NTNU – Trondheim
Norwegian University of
Science and Technology

IT2901 Informatics Project II

UbiLearn

Group: SINTEF_training

Ingeborg Ødegård Oftedal
Ragnhild Jølstad Seim
Kyrre Laugerud Moe
Espen Hellerud
Vegar Lerpoll
Kenneth Pettersen Lund

Spring 2014

Abstract

This report was made during the course IT2901 Informatics Project II at Norwegian University of Science and Technology, NTNU. The purpose of this report is to describe the development process of UbiLearn, an advanced e-learning application, targeted at home- and geriatric-care.
UbiLearn was developed in the spring of 2014 by a group of six students. The android application consists of two main learning methods; a practical method and a theoretical training method.
From the user perspective, the goal is to conduct the quizzes and practical exercises to get to the next level of case challenges. The completed cases are evaluated with a score.
One of the main goals of development of the application, was to offer both educated homecare personnel and non-educated personnel, a new and creative way of learning.

Contents

List of Tables

List of Figures

Chapter 1

Project introduction

The first chapter of this report will give a brief introduction of the project. We will start by giving a short overview of the report structure by summarizing the contents of each chapter.
This chapter also provides a short presentation of the course, our customer and other stakeholders involved in the project. In addition, we will give a presentation of the project group, including organization and management of the project.

1.1 Report structure

Chapter 1
The first chapter is an introduction to the project. It contains a short presentation about the project, including information about the course, the team and stakeholders.

Chapter 2
The aim of this chapter is to introduce relevant information about the research and prestudy phase of this project. This includes the stated reasons for the chosen process model, development tools, and domain knowledge. The chapter is placed early in the report because it contains relevant information to get a full understanding by further reading.

Chapter 3
This chapter aims to present the project management for this project. Including how the team has planned and organized tasks throughout the project duration.

Chapter 4
This chapter is a presentation of the system requirements, including an explanation of the process of specifying the system requirements.
The requirements are presented from both a user perspective, and a functional perspective.

Chapter 5
Based on the system requirements, this chapter present an overview of the system architecture. This chapter provides both a graphical and written description of logical architecture and physical deployment.

Chapter 6
This chapter contains information about the system testing. An explanation of the testing process is presented, including which tests have been performed on the system, how these have been performed, and the retrieved results.

Chapter 7
Chapter 7 addresses the development process of the project, from start to finish. This chapter reviews the entire project process, reflecting on each completed sprint.

Chapter 8
In chapter 8, the team evaluates the project process.

Chapter 9
Chapter 9 is an overall conclusion of the project.

1.2 The course IT2901

The course IT2901 is a practical programming project course, which gives the students the opportunity to develop a software product for a given customer. The project is conducted by a group of students collaborating to develop the final product and project report.The course is 15ETCS, and is a mandatory part of the bachelor degree in computer science at NTNU.
The purpose of this course is to provide the students with practical, realistic experience in the implementation of all phases of a larger system design project.

1.3 SINTEF

The customer for this project is SINTEF Group for Social Inclusion Technologies (SIT). SINTEF is Scandinavia's largest independent research organization and was founded in 1950 and develops technological solutions for practical use.

SINTEF SIT conducts research in technologies and processes to be used to increase stakeholder inclusion and influence. They have a main focus on innovative methods of inclusion with new technologies and processes.

1.4 SINTEF_training

1.4.1 Presentation of the project

The society has a great need of improvement in health service and homecare offers, especially amongst elders. Unfortunately the quality and expertise within this services are varying, and this is something SINTEF SIT is working to improve by developing a training platform for homecare personnel.
The purpose of the project is to develop an e-learning platform targeted at home- and geriatric-care.
The objectives for the application is to make it easy to use, and engage the users by providing learning-by-doing. As well as learning, the application contains a practice part, allowing the users to practice what they have learned in a real life environment.

An important goal for the end product is that the users find the application entertaining, and understands the purpose of the different tasks it provides, both theoretical and practical.
The customer stated early on that the team should spend time on researching for possible solutions for the final product. Due to this approach, the project has approximately been as much a research project as a development project.

The customer's expectations for the project, was that the deliverable application can serve as a prototype, and be fully implemented later on. Assuming the product will be of interest for further development.

1.4.2 The goals for the project 

The main goal for this project was to create a product that the user group will find both useful and educational. Because the final product will function as an aid in such an important area of society, makes demands of the delivered product to meet the requirements set by the customer.
The main target group is non-ICT users, focusing on those without sufficient education. The age group can vary in between 20-60 years.
The group members had both individual and common goals. The team strived to achieve the customers acceptance by developing a product that corresponds to the functional requirements and the customers expectations. The team wanted to learn more about application development, taking time to learn new tools and working environments, in addition to get the experience of working on a project together as a team.

1.4.3 A short presentation of the application

This subsection will give a brief presentation of the application, to provide the reader with a better understanding when reading the rest of the report. For a more detailed description of the different sections the application is composed of, see the *User Guide* found in Appendix A.

The application consists of two main parts; the practical part and the training part.

Training: The training section of the application works as the theoretical learning method. The user is presented with a game board, depicting a neighbourhood where the houses function as a patient house.
When entering a house, the user will be presented a case description, and should attempt to complete tasks related to the case. The user will recieve feedback, as number of points, scored after completing the case.

Practice: In the practical section of the application, the user will get the chance to practice what they have learned from the training part.
In this section, the user can access a variety of exercise programs and relevant health tests, to conduct with a patient.
This part will be used when the user is at a patient house, in real time environment, and is as much a way of learning for the user, as for the patient.

Profile: When entering the application, the user can choose between user registration, or access as an unregistered user (offline access).
When registered, the user will be able to access more functionalities. This includes recieving a profile in the application, which contains the users progress throughout the training part, saved reports, and general user information.

Handbook: The handbook contains different articles about diseases, symptoms and preventive actions. The handbook provides the user with useful theory, and can be used for pre-reading before solving the different tasks in the training-game.
This page will be more relevant to the non-educated users.

1.4.4 Final product

The product developed by the team, is a part of a more comprehensive project managed by SINTEF ICT.

At startup of this project, the customer requested the team to develop a prototype, including the features needed to test the main functionalities of the application.
The final product delivery should not function as a finished system, but rather function as an extended protoype. Providing the developers at
SINTEF who will continue this project, a starting point for further research and development.
Although the end product of this project is defined as a prototype, it will still be referred to with terms such as "product" and "system" in this report.

The project has implied the team to collaborate with SINTEF, as the main customer, but also with the relevant user group representatives from St. Olavs hospital. Input from domain experts have been important to create a product that can meet the user's needs.
The strategy used, is an iterative process where ideas, prototyping, data collection, analysis and learning is central.

1.5 Stakeholders

This section provides a brief presentation of the stakeholders in this project.

In this report, a stakeholder is defined to be a person or a group of people, who affects, or is affected by the project. By this mean, a stakeholder can be someone who works actively on the project, or be in a position to influence the project's success, such as deliverables.

The project team is shortly presented in section 1.5.1, and the team organization will be presented more thorougly in Chapter 3.
Project Management.

1.5.1 The team

The team consisted of six students belonging to the Institute of Computer and Information Science (IDI) at NTNU. All members have knowledge about the system development process and programming through NTNU subjects, but with different expertise in these fields.

Different skills and knowledge has been an advantage for the group, making everyone able to contribute in different areas. All team members were very motivated to work towards a good end result.

The team consisted of the following students:

- Ingeborg Ødegård Oftedal
- Ragnhild Jølstad Seim
- Kyrre Laugerud Moe
- Espen Hellerud
- Vegar Lerpoll
- Kenneth Pettersen Lund

1.5.2 External stakeholders

Table 1.1 presents an overview of the external stakeholders for this project. The table includes contact information and the stakeholders role during the project process.

Table 1.1: Overview of external stakeholders

Name	**Title**	**Responsibility**	**Contact information**
Monica Divitini	Professor	Coordinate the course	divitini@idi.ntnu.no
Alfredo Perez Fernandez	Supervisor	Supervise the group during the project process	alfredo.perez.fernandez-@idi.ntnu.no
Babak A. Farschian	Customer	Customer contact. Coordinating the project from the customer point of view	baf@idi.ntnu.no
Jorunn L.Helbostad	Domain expert	Advice the customer and the team	jorunn.helbostad@ntnu.no
Randi Granbo	Domain expert	Advice the customer and team in falls and mobility among the elderly	randi.granbo@stolav.no

Chapter 2

Prestudy

In this chapter the initial research phase of the project is presented. The team also justifies the choice of process model, development- and management tools.

2.1 Process model

The choice of which process model to implement during software development, is an important part of the initial phase. A process model provides a good base for the development process, and there are various models to choose from.

In this project it was found appropriate to implement a process model based on agile development. The basis for choosing an agile methodology was the fact that it as not established any product requirements from the start. Also the customer xpressed early on, that the project should be an agile process to make the team prepared to handle changes during the process.

The team has experience with agile process methodology from previous projects and saw the benefits of utilizing the expertise of the customer and own experiences to implement in the process model.
After thorough evaluation with the customer, and amongst the team members, it was decided that SCRUM would be an appropriate process model to adapt to the project.
An elaboration of the choice is made in the following subsections.

2.1.1 SCRUM

SCRUM is an incremental and agile process model which emphasises on being more adaptive, than predictive, to the changing requirements of a project. Being able to handle the changes in customer requirements made it a suitable process model for this project. The team did not spend much time looking into other process models, as the whole team has previous experience with SCRUM and found the main concepts the model to be suitable for UbiLearn as a project.

SCRUM consists of a team with different roles: product owner, SCRUM master and development team. The functionality of the product to be developed is put in a prioritized product backlog.

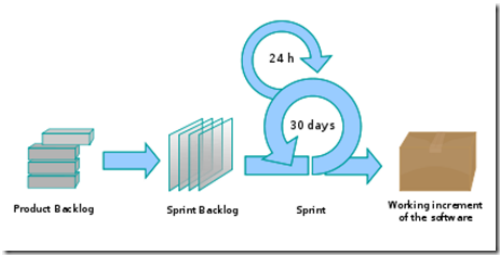


Figure 2.1: Illustration of the SCRUM process. Source: Daniel Root

The SCRUM-team works in a series of iterations called sprints, which last from one to four weeks. Each sprint begins with a planning meeting, where a set of functions from the product backlog is put in a sprint backlog, which then will consist of the goal of the iteration. A time and cost estimate will be assigned to each task.
The sprint backlog specifies a detailed list of what should be implemented during this sprint. Each sprint concludes with a review. The review makes the team able to get valuable feedback from the customer, and summarizes the iteration.

Although SCRUM has many good concepts, the team had to make some changes to adapt the model to match this projects needs. These adaptations will be explained in subsection 2.1.2.

2.1.2 Development model: grounds for choice

The team chose an agile development process because it permits great flexibility, both in terms of changes in system requirements, but also when it comes to development.
Agile development facilitate customer involvement by letting the customer take part in the process throughout the entire duration of the project. The customer for this project wanted to take part in the process by reviewing the team's deliverables for each sprint. This encouraged a high level of feedback, and allowed for innovative ideas, new knowledge and changes in requirements even late in the process.
Initially the agreement with the customer was to have weekly meetings on Fridays, but this was not always possible for the customer, and the team had to be flexible in terms of changing the meeting schedule in the last minute. The group found the cancellation of meetings, and lack of feedback,
challenging in the initial phase of the project, but one of the key principles of SCRUM is the recognition that customers can change their minds about the product during the project. As such, by working iteratively, the team minimized the risk of not being able to deliver a satisfactory product, and could adapt to the changes of meeting times by working on the requirements that was agreed upon in advance.

This way the team could continue on the project development even though the customer was not able to give feedback on a sprint, and if the customer wanted to change something it would not cost the group a big risk.

The team has not been a standard development team, as it consisted of six students, whom the everyday life involves other obligations and work in addition to the project. By such, there were some concepts SCRUM offers, the team concluded to not inaugurate.
These concepts will be listed and reflected on in the following sections.

Stand-up meetings

In SCRUM it is normal to have daily meetings called stand-up meetings. These meetings is used to inform the rest of the team about each team members work progress since last meeting, plan what will be done until next meeting and review any problems that may have occurred. Because the team was not able to have daily meetings, this concept was not used to its full, but rather implemented as a part of the weekly sprint meeting.

Planning poker

Planning poker is a process where the members of the group estimate tasks individually by using numbered cards. SCRUM make use of this time estimation for each task in a sprint.
This estimation process was found to be too time consuming for the project team, after trying to implement it the first sprint. As a result, the estimation process was rather discussed in plenary each sprint meeting, instead of using planning poker.

It may be that planning poker would have become more effective with more experience, but the team found plenary planning to work so well that it was not necessary to spend more time trying to streamline the use of planning poker. One can also see from the results of the burndown charts in Chapter 7, that it has not been experienced underestimation of sprints, although an alternative method of estimation was used.

Cost estimate

For each sprint or iteration, a related cost estimate is made. This is to keep track of the amount of required resources a given budget can exploit. This estimation was considered redundant for this project because the project is based on a subject, and by such, the only resources available are based on the team members hours and not a customer budget.

2.2 Domain knowledge

In order to meet the product requirements, and develop the best possible solution for the project, the team found it important to achieve knowledge about the content area for the environment the product will be used.
As requested by the customer, the team spent the initial phase of the project on researching both on homecare, with different areas of expertise, and how e-learning works. This was to get an overview of the task, and to have a good knowledge base to work from.
The customer also provided the team with a meeting with a domain expert, Jorunn L. Helbostad. Jorunn is a professor of movement science, and is currently doing research on movement difficulties and falls amongst elderly. At the meeting with Jorunn the team held a demonstration of the paper prototype, and got some constructive and useful feedback in regards of the design and application contents.
Jorunn showed interest in taking the project further, and introduced other researchers at St. Olavs hospital who were working on a similar project.
For the team, it was very useful to get feedback and directions by a domain expert and to be provided with quality assured exercise templates and patient cases for the application.

2.3 Development tools

The following subsection will give an overview of the different development tools and technologies used in this project. The reason for choosing the listed tools will also be justified.

Eclipse is a development tool for software consisting of an integrated development environment (IDE) with extended functionality using program extensions. This tool were selected by taking into consideration the team's experience of using this software, and also because Java was the required programming language for this project. Because the product is an Android application, the team used the integrated development environment provided by Eclipse, ADT. Android Development Tools (ADT) is a plugin for Eclipse IDE that provides the tools necessary to develop an application in Android.

LaTeX is a document preparation system for typesetting. It was used for

typesetting the report. Because none of the team members had experience in using this tool, it was conducted some time to learn how to use the system properly.

2.4 Management tools

In order to communicate and collaborate in the best way possible, the team adopted various management tools.

Table 2.1: Summary of management tools

Tool	**Purpose**
Google Drive	File sharing and organization
Google Spreadsheet	Sprint and product backlog
Facebook	Group communication outside meetings
Mail	Communication with customer, supervisor and domain expert
Git	Programming collaboration
MindMeister	Brainstorming tool

Google Drive

To easily collaborate in editing and storing the report and other project documents, Google Drive was used as an online storage space. Google Drive enables the users to access the shared files and Google documents anywhere and allows simultaneous editing. This was a big advantage for the group who often worked simultaneously but from different places.

Google Spreadsheet

To keep track of the progress and time spent on activities throughout each sprint the team created a Google Spreadsheet. Here every group member could fill in their working hours, in addition to keeping all group members updated and synchronized. It was also easy to make a burndown chart for each sprint based on the table data.

Facebook

The team shared a Facebook group with the purpose of easily communicate within the group, outside meetings. Whenever a team member needed to update the rest of the group, for example meeting times, or exchange other information, everyone would get a notification that something had been posted, and one had the possibility to see which members actually had seen the post. Therefore the private Facebook group was used as the main forum to exchange information.

Mail

Communication with the supervisor, customer and domain expert was mainly done by mail, in addition to meetings. The team had one person in charge of communicating with the customer and supervisor. Mainly because it was easy for the supervisor and customer to only relate to one person from the group, and also more easy-to-grasp for the team, with one responsible contact person.

Git

Git is a distributed version control and source code management system. Git was used by specifying every working directory as a standalone repository with complete history and full version tracking capabilities, not dependent on network access or a central server.

The reason for using Git was because several of the team members already had experience with this system. In addition, the customer had already created a repository for the project that the team hosted as a remote repository on github.com. This way he was able to keep track of the development process by watching the commits done by the team.

MindMeister

At the beginning of the project MindMeister, an online cloud service brainstorming tool, was used to brainstorm on ideas. At this point the team did not have access to a room with white board, so this was a good alternative solution.

2.5 Alternative solutions

The purpose of this subsection is to present some of the considered solutions associated with this project. The research and background work is based on problem elaboration and analysis, and was done to inspire the team to specify a product solution. The basis for the chosen solutions has been agreed upon with the customer.

The prestudy phase of this project revealed that there are various mobile health applications on the market. However, the team did not find any applications specializing in training homecare personell.
The customer has, in 2012 and 2013, supervised a thesis on fall prevention, and how mobile and ubiquitous computing can support these measures for the elderly. The work done in these reports was focused on a specific part of the geriatric care, fall prevention. Because this project was to integrate more geriatric aspects, the team used the ideas on fall prevention as a basis for the first implementation and prototype of the application.

The team has evaluated several alternative solutions for this project. As described in subsection 1.3.3, the product consists of two main parts. By such, if the team would have discovered that the project was to comprehensive, smaller subtasks in the practical or theoretical part could have been excluded or reduced, and still allowed the team to deliver an appropriate end product. The way this project has been put together, made it possible for the team to both decrease and expand the size and scope of the project along the way.
One method used to reduce the workload of this project, was by excluding the requirements categorized with importance = 1, as these were not necessary to complete the product. At the same time, the team could easily have added more functionality by including the lower prioritized requirements.

Research and analysis was conducted on several occasions as needed. As a result, the solutions chosen for this project is based on the teams evaluation, and in agreement with the customer.
One example of an alternative and extended solution that was evaluated with the customer, related to the educational part of the applicaiton. When a user performs a patient case, a possible solution could have been to extend each case question, so that one patient case includes multiple sub-tasks and related

quizzes. The team did not have enough time to implement this extensions, but consider it as an opportunity for further development.

The team also looked into possible backend solutions for UbiLearn. Moodle, an open source learning management system, was one of the solutions considered. Moodle offers the user a complete core which can be further developed, adapted to users needs. The system also includes a well documented API. The cons of implementing Moodle as a solution for this project, was that Moodle is very comprehensive, with a lot of of unnecessary functionality. It was also found complicated to get started as it required a lot of configurations and dependencies. For the team, it was not clarified how one would go about adapting to the project requirements.

Another backend solution that was evaluated was Django, an open source web application framework. This framework emphasizes reusability and pluggability of components, by allowing the user to write a small amount of code, and get automatically generated code for running a website with a database. Django also automically generates an admin page for inserting contens into the database.
Although this solution has many good features, the main concern was that Django is written in Python. The use of Python throughout the framework made this solution complicated for the team, as none of the team members had prior experience with the language. In addition, the project description for this project, states that the backend should be written in Java.

Based on the backend analysis presented to the customer, it was agreed upon that neither Moodle or Django provided the optimal solutions for this project and was disregarded.

2.6 Risk analysis

The group assembled in fellowship a risk analysis to give an overview of risks likely to occur during the project.
The analysis presents risks the team see as potential obstacles for this project, and each risk is presented with a plan of action to be performed if the risk should occur.
The preventive and remedial action provides the group with methods to avoid the listed risks, but also possible solutions if the risk should occur, and functions as an aid to minimize impacts of the risks.
The risks are presented with a number that represents its importance.

The degree of importance is calculated with the formula:

$$Likelihood * Impact = Importance. \tag{2.1}$$

This is a measure of the probability of an risk occurring multiplied with the estimated consequence for a risk; the factor of loss in development progress. These measurements provided the team with an estimation of the importance of each risk.
The higher the importance of a risk is set to, the more important it is to find good preventive and remedial actions. By such, the team had procedures on how to avoid these situations, but also what can be done if a situation can not be prevented, and the problem occurs.
The calculations of the risks is based on risk analysis from earlier project reports, taking into consideration their experiences, in addition to the teams own experiences and assumptions.

Table 2.2: Risk analysis

ID	Description	Likelihood	Impact	Importance	Preventive action	Remedial action
1	Underestimation of time needed to finish the project	5	5	25	Estimate tasks as a group, add a buffer (overestimate)	Downsizing the tasks. Reevalute if the task is necessary for the final product. If needed, drop low priority requirements
2	Customer is unavailable	7	3	21	Keep regular contact with the customer, plan meetings ahead. Also use a project management methodology that minimize the impact	Keep working on the project based on earlier agreements and requirements

3	Group disagreement or conflict	3	7	21	Good communication within the group,let all members voice their opinion	Try to resolve, if necessary meet with supervisor
4	Project complexity	3	6	18	Make sure that the project specifications are reasonable. Define project requirements in advance. Define project responsiblitiies,within the team	Contact supervisor and customer, make an agreement on a solution. Reevalute what we can do. If necessary, consider dropping low priority requirements
5	Short-term absence (1-2 weeks)	9	2	18	Keep a good working environment and group communication. Use a working buffer, so loss of resources will not impact progress	Keep regular contact, update the group on current tasks being worked on. If necessary, divide the relevant tasks on the remaining group members
6	Group member drop out of the course	2	9	18	Involvement of all group members in the process. Critical procedures should be secure and accessible for all members to minimize the impact of amember dropping out	Contact professor, supervisor and reassign tasks. Consider downsizing the project by re evaluating requirements and complexity

7	Customer conflict	2	7	14	Maintain good communication with customer, and have regular meetings. Create mutual understanding to prevent and minimize misunderstandings	Try to resolve conflict, if not possible involve supervisor
8	Group member doesn't participate	2	7	14	Good distribution of tasks is planned during sprint meetings. Follow up the determined delivery deadlines during sprint meetings to make sure the tasks have been done.	Contact supervisor and professor at an early point. Reduce workload if needed
9	Long-term absence (illness, injuries etc.)	3	4	12	Provide a good working environment to prevent illness. Keep good health	Maintain contact through the absence. If necessary, divide the tasks of the ill person to the rest of the group, or drop a low priority requirement

10	Internal misunderstanding of project	2	5	10	Develop a specific project description and requirements specification. Define stakeholder needs. Develop clear planning with well defined deliverables. Regular meetings as a group to keep everyone up to date. Use a process model that avoids the risk of misunderstanding (SCRUM) by regular meetings and specified deliverables.	Gather team members for a meeting to clarify project details
11	Customer makes changes to the project requirements	7	1	7	Use an appropriate process model(SCRUM). Keep regular contact with the customer	Make necessary changes to the new requirements and calculate new time estimates
12	System failure (loss of work)	1	7	7	Maintain computer and make regular backups for example by using Dropbox (cloud storage) to have files avaliable from anywhere.	Use failure recovery, locate the problem if possible, then debug

Chapter 3

Project management

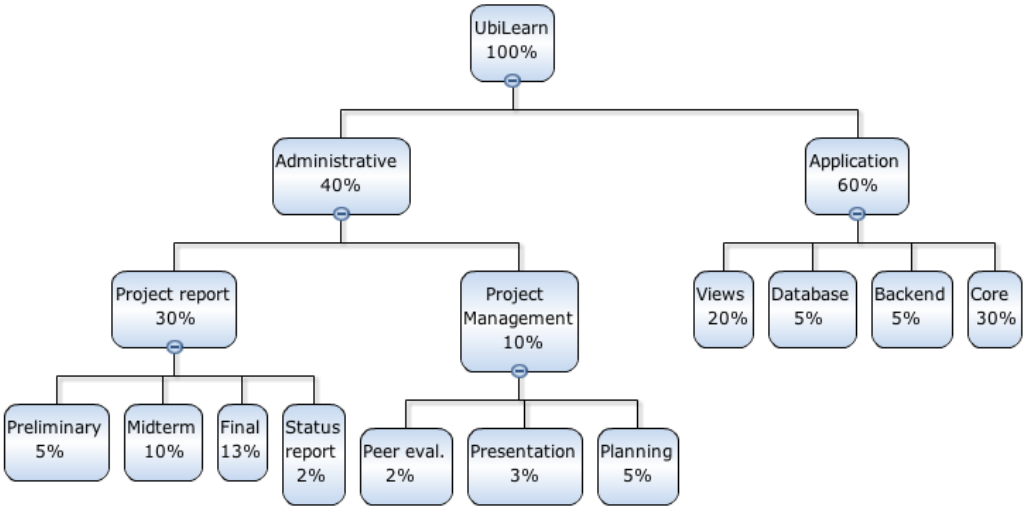
Project management is the process of planning, organizing and controlling the project resources to achieve specified goals. A good project organization is essential for a project to be successful.
This chapter introduces the project organization for the project, including the team organization and roles.

3.1 Overall project plan

To get a better overview of what phases this project is composed of, the team created a work breakdown structure, figure 3.1. The WBS outlines in percentage, how much work each part of the project requires. The WBS has been decomposed into to several layers. Where the combined percentages of the child nodes, equals the total percentage of the parent.

This was useful for allocation of resources to different tasks in the project, taking advantage of the estimation of tasks and subtasks.

Figure 3.1: Work breakdown structure



Only work that produces a deliverable is represented in the WBS. This is to simplify the diagram and letting it focus on the most important aspects of the project. This means that hours spent on meetings, research and other administrative tasks, are not represented.
Planning is represented in the WBS because of the UML diagrams and paper prototypes, categorized as deliverables.

3.1.1 Activity planning

The group chose to create their own version of the activity plan for task scheduling. The activity plan provided the team with a more detailed view of the work progress for each task. This allowed members to keep oneself posted in a better way during the sprint, and gave more control of the time spent.
At each sprint meeting the team picked certain activities that would be worked with for the next sprint, and agreed who should have the main responsibility for getting each activity done, although all team members could help if it was needed. This became a necessity to make sure tasks were completed with the expected quality.
An example of the activity plan is provided as a part of the status report in Appendix C, under paragraph 4 as a scheduled work period, and under paragraph 6, completed with the spent working hours.

3.1.2 Weekly schedule

The group decided to set Monday as a regular working day for collaboration. Each Monday started at 9.15 with a group meeting. The purpose of the meeting was to update each other on the work done and continue with planning of the next sprint. After completing the meeting, the team started working on the part of the sprint that was best suited for teamwork, and the work session usually lasted for 8-9 hours.
In addition, a a short meeting was held on Thursdays at 12.15. The rest of the work on each sprint was done in smaller groups or by individuals through the week.
The plan was to have weekly meetings with the customer on Fridays at 14.15, as this was requested by him, but this was not always possible from the customers side.
Supervisor meetings were set to every other week and agreed upon a few days in advance.

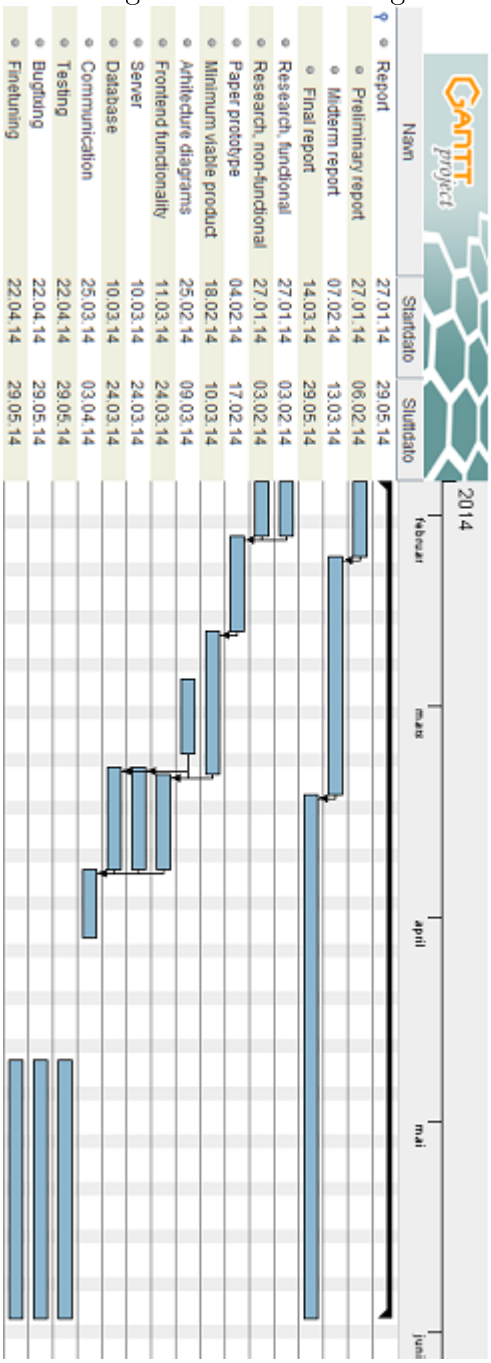
3.1.3 Milestones

Table 3.1: Project milestones overview

Planned finish date	Actual finish date	Milestone	Delivery
09.02.14	09.02.14	Project report - preliminary version	Preliminary report
14.02.14	10.02.14	Finish a prototype of the system	Paper prototype
10.03.14	16.03.14	Minimum viable product	Application
16.03.14	16.03.14	Project report - mid-semester version	Midterm report
19.03.14	19.03.14	Project presentation	Presentation
23.03.14	23.03.14	Peer evaluation	Evaluation form
11.05.14		UbiLearn	Final application
13.05.14		Acceptance test	Test result
15.05.14		Project report - final version	Final report

Figure 3.2 presents a Gantt diagram for this project. The diagram illustrates the start- and finish dates for each milestone, and provides an overall summary of the project presented as milestones.

Figure 3.2: Gantt diagram



3.2 Team organization

Using an agile development process, the team decided to part the project in smaller subtasks. The group has during this project experienced the importance of allocating responsibility for different development tasks between the group members to get the best possible cooperation and work structure.
With a segmented approach, each team member was assigned with a main responsibility for a certain subtask. This was done to ensure that the part would be due on time, and have the expected quality.
The responsible person of one part was still included in other subtasks. This approach gave flexibility to our work, and provided all group members with a better overview of the project.

3.2.1 Experience and knowledge

Because all the group members are in the third year of computer science at NTNU, the team have pretty similar backgrounds expertly. Although some group members may have more to contribute in one area, there were other group members who completed roles in other areas.
When tasks for a sprint were distributed, the team focused on
exploiting persons interests, experience and existing knowledge. In terms of assigning tasks, the team found it considerably wiser to have the person with most experience in an area, take responsibility of getting it done, but at the same time, let other group members who wanted to learn more contribute.

3.2.2 Responsibilities

To make the team structure work, the project was parted in different
disciplines and jointly assigned a responsible person for each part. The person responsible was not necessarily the only one working on this part, but was responsible to make sure all parts would be done. In addition, this person would have to devote more time on getting familiar with this discipline. It was found to be most efficient for the team to divide the project this way.

Table 3.2 define the partitioning of the project, including what the responsibility entailed and which team member was responsible.

Table 3.2: Team role description and responsibility

Part	**Descripiton**	**Responsible**
SCRUM master	The SCRUM master makes sure that the team works efficiently and gets the work done for each sprint. The SCRUM master is also responsible for leading the group meetings, and have to ensure there are no internal conflicts or problems in the group.	Espen Hellerud
Design	The design responsible is in charge of making sure the design and user interface is created as planned with the customer. This team member have to ensure that the user interface is understandable, user friendly and consistent	Ingeborg Ødegård Oftedal
Testing	The test responsible must verify that all modules of the system work as intended. This is done by designing usability tests, software tests and also executing these during the product development.	Kenneth Pettersen Lund
Back-end	This team member is responsible for making sure the server will handle all needed requests towards clients.	Kyrre Laugerud Moe
Contact person	Responsible for arranging meetings with supervisor and customer, also handles communication with external subjects.	Ragnhild Jølstad Seim
Report	Making sure the report covers all required points and that the experiences and issues of the group are well documented. In addition to quality assure that the report maintain high quality, both by content and design, before it is delivered.	Ragnhild Jølstad Seim
Front-end	The responsibility is to make sure that the application includes all modules from the paper prototype. In addition ensure that the application has fulfilled the functional requirements.	Ingeborg Ødegård Oftedal
Database	Responsible for making the database store all necessary models, and that the server has the right SQL code to retrieve insert and update need data.	Espen Hellerud

System communication	Ensure that the communication between client-server and server-database works and making sure the proper technology is used.	Kyrre Laugerud Moe
Third-party services	Ensure that the proper API is utilized when using third-party software and that the third-party service used, is optimal for the project needs.	Vegar Lerpoll

3.3 Process guidelines

Chapter 2.1, justifies the grounds for choosing a customized agile process model based on SCRUM. This section will explain the process model guidelines that has been followed during the project.

SCRUM provided the team with a good framework to follow for the process to be efficient, well monitored and evaluated. The team decided to have each sprint duration to be seven days. This was found to be an appropriate duration, both because it was a good match for the project size, the sprints could be parted in suitably small sub-tasks, and customer feedback was a part of each frequent sprint deliverable.

Sprint structure

The sprint iterations made sure everyone was up to date on what was being done and allowed the team members to work efficiently on their own. The team expected each member to list the number of hour spent on a task in the activity sheet, and also mark the related issues with an appropriate status on GitHub along the way.
To keep track of who did what, every person had a subtask to focus on, but also the responsibility to spend enough hours each week on the project. The team estimated approximately 20 hours per person each week to be an appropriate workload.

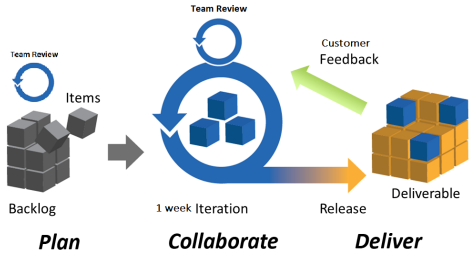


Figure 3.3: Illustration of the agile process. Source: Own work

Plan

A sprint started by a planning meeting each Monday, where tasks for the sprint were chosen and working hours estimated jointly by evaluating the scope of a task.

The tasks for the sprint were picked from the product backlog that the customer contributed to prioritize. The backlog was divided into milestones and recorded in github. Each milestone was again divided into smaller sub-tasks, called issues on GitHub. When a new sprint was planned, the related issues or tasks needed to get done, was added to GitHub. Whenever a team member worked on an issue, it was labeled based on what had to be done, whether it was enhancement, feature, bug etc.

Collaborate

When a sprint was planned the group started working with the chosen tasks. Mondays were the main collaborative working day, and the team also met on Thursdays for a short update and worked together after that.

The team review was based on the customer feedback on the deliverables for a sprint. This was done after customer meetings that usually were held on Fridays.
The purpose of the review was to evaluate the work done so far, and make any necessary changes before a sprint ended.

Deliver

At the end of a sprint, the team went through the deliverables with the customer, focusing on the changes made since last iteration.

The customer feedback contributed with making changes to the deliverables, but also provided the team with new aspects on the backlog prioritization. The team reviewed the feedback in two processes, both during the iteration, and as a part of the sprint planning. This made the progress of the project customer-driven based on the received feedback.

In addition to customer feedback in regards of the product, the project had comprehensive requirements for documentation in form of a detailed report, and by such, the supervisor has been an important resource for documentation feedback. We planned to have meetings with the supervisor every other week, or by email when needed.

Sprint completion

The sprint was ended by a sprint review-and-retrospective meeting. This was done on Mondays before the planning process started.

The completion meeting was used to review the progress for the previous sprint, and identify the lessons for the next sprint.

Management rules

To make sure the group worked together through the project, included all team members and followed the time schedule, the team decided some simple rules within the group.

- All working hours were expected to be updated within the end of the day.
- Communication outside the group meetings was done using Facebook. This way the team could make sure everyone were updated and included. Also, if one team member could not attend a meeting, or got sick, this had to be reported to the rest of the group as soon as possible.
- Working together as a group, it was expected that everyone performed their share of the load. This included helping those in the group who found one part difficult, make sure oneself worked enough hours based on the weekly planned workload, and keep focused on creating a good final result.

Chapter 4

Requirements

This chapter has the purpose of giving an understanding of the product requirements, both functional and nonfunctional. The requirements represent the purpose and behavior of the product.
The functional requirements specify particular features of the systems functions. Describing the behavior relating to the systems functionality.
The nonfunctional requirements elaborate a performance characteristic of the system.
Because the customer focused on research, we had to take into consideration that there would be changes made to the system requirements during the project process.

4.1 System functionality

The product will be an Android application, targeted for non-ICT users, with focus on those without university education. As a result, the customer wants the product to be very user-friendly, and provide the user with an e-learning process. The system comprises a combination of practical and theoretical learning methods.

4.2 Requirements process

This section will provide insight related to the process of specifying the system requirements.

4.2.1 Research study

The customer did not have any specific requirements for the system at the start of this project.
In order to get an overview on what was required of the final product, the team spent time on a research study. The result of this study was presented for the customer to get feedback on the findings.
After the research study, the customer communicated that the most important aspects of the product would be research, focusing on integrating practical and theoretical e-learning.

4.2.2 Requirements specification

When the group had a clearer view on what the final product should contain, the customer provided a written requirements list for both functional- and non-functional requirements. The group analyzed the requirements during a workshop, and made changes where the team felt it was necessary.
The analyzation and elicitation were conducted, taking into account the short duration time for the project. After the workshop, the updated requirements list with prioritized order of the requirements, was sent to the customer for feedback.
The prioritization of the requirements was done jointly, based on the teams opinion of what was most important to implement early. By also taking into account the customer-validation of the list, the team made sure that both the customers and the teams opinion on the requirements for the product were on the same page.
With the customer approval, the importance of the requirements is listed in table 4.1 and 4.2. When a requirement is categorized with importance = 1, it is of less
importance, and is not necessary to implement to get the system to work. Although the system will lack functionality without it.

The functional requirements are related to use cases to show the functionality from a user perspective.

4.3 Use cases

An overview of the functionalities of the system from a user perspective, is presented as use cases.
To access the system, the user needs a mobile device where the application can be downloaded. Each activity is represented as a use case, describing what the user can do.
The team did not find it necessary to set an "actor field"for each textual use case, as the user does not differ from a user with a mobile device.
The system only operates with one user, because the customer have not specified that there should be any administrator for this edition of the application.
For a user to be able to access the application, it has to be downloaded and installed from Google Play in advance.

Figure 4.1: Use case diagram

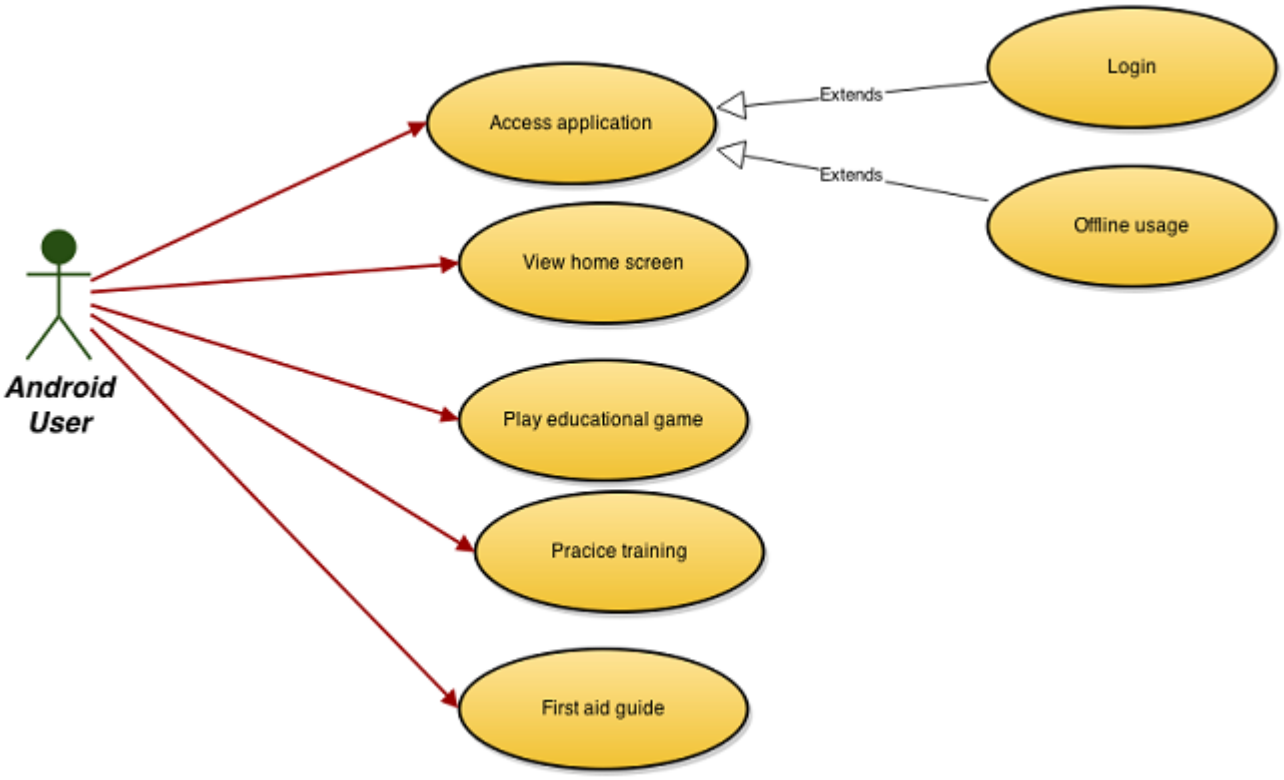


Table 4.1: Use case 1.1: Access application

Use case: 1.1	**Offline access**
Description	A user should be able to access the application without being a registered user
Pre-condition	Application is installed
Post-condition	The application supports offline individual usage
Normal event flow	The application is accessed
Variations: Possible variations to the normal event flow	The user ends the application before it is accessed

Table 4.2: Use case 1.2: Access application

Use case: 1.2	**Access application as a registered user**
Description	The user should be able to register as a user, and login to get the benefits a registered user has
Pre-condition	Application is installed
Post-condition	The application supports online collaborative usage
Normal event flow	The user is registered as a new user and logged in with username and password
Variations: Possible variations to the normal event flow	1.2 The user is not registered 1.2 Wrong username or password

Table 4.3: Use case 2: View user status

Use case: 2	**View user status**
Description	A registered user can access an overview of the users profile and progress for saving and reviewing. This view will also function as the home screen.
Pre-condition	Application is installed, the user is registered and logged in
Post-condition	The user can view the home screen with updated user progress
Normal event flow	User can access the home screen
Variations: Possible variations to the normal event flow	If the user is not logged in before accessing the application, return to use case: 1.2

Table 4.4: Use case 3: Play educational game

Use case: 3	**Play educational game**
Description	The user is able to access different houses, where each house represents a patient house, with a case description. This case works as a task to be completed
Pre-condition	The user should be registered and logged in, and the educational game feature accessed
Post-condition	The user will be given a score which will indicate how well the case was solved
Normal event flow	3.1 Push a house 3.2 Choose enter 3.2.1 If enter is pressed, a case will be presented
Variations: Possible variations to the normal event flow	The user pushes cancel and returns to the level screen

Table 4.5: Use case 4: Practice training

Use case: 4	**Practice training**
Description	The user can perform practical exercises and training programs with a patient. The practical training programs is provided by the application
Pre-condition	Application installed and practice training accessed
Post-condition	The user is provided with basic programs to be completed with the patient
Normal event flow	4.1 The user access the practical application activity 4.2 The user is provided with exercise templates 4.3 The program is conducted and the user can score the session
Variations: Possible variations to the normal event flow	The user pushes the "back"button and returns to the previous screen

Table 4.6: Use case 5: First aid guide

Use case: 5	**First aid guide**
Description	The user should be able to access a first aid guide
Pre-condition	Application installed and first aid guide accessed
Post-condition	In case of emergency, the user can now quickly access instructions of first aid
Normal event flow	5.1 . Click on "first aid" 5.2 A selection of first aid will then appear
Variations: Possible variations to the normal event flow	The user pushes the "back"button and returns to the previous screen

4.4 Non-functional requirements

Table 4.7: Non functional requirements

ID	Title	Description	Importance (1-3)
01	Usability	The mobile application should be intuitive and easy to understand. It should also provide an user manual.	3
02	Programming language	The system will be implemented as an mobile application on the android platform, using Java. Java will also be used to implement backend.	3
03	Backend simplicity	Backend code should be kept as small and simple as possible. This is an effort to keep maintenance at a minimum.	2
04	Maintainability	As this product is meant as a proof of concept, it should be easy to maintain.	1
05	Development time	The system should be implemented in one semester.	3
06	Engagement	The system should be both inspirational and entertaining, so that it motivates the users to continuous learning and practising.	2
07	Performance	The system should be responsive. When the user make different selections it should not take several seconds before for example a page is loaded. The smartphone requirements should not be too high.	1
08	Accessibility	The application should be available on Google Play (Google's app store). It needs to be accessible to both homecare personnel (with login) and regular users (without login).	1
09	Licensing and copyright	This project should be open source, and licensed under the Apache License 2.0.	3

10	Documentation	The source code should be documented using the javadoc format.	2
11	Offline	The application should provide basic functionality also when the user is not connected to the internet.	3
12	GUI	The application should follow Android GUI design guidelines corresponding to the provided information from http://developer.android.com/design.	2
13	Android OS	The application should be developed for smart phones running Android OS.	3
14	User database	The application should store user data in an online database.	2

4.5 Functional requirements

Table 4.8: Functional requirements

ID	Title	Description	Use Case source	Importance (1-3)
01	User profile	The application should support setting up a learning program with learning goals. It should also include an overview of the users profile and progress for saving and reviewing. This will also function as the home screen.	2	3
02	Training game	The application should include an educational game, which will use patient cases to simulate possible situations.	3	3
03	Save state	The application should support saving of learning programs as templates.	4	1
04	First aid	The application should include a first aid part which users could use if they end up in different accidental situations.	5	1
05	Handbook	The application should include a handbook which should contain articles relevant to the training game and the domain.		2
06	Expert help	A number to an expert should be available in the application		1
07	Menu	The menu should contain all the main activities and be available at all times.	1-5	3
08	Practice	The application should support homework type of training and practice based (field) training, like practical exercises.	4	3
09	Communication	The application should have basic support for communication with classmates (social computing) and support for viewing and sharing of progress reports.	2	2

10	Offline usage	The application should support offline individual usage and online collaborative usage. This means a student should be able to download and use the application without the need for registering with a service. but that registration with a service will provide additional benefits such as social networking and feedback from an expert.	1	2
11	Domain generalized	The application should have full support for different types of domain. It should be able to store and show content from different domains in a generalized manner.		2

Chapter 5

Architecture

This chapter describes the systems architecture, giving an overview of the logical architecture in addition to the physical deployment.

5.1 Overview

The system was originally designed using a client-server architecture, as shown in figure 5.1.

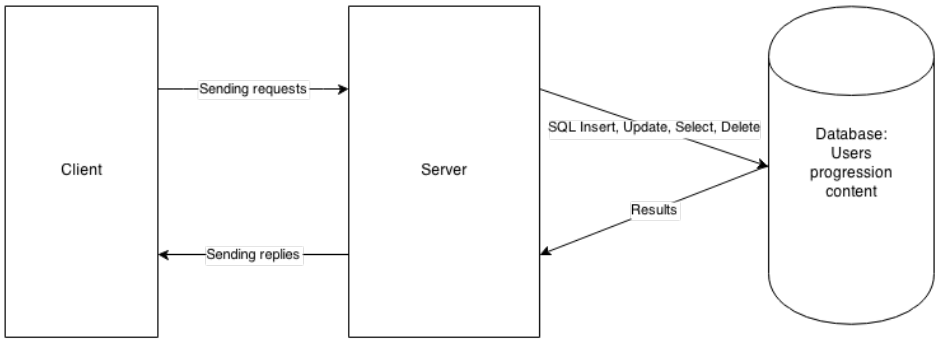


Figure 5.1: Original architecture

In the original architecture a server handled communication between the database and the client. The system did however turn out to be mainly frontend centered, which means that most of the system logic is coded into the client, an android application.
The server only handles data-requests from the client. As this system

functions as a prototype, and by such require very little backend logic, it made sense to abstract and simplify the backend part as much as possible.

After completing a backend-analysis, and presenting this to the customer, it was jointly decided that utilizing Backend-as-a-service would be of great benefit for this project.
To minimize the amount of requests sent to the server, and greatly improve offline functionality, the team decided to implement a SQLlite database locally. This database stores information retrieved from the server, so that only new or updated content would be pulled from the server.

In figure 5.2 the a general overview of the new architecture design is presented. Here the client is in the center, and implementation is heavily focused around it.

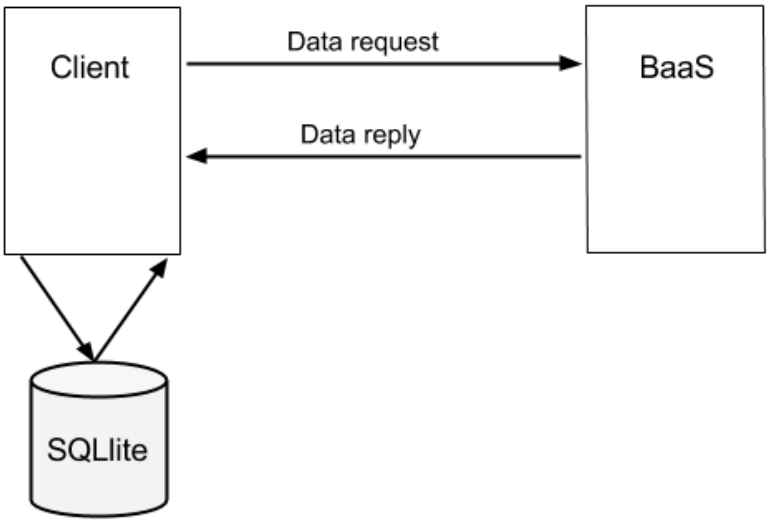


Figure 5.2: Improved architecture

5.1.1 Client

The client side of the system consists of a standalone Android application. The first time the application is launched, it will need to download content from the BaaS, i.e. Parse. After this initial download, only new or updated content will have to be downloaded. This is possible because date that has already been downloaded, is stored locally in a SQLlite database.

The client focuses around a main activity, which controls most of the other views. The first activity the user sees, is the login activity. The login activity will lead the user to the main activity view. The main activity refers to all the other views by using a navigation drawer. All the other views are fragments, a simplified version of an activity. Androids fragment class is very well suited for a structure where you have one main view, which leads to all others.
A visual representation of this implementation can be seen in figure 5.3.

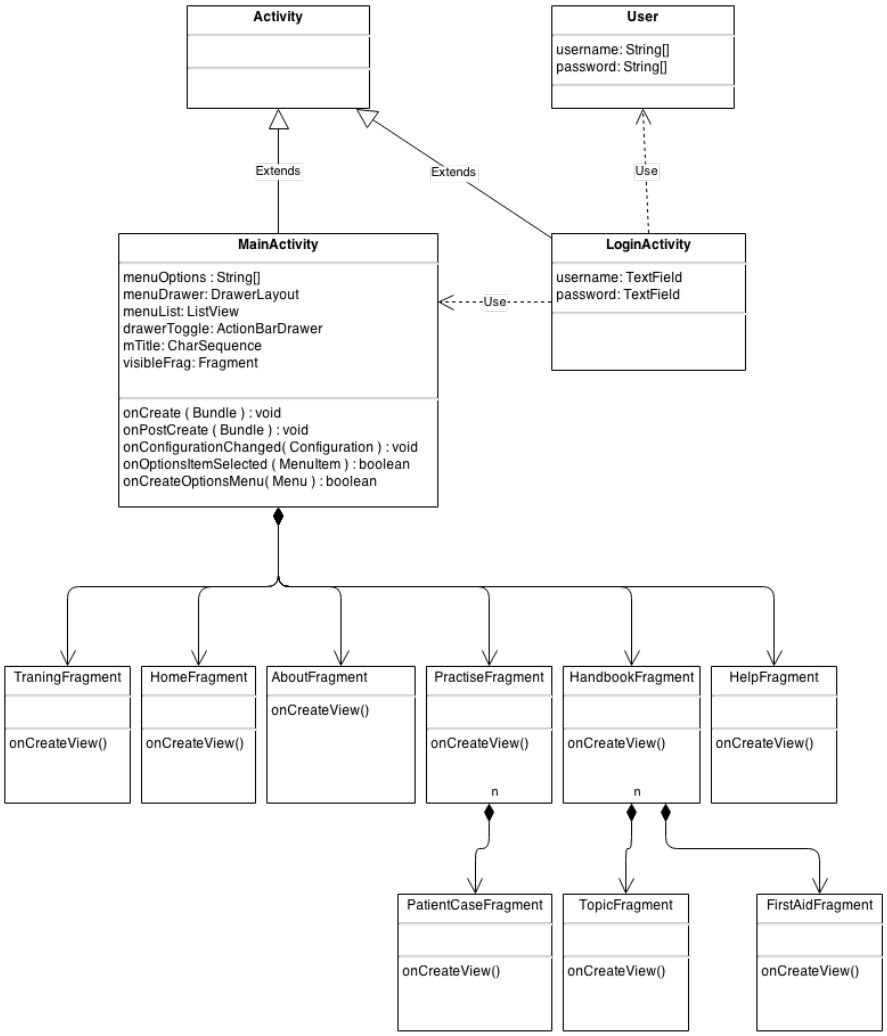


Figure 5.3: Original class diagram

The figure shows a simple class diagram containing the most important classes, and how the developing team thought of implementing these at the start of the project. An accurate presentation of the finished client will be shown in section ??

5.1.2 Local database

The database stores needed data, based on the data model which is presented in section 5.1.4. The BaaS serves as the central database, keeping an up-to-date version of the applications new content. As shown in figure 5.2 each client has its own database. Every time the user logs in successfully, the local database is synchronized with the BaaS.

The client database use SQLite, as this is recommended from Androids development site. This has been implemented and designed by the team.

5.1.3 Backend

As mentioned previously, Backend-as-a-service was used for the backend, where Parse.com functioned as the host for providing this service. Parse allows developers and administrators to use a simple web interface to create tables and insert data. It provides a range of APIs to transfer data between the server and the client, inlcuding: REST, Android API, IOS API and javascript API. This solution requires more code on the client side, but in return there is no need to develop the server and a central database. Therefor, the backend is a simple web-interface, providing access to everything needed for development and administrating. This means that it embeds all the normal parts of a backend system, into one system.

Parse's Android API was used for communication between the client and backend. This was done using a Parse library designed for Android. This library provides well documented methods for sending and receiving data from the client, as well as keeping a logging session.

Data storing in Parse is very similar to how you would store data in a SQL database. This means that the data models is represented much in the same way, and with the same relations as the SQLlite database in the client.

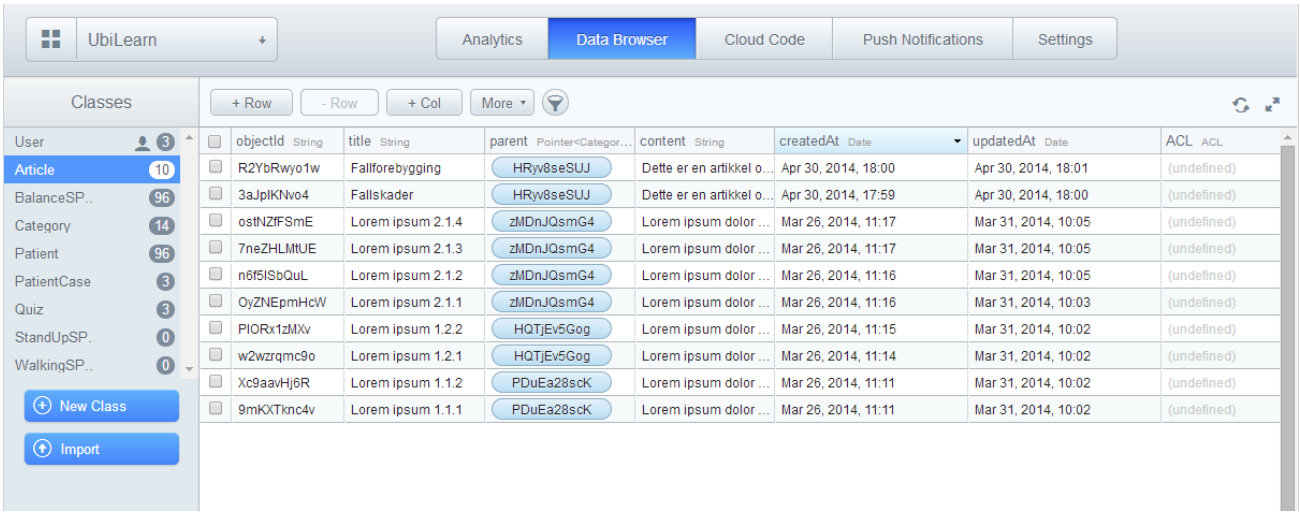


Figure 5.4: Parse web interface for data browsing

5.2 Data model

The data model represents the data that is needed for the system modules. Because data can be stored both locally and on the backend database, the data modules are presented in their own sections. These sections includes: backend data, client data and common data.
Common data, describes data that is stored both on the client and in Parse, as data is created by an administrator and stored on Parse. Each client will store a copy of this data locally. This is done to save API calls and time, and to have access to data without internet connection.

The following sections presents all the data needed, separated into classes. Each class represents an entity with attributes. The attributes holds the data, while the entity holds relations to other entities, and contains the attributes. The entities can then effortlessly be translated to Java classes.

5.2.1 Backend data

Figure 5.5 shows user data, which will only be stored on the server and brought down to the clients session, when the user logs in. This means that to obtain user data, the user needs to establish internet connection, and then log in.

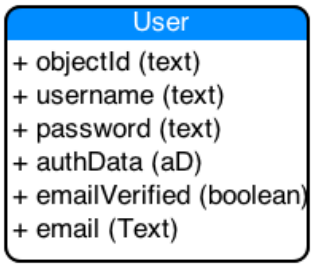


Figure 5.5: User data

5.2.2 Client data

Figure 5.6 shows client data, which contains information on different patients. The model contains SPPB tests, that are performed on a patient, allowing the user to record the patients performance in the application.

There is a one-to-many relation between Patient and SPPBTest. This shows that one patient can have several tests. There are three different tests. The model shows this by having one generalized class and three other specialized classes that extends the first. This data is only need for the user and is private. Because of this there is no need to upload this data to the backend. In future releases of the application, there should be a possibility to store this data on the server. This will allow the user to retrieve patients and test data from other devices, and serves as a backup.

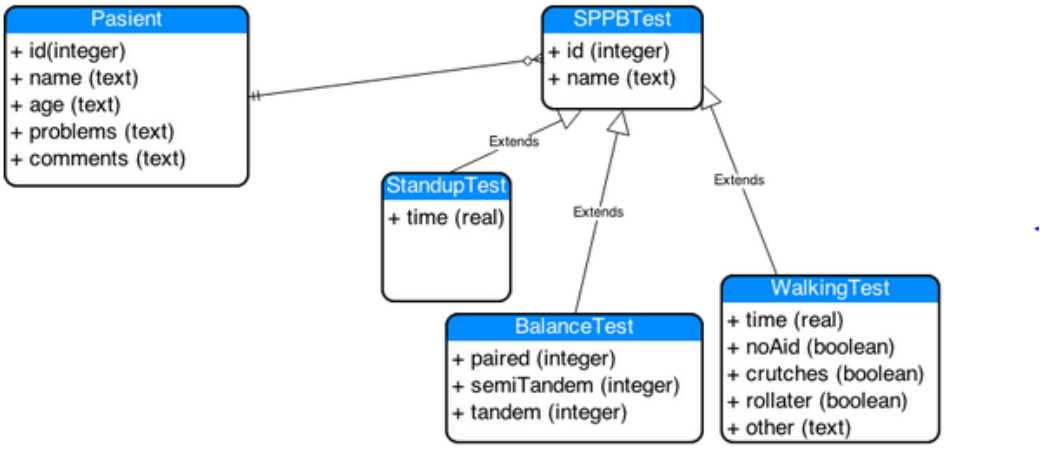


Figure 5.6: Client data

5.2.3 Common data

Figure 5.7 shows data that is stored both backend and on the client, where the leftmost classes cover the training part of the application.

As explained in chapter 1.4.3, the training part of the application, presents different CasePatients for the user. This is presented in figure 5.7 as a one-to-many relation from CasePatient, to Quiz and Interview.
The app also contains a handbook, where the user can read useful articles. The articles are sorted under different categories, where a category can contain several articles. This is presented by having a one-to-many relation between article and category. A category can also be sorted under a different category, presented in the figure as a relation from Category to itself.

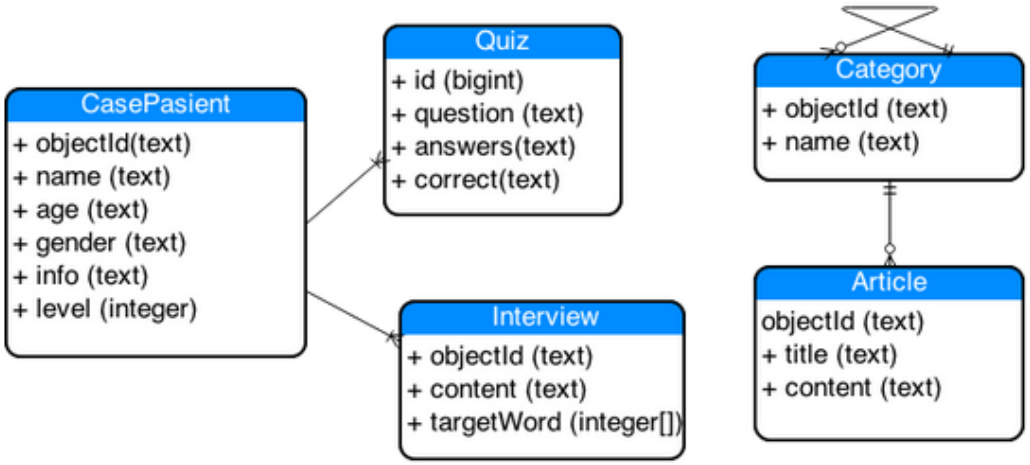


Figure 5.7: Common data

5.3 Physical deployment

5.3.1 Client

Ultimately the Android application should be available on Google Play, Androids standard market place for downloading applications. When the application is uploaded to Google Play all users who satisfy the applications requirement, i.e. Android version, will be able to download and install the application. Currently the application is only available as an Android APK file in a Dropbox folder owned by SINTEF UbiCollab. In order to install the application using the APK file hosted on Dropbox, the user must first allow installation from unknown sources. This setting can be changed via the security tab in the settings menu. Once this option has been checked the application can be installed by downloading and running the APK file.

5.3.2 Backend

As the backend of the system is only using Parse, Parse will take care of the deployment of the entire backend. Developers and administrators will only need to maintain an active user to Parse.com, including keeping a working API key, and access to the tables.

5.4 Architectural justification

Android application

Development of an Android application was the customers request for this project, but it was also in the team's best interest. Every member of the team has at least intermediate skills in Java development, and two or more years of experience. Since Java is the native language for Android, this suited the team perfectly.

5.4.1 RESTful communication

For communication with the backend, there are two main possibilities. Either a constant connection using Java sockets, or by having an asynchronous communication using HTTP.

Since the client is deployed on a mobile device, requiring a constant connection will quickly become challenging. When constantly loosing the connection a lot of bandwidth and waiting time will be used to maintain the communication.

When using a REST API, the android application will have a HTTP interface towards the server with POST and GET request, using JSON to format the data. This means no connections need to be either established or maintained. This is highly dynamic and separates the concerns of the server from the client.
With a REST API, the clients communication towards the backend was quite easily developed, by just requiring JSON objects to be sent from the server. JSON as data formatting also makes the API very lightweight and simple to understand, which is a great advantage for a smaller system, like this.

5.4.2 Backend as a service

The team needed a way to communicate between the server and the client, but it felt that a persistent connection was unnecessary. Therefore, RESTful communication became the natural choice.

The developer team felt that a custom server was needed in order to satisfy the teams expectations to what the server was able to do. There were however some disagreement and confusion between the developers and the customer, as to what would be the role of the backend system. The customer

wanted the backend to be very lightweight and utilize existing frameworks for data storage and retrieval.

The developing team felt that a framework that suited the project did not exist. To resolve the problem of finding a suitable backend solution, it was decided that we would conduct a backend analysis. One of the requirements of this backend analysis was that the proposed solution would deliver a RESTful service. To make sure the best backend solution was implemented, a thorough analysis of possible alternatives was done. Since a requirement for the server was to deliver a RESTful service, the client could be developed without knowing the outcome of the server analysis. The analysis discussed options like; Moodle, Django server and Java server, to be used as basis for the backend. The developer team even tried using a REST library for Android, called Dropwizard.

The backend analysis did however reveal that these solutions either were too time consuming to implement, or did not offer the flexibility required. The backend analysis did however explore the possibility of using Backend as a service (BaaS) providers. BaaS turned out to be more or less what the entire team and the customer was looking for.
As explained in 5.1.3, BaaS provides a simple yet flexible backend solution with pre built API's and libraries, for communication. This meant that the backend system would be very easy to maintain, and keep the customizability needed for the project.

When it came to choosing BaaS providers the team had very little experience to go on. Research revealed Parse to be a good option. Parse is considered to be one of the leading providers in its market, i.e. Backend-as-a-Service. Parse is known to be one of the simplest and easiest BaaS providers, both to learn and to use. This was in the interest of the team, as well as the customer, since using as little time backend as possible was preferable.

5.4.3 SQLite as local storage

Since the data would benefit from a structured storage with class, attributes and relations, using a relational database were found to be the optimal solution. Android supports a SQLite DBMS. It has a API for sending queries and returning the results. This means that by using SQL syntax, it is possible to create, use and manage a local database on the device. It is a bit more complicated than just saving data as key value pairs, but the structure and usability makes it well worth it.

5.5 Implementation

5.5.1 Views

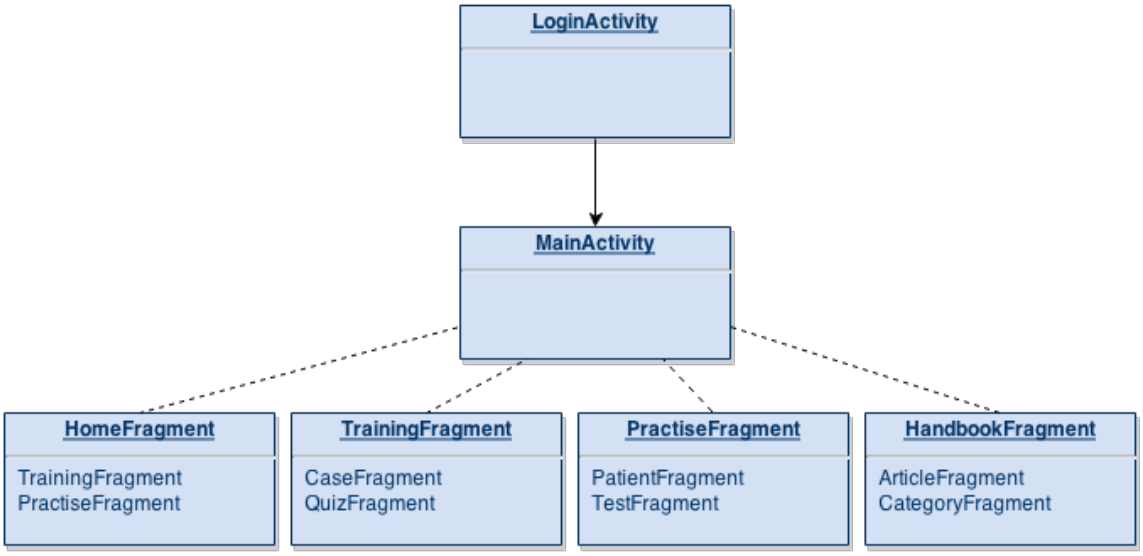


Figure 5.8: Overview of the main screens

The first screen the user will reach is LoginActivity. If the user manages to login in successfully, the user will be taken to MainActivity. MainActivity does not have its own screen but instead use Fragments. A Fragment is a lightweight version of an Activity, and is better to use for hierarchical structure under an Activity.

A screen is mainly separated into two files; a fragment which acts as the controller, and contains all the logic, and the view, which is a XML file. A view defines the objects that can be seen on the screen, and the objects in the XML file are manipulated by the controller. When an Activity or an Fragment is loading, it will inflate a view to cover the screen.

Login

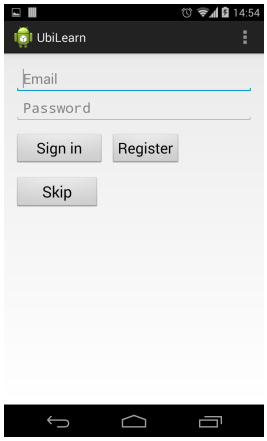


Figure 5.9: Login screen

Home

Figure 5.10: Home screen

Training

This view is controlled by the training fragment, and is the main training screen. The fragment controls which patients belong to which house, and is also responsible for switching images, indicating which level the user is currently in. The view receives information from User.java(singleton connected to Parse), so it can set the right level and make sure the user has access to the correct level. When clicking a house, you get access to CaseFragment, and a patient object is sent from training to CaseFragment.

CaseFragment with its views, shows the case information about a patient. This view also shows the users progression in a patient case, if any. This information is retrieved from User.java. This fragment is connected to a model, CasePatient model, which again the database has a connection with. When clicking the quiz button, you get access to the QuizFragment, and a patient object is sent from CaseFragment to QuizFragment.

QuizFragment controls all the quiz views. It gets all the quizzes in one case from the database, and is connected to a Quiz model. The Quiz model sets the quizzes, returns the correct answer and return a random list of the answer alternatives. When a user has the right answer, the User.java is updated, and sends it to Parse, so that the progression is being saved.

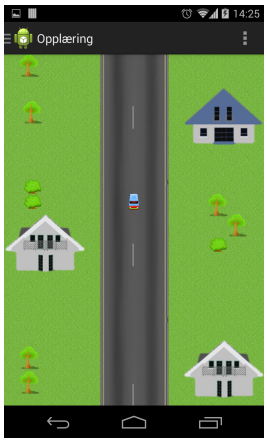


Figure 5.11: Training screen

Practice

The practice fragment view, presents the user with a list of the patient this user has responsibility for. When a user add a new patient, this is stored in the database. When you click a patient, you get access to a new view, which shows patient information, different tests, exercises and tests results. All of this information is retrieved from, and stored in the database.

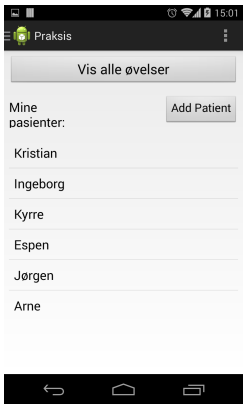


Figure 5.12: Practice screen

Handbook

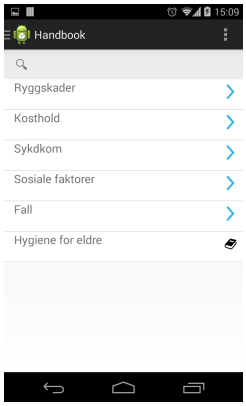


Figure 5.13: Handbook screen

5.6 Storage

5.6.1 SQLite interface

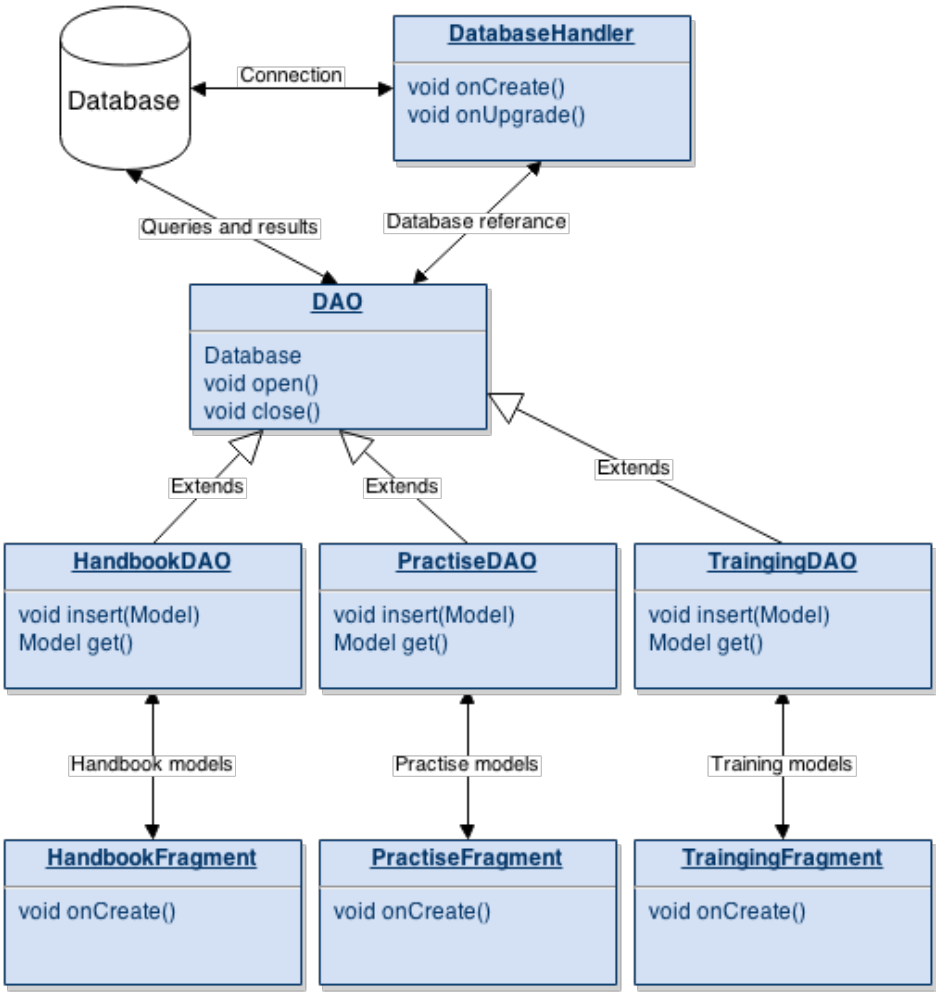


Figure 5.14: DB class overview

The custom made interface used by the Fragments to insert data into the database, have three levels. At the lowest level there is an Database handler. This class creates the database and all the tables and attributes. It also opens and closes the connection to the database. This class is used by DAO(Data Access Object) to communicate with the database. DAO is an abstract class,

so it will never be initiated, instead there are several subclasses with methods that are specifically made for their Fragments. Fragments can then use these methods to insert and retrieve data by using a model. A model is an entity which with properties that exist both as attributes in a table and as a java class.

5.6.2 Parse interface

Chapter 6

Testing

This chapter describes the strategy for testing on different levels of this project, including integration testing, system testing, usability and acceptance testing.

6.1 System testing overview

The system testing is conducted with the purpose of confirming that the system meets the requirements set, with a level of quality equivalent to the customer expectations. System testing has been conducted to reveal variations in the expected output of the system.
When the testing revealed such defects, the team worked specific to correct these and improve the system quality.

For this project the objectives of the testing process was:

1. To demonstrate for the customer at SINTEF, that the system meets the requirements. This means that the testing is based on the requirements specified in chapter 4.

2. To discover situations where the software did not conform to its specifications, where the behaviour of the system was incorrect or not operating in a desirable way. When such discoveries were made, the team worked to correct these.

6.1.1 Testing methodology

The agile process model that was adapted for this project, based on iterations, allowed for continual enhancements during the process. At the end of each iteration all team members were allowed to identify problem areas, if found, and suggest improvements for this iteration deliverables.
The agile model focuses on recognizing that the development process can evolve, this was experienced by all team members. The team gained insight after every iteration, both on how to improve the productivity, but also by receiving feedback from the customer and the overall team on what improvements could be integrated during the next iteration.

System tests were performed continuously as each deliverable received performance or functionality changes. The team member responsible for the specific deliverable was in charge of running necessary tests to the module before it was merged into the main project.
If a bug or error was detected but not handled, these were added as issues to be solved in GitHub. These issues were "open" until the assignment was been solved. Then the state of the issue was marked as closed.
The high frequency of defect corrections and system improvements allowed the overall team to improve the system efficiency during the process, and experiment with different solutions if necessary. This was found to be a good approach for the team, with the goal of achieving a good end-product.

6.2 Development testing

Development testing includes all testing activities carried out by the development team. During development, testing has been carried out at different levels of granularity which will be explain more thoroughly in the next subsections.

6.2.1 Unit testing

Unit testing was used to test the smallest components of the application, before pushing the code to GitHub. Whenever new functionality was implemented, a unit test was carried out to verify this exact part of the system. The unit tests in this project has been conducted as a form of white box testing, as they required some insight and knowledge of the code.

6.2.2 Integration testing

When unit tests completed, and the units was approved as small components, integration testing was done to test that the components would merge well and run as planned.
In this application, this was tested by for example clicking a button from one page, and check that the output was according to the expected response.
If the response differed from the expected output, this was identified as a bug that was not found as part of unit testing. Such results were discussed in the iteration they were performed.

6.2.3 System testing

System testing has been conducted to test the overall functionality and integrated system. To make sure that the functional and non-functional requirements are fulfilled, each testcase have been based on these specifications. The system testing has been performed by the group. The overall testplan is provided in appendix B.

System testing was performed once a week from the middle of March, when the team had implemented enough functionality. An example of a completed test is provided in Appendix B.1.

6.3 Usability testing

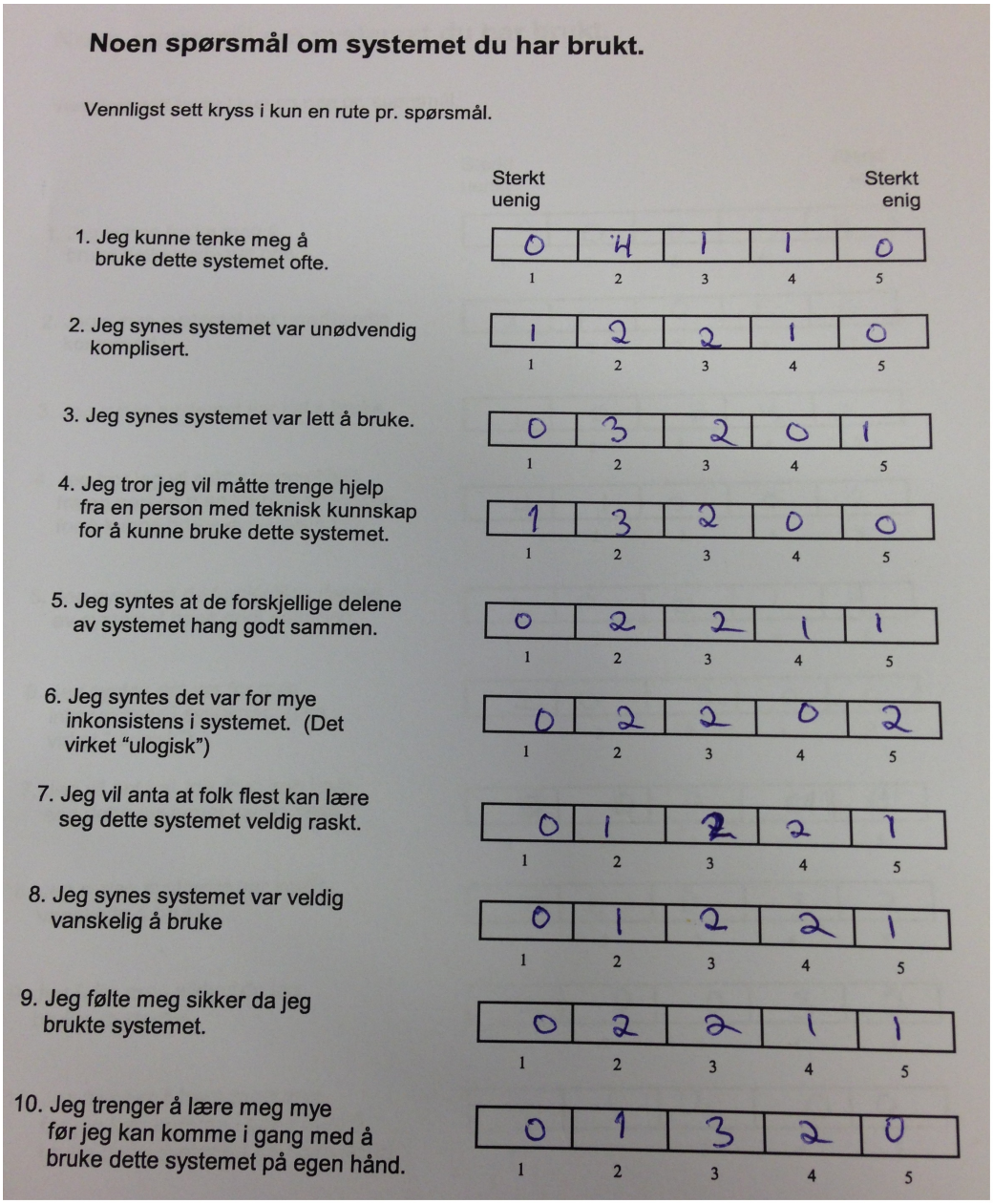
The customer has throughout the entire project specified the importance of usability. Consequently, the team has spent time on creating a user interface that is comprehensive and easy to grasp for the user, although the finished appearance of the final end product will be redesigned by the customer.

The usability testing was created by the team, who designed a case with tasks the users had to solve without any help. While the user was working on the tasks, the responsible testperson made notes about the users reactions throughout. After completing the case, the user was given a schema with related questions about the system.

Process

Usability testing of UbiLearn has been conducted in two phases. The first test was performed located at the school, with classmates as testers. The purpose of the first test was only to check the system intuitivity.
During the first test, the team was made aware of parts of the system that some test subjects struggled to understand. This revealed weaknesses in the overall application, aimed at the ease of use and system intuitivity.
In figure 6.1 is the results from the first SUS test. The numbers in each cell represents the number of participants who checked this cell.
The score has been calculated by following rules:

- **Odd items:** One is subtracted from the user response.
  Even-numbered items The user response is subtracted from 5.
  To convert the range of values from 0 to 100, the overall responses has been added up, and multiplied by 2.5.



Noen spørsmål om systemet du har brukt.

Vennligst sett kryss i kun en rute pr. spørsmål.

	Sterkt uenig				Sterkt enig
1. Jeg kunne tenke meg å bruke dette systemet ofte.	0	4	1	1	0
	1	2	3	4	5
2. Jeg synes systemet var unødvendig komplisert.	1	2	2	1	0
	1	2	3	4	5
3. Jeg synes systemet var lett å bruke.	0	3	2	0	1
	1	2	3	4	5
4. Jeg tror jeg vil måtte trenge hjelp fra en person med teknisk kunnskap for å kunne bruke dette systemet.	1	3	2	0	0
	1	2	3	4	5
5. Jeg syntes at de forskjellige delene av systemet hang godt sammen.	0	2	2	1	1
	1	2	3	4	5
6. Jeg syntes det var for mye inkonsistens i systemet. (Det virket "ulogisk")	0	2	2	0	2
	1	2	3	4	5
7. Jeg vil anta at folk flest kan lære seg dette systemet veldig raskt.	0	1	2	2	1
	1	2	3	4	5
8. Jeg synes systemet var veldig vanskelig å bruke	0	1	2	2	1
	1	2	3	4	5
9. Jeg følte meg sikker da jeg brukte systemet.	0	2	2	1	1
	1	2	3	4	5
10. Jeg trenger å lære meg mye før jeg kan komme i gang med å bruke dette systemet på egen hånd.	0	1	3	2	0
	1	2	3	4	5

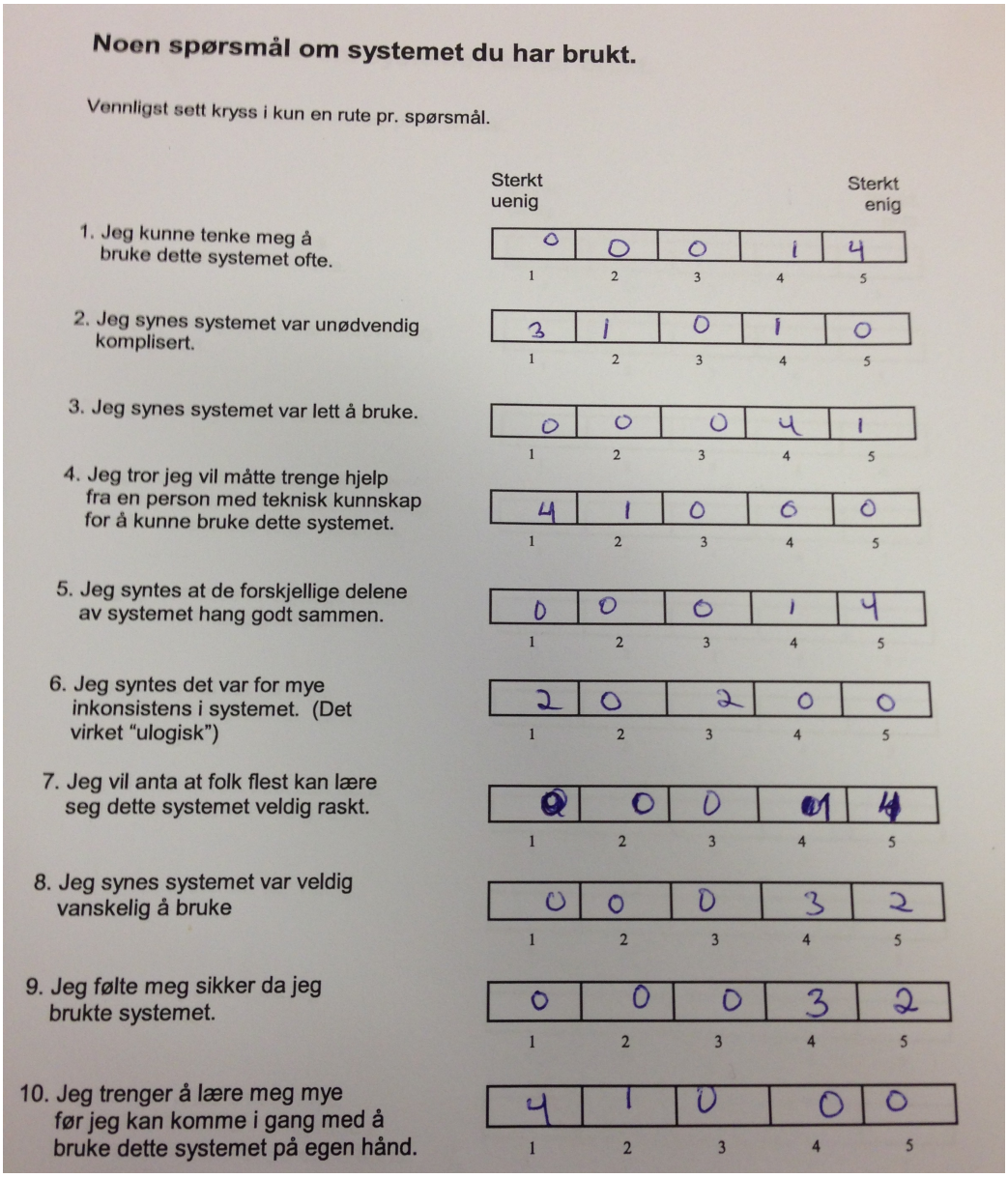
Figure 6.1: SUS score test 1

The results from test 1 gave a score of 58,7 out of 100 points. With a score above 68 as the goal, it showed that improvement could be made.

Test 2

To improve the usability of UbiLearn, more focus was spent on creating a more holistic design, make improvements to the user profile and login functionality.
These were the main parts that were corrected before a new test was conducted near the end of the project. The last test had the purpose of revealing whether the application was improved and more user-friendly. The second test was completed with new test subjects to have the same starting point as the initial test.
Test results from the second test is provided in figure 6.2.

In the second test the overall score went up 90,2, which is above the average score of 68, and a big improvement from the first test.
As a result, the team was very happy with the improvements of the product, and the test subjects response. Although there were fewer test subjects in the second test, the team still sees the results as valid.
The testresult ensured the team that the application most likely would measure up to the customer's requirements.



Noen spørsmål om systemet du har brukt.

Vennligst sett kryss i kun en rute pr. spørsmål.

	Sterkt uenig 1	2	3	4	Sterkt enig 5
1. Jeg kunne tenke meg å bruke dette systemet ofte.	0	0	0	1	4
2. Jeg synes systemet var unødvendig komplisert.	3	1	0	1	0
3. Jeg synes systemet var lett å bruke.	0	0	0	4	1
4. Jeg tror jeg vil måtte trenge hjelp fra en person med teknisk kunnskap for å kunne bruke dette systemet.	4	1	0	0	0
5. Jeg syntes at de forskjellige delene av systemet hang godt sammen.	0	0	0	1	4
6. Jeg syntes det var for mye inkonsistens i systemet. (Det virket "ulogisk")	2	0	2	0	0
7. Jeg vil anta at folk flest kan lære seg dette systemet veldig raskt.	0	0	0	1	4
8. Jeg synes systemet var veldig vanskelig å bruke	0	0	0	3	2
9. Jeg følte meg sikker da jeg brukte systemet.	0	0	0	3	2
10. Jeg trenger å lære meg mye før jeg kan komme i gang med å bruke dette systemet på egen hånd.	4	1	0	0	0

Figure 6.2: SUS score test 2

6.4 Acceptance testing

(To be completed)

Chapter 7

Development process

This chapter addresses the development process of the project, from start to finish. The team explains more thoroughly how difficulties have been resolved, and presents challenges faced during the project process.
As explained in chapter 3, project management, the project process was based on weekly meetings with the customer, in addition to the SCRUM team's sprint meetings. The team also met with the supervisor approximately every other week for feedback.
The purpose of this chapter is to give an overview of the entire process, reflecting on each completed sprint, summarizing the work done and number of hours used.

7.1 Startup

An introductory meeting for the team, was established and planned through a common Facebook group. During the first meeting, the team went through some key points to get to know each other better, and gain understanding about what expectations everyone had to the project.
Other key points reviewed, were the team's ambitions for the project, planning of meeting with the supervisor and the customer, responsibility areas, and a quick assessment of development tools was done.
The team decided on meeting times by taking into account all the team members other subjects and timetable.

7.2 Sprint 1

Summary

The starting date of the first sprint was set to the 3rd of February, even though the project process was started earlier with the introductory meetings with the customer and supervisor. The introductory process was not considered as a sprint, because of uncertainties in regards of management decisions at this stage.

In addition, the team was lacking detailed information about what the project would include, and found it challenging to decide on a process model before this was established.
During the first customer meeting the customer stated that it was not decided any specific requirements or specifications for the application. He recommended that an agile process model would be a good choice for the team to be prepared for changes in the project along the way.
The first sprint was used to do relevant prestudy and research on the domain. The team also worked on creating a preliminary draft for the project report.

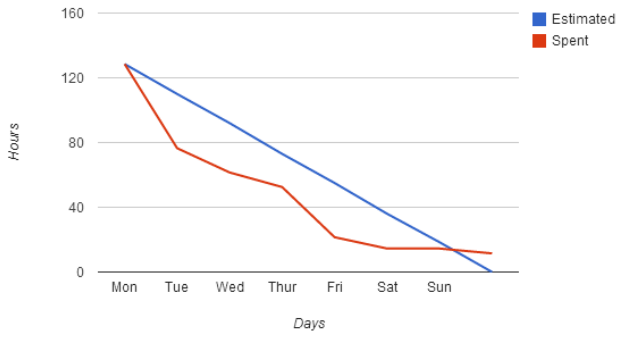


Figure 7.1: Burndown chart: Sprint 1

Review

The team found it challenging to know where to start the process, when the task was first awarded. Doing research and analysis without really knowing what to look for felt a bit useless, but later in the project, the team found the prestudy to be quite useful after all.
By having the whole group perform brainstorming on what everyone envisaged for the applications contents, gave a better insight of the functionalities the team found neccessary for the system.
All in all, looking back on the first sprint, the prestudy and research were very useful. It gave the group the opportunity to voice opinions and suggestions, and in turn improving the group communication.

7.3 Sprint 2

Summary

During this sprint the team presented the research results and a draft of a paper prototype for the customer. The presentation also included some screenshots of existing applications the team discovered during the research process.
The team also spent time on researching what kind of development tools could be appropriate to use, and started to set up a proper development environment.

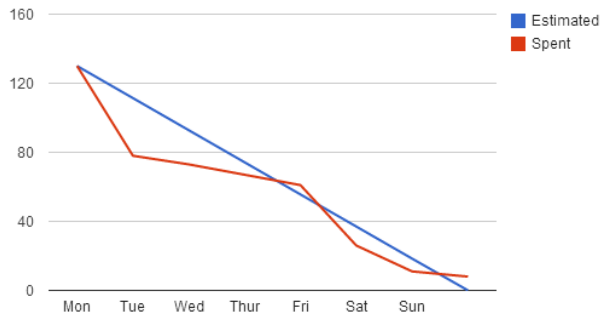


Figure 7.2: Burndown chart: Sprint 2

Review

The team has in retrospect, knowledged that this sprint could have been performed better. By stating the difficulties of lacking system requirements to the customer more clearly, it might have been possible for the team to start the development process earlier.
In the second sprint, it was experienced some trouble with setting up Android Studio properly. As a result, the group spent over twice as much time than what was estimated for development environment, and never got it to work.

All in all the team experienced some complications and starting difficulties, resulting in the team spending a lot of time on things that did not result in much afterwards.

7.4 Sprint 3

Summary

This Sprint has been characterized by both changes and progress.
The team met with a domain expert at St. Olavs, and got a clearer view of the requirements for the application. In addition, more time was spent working on the report, as this had been somewhat abandoned, and requested by the supervisor.
The team also came to the conclusion that the sprint backlog structure was not working optimally, and chose to use the necessary time to create a customized backlog, based on this projects requirements.
Early in the sprint, the team decided on changing the development environment from Android Studio to Eclipse, as it was agreed upon that enough time was spent on the issues with Android Studio.
Using Eclipse as a development enviroment was much better and was easily set up for everyone.

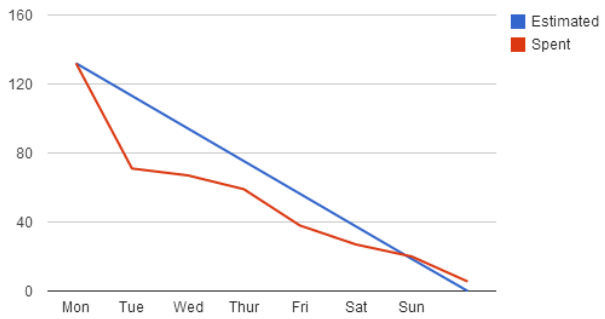


Figure 7.3: Burndown chart: Sprint 3

Review

The team had a meeting with the supervisor and received very useful constructive criticism in regards to the preliminary version of the report. It was realized that it was important to estimate, and put more effort into the report work, which resulted in a revaluation of using only Google Docs for the report layout. None of the group members had experience with text layout tools, but it was decided that learning to use LaTeXwould benefit the team.

The team also developed a specified requirements list in cooperation with the customer. This made it a lot easier to evaluate what tasks should be prioritized, and what functionality was expected in the final product.
The decision of going away from Android Studio, and rather use Eclipse, affected the work progress positively. By having both the specific requirements in place, as well as development tools that functioned optimally, contributed the implementation positively.
For the group, this sprint worked almost like a turning point in the adversity experienced at first.

7.5 Sprint 4

Summary
The report and relevant diagrams was in focus during this sprint, in addition to the continuation of developing a minimum viable product draft.

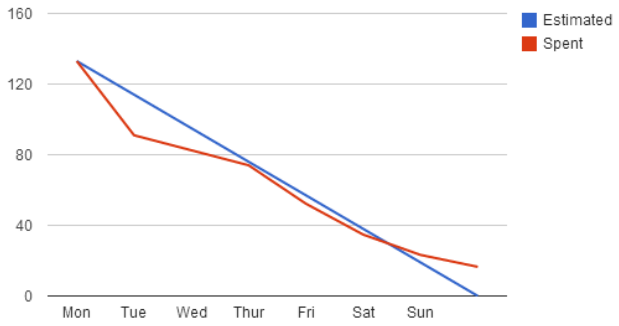


Figure 7.4: Burndown chart: Sprint 4

Review

Sprint 4 went very well, with a more stable workflow and improved time estimation. The planned customer meeting was postponed, but as the team had planned work process further, this did not effect the group as experienced in earlier sprints.
In the report, several sections were improved, and transferred to LaTeX.
Because a majority of the group struggled with using Git as an efficient tool, the team held a Git workshop. During the workshop those with more experience and knowledge learned the rest of the group how to use Git for a good workflow. This provided all group members with useful knowledge, and improved communication and cooperation within the team.

7.6 Sprint 5

Summary

During sprint 5, the report and minimum viable product still was the main focus. The entire report draft for the preliminary version was completed as a LaTeXfile, and sent to the supervisor for feedback.
The team developed a working prototype to show the customer for feedback during this sprint. But, as the customer meeting was canceled again, the apk file was sent to the customer so he could look at the product process so far.

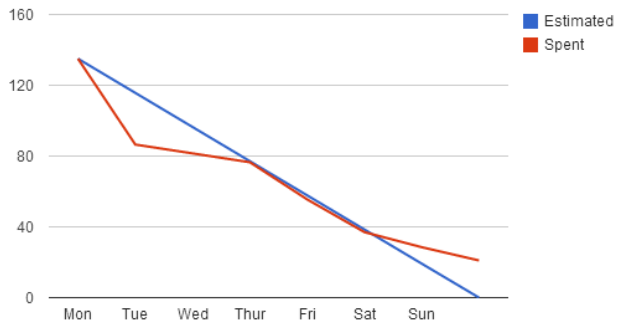


Figure 7.5: Burndown chart: Sprint 5

Review

This sprint could have been utilized better. Because of two cancelled customer meetings in a row, the team found it challenging to continue further development of the minimum viable product. The customer did not give feedback on the prototype in a few days as agreed, so the implementation process stagneted. The challenges experienced with lacking feedback was expressed to the customer.

7.7 Sprint 6

Summary

Delivery of the midterm version of the project report was due this week, and a lot of time was used on preparing, improving and complement the report. Backend implementation was commenced, but later delayed due to new requirements from the customer.
During the customer meeting, the team was given some pointers on what to improve, and created a roadmap for the future development jointly with the customer.

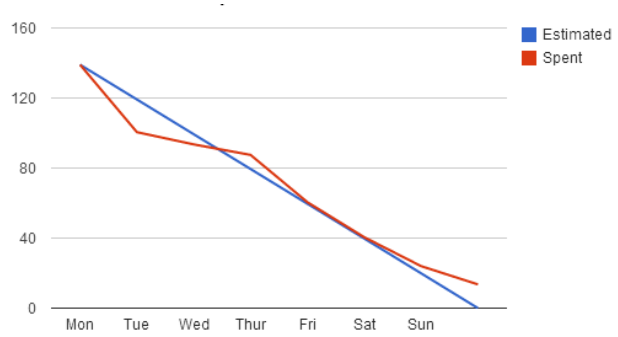


Figure 7.6: Burndown chart: Sprint 6

Review

The team underestimated the planned report working hours. Mostly because the supervisor guided the team on using time proofreading in addition to fill in the lacking sections before delivery of the midterm version.
The team still had difficulties on getting started with the backend implementation. Postponements of customer meetings and new requirements delayed the implementation process. A new requirement from the customer, requesting an interface for adding more content, resulted in the team starting from scratch with a backend analysis. The purpose of the analysis was to determine a final choice for backend implementation.

7.8 Sprint 7

Summary

This sprint it was time to conduct a process presentation for the class, and to do a peer evaluation of another group's report. The purpose was to share and learn from each other's experiences.

The customer meeting was conducted at St. Olavs hospital with domain experts, Jorunn Hellbostad and Randi Granbo. An improved version of the design theme was also presented during the meeting.
The report has also been a focus, taking into account the provided feedback from the supervisor. In addition, a general system test has been conducted.

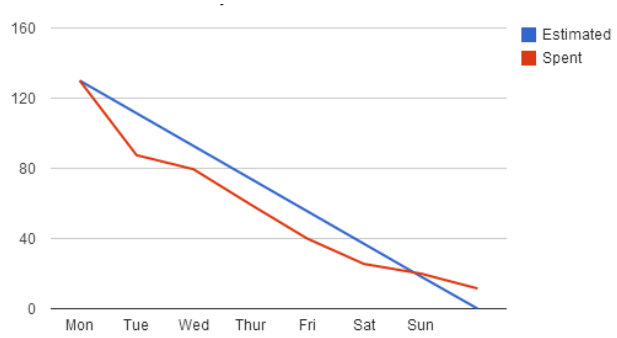


Figure 7.7: Burndown chart: Sprint 7

Review

The process presentations, focusing on complications and possible solutions, was a good way to reflect on the process so far. The team found this to be helpful for planning the process further.
The customer and domain expert meeting went very well. Randi gave ideas on how the practice part should be, and examples of well known exercises in the home care environment that could be implemented.

After the last meeting with the customer, he expressed a desire to see an improved design layout and comprehensive theme throughout the application. The improved design was presented in addition to the backend analysis. The team and the customer agreed upon using Parse, which is a "backend-as-a-service"solution.

The general system test was performed taking into consideration that the application was not completed. At this point, much of the application consisted of moving from page to page and loading data from files. The test checked if it was consistency between output in the application and the actual file. The errors found, was put as issues on GitHub.

7.9 Sprint 8

Summary

There was not conducted a customer meeting this sprint, as he was away. But with the decision in regards to the backend, the implementation was finally commenced.

The reportwork was mainly focused on improving sections. The team took into account both the latest supervisor feedback and the peer evaluation.

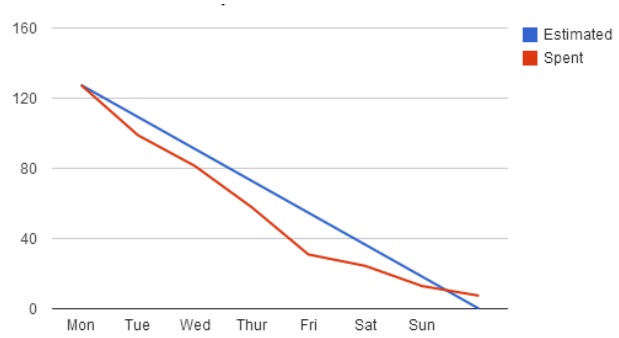


Figure 7.8: Burndown chart: Sprint 8

Review

The team made important progress this period. With the established backend implementaion and newly provided information in regards to the practice fragment, increased the work progress a lot.
The time estimation was been relatively good, only a cancellation of the customer meeting made a difference in the total estimation.
As can be seen in figure 7.8, the team worked evenly throughout the whole week, but ended up overestimating a bit.

7.10 Sprint 9

Summary

Because five of the team members were going on an educational excursion before the easter holiday, the team decided to merge the half week before departure, with the half week left after arrival to Trondheim. This resulted in a nine days long sprint, instead of the usual seven days.
Unfortunately, the team experienced some difficulties in reaching the estimated working hours because of unforeseen travel-related tasks, and after the trip, jetlag was an issue.

Before departure,the team met with the supervisor. As the supervisor will not be able to have more meetings with the team before the project deadline, the team utilized the opportunity to get a good overview for future work.
There was not any customer meeting this sprint, as there was some difficulties in finding an appropriate meeting time because most of the team was leaving so early, so the customer provided us with feedback via email.

A lot of the parse communication came together this sprint, but there has been a lot of bugs followed by the backend implementation. This required a lot of time to fix.

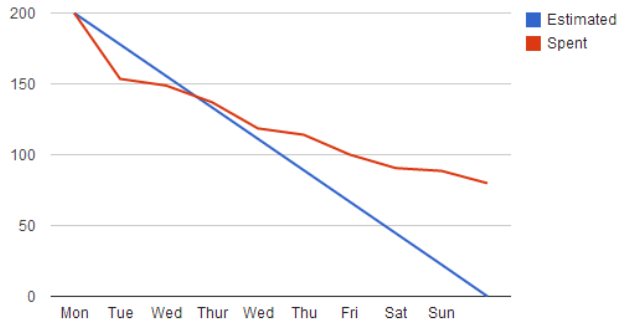


Figure 7.9: Burndown chart: Sprint 9

Review

This sprint was overestimated with about 80 hours, and could thus have been implemented in a better way. The main reason for why the workflow stagnated, was because 5 of 6 team members went on a educational excursion in the middle of a sprint, and was supposed to take up the sprint work 20 days later. Instead of creating a 9 days long sprint, the team would have benefited more from shortening the work period and start up fresh after the holiday.
Although sprint 9 was not an optimal iteration, the team started out very well, and took the experience to improve continuous sprint iterations.

7.11 Sprint 10

Summary

Sprint 10 was characterized by a lot of progress on both frontend and backend implementation.

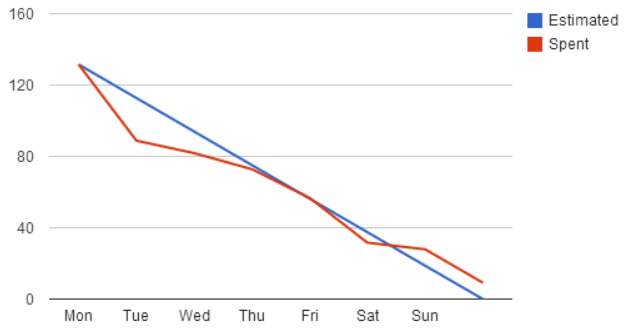


Figure 7.10: Burndown chart: Sprint 10

Review

As the project was getting close to the planned finish date, the team decided to postpone further development on correctly syncing classes related to practice in Parse.
It was agreed with the customer that the planned implementation the team did not find time to complete, should be added to GitHub. This way, SINTEF is able to keep working on the prototype, focusing on the lacking functionality.
The team had a steady workflow this sprint as shown in figure 7.10.

7.12 Sprint 11

Summary

Bugfixing, reportwork and overall improvement of the deiliverables, has been the main focus this sprint.

Review

The team agreed with the customer that sprint 11 would be the final development sprint. It was planned that the acceptance test should be performed after this sprint, and by such, the team had to finish the tasks necessary to achieve the customer acceptance.
To complete the planned tasks, the team met up in the weekend to collaborate.
As agreed with the customer, the team members created issues on GitHub, representing functionalities that can be implemented further.

Chapter 8

Evaluation

This chapter is as an evaluation of the project. The team make an assessment of the project management, the process, the project team, customer interaction and also the tools and aids used during the process. The chapter is ended by summing up the experiences and lessons learned throughout.

8.1 Development process

8.1.1 Process model

The development process has been based on SCRUM principles, with some modifications as explained in Chapter 3.3 Process guidelines.

For this project the main advantages of using SCRUM has been the continuous customer interaction and feedback, enabling the team to make necessary changes and adapt to the customers requirements to create a satisfying product. The team has experienced some challenges of using SCRUM. At first the team found it somewhat difficult to adapt to planning, structure and organizing a project without a very clear definition, as this was a new approach for the whole team. Also, frequent changes and uncertainty regarding the finished product, has been both challenging and educational. Furthermore, even though weekly SCRUM meetings worked for this project, the team believe SCRUM will work best if one has the opportunity to meet every day. As this ensures consistent progress and high level of cooperation and communication. At some points, the group experienced that there was

room for improvement on this area during the project. Overall, SCRUM has worked very well for this project development, keeping the process stable and prepared the team for changes along the way. As shown in the burndown charts in Chapter 7, the team has not underestimated tasks, and the work estimation has gone smoothly.

8.1.2 Time management

For the weekly sprints, it was estimated approximately 20 working hours per team member. This was found to be an appropriate workload, to keep the progress continuous, and still not overloading the team with too much work each week. This estimation meant that the group should work about 120 hours in total each sprint.

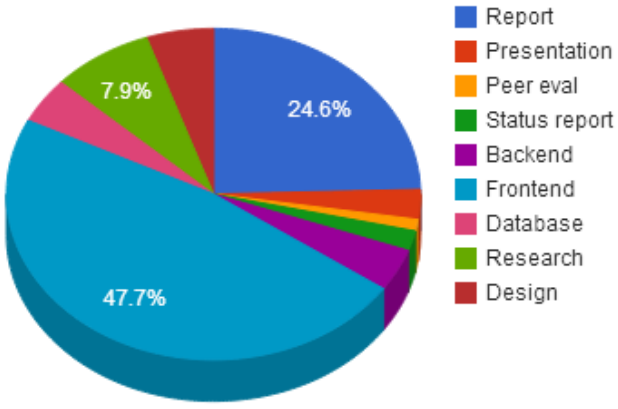


Figure 8.1: Acitivity summary

Although the team was not able to maintain the exact estimated working hours each week, the overall resource distribution has been successful.
Comparing to the WBS in figure 3.1, it is clarified that the planned estimation of resources has been maintained quite well.
During some sprints it was experienced that some team members worked more than 20 hours, and others less. This was discussed in plenary to better

balance the workload, and worked better for the final sprints. The team decided to change reporting platform early on, as the first choice did not meet the teams needs. As soon as the reporting platform was established, the time management worked well.

8.2 Project Management

The group collaboration became better and better throughout the project. Most of the team members knew each other from before, but through the project process all members has become better acquainted with each other. The team based on students with different knowledge and abilities, has utilized each other's knowledge, and attempted to distribute the workload equally amongst all.

The team has not experienced any major drawbacks, but there has been some minor complications along the way. These complications has mainly been communication and meeting related. Some late responses and changes of meeting times with the customer was in the beginning an issue for the team, but it was resolved and handled in a better way after some guidelines from the supervisor. Also, at some group meetings, the team did not always discuss the issues that needed to be handled, but exposed these to a later date. To improve this situation, the team implemented a better meeting organization to keep the progress continuous.

The team organization and responsibility roles has worked quite well. There was made some adjustments to the distribution of responsible areas after working on the project for a while, and discovering who contributed the most in which areas. Beyond this, the team structure has been optimal for the group.

8.3 Customer interaction

The biggest challenge with the customer interaction, was postponement of appointments and late responses. In the beginning it seemed like the customer might have a bit too much on his agenda to give feedback as rapidly as expected by the team. Also, because the customer stated that the project should involve a lot of research and would require continuous changes of functionality, the team had to take this into consideration because the workload

increased every week. The challenges was handled by keeping regular contact with the customer, and try to have weekly meetings when it was possible. The customer was made aware that the team needed feedback to keep the progress as good as possible, and he worked on handling this to help the team. The customer interaction improved as one got to know each other better, and the team did not only see the customer as a customer, but as a resource. The customer also provided the team with relevant domain experts to give us feedback and input through the process.

8.4 Tools and aids

At first, the team decided to use Android Studio for frontend development. Unfortunately it was experienced difficulties on getting the development environment to work properly on all of the team members computers. After using a lot of time on getting Android Studio to work, the team decided to not spend more time on this, and to use Eclipse with an Android Development Tool (ADT) plugin instead. The decision on using Eclipse for development instead of Android Studio, allowed for improvement on the development progress, which first was somewhat problematic.

To ease the project management, the team first used Planbox, an online agile development tool. This was used to plan the Scrum activities at the beginning of the project, but after using Planbox as the activity planner for the two first sprints, a reevaluation on using this tool was made. Planbox was not suitable for the team, and Google spreadsheets was used to document the Scrum activities instead. This worked as an appropriate and useful tool for the team in the continuous sprints.

8.5 The team

The team structure, with appointed responsibilities, has not been followed at all times. The team found it to be more useful to utilize the knowledge of the different members and rather share the responsibilites to keep the progress going. This resulted in different team members experiencing different roles through the project.
The team has experienced the importance of communicating when collaborating. It was also experienced how important it is to create a good manage-

ment structure. In order to plan, allocate and execute, the team has to have the same conception of what needs to be done. By establishing management rules, makes it easier to handle possible conflicts in the team.
The group did not experience any misunderstandings or conflicts, and concludes that the project management and teamwork has been successful.

Chapter 9

Conclusion

The project started as a very open task, with few specific requirements, letting the team provide input and ideas for the final product. The initial phase was mainly centered around brainstorming, research and prototyping, with the objective to determine more specific system requirements. When the requirements and system functionality was established in agreement with the customer, the team was able to work more independently and efficient with development. From this point, the progression was stable and solid until the end of the project.

The team has gained a lot of useful experience when it comes to the process of developing a working system; interaction as a team and with a customer, collaboration and implementation of necessary management resources. During the project, the team has learned to use some new technology, considering none of the team members had any experience in developing mobile apps, this was a new and exciting domain to work with.

The team has experienced unexpected changes during the project, and although it has been challenging at times, the experiences has helped the team evolve in a positive way as developers.

The experience of cooperation and teamwork is reviewed as very useful for further projects and work contexts. The team has realized the importance of this subject, that provide students with practical experience. As a conclusion, the project, in its entirety, has been a very good learning experience for all of us.

9.1 Product

As the team was made aware of that the product would function as a prototype and be used in the healthcare sector, it was in the teams best interest to create a product of good quality. UbiLearn should be a application that is very easy to use, and with a minimum of errors.
The team feels that these goals has been reached to a certain extent. As there is some functionality that is lacking, and there is always room for improvement, the term "to an extent" is used. The team is overall satisfied with the delivered product, and the customer has also stated that he is satisfied.

9.2 Future work

The delivered product is a complete prototype, and will require further development and functionality to function as a complete application for the intended use.

Because the delivered system would not be a complete application at the end of the project, the team has taken into account that there is potential for further development when frontend and backend solutions was determined. The team has agreed with the customer that he can contact a team member if there is any uncertainties or questions in regards to changing or improving the system. Although the team have assumed there will be created a new version of the application, based on the functional prototype made during the project. Based on customer feedback and interest in getting the product on the market, the group believes that the development will not be shelved at the end of the project, but further developed and improved.

Bibliography

[1] Ian Sommerville, Addison Wesley *Software Engineering*, 9^{th} Edition. (2011)

[2] Agile Project Planning - *http://en.wikipedia.org/wiki/File:AgileProject_Management_by_Planbox.png*, (2012)

[3] SCRUM image - *http://www.danielroot.info/2009/08/how-to-apply-scrum-using-fogbugz-7.html*, (2009)

[4] Requirements specification - *http://stackoverflow.com/questions/16475979/what-is-functional-and-non-functional-requirement*, (2013)

[5] SINTEF - *http://sit.sintef9013.com/about*, (2014)

[6] Course Information - *http://www.ntnu.no/studier/bit/oppbygging/prosjekt2*

[7] UbiCollab I - *http://files.ubicollab.net/pdf/Wang2012a.pdf*, (2012)

[8] UbiCollab II - *http://files.ubicollab.net/pdf/Wang2013a.pdf*, (2013)

[9] Stakeholder - *http://blogs.pmi.org/blog/voices_on_project_management/2009/09/who-is-a-stakeholder.html*, (2009)

[10] Healthcare Applications - *http://tbrelearning.org/apps/top-20-free-medical-apps-health-care-professionalsl*

Appendix A

User manual

Oppstart

Velkommen til UbiLearn! Når du starter opp applikasjonen vil det dukke opp et innloggingsvindu. Skriv inn brukernavn og passord, og trykk på "Logg inn" for å logge deg inn med din bruker.
Hvis du ikke har en UbiLearn-bruker, kan du opprette en ny bruker ved å trykke på "Registrer". Skriv deretter inn all informasjon du blir bedt om i vinduet som dukker opp, og trykk "OK". Ønsker du å bruke applikasjonen uten å logge inn, trykker du på "Skip".
Etter du har fullført innloggingen blir du tatt videre til Hjem-siden.

A.1 Hjem

Dette er hovedsiden til applikasjonen. Her finner du en oversikt over din brukerprofil og oppnådd poengsum i de ulike levelene fra opplæringsdelen. Trykker du på "Prestasjoner" kan du se alle utmerkelsene du har oppnådd hittil.

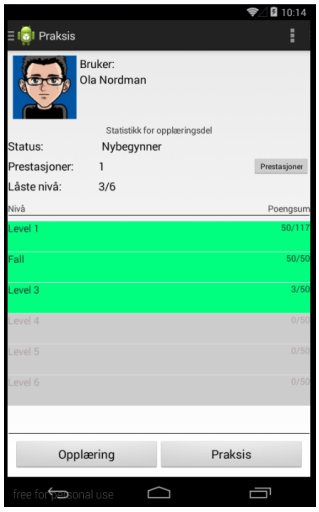


Figure A.1: Hjem

Nederst i bildet kan du navigere deg direkte til applikasjonens to hoveddeler; "Opplæring" og "Praksis".

Navigering

For å forflytte deg mellom de forskjellige delene av applikasjonen, bruker du navigasjonsskuffen. Denne kan når som helst dras ut fra venstre side av skjermen. Den inneholder valgene: opplæring, praksis, håndbok og førstehjelp.
Du kan også bruke navigasjonsskuffen for å komme deg tilbake til hjem-siden, og for å logge deg ut av UbiLearn.

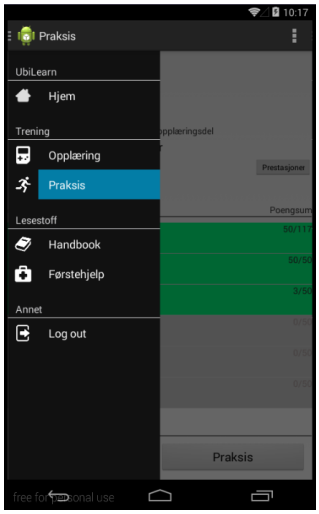


Figure A.2: Navigering

For å gå en side tilbake kan du trykke på tilbake-tasten på telefonen din.

A.2 Opplæring

I opplæringsdelen av UbiLearn skal du lære å håndtere forskjellige problemer som kan oppstå i din arbeidshverdag. Dette gjøres ved å fullføre en rekke mini-spill som simulerer realistiske situasjoner du kan møte på.
Opplæringsdelen viser et kart over et nabolag. Bilen på veien illustrerer din progresjon i spillet. Husene over bilen er oppgaver du har gjennomført, mens husene under bilen er oppgaver du ikke har låst opp ennå.
For å flytte bilen nedover langs veien, må du fullføre oppgaven i huset ved siden av bilen. Ved å dra fingeren opp og ned på skjermen, vil du kunne utforske hele nabolaget.

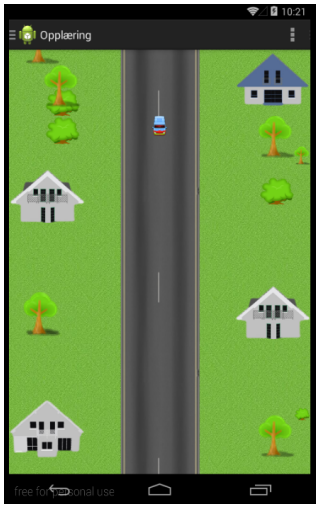


Figure A.3: Nabolag

For å starte en ny oppgave må du trykke på et av husene. Du vil da få opp en dialogboks med informasjon om hvem som bor i huset, og valgene "Gå inn" eller "Avbryt".

Figure A.4: Dialogboks

Når du går inn i et av husene, får du opp en informasjonsside om personen som bor der. Du kan deretter trykke på "Quiz" for å svare på spørsmål som er knyttet til informasjonen du har fått.

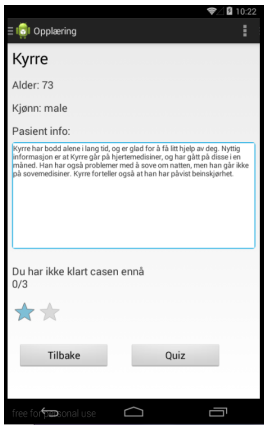


Figure A.5: Case

Du får fire alternativer på hvert spørsmål. Når du har trykket på et av dem, får du vite om du svarte riktig eller galt og det blir mulig å trykke på "Neste".

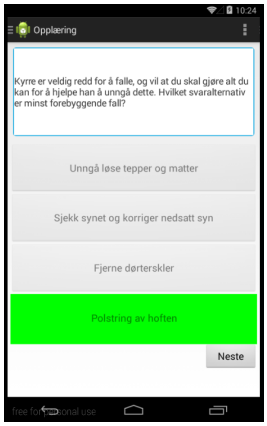


Figure A.6: Case

Når alle spørsmålene er besvart vil du få opp en dialogboks som forteller deg hvor mange av spørsmålene du svarte riktig på.
Du må svare riktig på alle spørsmålene for å komme deg videre til neste hus.

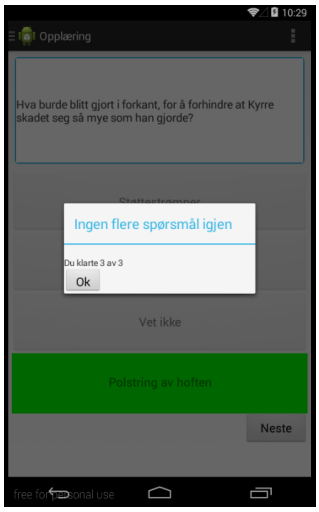


Figure A.7: Case slutt

A.3 Praksis

Praksisdelen av UbiLearn hjelper deg med å holde oversikt over pasientene dine. Her kan du opprette treningsprogrammer og gjennomføre SPPB-testene for hver pasient. Hovedsiden til praksisdelen viser en oversikt over dine pasienter. Du kan legge til en pasient ved å klikke på "+ " øverst i bildet. Ved å trykke på "Vis alle øvelser", vil du få opp en liste med alle profound-øvelsene.

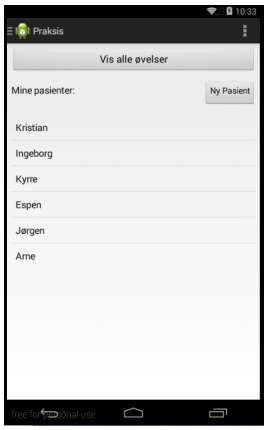


Figure A.8: Praksis

Hvis du trykker på en pasient, blir du presentert for en ny side som inneholder informasjon om denne pasienten. Pasientinformasjonen kan endres ved å trykke på blyant-ikonet øverst i høyre hjørne.

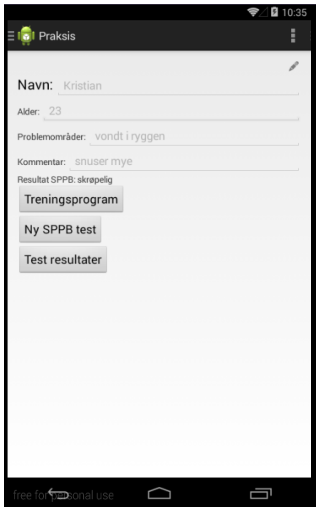


Figure A.9: Oversikt

Trykker du på "Treningsprogram", kan du opprette eller endre på treningsprogrammet til pasienten.
For å utføre SPPB-testene, trykker du på "SPPB test"og deretter velger du en av de tre testene. Når testen er fullført vil du bli tatt tilbake til denne siden, og ha muligheten til å velge en ny test.

Figure A.10: SPBB tester

A.3.1 Balansetest

Balansetesten består av tre deltester. Øverst kan du se hvilken del av testen som skal utføres, og fotstillingen pasienten skal ha. For mer informasjon, kan du trykke på info-ikonet øverst.
Trykk på "Start" for å starte hver testdel. Etter 10 sekunder, vil deltesten stoppe automatisk, og midlertidig poengsum vi regnes ut. Du kan avslutte deltesten før det har gått 10 sekunder ved å trykke på "Stopp". Når du har fullført alle deltestene, vil du kunne trykke på "Neste" for å komme videre.

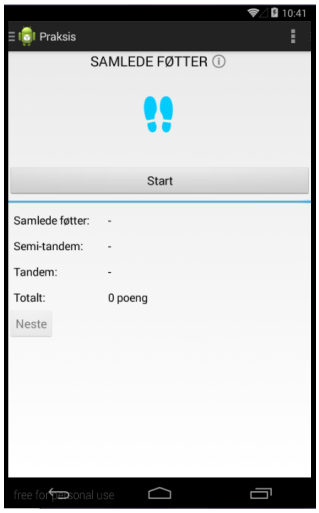


Figure A.11: Balansetest

A.3.2 Gangtest

Gangtesten skal utføres to ganger. Trykk på "Start" for å starte hver test, og "Stopp" for å stoppe dem. For mer informasjon om gangtesten, kan du trykke på info-ikonet øverst. Før du trykker på "Neste" kan du krysse av for hvilke hjelpemidler deltakeren brukte for å gjennomføre testen.

Figure A.12: Gangtest

A.3.3 Reise/sette seg

Trykk på"Start" for å starte testen, og deretter på "Repetisjon" for hver repetisjon pasienten gjør. For mer informasjon om denne testen, kan du trykke på info-ikonet øverst.

Figure A.13: Reise/sette seg

Når du har registrert 5 repetisjoner, vil testen avsluttes automatisk, og du kan trykke på "Neste". Når du trykker på "Neste" vil du bli tatt videre til en ny side. Hvis testen ble avbrutt, skal du her krysse av for grunnen til at testen ble avbrutt.

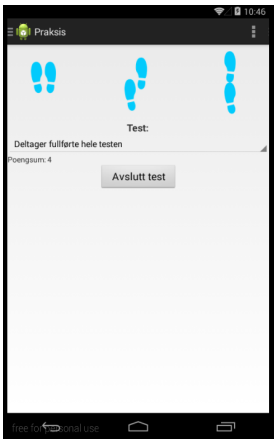


Figure A.14: Avslutt test

A.4 Håndbok

I håndboken kan du slå opp og lese forskjellige artikler som kan være relevante i din jobb i hjemmetjenesten. Du kan bruke søkefeltet øverst til å finne raskere fram til den artikkelen du ønsker å lese.
Artikkelen om førstehjelp kan i tillegg nås fra navigasjonsskuffen.

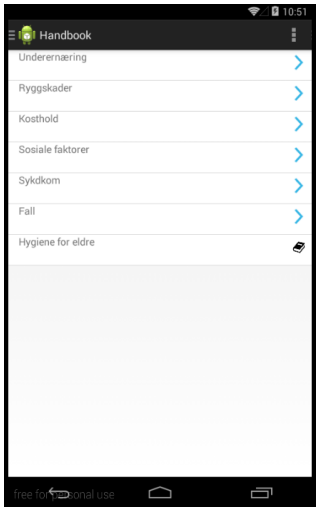


Figure A.15: Håndbok

Appendix B

System testing

The number which is part of the ID is referencing to the ID of the functional requirement. Some of the test cases are not complete, as a result of uncertainty about the system functionality in the final product. The test cases influenced by this will be updated.

Table B.1: User profile test

Test name	**User profile test**
Test ID	FR 01
Purpose	Showing datafor the specified user
System condition	User is logged in
Input	Email and password
Expected response	Data for the specified user: name, status, achievements, unlocked cases, locked cases. If the score of the case is high enough, the marked case will contain a medal icon.

Table B.2: Training game test

Test name	**Training game test**
Test ID	FR 02
Purpose	Select different cases, represented by houses, in the training page of the application
System condition	The user could be logged in, but this is not required
Input	Clicking a particular house representing the case
Expected response	Given the opportunity to enter house or abort

Table B.3: Save state test

Test name	**Save state test**
Test ID	FR 03
Purpose	The user should be able to save different training programs for each patient, as templates in the application
System condition	The user must be logged in to access this functionality
Input	Clicking the training program button in the practice part
Expected response	The user will enter a page where he will be able to edit or add new training program templates for patients. These templates should be saved, if chosen by the user, when logging out

Table B.4: First aid test

Test name	**First aid test**
Test ID	FR 04
Purpose	Test to see if a user can select different categories of accidents and get a description of what to do
System condition	The user could be logged in, but this is not required
Input	Clicking the menu icon representing first aid
Expected response	The user will enter a page where he should be given the opportunity to select different categories in first aid and finally get a description according to type of accident

Table B.5: Handbook test

Test name	**Handbook test**
Test ID	FR 05
Purpose	Check if a selected page will be shown with the specified article
System condition	The user could be logged in, but this is not required
Input	Clicking the menu icon representing the handbook
Expected response	We will enter a page with a list of selectable categories. Selecting a category gives a new specified list within that category, and clicking one of these should open a page with relevant information

Table B.6: Expert help test

Test name	**Expert help test**
Test ID	FR 06
Purpose	Provide a telephone number to an expert. (Possibly also an email address)
System condition	The user should be logged in
Input	This has not yet been implemented. Could possibly be implemented as an icon in the menu which the user clicks
Expected response	Open a page with a phone number to an expert. (Possibly an email address)

Table B.7: Menu test

Test name	**Menu test**
Test ID	FR 07
Purpose	To see if the menu is available on all pages
System condition	User must be logged in to have access to all pages in the application
Input	Email and password to log in
Expected response	Menu available on all pages

Table B.8: Practice test

Test name	**Practice test**
Test ID	FR 08
Purpose	Check to see if the user can select different practice exercises
System condition	User can be logged in, but is not required to
Input	Select practice from the menu or from the home page
Expected response	A page which gives the user the opportunity to select exercises and/or patients. Selecting exercises brings the user to a page where he can watch pictures or movies of exercises, or be given a program to perform with the patient.

Table B.9: Communication test

Test name	**Communication test**
Test ID	FR 09
Purpose	Test if the user can communicate with other users and is able to see the progress of other users
System condition	User must be logged in
Input	Email and password to log in and a user must be selected
Expected response	Page with different users that can be selected. Selected user can be communicated with and profile shown for this user

Table B.10: Offline usage test

Test name	**Offline test**
Test ID	FR 10
Purpose	To see if the user can use the application without being logged in. Offline usage of the application will be limited in terms of different features
System condition	User not logged in
Input	No specific input expected
Expected response	User has no access to an expert and cannot contact other users. Limited set of features

Table B.11: Domain generalized test

Test name	**Domain generalized test**
Test ID	FR 11
Purpose	To see if the system supports different domains
System condition	User can be logged in, but is not required to
Input	Not yet decided how to implement this.
Expected response	A case or practice options related to a specific domain

B.1 System test example

An example of a system test conducted Wednesday 07.05.2014.
Code pulled from GitHub at: 19.30

Table B.12: Test intro

Code	**Description**
A	Accepted
F	Failure
U	Missing some features / unfinished

Table B.13: activity_login.xml

Page loaded	A	
Sign in or Register button	A	
Skip button	A	Home page loaded

Table B.14: fragment_home_cases.xml

Page contains data in accordance to the level selected in fragment_home.xml	A	
"Back"button (on the phone) leads to the home page.	A	As a note, the back button in the application has been removed.
Selecting another page from this page, and then clicking the "Back"button returns the user to this page	A	

Table B.15: home_fragment.xml

Page loaded	A	A bug was found related to the Home pages. When the user returned to one of the Home pages from another page, new data where added to previously loaded data in these pages. This error were related to all Home pages: home_fragment.xml, fragment_home_cases,xml, fragment_home_achievements.xml. (The errors are related to the java files, not the .xml files, but as part of testing we refer to the .xml files). This error has now been fixed.
General text elements loaded with correct data according to testdata loaded in objects	A	
"Prestasjoner"button loads a new page containing achievements	A	
Each level in list has a name	A	
Each level has a score	A	
"OpplÃęring"button leads to training page	A	
"Praksis"button leads to practise page	A	
Pushing each element in the list with levels leads to a new page which has more detailed data for the selected level	A	
Selecting another page from the Home page, and then clicking the "Back"button returns the user to the Home page	A	

Table B.16: fragment_home_achievements.xml

Page contains data according to the achievements obtained	A	
"Back"button leads to the home page	A	As a note, the back button in the application has been removed.
Selecting another page from this page, and then clicking the "Back"button returns the user to this page	A	

Table B.17: training_case.xml

Page loaded	A	
Each of these cases correspond to the house selected in fragment_training.xml	A	
"Tilbake"button takes the user back to previous page, fragment_training.xmlSkip button	A	
"Quiz"button takes the user to the next page	A	

Table B.18: fragment_training.xml

Page loaded	A	
Page is loaded with image of the road, a car and several houses	A	
Selecting a house will open a dialogbox where the user can choose “Gå inn” or “Avbryt”	A	
Selecting a house will move the car next to the selected house	A	
Selecting “Gå inn” space opens a window for that specific case	A	
Selecting “Avbryt” space will close the window and we return to the page where we can select another house	A	
Some of the houses are locked, leading to the information that no one is at home (“Ingen hjemme”)	A	
Selecting another page from the Training page (from the menu), and then clicking the “Back” button returns the user to the Training page	F	The application stops. Bug has been made as an issue to be solved on GitHub.

Table B.19: fragment_training_quiz.xml

Page loaded	A	
An option can be selected among several which leads to a text output indicating right/wrong answer	A	
A “next”button pops up after selecting an answer, such that next question can be given	A	
“Back”button button leads to training_case.xml	A	

Table B.20: fragment_practice.xml

Page loaded	A	
A list of pasients shows up	A	
Clicking a pasient among “Mine pasienter”leads to a new page	A	
Clicking “Vis alle øvelser”leads to a page with exercises	A	
Selecting another page and clicking “Back”leads to this page	A	

Table B.21: fragment_practice_exercises.xml

Page loaded	A	
The page contains some execises	A	
Selecting another page and clicking “Back”leads to this page	F	When the user selects a page from the menu and select “Back”, the user is not taken back to this page, but to other pages. Bug has been made as an issue to be solved on GitHub.

Table B.22: fragment_practice_patient.xml

Page loaded	A	
“Treningsprogram”button leads to a new page	U	Clicking the button does not work yet.
“Ny SPPB test”leads to a new page	A	
Test “resultater” button leads to a new page	U	Clicking the button does not work yet.
Selecting another page and clicking “Back”leads to this page	U	When the user selects a page from the menu and select “Back”, the user is not taken back to this page, but to the Home pages. Considering that many of the pages leads to the Home page from the Practice part, this is something we need to find out if we want to keep, or change.

Table B.23: fragment_practice_spbb.xml

Page loaded	A	
"Balansetest" button leads to a new page	A	
"Gangtest" button leads to a new page.	A	
"Reise/sette seg" button leads to a new page	A,U	No real test is yet implemented.

Table B.24: fragment_practice_balance.xml

Page loaded	A	
The user can do a test	A	
Clicking the "Information" text leads to a page with description of the test	A	

Table B.25: fragment_practice_gangtest.xml

Page loaded	A	
The user can do a test	A	
Clicking the "Information" text leads to a page with description of the test	A	

Table B.26: gangtest_resultat.xml

Page loaded	A	
Clicking the button "Avslutt test" leads to the previous page	A	

Table B.27: fragment_practice_standup.xml

Page loaded	A, U	The page is not yet finished

Table B.28: fragment_handbook.xml

Page loaded	A, U	The page is not yet finished
The page contains a selectable list with categories that eventually leads to an article	A,U	The articles contains only dummy text, and there does not exits any particular categories

Table B.29: fragment_firstaid.xml

Page loaded	A, U	The page contains only dummy text

Table B.30: Menu

The menu is available on every page	A, U	The page contains only dummy text
The user can go to the home page	A	
The user can go to the training page	A	
The user can go to the practice page	A	
The user can go to the handbook page	A	
The user can go to the first aid page	A	
The user can logout	A	

Table B.31: Logout

The user is logged out and taken to the page given by activity_login.xml	A	

Appendix C

Example of a status report

See next page

PROJECT: SINTEF_training

SUMMARY STATUS REPORT NO. #3

1 Introduction

This report includes the status after working on, and finishing sprint #05 and #06. We also present the estimated working hours for sprint #07.

2 Progress summary

Report work

We have during these sprints spent more time on report layout and content. The google docs draft has been filled into the LaTex-file. We also delivered a midterm version of the report to the supervisor for feedback.

During sprint #06 we added testing and architecture sections. We also fixed issues based on the feedback from the supervisor. We finished and delivered the midterm version of the report on time.

Project planning

Class diagram:

An abstract class-diagram was made. Also a diagram to show the overall architecture, and a relation diagram for the database.

Finish usecases

The use cases were updated after receiving feedback from the supervisor.

Finish WBS

WBS was finished but after feedback from the supervisor we decided it needed some more work. We decided to push this over to the next iteration

Finish Gantt-diagram

The Gantt-diagram was completed and added to the report.

Finish Architectural overview

Also completed and added to the report.

Minimum viable product

Menu drawer

Icons where added succesfully.

Training fragment

Quiz feedback added. Car also implemented

Practice fragment

Lacking some information on what this fragment should contain, but the GUI has been implemented.

Main/profile fragment

A list has been added to the homepage, containing different cases represented with a medal if the user has enough score for that particular case, the name of the case and finally the score achieved. There was also a bug that was related to this page that was solved. We have also aligned text elements on this page.

Handbook fragment

Icons were added. General styling was improved. Navigation now works with a “go back” button.

Consistent styling

Some attempts at abstracting style to general xml files were made. We also spent time discussing general styling strategies

Meetings

Sprint meeting

We have tried to estimate the tasks as accurately as possible, but ended up with overestimating on some parts and underestimating on others. We also used the meetings to go through and discuss work tasks for the next sprint.

Customer meeting

We had a customer meeting during sprint #05 that was really planned during sprint #04, but had to be rescheduled because of travel. During the meeting that was held at the offices of SINTEF, we first went through the product progress so far, and received good feedback on this. Babak also gave us some pointers to what to improve.
He also requested to get the updated APK file when we added new functions to the application.
The second customer meeting was held Friday 14/3. We hoped to come to an agreement in regards of the implementation of the system backend, but was encouraged to spend more time on researching open source solutions to use. Babak also wanted to be more involved in the "sprint meeting" process, and we used time on creating a roadmap for our future development in cooperation with him.

Supervisor meeting

We received a lot of useful feedback during the supervisor meeting in regards to the report. Because we had the meeting before the submission of the midterm report, we were able to focus on improving the parts of the report that needed extra work.

3 Open / closed problem

- We find it difficult to implement the practice part because we are lacking information on what it should contain. We plan to have a meeting with the domain expert during sprint 7 to resolve this issue.
- The customer requested us to use more time on analyzing possible backend solutions. We find this challenging because we feel like this is an important part, that is urgent to get into place. We hope to resolve this problem within the next customer meeting.

4 Planned work for next period

Sprint #7 - 17-23/3

ID	Description	Responsible	Estimate	Daily hours logged							Total Logged
				mon	tue	wed	thu	fri	sat	sun	
1	**Report work**										
1.3	Status report		2								0
1.4	Final Report		20								0
1.5	Sprint retrospect		3								0
1.6	Peer evaluation		10								
7	**Project Planning**										
7.5	Rework WBS		4								0
2	**Meetings**										
2.1	Sprint meeting		6								0
2.2	Customer meeting		6								0
7	**Backend**										
7.1	Server side REST api		12								0
7.4	Implemention analysis		10								
8	**Frontend**										
8.3	Design theme		10								0
8.2	RESTful communication with server		10								0
9	**Unforeseen**										
9.1	Bug fix										
10	**Presentation**										
10.1	Process presentations		22								
11	**Testing**										
11.1	General testing		5								
	Sum:		120	0	0	0	0	0	0	0	0

5 Updated risks analysis

We got feedback on the last version of the risk analysis, and was encouraged to make some adjustments. After taking into consideration the understanding of preventive and remedial actions we looked over the analysis again. Most of the risks have been updated, either by reevaluating the degree of likelihood, impact and importance, or by changing the remedial or preventive actions.

The risks are still based on our teams experiences, but also taking into consideration feedback from the supervisor and looking at earlier groups risk analysis and experience.

6 Time spent on project

Here is an overview of the logged hours from the last two sprints, #05 and #06.

Sprint #05

ID	Description	Responsible	Estimate	Daily hours logged							Total Logged
				mon	tue	wed	thu	fri	sat	sun	
1	**Report work**										
1.3	Status report		2				1			0.5	1.5
1.4	Mid term Report		40	10.5	2	2	6	5		3	28.5
1.5	Sprint retrospect		4	1			1		0.5		2.5
7	**Project Planning**										
7.7	Create class-diagram		4	2	1						3
2	**Meetings**										
2.1	Sprint meeting		9	9							9
2.2	Customer meeting		6	6							6
6	**Minimum viable product**										
6.2	Menu drawer		5	4			2		2		8
6.7	Handbook fragment		10	3			1				4
6.3	Training fragment		20	6	2	3	2	2	2	2	19
6.4	Practice fragment		15	4			2	5			11
6.5	Home/profile fragment		12	3			6	6.5	1		16.5
6.10	Consistent styling		8						3		3
	Sum:		135	48.5	5	5	21	18.5	8.5	5.5	112

- We planned on working more on the report than what was actually conducted during this sprint. The reason for why we downgraded the report work, was because we spent time on enhancing different project areas like the minimum viable product. We decided to spend more effort on the midterm version during the next sprint.

- Some of the system fragments required more time than estimated, mainly because of bug fixes and unforseen errors.

#Sprint 06

				mon	tue	wed	thu	fri	sat	sun	
1	**Report work**										
1.3	Status report		3				0.5			0.2	0.7
1.4	Mid term Report		40	9.5	2	2	11	11.75	8.5	6	50.75
1.5	Sprint retrospect		3	1						0.25	1.25
7	**Project Planning**										
7.4	Finish Usecases		2			1					1
7.5	Finish WBS		1	1							1
7.6	Finish gantt-diagram		2	2							2
7.8	Finish architecture overview		1	1							1
2	**Meetings**										
2.1	Sprint meeting		9	9							9
2.2	Customer meeting		6					6.25			6.25
2.3	Supervisor meeting		6	5							5
6	**Minimum viable product**										
6.3	Training fragment		10	3	3		3.5			2	11.5
6.4	Practice fragment		10	4			5			2	11
6.5	Home/profile fragment		1	1							1
7	**Backend**										
7.1	Server side REST API		10				4		6		10
7.2	Server side SQL		10	1	2	1	3				7
7.3	DBMS		10	1				1	2		4
8	**Frontend**										
8.1	Store data in lightSQL		10								0
8.2	RESTful communication with server		5								0
	Unforeseen										
	Bug fix					2		1			
	Sum:		139	38.5	7	6	27	20	16.5	10.45	125.45

- We underestimated time needed to finish the midterm report, and found the report work to be more comprehensive than expected. Also we used time on proof reading and correct some minor errors throughout the report.

- We also spent a little more time on the training and practice fragment of the minimum viable product than estimated.

- After the customer meeting on Friday 14th, we came to agreement with the customer that we should research on solutions to be used on Backend, and then present this on the next customer meeting. Because of this the process has somewhat stagnated regarding frontend and backend development.

Appendix D

Example of minutes with the customer

Customer meeting 07.02.14
Present: Babak and all team members

Agenda
Issue 1. Give the customer a status update, and conversely.
Issue 2. Clarify some system functionality.
Issue 3. Present and get feedback on the research results with screenshots and paper sketches
Issue 4. Establish more specific system requirements

Summary

Agree on dividing the application in three main categories or phases
Phase 1. training
Phase 2. practical learning
Phase 3. Community of practice, handling reflection and experiences

- Babak tells the group to conduct more time in looking into specified techniques for each phase. We should also work with different concepts, and present even more examples for the next meeting.

- A more concrete requirements specification will be provided when we have done more research.

- The customer is very focused on agile development, as he finds it to be most suitable for this project.

- The customer gives us an example to use in the practical learning in phase 2. If we manage, we can use embedded video as a part of this phase.

- The customer plans for us to have a meeting with Jorunn at St. Olavs next week to get specified cases to work with and other suggestions for the system.

- If the team finds time, we should have both a paper protoype, class diagram and a basic draft of the application ready for the meeting. The core functionality

- The customer states that the application should be a minimum viable product, with a great focus on usability.
- When it comes to the use of forum, sharing experiences with other people online, we have to be careful in regards of anonymity and privacy. This is a part we have to look more into.

- We also have to take into consideration that the user might not be online all the time, and the core functionality of the appliction has to work offline. It must be able to save things locally and synchronize the changes when the user is online.

Appendix E

Stakeholder meetings

E.1 Customer meetings

Table E.1: Customer meetings

Date	**Agenda**
31.01.14	Get to know the customer, and get a better overview of the project specifications.
07.02.14	Present result of the research done since last meeting, including screenshots and some simple paper sketches. Establish more specific requirements and regulations for the system.
14.02.14	Determine what requirements the system should include. Get feedback on the paper prototype we sent as an presentation by email.
21.02.14	Domain expert meeting at St. Olavs hospital to present the paper prototype and get feedback on this.
03.03.14	Present the preliminary minimum viable product to get feedback on this.
14.03.14	Present results on back-end research, and come to agreement in regards to the back-end implementation. Get feedback on the improved minimum viable product.

21.03.14	Customer and domain expert meeting. The agenda was to present results on back-end research. Get feedback on the improved minimum viable product. In addition we needed some input on parts of the application where we are lacking relevant data, like the practice part.
28.03.14	Customer feedback per email because we did not find a suitable meeting time. General update about the progress.
30.04.14	Feedback on the process since last meeting. Plan when to conduct the acceptance testing
09.05.14	Updates on the application, agree on last minute changes before acceptance testing.
13.05.15	Conduct the acceptance test with the customer

E.2 Supervisor meetings

Table E.2: Supervisor meetings

Date	**Agenda**
30.01.14	Get to know each other and get information on what to expect of the project, supervisor role and general startup advice.
17.02.14	Report feedback. Discuss the problems with getting in touch with the customer. Get some guidelines on what we can expect from our customer regarding feedback and contact.
10.03.14	Feedback on the preliminary version of the report.
21.03.14	Received feedback on the midterm report per email because the supervisor was out travelling
31.03.14	Report feedback, focusing on process model, architecture and testing
03.04.14	Email feedback on changes to the process model chapter