowledge to configure new areas and all brands of beacon products can be used.

8.3 Privacy of the user

One of the goals in the project was to keep track of how many people there are within an area of a building without being able to track individual persons specifically. To do this, the data needs to be anonymized, but is that enough?

It could be easy for a manager just to find out where one of his / her employees normally will be seated in one of the hotdesk areas and afterwards just consult the system to find out whether the person went home earlier than the time-registration said. This scenario will not be possible since the web-interface does not provide any unique user IDs but only a person count for each area.

If this was the goal for the manager to locate the registration for one of his / her employees he / she needs access directly to the system database to get hold on this information. Afterwards these found data has to be associated with the beacon numbers from the working area of the employee.

There is a greater risk for this scenario to be a reality, the manager could use the surveillance cameras which often are set up around the building to catch unwanted activities (e.g. burglars or vandalism).

But since the system technically is supervising / surveillance the person who uses it, it must comply the danish law of Personal Data Protection Act⁵

This project is not the first of its kind. In January 2016 the Daily Telegraph in Britain had a similar system installed under the desks of the employees without telling them before hand. The management said that the boxes were installed to monitor building heating systems. The boxes has monitoring and temperature sensors and can be linked wirelessly to e.g. HR so they can see whether the desk is occupied or not. The system was removed shortly after due to a storm of protest from the employees.

This is just one story, the article mentions other stories about other products to monitoring employees [49]. This story illustrates the importance of having the employees supporting and understanding the reasons and benefits of these kinds of monitoring systems before installing them.

⁵In danish "Persondataloven".

8.4 Requirements fulfilled

In order to verify that all the requirements are fulfilled, a check-list are made for both functional and non-functional requirements, see table 21 and 22.

Functional Requirements (FR)	
FR	Is FR fulfilled?
Register person	Yes, this is implemented and fully tested in the acceptance test.
Assign person	Yes, implemented and tested. But not fully implemented as described in section 7.2 - Acceptance test. It was not possible to locate a person at an object, due to precision errors between client and beacons.
Configuration of area	Yes, implemented and tested. However there are still some bugs in the functionality and the graphical representation.
Show data	Yes, implemented and tested. But locating a person at a desk was not technically possible, so the graphical representation of which desks are occupied does therefore not work.
Show statistic	Yes, is implemented with the show data functionality.
Login	No, this part is not implemented.
Reserve a seat	No, this part is not implemented.
Reserve area	No, this part is not implemented.
Assign seat	No, this part is not implemented.
Assign person	No, this part is not implemented.

Table 21: A check-list showing if the functional requirements are fulfilled.

Non-Functional Requirements (NFR)	
NFR	Is NFR fulfilled?
The system have to benefit the user	No, some essential FR are not implemented and the system is not currently linked to any context systems such as ventilation or heating. The most essential function that would benefit the user, would be to reserve a seat, an as long as this function is not implemented, this NFR can not be considered fulfilled.
Non / low user interaction with the system	Partially, the system clients runs without any user-interaction except for a manual start-up. However the headless version of the client application is not implemented so the user can be bothered by the debug screens on the mobile device.
Cross-Platform	Yes, by selecting the Xamarin platform for programming the system, all the client code only have to be programmed once and then transferred to each different client without having to re-write program code. The client code can be re-used, but additional project set-up might be needed when making a client for a new platform e.g. screen and device set-up.
Low battery usages on clients mobile devices	Yes, it was fully tested that the bluetooth module had a low impact on the battery when it was constantly turned on. The client application itself had some bugs that used more battery than normal. If this is fixed it is assumed, that the battery usage of the client application will fulfil the requirement of low battery usage too.
Power-supply or long battery lifetime for transmitters	Yes, by the selection of the Beaconinside BEACON model discussed in section 6.1 - Beacon, the requirements of having the possibility of a long battery lifetime and a additional power-source is fulfilled.

Table 22: A check-list showing if the non-functional requirements are fulfilled.

9 Future Work

After the project has been handed in there is still a lot of improvements and new initiatives that could make the product better and reach the final goal of being able to register, locate and make statistics for the persons in the free seats area in order to get the rest of the functionalities as reserve an area and book an area for employees.

In the list below in table 23, it can be found what needs to be improved.

- **Stabilize REST web-service**
During the test, it was found that the REST web-service receiving the client data was unstable and could be overloaded by sending to many request to it at the same time.
- **Optimize database**
The current database implemented do not fulfil the rules of database normalization. This can be solved by implementing the database structure found in appendix D.
- **Fix battery usage problem of client application**
The mobile application for the Android mobile devices have been giving some system errors in the OS stating that the application uses an larger amount of battery power than expected.
- **The problem with interference of the signals**
The project product is supposed to be used in office environments inside buildings. So something has to be done to reduce the interference of the signals.
- **Make the client screen update faster**
During the tests it was seen that the client screens from figure 18, for the Android mobile device did not have the update time needed to show all the debug data the screen should provide to the user. To optimize the system for later use this problem should be fixed to get better debug output.
- **Test how fast the beacon is transmitting**
In the test of both PC and Android clients it was seen that a beacon signal can be received more than once within the period of 350ms. It should be controlled weather it is an error in the beacon or in the project software.
- **Optimize the scaling of the maps of the web-interface**
The scaling of the maps does not fit to the web-interface e.g. maps are too large. To solve this, a map-scaling function should be implemented so high resolution maps still can fit every screen type without disturbing the interface layout.

Table 23: List of the project problems that can be improved in the future work at the project.

10 Conclusion

To solve the problem at hand, there was first listed the main stakeholders of the project and afterwards created as many user stories as possible for those stakeholders. The user stories were then collected in to functionalities and was prioritized.

It was controlled if the implemented system fulfilled the functional requirements by creating acceptance tests based on the use cases. These proved that all that was expected to be implemented also was implemented. So the acceptance tests confirmed that the implemented functions works as intended.

To make the PC client for Windows 7, a universal driver was used, in order to collect the beacon-data from the bluetooth chip. A side effect of this universal driver, is that the chip can then not be used for other purposes, such as a bluetooth headset or other wireless bluetooth devices.

It is possible to make the client without the universal drivers in Windows 10, since the functionality of collecting beacon-data comes native with Windows 10.

From the precision tests of the mobile devices it was found that most of the readings from all the devices were 50-60% off and it was assumed that a part of the error is due to signal interference coming from other 2.4GHz devices and building materials such as concrete, plaster and metal.

By looking at the results of the battery test, it was found that having the bluetooth module turned on all the time, it used less than 1/10 of the battery use that the normal GSM antennen module used in the same time period. This means that the battery usage will not affect the daily use of the mobile device and thereby the non-functional requirement was fulfilled.

In general all the functional and non-functional requirements have been tested and except a few, they are all fulfilled in the implemented system.

The implemented system functions as intended and are able to solve the problem of knowing how many employees are using the hotdesk areas, but the system will require some additional work in the future in order to work as designed.

With the additional work the system can be used as designed in the requirement specification (Design section), but if a higher accuracy of tracking persons is needed, the beacon technology have to be changed to something else.

It has to be noticed that it is important to inform and make the employees understand the benefits of the system, and thereby getting the support from the employees before implementing such a system. Otherwise the employees might feel watched and controlled. This can create an unpleasant working environment.

11 Glossary

List of words and expressions used in the report.

Administrator

This user can create, read, update and delete everything within the system.

Assign person

The name of use case 2.1.

Beacon

The device used to transmit the distance between the beacon and the mobile device.

Beacon-data

The data that are transmitted to and from the mobile device.

Building personnel

The people who maintain a building.

Client application

The software installed on the users mobile devices.

Configuration of area

The name of use case 3.1.

De-serialize

Converting a stream of bytes into a data object.

Developing process

The pattern of how the developing of the project has been done.

DTU

Short for the Technical University of Denmark,

Employee

User of the system.

Employer

User / administrator of the system.

FR

Short for functional requirements.

Free seating

See hotdesk area.

Headless mode

A piece of software running without any user interface.

Hotdesk area

An area where people share a desk and normally not working full time.

iBeacon

The Apple name for the beacon technology.

MAC address

A universal address for an electronic device. In this project used as an identifier after it is being hashed.

Manager

See employer.

Measuring stick

Drawing that shows the scalation of the map.

Mobile device

The term for all the devices that can register beacon-data.

Mock-up

A graphically representation of the interface to be implemented.

Mssql

The database technology from Microsoft.

NFR

Short for non-functional requirements.

Person

The person is the one who collects the beacon-data. Is also known as the user.

Person registration

The term is used for a person collecting beacon-data and send it to the REST web-service.

Project product

A term covering all the pieces of software created and implemented in this project.

Raspberry Pi

A single board ARM-computer.

Register person

The name of use case 1.1.

Registration

See Person registration.

REST web-service

The piece of software that gets the data from the client application and store it in the database.

Serialize

An object of data is converted into a stream of bytes.

Show data

The name of use case 3.2.

Signal interference

When a signal gets interrupted by things like other signals, concrete of plaster.

SRE employee

See Building personnel.

Stakeholders

The persons who have interest in the system.

Sub-area

An item or object within the area, such as a table.

Sub-use case

A use case that depends on another use case.

System

The term of the full physically project.

Team leader

See employer.

Use case

A description of a functionality of the system.

User data

The data the client application provides to the REST web-service.

User ID

A unique ID of each user.

User story

A model used to find possible usage for a system from wishes from the users.

Winbeacon

A software library used to collect beacon-data on a Windows computer.

12 Figure reference

Some of the figures in this report contains images or parts of images there are not fully created by the report author. Below are listed where the figures occur and the source of the original image or graphics.

Figure 1 Page: 6

Reference: The picture Comes from the internal website for Siemens.

The internal path is:

Home > Process house(mobility) > System Development & Engineering

Figure 2 Page: 7

Reference: The picture Comes from the internal website for Siemens.

The internal path is:

Home > Process house(mobility) > System Development & Engineering > Software Development

Figure 3 Page: 12

Reference: created by author.

Figure 4 Page: 22

Reference: The picture icons used is from the Cisco website [10].

Figure 5 Page: 23

Reference: created by author.

Figure 6 Page: 24

Reference: created by author.

Figure 7 Page: 24

Reference: created by author.

Figure 9 Page: 25

Reference: created by author.

Figure 9 Page: 25

Reference: created by author.

Figure 10 Page: 26

Reference: created by author.

Figure 11 Page: 26

Reference: created by author.

Figure 12 Page: 27

Reference: The picture icons used is from the Cisco website [10].

Figure 13 Page: 30

Reference: created by author.

Figure 14 Page: 31

Reference: created by author.

Figure 15 Page: 32

Reference: The picture used is from the Cisco website [10].

Figure 16 Page: 34

Reference: created by author.

Figure 17 Page: 35

Reference: created by author.

Figure 18(a) Page: 37

Reference: created by author.

Figure 18(b) Page: 37

Reference: created by author.

Figure 18(c) Page: 37

Reference: created by author.

Figure 18(d) Page: 37

Reference: created by author.

Figure 18 Page: 37

Reference: created by author.

Figure 19 Page: 38

Reference: The picture used is from the Cisco website [10].

Figure 20 Page: 39

Reference: The picture used is from the Cisco website [10].

Figure 21 Page: 41

Reference: created by author.

Figure 22 Page: 42

Reference: created by author.

Figure 24 Page: 45

Reference: created by author.

Figure 25 Page: 46

Reference: created by author.

Figure 26 Page: 46

Reference: created by author.

Figure 27 Page: 47

Reference: The picture used is from the Cisco website [10].

Figure 28 Page: 52

Reference: created by author.

Figure 29 Page: 55

Reference: created by author.

Figure 30 Page: 56

Reference: Screenshot youtube video from Estimote [52].

Figure 31 Page: 56

Reference: Screenshot youtube video from Estimote [52].

Figure 32 Page: 81

Reference: created by author.

Figure 33 Page: 82

Reference: The picture is copied from Raspberry Pi webpage [50].

Figure 34 Page: 84

Reference: created by author.

Figure 35 Page: 87

Reference: created by author.

Figure 36 Page: 88

Reference: created by author.

Figure 37 Page: 89

Reference: created by author.

Figure 38 Page: 90
Reference: created by author.

Figure 39 Page: 91
Reference: created by author.

Figure 40 Page: 92
Reference: created by author.

Figure 41 Page: 93
Reference: created by author.

Figure 42 Page: 93
Reference: created by author.

Figure 43 Page: 93
Reference: created by author.

Figure 44 Page: 93
Reference: created by author.

Figure 45(a) Page: 94
Reference: created by author.

Figure 45(b) Page: 94
Reference: created by author.

Figure 45 Page: 94
Reference: created by author.

References

- [1] Maplesoft a division of Waterloo Maple Inc. The essential tool for mathematics. "<http://www.maplesoft.com/products/Maple/>", 2016. Last visit the 23. August 2016.
- [2] Aislelabs. The hitchhikers guide to ibeacon hardware: A comprehensive report by aislelabs (2015). "<http://www.aislelabs.com/reports/beacon-guide/>", 04. May 2015. Last visit the 05. August 2016.
- [3] Michael Ashbridge. Eddystone protocol specification. <https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RSSI-and-other-characteristics-of-beacon-s-signal->, 14. April 2016. Last visit the 3. August 2016.
- [4] Tim Baker. What is the positioning accuracy of ibeacon (and other ble) technology? - quora. <https://www.quora.com/What-is-the-positioning-accuracy-of-iBeacon-and-other-BLE-technology>, unknown. Last visit the 31. July 2016.
- [5] Pete (Akeor) Batard. Zadig - usb driver installation made easy. "<http://zadig.akeo.ie/>", 22. January 2016. Last visit the 03. August 2016.
- [6] Inc BKON Connect. What are the differences between ibeacon and eddystone beacons? <https://bkon.com/beacons/differences/>, 2016. Last visit the 3. August 2016.
- [7] Inc. Borowicz, Wojtek from Estimote. How precise are estimote beacons? <https://community.estimote.com/hc/en-us/articles/201302836-How-precise-are-Estimote-Beacons->, 11. September 2015. Last visit the 5. August 2016.
- [8] ChinaPaul and davidgyoung (stackoverflow users). What is the ibeacon bluetooth profile. <http://stackoverflow.com/questions/18906988/what-is-the-ibeacon-bluetooth-profile>, 26. Septemperl 2013. Last visit the 02. August 2016.
- [9] Javier Chávarril. Stackoverflow - triangulate example for ibeacons. <http://stackoverflow.com/questions/20332856/triangulate-example-for-ibeacons>, 07. January 2014. Last visit the 5. August 2016.
- [10] Inc. Cisco Systems. Network topology icons - doing business with cisco. <http://www.cisco.com/c/en/us/about/brand-center/network-topology-icons.html>. Last visit the 15. August 2016.
- [11] Steven (Oracle A-TEAM Chronicles) Davelaar. Performance study – rest vs soap for mobile applications. "<http://www.ateam-oracle.com/performance-study-rest-vs-soap-for-mobile-applications/>", 27. February 2015. Last visit the 03. August 2016.
- [12] eBay Inc. Ebay pris for beacon model hm-10. "http://www.ebay.com/sch/i.html?_odkw=beacon+hm-10&_sop=15&LH_BIN=1&_osacat=0&_from=R40&_trksid=p2045573.m570.11313.TR0.TRC0.H0.Xhm-10.TRS0&_nkw=hm-10&_sacat=0", 04. April 2016. Last visit the 04. April 2016.
- [13] Estimote. Estimote - real-world context for your apps. <http://estimote.com/>, 2012-2015. Last visit the 15. August 2016.
- [14] farproc. Wifi analyzer - google play. "<https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=da>", 10. April 2016. Last visit the 20. August 2016.

-
- [15] flurp laboratories. ibeacon and eddystone scanner - google play. "<https://play.google.com/store/apps/details?id=de.flurp.beaconscanner.app>", 19. April 2016. Last visit the 26. August 2016.
- [16] INC FUJITSU AMERICA. Lifebook e752 [dscn010224]. <http://solutions.us.fujitsu.com/pdf/notebooks/Notebook-computer-LIFEBOOK-E752-datasheet.pdf>, Unlnown. Last visit the 1. August 2016.
- [17] Eva Gaudlitz. Indoor navigation using wifi as a positioning technology. <https://www.infsoft.com/blog-en/articleid/40/indoor-navigation-using-wifi-as-a-positioning-technology>, 29. September 2015. Last visit the 5. August 2016.
- [18] Sunny Gleason and Norvan Sahiner. Building an android beacon (android ibeacon tutorial overview) [1/3]. "<https://www.pubnub.com/blog/2015-04-14-building-android-beacon-android-ibeacon-tutorial-overview/>", 14. April 2015. Last visit the 04. August 2016.
- [19] Beaconinside GmbH. Beaconinside beacon. <http://www.beaconinside.com/beacon/>, 2015. Last visit the 31. July 2016.
- [20] Beaconinside GmbH. Beaconinside beacon - shop featured. <http://shop.beaconinside.com/>, 2016. Last visit the 31. July 2016.
- [21] Robert Bosch GmbH. Power tools for diy enthusiasts. https://www.bosch-do-it.com/gb/en/diy/tools/plr-15-3165140727754-199929.jsp#tab_1, Unlnown. Last visit the 1. August 2016.
- [22] Belinda la Cour fra Trafitec Herrstedt, Lene og Lund. Ganghastigheder – med særlig fokus på ældre fodgængere - trafik og veje • 2010 juni/juli. "<http://asp.vejtid.dk/Artikler/2010/06-07%5C5797.pdf>", June / July 2010. Data from page 7 - table 1. Last visit the 31. July 2016.
- [23] Vincent (Github user VincentH-Net) Hoogendoorn. Android ibeacon app - findthemonkey. "<https://github.com/MacawNL/FindTheMonkey-Android>", 16. April 2014. Last visit the 5. August 2016.
- [24] Huawei. P8 lite. <http://consumer.huawei.com/en/mobile-phones/p8lite/index.htm>, 1998-2016. Last visit the 15. August 2016.
- [25] Wouter Huysentruit. Winbeacon - github, inc. "<https://github.com/huysentruitw/win-beacon>", 08. June 2015. Last visit the 03. August 2016.
- [26] Apple Inc. Potential sources of wi-fi and bluetooth interference. <https://support.apple.com/en-us/HT201542>, 2016. Last visit the 20. August 2016.
- [27] ASUSTeK Computer Inc. Bluetooth 4.0 usb adapter. <https://www.asus.com/Networking/USB400/>, Unlnown. Last visit the 1. August 2016.
- [28] GSAM Labs. Gsam battery monitor - google play. "<https://play.google.com/store/apps/details?id=com.gsamlabs.bbm>", 11. July 2016. Last visit the 11. August 2016.
- [29] Tutorialspoint Simply Easy Learning. Learn wcf - wcf tutorial. <http://www.tutorialspoint.com/wcf/index.htm>, 2016. Last visit the 1. August 2016.
- [30] Eric Leschinski and Falko Schindler. How do i run a script at start up? "<http://askubuntu.com/questions/228304/how-do-i-run-a-script-at-start-up>", 12. December 2012. Last visit the 07. August 2016.

-
- [31] Xin Li, Jian Wang, and Chunyan Liu. A bluetooth/pdr integration algorithm for an indoor positioning system. *Sensors*, 15(10):24862–24885, 2015.
- [32] Microsoft. Bluetoothleadvertisementwatcher class. <https://msdn.microsoft.com/en-us/library/windows.devices.bluetooth.advertisement.bluetoothleadvertisementwatcher.aspx>, 2016. Last visit the 1. August 2016.
- [33] Xamarin Inc. (Microsoft). Build native ios, android and windows apps in visual studio. "<https://www.xamarin.com/visual-studio>", 2016. Last visit the 04. August 2016.
- [34] Adrian Mullally, Nigel McKelvey, and Kevin Curran. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *TELKOMNIKA*, 9(1):1–8, 2011.
- [35] Chris (Product) Necklas. Understanding advertising interval and beacon battery life. <http://www.swirl.com/blog/understanding-advertising-interval-and-beacon-battery-life/>, 20. November 2015. Last visit the 2. August 2016.
- [36] Microsoft Developer Network. Sql server drivers. "[https://msdn.microsoft.com/en-us/library/mt654049\(v=sql.1\).aspx](https://msdn.microsoft.com/en-us/library/mt654049(v=sql.1).aspx)", 2016. Last visit the 03. August 2016.
- [37] Newtonsoft. Json.net - popular high-performance json framework for .net. "<http://www.newtonsoft.com/json>", 2016. Last visit the 04. August 2016.
- [38] Newtonsoft. Json.net - popular high-performance json framework for .net - version 9.0.1. <http://www.newtonsoft.com/json>, 2016. Last visit the 23. August 2016.
- [39] Ola Puchta. What are broadcasting power, rssi and other characteristics of beacon's signal? <https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RSSI-and-other-characteristics-of-beacon-s-signal->, 25. January 2016. Last visit the 3. August 2016.
- [40] Ola Puchta. What are broadcasting power, rssi and other characteristics of beacon's signal? <https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RSSI-and-other-characteristics-of-beacon-s-signal->, 25. January 2016. Last visit the 5. August 2016.
- [41] Samsung. Galaxy s5. <http://www.samsung.com/uk/consumer/mobile-devices/smartphones/galaxy-s/SM-G900FZKABTU>, 1995-2016. Last visit the 15. August 2016.
- [42] Samsung. Galaxy s6 edge. <http://www.samsung.com/global/galaxy/galaxys6/galaxy-s6-edge/>, 1995-2016. Last visit the 15. August 2016.
- [43] Samsung. Galaxy xcover 3. <http://www.samsung.com/uk/consumer/mobile-devices/smartphones/others/SM-G388FDSABTU>, 1995-2016. Last visit the 1. August 2016.
- [44] Michael Shrivathsan. Product management insights - use case template and an example. <http://pmblog.accompa.com/2009/10/08/use-case-template-example-requirements-management-basics/>, 8. October 2009. Last visit the 19. July 2016.
- [45] Tom Simonite. Wi-fi trick gives devices super-accurate indoor location fixes. <https://www.technologyreview.com/s/542561/wi-fi-trick-gives-devices-super-accurate-indoor-location-fixes/>, 16. October 2015. Last visit the 5. August 2016.
- [46] Mountain Goat Software. User stories. <https://www.mountaingoatsoftware.com/agile/user-stories>, 1998-2016. Last visit the 31. July 2016.

- [47] SourceMaking.com. Uml - modeling business systems - activity diagrams. <https://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams>, 25. January 2007-2016. Last visit the 5. August 2016.
- [48] John Sprunger. Mobile web services – json/rest vs. xml/soap. "<https://blog.westmonroepartners.com/mobile-web-services-jsonrest-vs-xmlsoap/>", 08. November 2013. Last visit the 03. August 2016.
- [49] Lain The Register Thomson. Hacks rebel after bosses secretly install motion sensors under desks. "http://www.theregister.co.uk/2016/01/12/bosses_install_motion_sensors/", 12. January 2016. Last visit the 04. August 2016.
- [50] Eben Upton. Raspberry pi 2 on sale now at 35usd. "<https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>", 02. February 2015. Last visit the 28. August 2016.
- [51] Wade Wegner. Create an ibeacon transmitter with the raspberry pi. "<http://www.wadewegner.com/2014/05/create-an-ibeacon-transmitter-with-the-raspberry-pi/>", 20. May 2014. Last visit the 07. August 2016.
- [52] Estimote (youtube). Estimote indoor location. <https://www.youtube.com/watch?v=wtBERi7Lf3c>, 24. September 2014. Last visit the 15. August 2016.
- [53] Paul A. Zandbergen and Sean J. Barbeau. Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. Journal of Navigation, J. Navig, J Navig, J Navigation, Journal of Navigation, 64(3):381–399, 2011.

A Appendix - Project Registration

Projektinfo

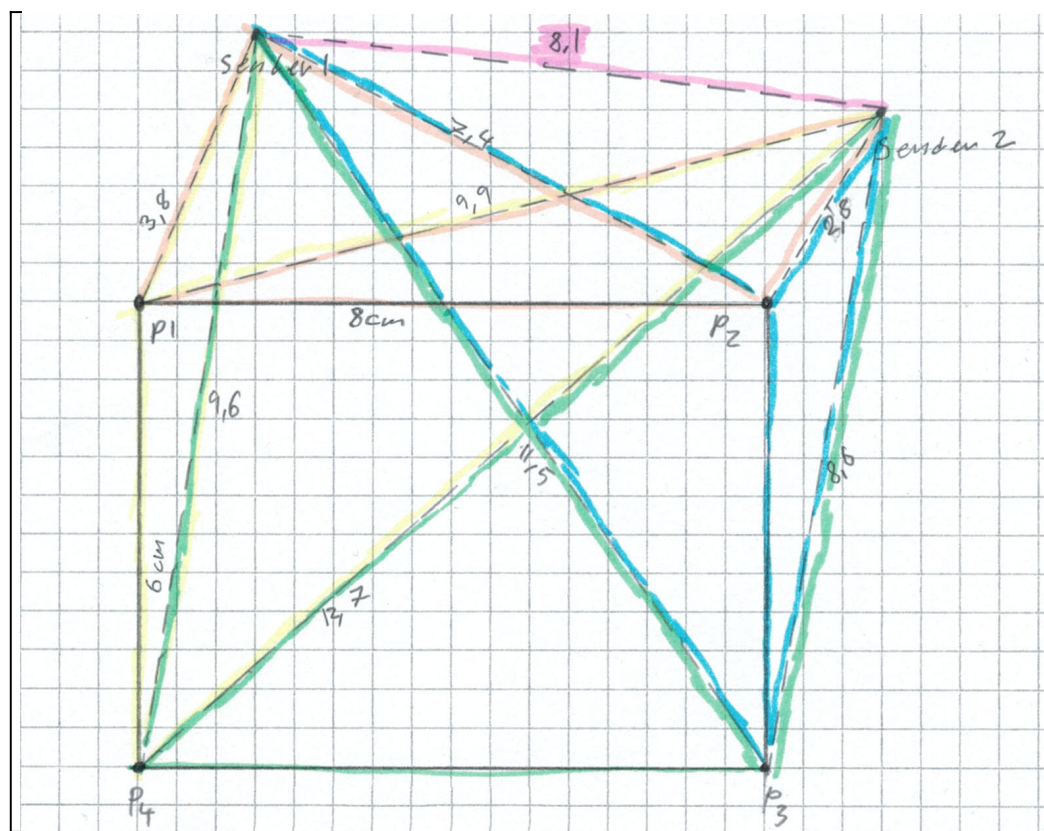
Dansk titel:	iBeacons anvendelser indenfor registrering og behandling af bruger-data
Engelsk titel:	iBeacons application in the registration and processing of user data
Studerende:	Magnus Bjarke Bang Nielsen(071853)
Kommentar(DK):	Projektet udføres og skrives på dansk. Se "Begrundelse for overskridelse af projekttid"
Kommentar(UK):	The project is written in danish. See "Begrundelse for overskridelse af projekttid"
ECTS Point:	35
Type:	Kandidatspeciale
Startdato:	2016-02-29
Afslutningsdato:	2016-08-29
Samarbejdsinstitutter:	Intet samarbejdsinstitut
Samarbejdsvirksomheder:	Siemens A/S - Industry Sector - Mobility
Samarbejdsform:	
Projekt udføres i:	Danmark
Antal måneder i udlandet:	(Ikke udfyldt)
Vejleder(e):	Stig Høgh (shog@dtu.dk)
Vejleder(e) (ekstern):	Jesper Lauge Kilde (Jesper-lauge.kilde@siemens.com) (23 72 43 38)

B Appendix - Calculation of area based on two beacons

The following calculations are made in Maple which is a calculation tool made by Maplesoft [1]. The version that is used for these calculations is Maple2016.

```

[First loaded libraries used for the calculations.
> restart;
> with(Gym) :
Then listed the distances between the two beacons and the four points. As seen in the image below.
    
```



It provides together eight points of data.
 The points are named b <beacon No> <points No>. This means that the beacon 1 distance to point 1 is named b11.

- ```

> b11 := 3.8;
> b12 := 7.4;
> b13 := 11.5;
> b14 := 9.6;

```
- $b11 := 3.8$  (1)  
 $b12 := 7.4$  (2)  
 $b13 := 11.5$  (3)  
 $b14 := 9.6$  (4)

Now the distance to the second beacon to the points.

$$\begin{aligned} > b21 := 9.9; & & b21 := 9.9 & (5) \end{aligned}$$

$$\begin{aligned} > b22 := 2.8; & & b22 := 2.8 & (6) \end{aligned}$$

$$\begin{aligned} > b23 := 8.6; & & b23 := 8.6 & (7) \end{aligned}$$

$$\begin{aligned} > b24 := 12.7; & & b24 := 12.7 & (8) \end{aligned}$$

Finally, the distance between the two beacons. This is also called reference distance named S.

$$\begin{aligned} > S := 8.1; & & S := 8.1 & (9) \end{aligned}$$

Now the distance between P1 and P2 calculated that forms one side of the square.

Then is calculated the first triangle formed between b11, b21 and S .

$$\begin{aligned} > COSb11b21 & := \frac{(b11^2 + S^2 - b21^2)}{2 \cdot b11 \cdot S} \\ & & COSb11b21 := -0.2917478882 & (10) \end{aligned}$$

Afterwards the angle between b11 and b21 is found.

$$\begin{aligned} > b11b21 & := \text{invCos}(COSb11b21); \\ & & b11b21 := 106.9626286 & (11) \end{aligned}$$

Then is calculated the first triangle formed between b12, b22 and S .

$$\begin{aligned} > COSb12b22 & := \frac{(b12^2 + S^2 - b22^2)}{2 \cdot b12 \cdot S} \\ & & COSb12b22 := 0.9386886890 & (12) \end{aligned}$$

Afterwards the angle between b12 and b22 is found.

$$\begin{aligned} > b12b22 & := \text{invCos}(COSb12b22); \\ & & b12b22 := 20.16750780 & (13) \end{aligned}$$

Now one can via cosinusrelationen find the distance between P1 and P2, as there are now two sides of the triangle and the angle has just found.

First, the two angles pulled apart to get the correct angle.

$$\begin{aligned} > PIP2Angle & := b11b21 - b12b22; \\ & & PIP2Angle := 86.79512080 & (14) \end{aligned}$$

$$\begin{aligned} > Cos(PIP2Angle); & & 0.05590652974 & (15) \end{aligned}$$

Now the distance can be found using the cosinusrelationen.

$$A := \sqrt{b^2 + c^2 - 2 \cdot b \cdot c \cdot \text{Cos}(X)} :$$

$$\begin{aligned} > PIP2 & := \sqrt{b11^2 + b12^2 - 2 \cdot b11 \cdot b12 \cdot \text{Cos}(PIP2Angle)} ; \\ & & PIP2 := 8.127472963 & (16) \end{aligned}$$

The distance between P1 and P2 is calculated to be 8.127472963 meter.



Now the distance between P2 and P3 calculated that forms one side of the square.

Then is calculated the first triangle formed between b12, b22 and S .

$$\begin{aligned} > \text{COSb12b22} := \frac{(b12^2 + S^2 - b22^2)}{2 \cdot b12 \cdot S} \\ \text{COSb12b22} := 0.9386886890 \end{aligned} \quad (17)$$

Afterwards the angle between b12 and b22 is found.

$$\begin{aligned} > b12b22 := \text{invCos}(\text{COSb12b22}); \\ b12b22 := 20.16750780 \end{aligned} \quad (18)$$

Now the distance between P2 and P3 calculated that forms one side of the square.

Then is calculated the first triangle formed between b13, b23 and S .

$$\begin{aligned} > \text{COSb13b23} := \frac{(b13^2 + S^2 - b23^2)}{2 \cdot b13 \cdot S} \\ \text{COSb13b23} := 0.6650563605 \end{aligned} \quad (19)$$

Afterwards the angle between b13 and b23 is found.

$$\begin{aligned} > b13b23 := \text{invCos}(\text{COSb13b23}); \\ b13b23 := 48.31335063 \end{aligned} \quad (20)$$

Now one can via cosinusrelationen find the distance between P2 and P3, as there are now two sides of the triangle and the angle has just found.

First, the two angles pulled apart to get the correct angle.

$$\begin{aligned} > P2P3\text{Angle} := b12b22 - b13b23; \\ P2P3\text{Angle} := -28.14584283 \end{aligned} \quad (21)$$

$$\begin{aligned} > \text{Cos}(P2P3\text{Angle}); \\ 0.8817497231 \end{aligned} \quad (22)$$

Now the distance can be found using the cosinusrelationen.

$$A := \sqrt{b^2 + c^2 - 2 \cdot b \cdot c \cdot \text{Cos}(X)} :$$

$$\begin{aligned} > P2P3 := \sqrt{b12^2 + b13^2 - 2 \cdot b12 \cdot b13 \cdot \text{Cos}(P2P3\text{Angle})}; \\ P2P3 := 6.077515701 \end{aligned} \quad (23)$$

The distance between P2 and P3 is calculated to be 6.077515701 meter.

Now the distance between P3 and P4 calculated that forms one side of the square.

Then is calculated the first triangle formed between b13, b23 and S .

$$\begin{aligned} > \text{COSb13b23} &:= \frac{(b13^2 + S^2 - b23^2)}{2 \cdot b13 \cdot S} \\ &\text{COSb13b23} := 0.6650563605 \end{aligned} \tag{24}$$

Afterwards the angle between b13 and b23 is found.

$$\begin{aligned} > b13b23 &:= \text{invCos}(\text{COSb13b23}); \\ &b13b23 := 48.31335063 \end{aligned} \tag{25}$$

Now the distance between P3 and P4 calculated that forms one side of the square.

Then is calculated the first triangle formed between b14, b24 and S .

$$\begin{aligned} > \text{COSb14b24} &:= \frac{(b14^2 + S^2 - b24^2)}{2 \cdot b14 \cdot S} \\ &\text{COSb14b24} := -0.02263374486 \end{aligned} \tag{26}$$

Afterwards the angle between b14 and b24 is found.

$$\begin{aligned} > b14b24 &:= \text{invCos}(\text{COSb14b24}); \\ &b14b24 := 91.29692882 \end{aligned} \tag{27}$$

Now one can via cosinusrelationen find the distance between P3 and P4, as there are now two sides of the triangle and the angle has just found.

First, the two angles pulled apart to get the correct angle.

$$\begin{aligned} > P3P4\text{Angle} &:= b13b23 - b14b24; \\ &P3P4\text{Angle} := -42.98357819 \end{aligned} \tag{28}$$

$$\begin{aligned} > \text{Cos}(P3P4\text{Angle}); \\ &0.7315491422 \end{aligned} \tag{29}$$

Now the distance can be found using the cosinusrelationen.

$$\begin{aligned} A &:= \sqrt{b^2 + c^2 - 2 \cdot b \cdot c \cdot \text{Cos}(X)} : \\ > P3P4 &:= \sqrt{b13^2 + b14^2 - 2 \cdot b13 \cdot b14 \cdot \text{Cos}(P3P4\text{Angle})}; \\ &P3P4 := 7.929940063 \end{aligned} \tag{30}$$

The distance between P3 and P4 is calculated to be 7.929940063 meter.

Now the distance between P4 and P1 calculated that forms one side of the square.

Then is calculated the first triangle formed between b14, b24 and S .

$$\begin{aligned} > \text{COSb14b24} := \frac{(b14^2 + S^2 - b24^2)}{2 \cdot b14 \cdot S} \\ \text{COSb14b24} := -0.02263374486 \end{aligned} \quad (31)$$

Afterwards the angle between b14 and b24 is found.

$$\begin{aligned} > b14b24 := \text{invCos}(\text{COSb14b24}); \\ b14b24 := 91.29692882 \end{aligned} \quad (32)$$

Now the distance between P4 and P1 calculated that forms one side of the square.

Then is calculated the first triangle formed between b14, b24 and S .

$$\begin{aligned} > \text{COSb11b21} := \frac{(b11^2 + S^2 - b21^2)}{2 \cdot b11 \cdot S} \\ \text{COSb11b21} := -0.2917478882 \end{aligned} \quad (33)$$

Afterwards the angle between b11 and b21 is found.

$$\begin{aligned} > b11b21 := \text{invCos}(\text{COSb11b21}); \\ b11b21 := 106.9626286 \end{aligned} \quad (34)$$

Now one can via cosinusrelationen find the distance between P4 and P1, as there are now two sides of the triangle and the angle has just found.

First, the two angles pulled apart to get the correct angle.

$$\begin{aligned} > P4P1\text{Angle} := b14b24 - b11b21; \\ P4P1\text{Angle} := -15.66569978 \end{aligned} \quad (35)$$

$$\begin{aligned} > \text{Cos}(P4P1\text{Angle}); \\ 0.9628535693 \end{aligned} \quad (36)$$

Now the distance can be found using the cosinusrelationen.

$$A := \sqrt{b^2 + c^2 - 2 \cdot b \cdot c \cdot \text{Cos}(X)} :$$

$$\begin{aligned} > P4P1 := \sqrt{b14^2 + b11^2 - 2 \cdot b14 \cdot b11 \cdot \text{Cos}(P4P1\text{Angle})}; \\ P4P1 := 6.029113001 \end{aligned} \quad (37)$$

The distance between P4 and P1 is calculated to be 6.02911301 meter.

## C Appendix - Database creation script

The creation script for the system database with the set-up of the database user included. The script is shown below in listing 1 and continued in listing 2.

```
-- Creating database Beacon if not exists
Use Master;

if db_id('Beacon') IS NOT NULL
BEGIN
 ALTER DATABASE Beacon SET SINGLE_USER WITH ROLLBACK IMMEDIATE
 DROP DATABASE Beacon;
END

CREATE DATABASE Beacon;
USE Beacon;

-- Creating test table users if not exists
IF OBJECT_ID('dbo.users', 'U') IS NOT NULL
 DROP TABLE users;

Create TABLE users (
 Username nvarchar(260)
)

-- Creating Beacon table users if not exists
IF OBJECT_ID('dbo.beaconregistration', 'U') IS NOT NULL
 DROP TABLE beaconregistration;

Create TABLE beaconregistration (
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
 RegistrationId nvarchar(260),
 RegistrationTimeDate DateTime
)

-- Creating beaconAera table users if not exists
IF OBJECT_ID('dbo.beaconAera', 'U') IS NOT NULL
 DROP TABLE beaconAera;

Create TABLE beaconAera (
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
 AreaName nvarchar(260),
 AreaMapName nvarchar(260),
 StartPointX float,
 StartPointY float
)

-- Creating beaconAeraDimensions table users if not exists
IF OBJECT_ID('dbo.beaconAeraDimensions', 'U') IS NOT NULL
 DROP TABLE beaconAeraDimensions;

Create TABLE beaconAeraDimensions (
 id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
 AreaMapName nvarchar(260),
 DimensionsOfMap float,
 StartX float,
 StartY float,
 SlutX float,
 SlutY float
)
```

Listing 1: First part of the database script for the system. The Second part continues on the next page.

```
57 -- Creating Found Beacon table users if not exists
58 IF OBJECT_ID('dbo.beaconfound', 'U') IS NOT NULL
59 DROP TABLE beaconfound;
60
61
62 Create TABLE beaconfound (
63 Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
64 BeaconRegistrationId INT FOREIGN KEY REFERENCES beaconregistration(Id),
65 BeaconAdress nvarchar(260),
66 BeaconCompanyId INT,
67 BeaconCalibratedTxPower INT,
68 BeaconApproximateIosRange float,
69 BeaconRange float,
70 BeaconIsAppleIBeacon bit,
71 BeaconMajor INT,
72 BeaconMinor INT,
73 BeaconRssi INT,
74 BeaconUUID nvarchar(260)
75)
76
77 -- Creating Found beaconInArea table users if not exists
78 IF OBJECT_ID('dbo.beaconInArea', 'U') IS NOT NULL
79 DROP TABLE beaconInArea;
80
81 Create TABLE beaconInArea (
82 Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
83 BeaconFoundId INT NOT NULL FOREIGN KEY REFERENCES beaconfound(Id),
84 BeaconAreaId INT NOT NULL FOREIGN KEY REFERENCES beaconAera(Id),
85 Point1 float,
86 Point2 float,
87 Point3 float,
88 Point4 float
89)
90
91 -- Creating Found beaconBetweenDistance table users if not exists
92 IF OBJECT_ID('dbo.beaconBetweenDistance', 'U') IS NOT NULL
93 DROP TABLE beaconBetweenDistance;
94
95 Create TABLE beaconBetweenDistance (
96 Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
97 BeaconFromId INT NOT NULL FOREIGN KEY REFERENCES beaconfound(Id),
98 BeaconToId INT NOT NULL FOREIGN KEY REFERENCES beaconfound(Id),
99 BeaconAreaId INT NOT NULL FOREIGN KEY REFERENCES beaconAera(Id),
100 DistanceBetween float
101)
102
103 -- Check if login exists
104 if exists(select * from sys.server_principals where name = 'BeaconAdmin')
105 -- you need to create a server login first
106 DROP LOGIN BeaconAdmin;
107
108 -- Create user for SQL Authentication
109 CREATE LOGIN BeaconAdmin WITH PASSWORD = '123456aA!'
110 ,DEFAULT_DATABASE = [Beacon]
111 GO
112
113 -- Now add user to database
114 USE Beacon;
115 CREATE USER BeaconAdmin FOR LOGIN BeaconAdmin;
116 GO
117 -- Grant User all WRITE permissions (INSERT, UPDATE, DELETE) on all tables
118 EXEC sp_addrolemember N'db_owner', N'BeaconAdmin'
```

Listing 2: Second and last part of the database script for the system.

## D Appendix - Final Database layout

Below in figure 32 is shown a layout for the final database. This database should respect the normalization rules of databases and have more logic names than the current implemented database shown in figure 14.

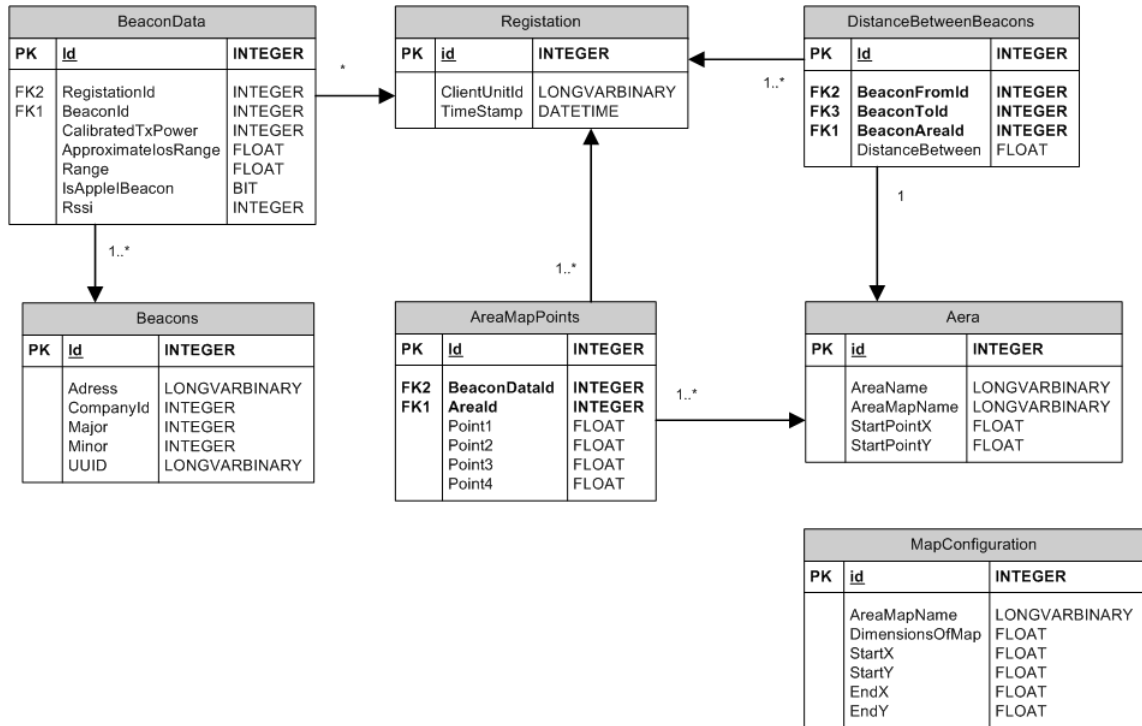


Figure 32: The database model with datatypes nullalbe. Not-nullable data fields is marked with **bold text**. Cardinality are 1 = 1..1, \* = 0..1.

## E Appendix - Raspberry Pi and PC Install guide to use as a Beacon

Beacons are not something consumers or companies need to buy from a manufacturer. It is possible to create their own beacon transmitter with a hardware platform that has x86, x64 or ARM architecture, a Linux distribution and has a bluetooth module that is at least a version 4.0.

The following guide describes how to build a beacon based on a Raspberry Pi or a regular PC. The Raspberry Pi is a single board computer and an example of one, can be seen in figure 33. If the Raspberry Pi is not a version 3 then a bluetooth usb-dongle of the class 4.0 or higher, is needed for the installation beacon to work. Beacon installation also works on an Intel Edison (a x86 single board computer). Just follow from step 5 in the instructions roughly and remove to "tool" folder from the file paths.

Below is found a step guide for how to install and use a Raspberry Pi as a beacon transmitter.

Step 3 to 4 is combined from the following sources [51] and the boot steps, step 5 to 8 from the Ubuntu community [30].

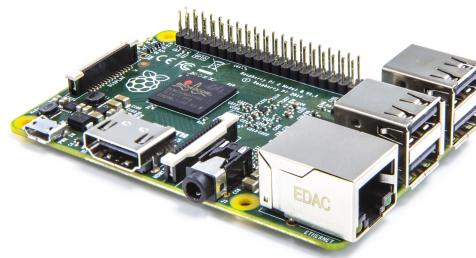


Figure 33: The Pitcher shows the single board computer Raspberry Pi version 2. This model requires a bluetooth dongle to function as a beacon.

1. **First install a Linux distribution of your choice. It is recommended to use Raspbian Lite (Jessie) or the Ubuntu Mate 16.04 for the installation.**
2. **Make sure the distribution is fully updated.**
3. **Installing the bluetooth library controlling the hardware. This is found in listing 3.**

```
sudo apt-get install libusb-dev
sudo apt-get install libdbus-1-dev
sudo apt-get install libglib2.0-dev --fix-missing
sudo apt-get install libudev-dev
sudo apt-get install libical-dev
sudo apt-get install libreadline-dev
```

Listing 3: Code for installing the hardware library for controlling the bluetooth module.

4. **Install the beacon controller software by downloading it. Make and install the bluez-library. This is shown in listing 4.**

```
sudo mkdir bluez
cd bluez
sudo wget www.kernel.org/pub/linux/bluetooth/bluez-5.18.tar.gz
sudo gunzip bluez-5.18.tar.gz
sudo tar xvf bluez-5.18.tar
cd bluez-5.18
sudo ./configure --disable-systemd
sudo make
sudo make install
sudo shutdown -r now
```

Listing 4: Commandlines for downloading and installing bluez-library.

5. Set-up the beacon. Start up with the Raspberry Pi boot up, without login, using a boot script. First login with root privileges and then create the file which is going to hold the boot code. This is done in listing 5.

```
sudo su
nano /home/pi/startbeacon.sh
```

Listing 5: Login as root and create an empty file that ends up holding the beacon startup code.

6. Copy and paste the script code from listing 6, change the parameters needed and save the script.

```
#!/bin/bash
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

#Beacon parameter:
UUID='63 6F 3F 8F 64 91 4B EE 95 F7 D8 CC 64 A8 63 B7'
Major='07 A3' #test numbers
Minor='00 00' # test igen
RSSI='C8' # C8 hex = 200 int

#UUID_NO_WHITESPACE="$(echo -e "${UUID}" | tr -d '[:space:]')"
Major_NO_WHITESPACE="$(echo -e "${Major}" | tr -d '[:space:]')"
Minor_NO_WHITESPACE="$(echo -e "${Minor}" | tr -d '[:space:]')"
RSSI_NO_WHITESPACE="$(echo -e "${RSSI}" | tr -d '[:space:]')"

Write to screen what settings entered
echo "UUID_hex: " $UUID
#echo "UUID: " $((0x$UUID_NO_WHITESPACE))
echo "Major: " $((0x$Major_NO_WHITESPACE))
echo "Minor: " $((0x$Minor_NO_WHITESPACE))
echo "RSSI: " $RSSI "or" $((0x$RSSI_NO_WHITESPACE))
echo

#Find Beacon location folder
cd /home/pi/bluez/bluez-5.18/

#http://linux.die.net/man/8/hciconfig
sudo tools/hciconfig hci0 leadv 3
#Open and initialize HCI device.
sudo tools/hciconfig hci0 up
Disable page and inquiry scan.
sudo tools/hciconfig hci0 noscan

Beacon advertising Data
#http://www.princetronics.com/ibeacon-minimal-setup-with-raspberry-
pi-pibeacon/

sudo tools/hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF 4C 00
02 15 $UUID $Major $Minor $RSSI

read
```

Listing 6: Script code used for setting up the properties of the beacon and start it up afterwards. The script is setup with the pi user (line 23).

7. As the last step, permissions are needed to be added to the script so it can be accessed and launched. This is done in listing 7.

```
chmod u+x /home/pi/startbeacon.sh
```

Listing 7: Adding the permissions so the script can be executable on boot up.



8. Make the beacon turn on at boot up, by inserting the boot script in the startup sequence of the Raspberry Pi.

```
nano /etc/rc.local
```

Listing 8: Opening the file holding some of the startup code in a text editor.

9. Inside the file, add the line `"/home/pi/startbeacon.sh || exit 1"` as show in listing 9.

```
#!/bin/sh -e
#
rc.local
#
This script is executed at the end of each multiuser runlevel.
Make sure that the script will "exit 0" on success or any other
value on error.
#
In order to enable or disable this script just change the
execution
bits.
#
By default this script does nothing.
/home/pi/startbeacon.sh || exit 1 # <----- Add the line here

Print the IP address
_IP=$(hostname -I) || true
if ["$_IP"]; then
 printf "My IP address is %s\n" "$_IP"
fi

exit 0
```

Listing 9: Inserting the boot script in the startup in line 13.

10. Then reboot(see listing 10) and the beacon will now boot up with the Raspberry Pi. A screen-shot of it can be seen in figure 34.

```
sudo reboot
```

Listing 10: commandline for rebooting the Raspberry Pi.

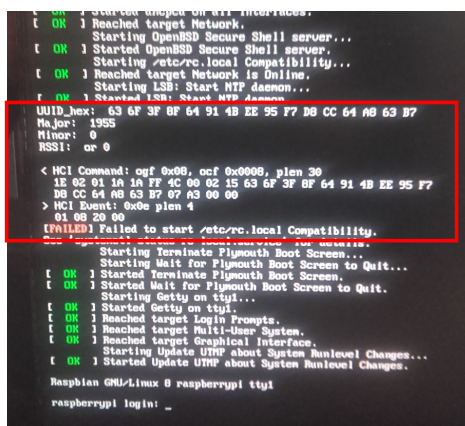


Figure 34: A screen-shot of the Raspberry Pi boot-up with the beacon script turned on. The script is marked with red square.

## F Appendix - Acceptance test of main success scenario

The following section contains the results of the acceptance tests created for the main scenarios of the use cases.

The first use case, Register person, is split into two acceptance tests, one for the Android implementation in table 24 and one for the PC implementation in table 25 .

The second test for Configuration of area is found in table 26, the third test for Show data, is found in table 27 and the fourth for Assign person, is found in table 28.

| <b>Black-box for UC1.1 - Android implementation</b>                                                                            |                          |                     |
|--------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------|
| <b>Main success Scenario</b>                                                                                                   | <b>Expected outcome</b>  | <b>Real outcome</b> |
| 1. The mobile device registers a beacon nearby. (It is possible to determine a position for the mobile device).                | Implemented and working. | As Expected         |
| 2. Waiting for a time period to try to find more beacons and have a higher precision of those all ready found. (mobile device) | Implemented and working. | As Expected         |
| 3. The mobile device sends the beacon-data to the system database.                                                             | Implemented and working. | As Expected         |

Table 24: Black-box test for UC1.1 Main success Scenario - Android client.

| <b>Black-box for UC1.1 - PC implementation</b>                                                                                 |                          |                     |
|--------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------------------|
| <b>Main success Scenario</b>                                                                                                   | <b>Expected outcome</b>  | <b>Real outcome</b> |
| 1. The mobile device registers a beacon nearby. (It is possible to determine a position for the mobile device).                | Implemented and working. | As Expected         |
| 2. Waiting for a time period to try to find more beacons and have a higher precision of those all ready found. (mobile device) | Implemented and working. | As Expected         |
| 3. The mobile device sends the beacon-data to the system database.                                                             | Implemented and working. | As Expected         |

Table 25: Black-box test for UC1.1 Main success Scenario - PC client.

| <b>Black-box for UC2.1</b>                                          |                          |                     |
|---------------------------------------------------------------------|--------------------------|---------------------|
| <b>Main success Scenario</b>                                        | <b>Expected outcome</b>  | <b>Real outcome</b> |
| 1. First select a map which the area is placed in.                  | Implemented and working. | As Expected         |
| 2. Select an area name.                                             | Implemented and working. | As Expected         |
| 3. Then select the area by adding the area beacons (beacon ID).     | Implemented and working. | As Expected         |
| 4. Select an area by reading distance for all beacons within reach. | Implemented and working. | As Expected         |
| 5. The position is determined to a sub-area (as a table or desk).   | Not Implemented.         | As Expected         |

Table 26: Black-box test for UC2.1 Main success Scenario - Configuration of area.

| <b>Black-box for UC3.1</b>                                                                        |                                                                               |                     |
|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|---------------------|
| <b>Main success Scenario</b>                                                                      | <b>Expected outcome</b>                                                       | <b>Real outcome</b> |
| 1. Choose a location.                                                                             | Not Implemented, default map is selected.                                     | As Expected         |
| 2. Show data for today, for the selected location.                                                | Not Implemented, the user have to select parameters before any data is shown. | As Expected         |
| 3. The user change the parameters and the screen is updated.                                      | Partially implemented, require a button press to update.                      | As Expected         |
| 4. The user is presented with a map with an overview of where the selected location is displayed. | Implemented                                                                   | As Expected         |
| 5. Screen automatically updates with new data (see frequency).                                    | Not Implemented, the update requires a button press every time.               | As Expected         |

Table 27: Black-box test for UC3.1 Main success Scenario - Show data.

| <b>Black-box for UC3.2</b>                                                                                             |                                                                         |                     |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|---------------------|
| <b>Main success Scenario</b>                                                                                           | <b>Expected outcome</b>                                                 | <b>Real outcome</b> |
| 1. The system finds all person registrations.                                                                          | Implemented and working.                                                | As Expected         |
| 2. The person registrations are linked to areas based on which beacons they have encountered and the distance to them. | Partially implemented, the distance is not used.                        | As Expected         |
| 3. The user registrations are linked to a sub-category such as a desk or another object.                               | Not Implemented, do to precision issues this feature is not programmed. | As Expected         |

Table 28: Black-box test for UC3.2 Main success Scenario - Assign person.

## G Appendix - Precision test of mobile devices clients

The following appendix contains the results of the precision tests where a mobile device have been registering beacon-data for up to 6 minutes. The results includes a graph of the received beacon-data over time and a calculation of error based on the minimum, average and maximum distance registered.

The 6 devices is listed below with their corresponding graph and table.

- Samsung Xcover 3 graph results in figure 35 and error calculation in table 29.
- Samsung Galaxy S5 graph results in figure 36 and error calculation in table 30.
- Samsung Galaxy S6 Edge graph results in figure 37 and error calculation in table 31.
- Huawei P8 Lite graph results in figure 38 and error calculation in table 32.
- Fujitsu FJNB23F with on-board blue-tooth module graph results in figure 39 and error calculation in table 33.
- Fujitsu FJNB23F with ASUS USB-BT400 graph results in figure 40 and error calculation in table 34.

### Samsung Xcover 3 - Precision test

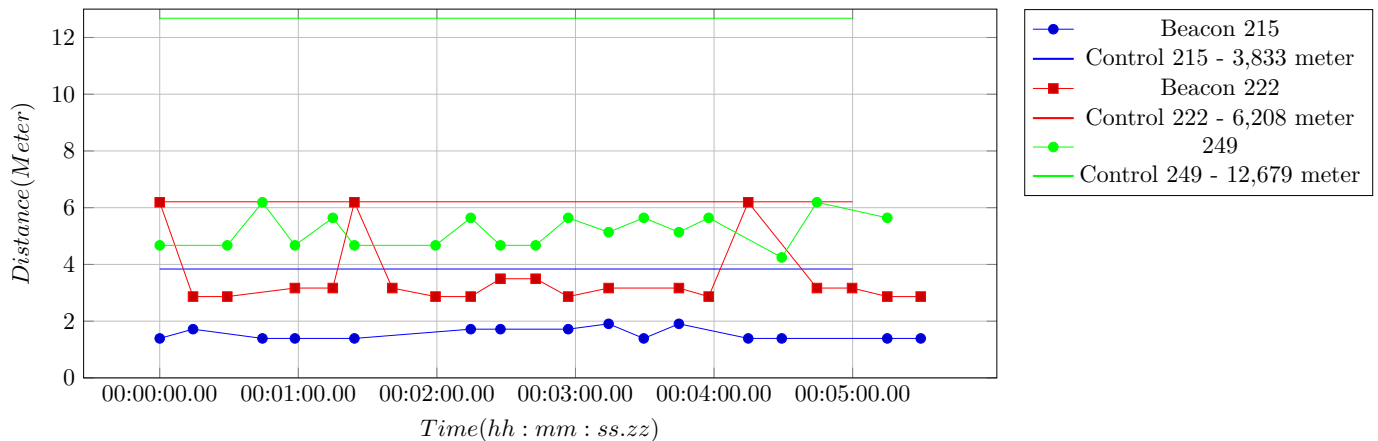


Figure 35: Precision test of the Samsung XCover 3.

| Error statistis for Samsung Xcover 3 |                         |                         |                         |                        |
|--------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                        | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                  | 1,39m<br>-63,73%        | 1,55m<br>-59,68%        | 1,91m<br>-50,33%        | 3,83m                  |
| 222                                  | 2,87m<br>-53,82%        | 3,53m<br>-43,08%        | 6,19m<br>-0,27%         | 6,21m                  |
| 249                                  | 4,25m<br>-66,51%        | 5,22m<br>-58,80         | 6,19m<br>-51,17%        | 12,68m                 |

Table 29: Table overview for the Samsung Xcover 3, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shorten to a two decimal number.

### Samsung Galaxy S5 - Precision test

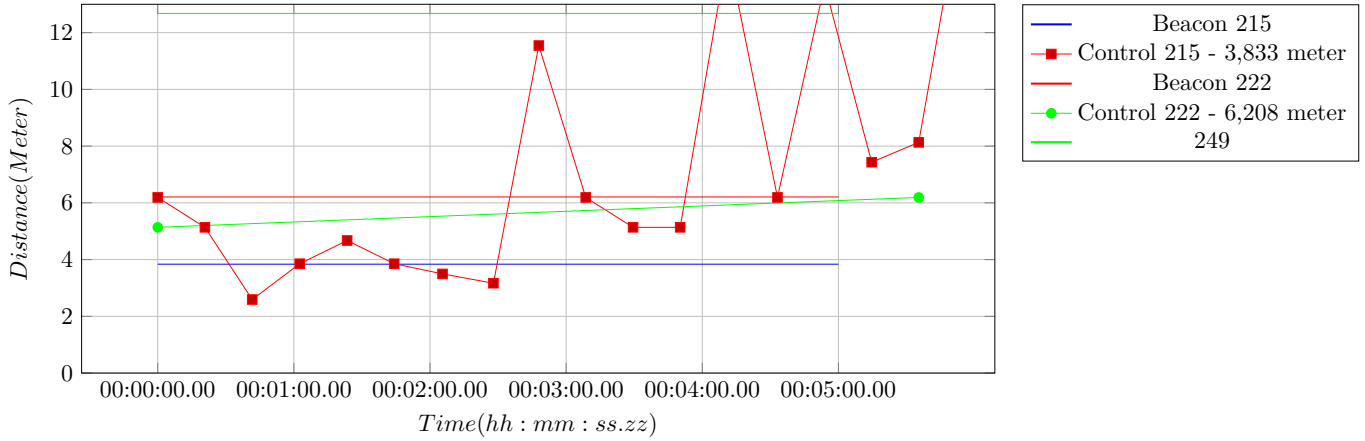


Figure 36: Precision test of the Samsung S5.

| Error statistics for Samsung Galaxy S5 |                         |                         |                         |                        |
|----------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                          | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                    | 0m<br>-100%             | 0m<br>-100%             | 0m<br>-100%             | 3,84m                  |
| 222                                    | 2,60m<br>-58,24%        | 7,00m<br>12,82%         | 17,54m<br>182,50%       | 6,21m                  |
| 249                                    | 5,14m<br>-59,48%        | 6,04m<br>-52,37%        | 6,79m<br>-46,47%        | 12,68m                 |

Table 30: Table overview for the Samsung Galaxy S5, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shorten to a two decimal number.

### Samsung Galaxy S6 Edge - Precision test

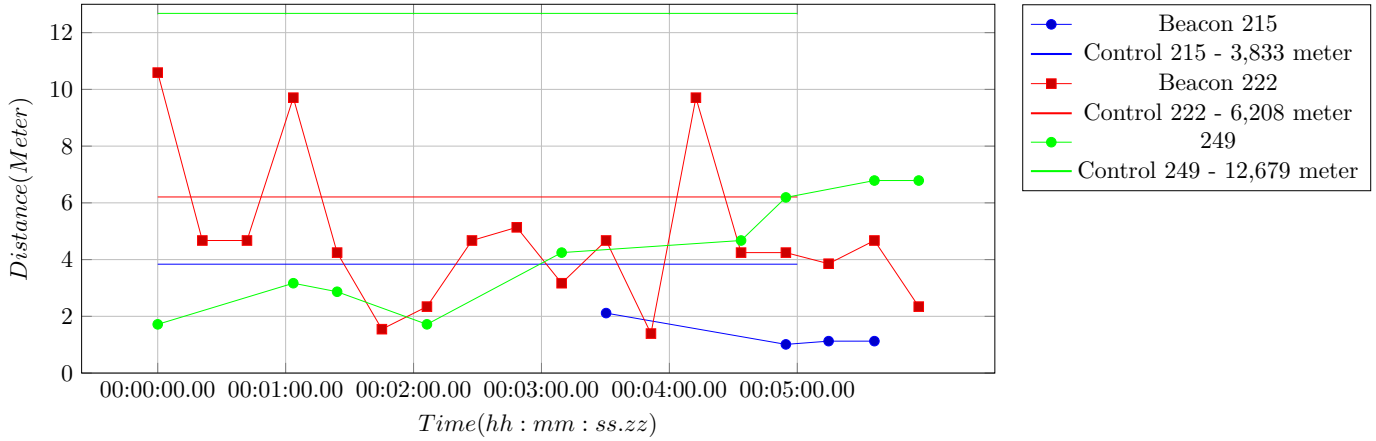


Figure 37: Precision test of the Samsung S6 Edge.

| Error statistics for Samsung Galaxy S6 Edge |                         |                         |                         |                        |
|---------------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                               | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                         | 1,0m<br>-73,66%         | 1,34m<br>-64,99%        | 2,11m<br>-44,92%        | 3,84m                  |
| 222                                         | 1,39m<br>-77,58%        | 4,77m<br>-23,14%        | 10,59m<br>70,61%        | 6,21m                  |
| 249                                         | 1,72m<br>-86,45%        | 4,24m<br>-66,56%        | 6,79m<br>-46,47%        | 12,68                  |

Table 31: Table overview for the Samsung Galaxy S6 Edge, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shortened to a two decimal number.

### Huawei P8 lite - Precision test

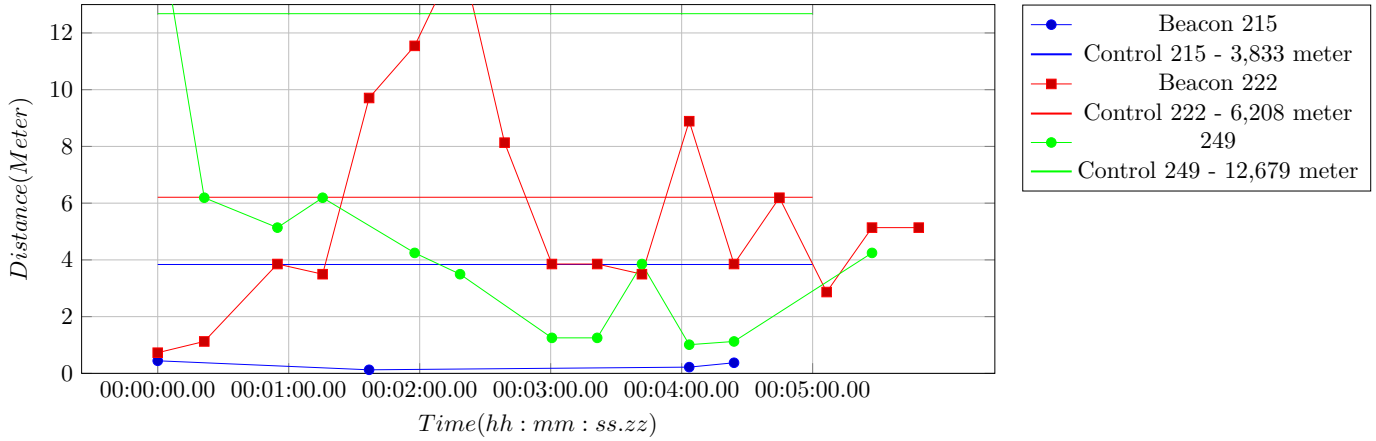


Figure 38: Precision test of the Huawei P8 lite.

| Error statistis for Huawei P8 lite |                         |                         |                         |                        |
|------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                      | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                | 0,07m<br>-98,20%        | 0,27m<br>-93,05%        | 0,44m<br>-88,45%        | 3,84m                  |
| 222                                | 0,73m<br>-88,27%        | 5,36m<br>-13,60%        | 14,88m<br>139,61%       | 6,21m                  |
| 249                                | 1,01m<br>-92,03%        | 4,03m<br>-68,19%        | 16,16m<br>27,44%        | 12,68m                 |

Table 32: Table overview for the Huawei P8 lite, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shorten to a two decimal number.

### Fujitsu FJNB23F with on-board blue-tooth module - Precision test

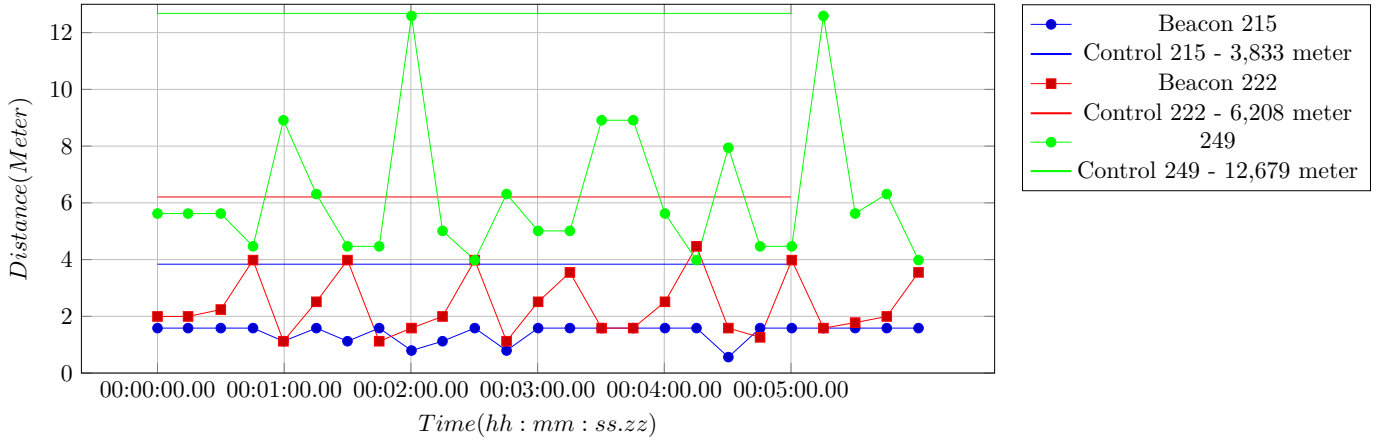


Figure 39: Precision test of the Fujitsu FJNB23F with on-board blue-tooth module.

| Error statistics for Fujitsu FJNB23F with on-board blue-tooth module |                         |                         |                         |                        |
|----------------------------------------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                                                        | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                                                  | 0,56m<br>-85,35%        | 1,41m<br>-63,17%        | 1,58m<br>-58,71%        | 3,84m                  |
| 222                                                                  | 1,12m<br>-81,93%        | 2,35m<br>-62,11%        | 4,47m<br>-28,05%        | 6,21m                  |
| 249                                                                  | 3,98m<br>-68,60%        | 6,35m<br>-49,91%        | 12,59m<br>-0,70%        | 12,68m                 |

Table 33: Table overview for the Fujitsu FJNB23F with on-board blue-tooth module, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shortened to a two decimal number.



### Fujitsu FJNB23F with ASUS USB-BT400 - Precision test

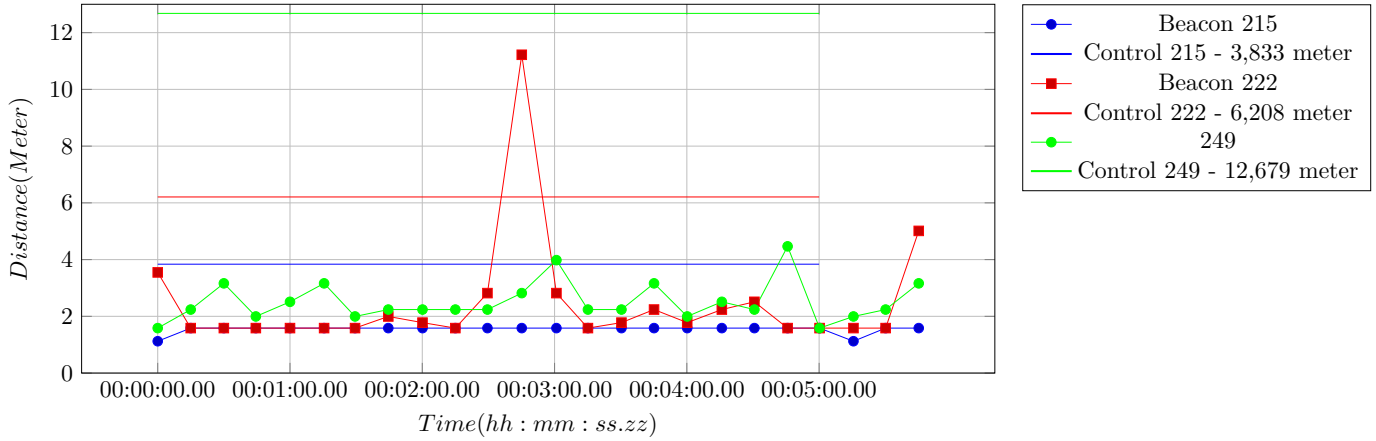


Figure 40: Precision test of the Fujitsu FJNB23F with ASUS USB-BT400.

| Error statistics for Fujitsu FJNB23F with ASUS USB-BT400 |                         |                         |                         |                        |
|----------------------------------------------------------|-------------------------|-------------------------|-------------------------|------------------------|
| Beacon Number                                            | min distance<br>%-error | avg distance<br>%-error | max distance<br>%-error | References<br>Distance |
| 215                                                      | 1,12m<br>-70,77%        | 1,55m<br>-59,51%        | 1,58m<br>-58,71%        | 3,84m                  |
| 222                                                      | 1,59m<br>-74,4%7        | 2,36m<br>-62,00%        | 11,22m<br>80,74%        | 6,21m                  |
| 249                                                      | 1,58m<br>-87,50%        | 2,56m<br>-79,79%        | 4,47m<br>-64,77%        | 12,68m                 |

Table 34: Table overview for the Fujitsu FJNB23F with ASUS USB-BT400, showing the minimum, the average and the maximum distance measured for the three beacons. The %-error beneath the distances is the percentage of the deviation to the reference distance. Please notice that the results have been shortened to a two decimal number.

## H Appendix - Battery test of the Android bluetooth module

Screen-shot of the battery test made for the different Android devices tested with the GSAM application [28], to see how much battery the bluetooth module used by being turned on full time. The results can be found in the figure 41 for the XCover 3, figure 42 for the Galaxy S6, figure 43 for the Galaxy S6 Edge and figure 44 for the Huawei P8 Lite. Please notice that the application used for the screen-shots contains advertisement. Those have been removed from the screen-shots below.

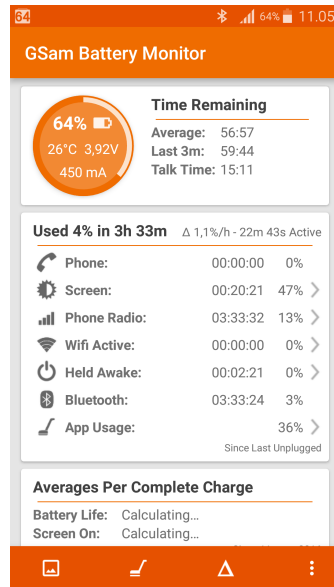


Figure 41: Battery usages of the Samsung Galaxy XCover 3.

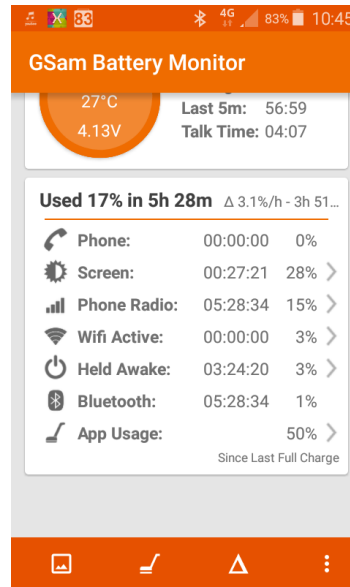


Figure 42: Battery usages of the Samsung Galaxy S5.

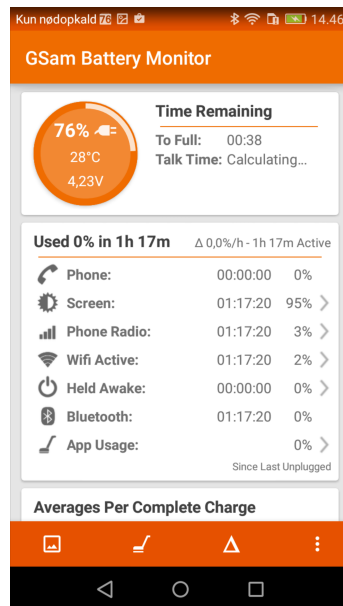


Figure 43: Battery usages of the Samsung Galaxy S6 Edge.

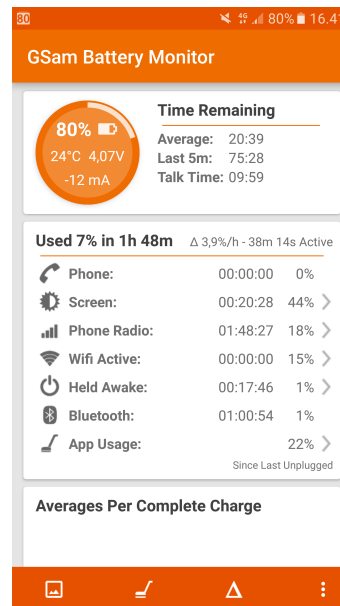
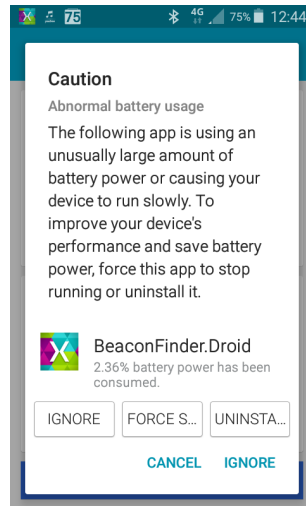


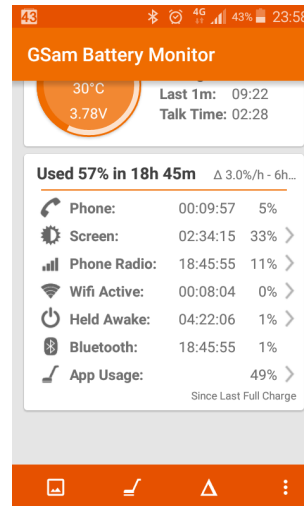
Figure 44: Battery usages of the Huawei P8 Lite.

## I Appendix - Battery test android application readout

Please notice that the GSAM application [28] used for the screen-shots contains advertisement. Those have been removed from the screen-shots below.



(a) Android native handling of battery usages for an Android application. This is an error report of the application.



(b) Screen show the test software used to calculate the battery usages of the entire phone. All apps including the client app usages 49% of the current battery usage.

Figure 45: Screen-shots of the client application running a battery test on an Android phone.

# iBeacons application in the registration and processing of user data

**Author** Magnus Bjarke Bang  
Nielsen, s071853  
s071853@student.dtu.dk

**University** Technical University of  
Denmark

**Institute** DTU Compute

**Supervisor** Stig Høgh  
**(University)**

**Project Type** Master Thesis

**Company** Siemens A/S Mobility

**Supervisor** Jesper Lauge Kilde  
**(Company)** Jesper-lauge.kilde@siemens.com

**Project start** 29/02-2016

**Hand-in date** 29/08-2016

**Danish Title** iBeacons anvendelser  
indenfor registrering og  
behandling af bruger-data

It has become more common for companies to have hotdesk areas. This means that many employees shares their office desks and other equipment. To make sure, that there are enough desks, sufficient air ventilation and heating, the company needs to know when the desks are occupied and how many persons there are in a given area.

Here the beacon technology is chosen together with an application to be installed either on the users smartphone or laptop. The client application is made both for Android and PC and can later be compiled to IOS.

When a person arrives to an area / location, the mobile device(smartphone or laptop) receives a signal from a beacon and the mobile device sends this data with user identification and timestamp to a database via a REST web-service, for later use. The data is anonymized and contains a unique user ID, a timestamp, a unique registration ID and location data from the beacons.

The data is presented in a web-interface, where statistics are displayed. The full system set-up can be seen in figure 1.

Results of the implementation shows that it was possible to locate a person from the implemented solution within a given area. The test results showed an error in precision of 50-60%, so it is not possible to register a person at a desk as initially intended, but only in an area. The major problem seems to be the signal interference in the office environment.

The solution is usable, but not as precise as intended. If higher precision is required, the technology has to be changed.

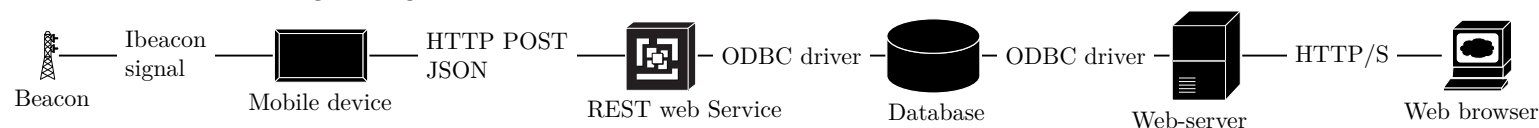


Figure 1: Overview of all system components with their selected interfaces attached.

This project is written as a cooperation with:



**DTU Compute**  
Department of Applied Mathematics and Computer Science