

Project iBeacon

ProjDat 2015

Report 1

March 23, 2015



Emil Slot Arakelian Jensen - 170494

Samuel Korn - 101186

Mark Jay Nielsen - 250694

Rune Bjerg Ono - 180187

TA: Kasper Passov

Contents

1	Problem definition	3
1.1	Overview	3
1.2	Problem statement	3
1.2.1	Problem domain	3
1.2.2	Solution domain	4
1.2.3	Functional requirements	4
1.2.4	Non-functional requirements	4
1.2.5	Deadlines	4
2	Initial Software Project Management Plan	5
2.1	Work breakdown	5
2.2	Roles and responsibilities	5
2.2.1	Internal roles	5
2.2.2	External roles	6
2.3	Project schedule	6
2.4	Budget	6
2.5	Risks	7
3	Initial software architecture	7
4	Project Agreement definition	8
5	Project establishment	9
6	Exercises	10
6.1	1-6.	10
6.2	1-8.	10
6.3	2-6, 2-7, 2-9.	10
6.4	2-10.	11
6.5	5-3.	12
6.6	7-1	13

1 Problem definition

1.1 Overview

The following text, is the original description of one of the many projects made available to us from the company Shape A/S.

How many people are in the office?

Use iBeacons to determine the amount of people currently in the office.

. . . Using iBeacons we would like to make an app that registers whenever the user enters or leaves our office. . . . This could e.g. be used to adjust the heating, automatically lock the doors and turn off the lights when the last person leaves, predict when we need to order more or less lunch, or simply to check whether a certain person is currently at the office.¹

These particular iBeacon devices are made by a company called Estimote, and are described as . . . *small wireless sensors that you can attach to any location or object. They broadcast tiny radio signals which your smartphone can receive and interpret, unlocking micro-location and contextual awareness.²*

On March the 12th, we had a meeting with Søren Ulrikkeholm, who is the developer responsible for student contact at Shape. We talked about the different projects and realistic acceptance criteria, taking into account our limited experience. We settled on the aforementioned project, because it was deemed the most appropriate for our level of skill, as well as being the one in which Shape had the most interest.

Søren explained that the company's biggest interest is learning how to use iBeacons with Android, while aggregation of the data and application of machine learning and data visualization are secondary objectives. We have decided to split the project into two development phases.

1.2 Problem statement

1.2.1 Problem domain

Our primary focus is on the first development phase, which consists of developing an app for Android that registers individual users entering and leaving

the office space.

The second development phase consists of developing the backend for the system and integrating it with the app, in order to aggregate the data from users entering and leaving the office in a database.

1.2.2 Solution domain

Our client will be using this system in their office, with the intention of aggregating data on employees entering and leaving the office. The plan is for them to apply machine learning and data visualizations to this data to discover usage patterns allowing them to implement automatic control of various functions within the office.

1.2.3 Functional requirements

The app must be able to register an iBeacon in the near vicinity.

The app must register a user entering or leaving the office.

The app must communicate with a database to collect this data.

1.2.4 Non-functional requirements

The app must be written in Java and XML for the Android operating system.

The app must be simple to use.

1.2.5 Deadlines

So far we only have a couple of deadlines.

March 22: Delivery of problem definition to client.

June 8: Delivery date for finished product.

2 Initial Software Project Management Plan

2.1 Work breakdown

As of now, we are only in the initial phase of designing our app, therefore the current management plans and schedules are rough drafts.

We have made a very rough draft of a partial work breakdown structure (see figure 1) for the project, but it is still very general and not very detailed.

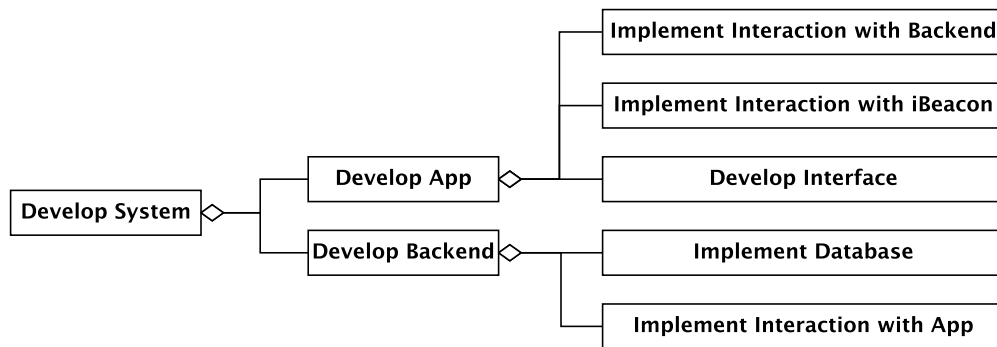


Fig. 1: Partial work breakdown structure for the project.

2.2 Roles and responsibilities

2.2.1 Internal roles

With regards to the development of the system we have not appointed any specific roles or responsibilities. We are a group of only four people and are confident that we do not need to appoint a project manager to supervise the remaining three. We will however, later on in the development, create detailed schedules and individual deadlines for smaller assignments, which we will use to keep track of our progress rather than having a single person be manager. This approach is not quite the same as mentioned in our textbook,³ however, given the size of our team we believe we can manage this project together.

See figure 2 for a skill matrix for our group.

●: Primary skill ○: Secondary skill △: Interest

	Emil	Samuel	Mark	Rune
Graphic design	△		△	△
Database design	○	○	○	○
Project management		●△		
Application development (Java)	●△	○△	○△	●△
Database interaction (Python)	○	○△	○△	○△
Database implementation (SQL)	●	○	●△	●
Object oriented design	●	○	○	○
Report-writing		○		○

Fig. 2: Skill matrix.

2.2.2 External roles

With regards to external roles, Rune is responsible for communication with the client and Samuel is responsible for communication with our appointed TA, Kasper Passov.

2.3 Project schedule

March 16 - March 22: Problem definition.

March 23 - April 5: Initial design phase.

April 5 - June 7: Main development phase.

2.4 Budget

We do not plan to have any expenses tied to the development of our project, nor does the development software pose any costs. Shape A/S will provide

us with access to the iBeacon devices.

2.5 Risks

INSERT

3 Initial software architecture

We are developing the system from scratch rather than building on top of existing software. As we are developing the app for Android, it will be programmed mainly in Java as well as XML. We expect to be implementing the backend of the system as some sort of SQL database.

4 Project Agreement definition

The following is a translation of the Project Agreement definition, agreed upon by both our client and ourselves:

Together with Shape A/S we have chosen to work on project 5 from their website www.shape.dk/projects. The project consists of developing an app for the Android and iOS operating systems that, using an iBeacon device, collects data showing how many people are present in their office at a given time. This data is collected with the intention of finding usage patterns which could potentially be used for automatic control of a number of functions in the office (turning the lights on/off, regulating the room temperature etc.).

During our meeting with the client on March 12th we discussed how the project can be split into 3 development phases. First phase involves designing and developing an app which collects data from an iBeacon. Second phase is to integrate the app with a backend database that stores all the collected data, and the third phase revolves around the use of the stored data for data visualizations and machine learning purposes.

An app for both iOS and Android would be too much for us to handle, which is why we have agreed to only develop the app for Android. Our limited programming experience means that we will be focusing solely on the first two development phases, i.e. we will only be designing and developing the app and implementing the database. A temporary schedule follows.

March 16 - March 22: Problem definition.

March 23 - April 5: Initial design phase.

April 5 - June 7: Main development phase.

June 8: Product delivery date.

We have made an arrangement with Shape A/S to be able to work on the project in their office one day of every week.

Our next meeting with the client will be on March 25th at 10 o'clock, where we will have our first look at the iBeacon device.

Aside from their time Shape A/S will have no expenses tied to this project.

5 Project establishment

We have reached an agreement between our group and our client, consisting of developing an app for the Android operating system and a backend database.

The project depends heavily on the iBeacon devices. These are relatively unknown to us for now, but the developers at Shape A/S already have some experience working with them. For the duration of the project, our client will be providing us with access to the iBeacon devices and any necessary guidance in the use of these, as well as letting us work on the project in their office about once a week.

Our client's interest in this project is to receive a finished product for use in their office. The interest of our group is to gain experience working in a corporate setting as well as extending our software development skills. No other parties have an interest in this project.

6 Exercises

6.1 1-6.

Specify which of these statements are functional requirements and which are nonfunctional requirements

Functional requirements:

- "The TicketDistributor must enable a traveler to buy weekly passes."
- "The TicketDistributor must always be available."
- "The TicketDistributor must provide a phone number to call when it fails."

Nonfunctional requirements:

- "The TicketDistributor must be written in Java."
- "The TicketDistributor must be easy to use."

6.2 1-8.

In the following description, explain when the term account is used as an application domain concept and when as a solution domain concept:

"...managing bank accounts for mobile costumers." - It is used as an application domain concept, saying what it is supposed to do. "...provide access to the accounts when the..." - It is used the same way as before. "One proposal is that accounts are made available on the mobile computer..." - Here it is used as a solution domain concept, as it is a possible solution to be evaluated. "...the accounts show the amounts from the last connected session." - It is used in connection with the previous use.

6.3 2-6, 2-7, 2-9.

Draw a class diagram representing a book. Add multiplicity to the class diagram you produced in Exercise 2-6. Extend the class diagram to include attributes.

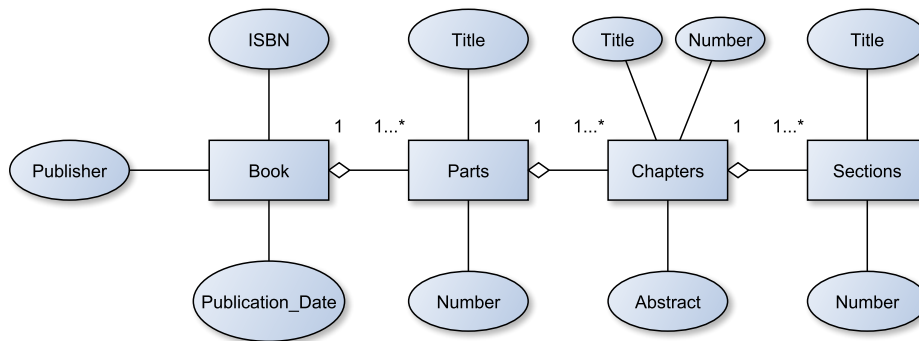


Fig. 3: Class diagram for exercise 2-6, 2-7 and 2-9.

6.4 2-10.

Add an abstract class and an inheritance relationship to factor out these two attributes into the abstract class

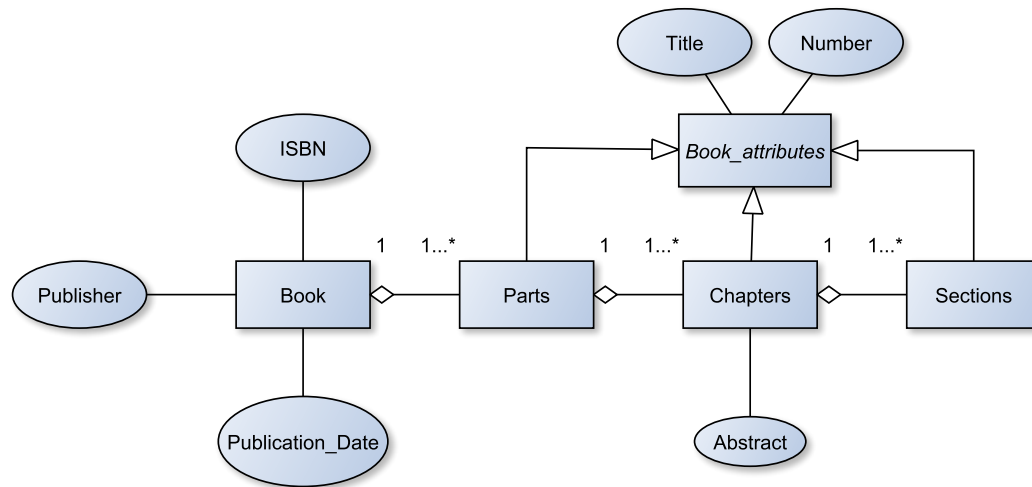


Fig. 4: Altered class diagram for exercise 2-10.

6.5 5-3.

Arrange the objects listed in Exercises 5-1 and 5-2 horizontally on a sequence diagram.

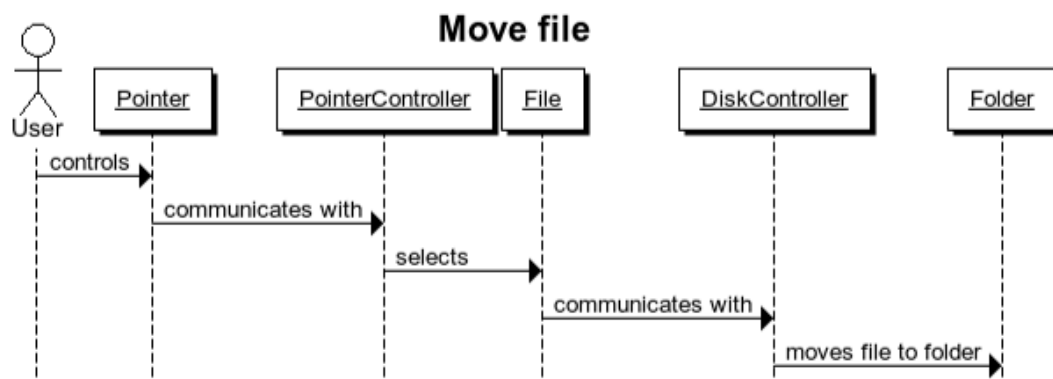


Fig. 5: Sequence diagram for exercise 5-3.

6.6 7-1

Draw a UML deployment diagram representing the hardware/software mapping.

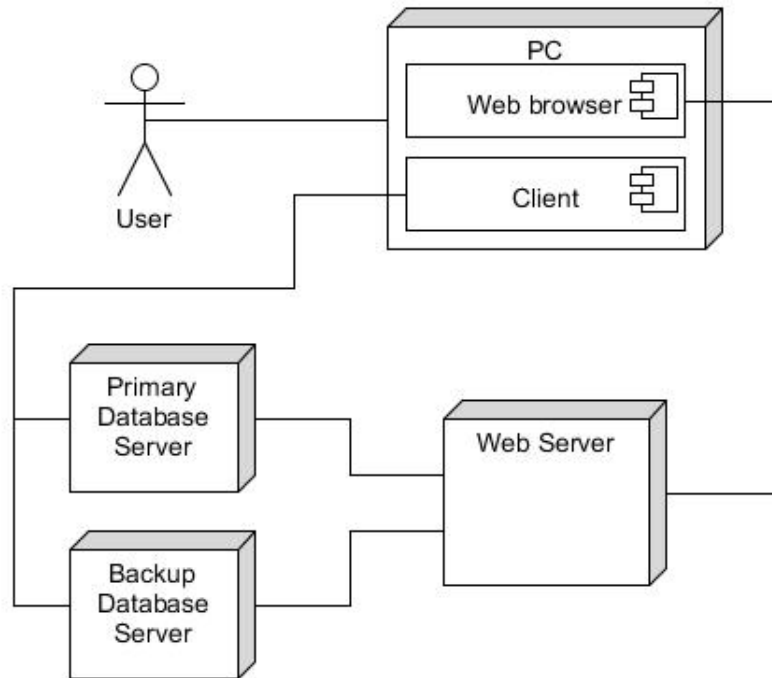


Fig. 6: Deployment diagram for exercise 7-1.

References

¹ <http://www.shape.dk/projects>

² <http://www.estimote.com>

³ Bruegge & Dutoit (2010), Object-Oriented Software Engineering - using UML, Patterns and Java, Third edition, Pearson 2010. Page 611.