

Giraf

Product Owner

Bachelor Project Report
By SW6-12-F17

Aalborg University



AALBORG UNIVERSITY

STUDENT REPORT

Computer Science

Aalborg University

Cassiopeia

Selma lagerløfs vej 300

9220 Aalborg Øst

<http://www.aau.dk>

Title:

Giraf

Theme:

Product Owner

Project Period:

Spring Semester 2017

Project Group:

SW612F17

Participants:

Krause-Kjær, Thomas Henrik Stagstrup

Lepka, Nikolaj

Sørensen, Anders

Sørensen, Daniel Thiemer

Abstract:

The Graphical Interface Resource for Autistic Folk (*Giraf*) is an application suite designed to help institutions that work with people suffering from Autism Spectrum Disorder (ASD). *Giraf* is a multi-group project developed by students from the *Software Engineering* course in their 6th semester and has been in development for six years. The group writing this (SW612F17) was responsible for all communication between the customers and developers; a role referred to as *Product Owner*.

Supervisor:

Bacci, Giorgio

Copies: 1

Page Numbers: 57

Date of Completion:

October 5, 2017

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Summary

This report focuses on issues related to managing customer contact in a large team of developers, who work on developing software. There are three institutions receiving the software—*Birken*, *Egebakken*, and *Enterne*—all handling patients suffering from various levels of Autism Spectrum Disorder (ASD).

Our motivation for this project, is to make a tool to help both the guardians and the people suffering from ASD. The developed tool should be used to improve the quality of life of those suffering, by helping them act more autonomously. Additionally, as this project has been handed down to us by students from the previous years, our goal is to improve what they already made, with focus on a small package solution including the Launcher and Week Planner applications.

The project is divided into five *Sprints*, to structure the process and organise communication between groups. During the Sprints the following was accomplished.

0. The Initial Sprint—We familiarised ourselves with the system. Representatives from *Birken* visited the university and gave a presentation. They explained what they knew about Giraf, and went into detail on the needs of citizens with ASD. The first set of user stories were generated based on this meeting.
1. The First Sprint – Progress on developing the system started. A mail was send to all customers, but a meeting did not occur due to some internal misunderstandings. However we got a meeting planned with *Egebakken*, but they did not have time before next Sprint.
2. The Second Sprint – We had the meeting with *Egebakken* and conducted the interview at their location, gathering information on how they would like to use Giraf. Additional user stories were developed for next Sprint.
3. The Third Sprint – We established an interview with *Enterne*, where we showcased the new additions and ideas to the system. New user stories were written from the feedback to the system, which were then relayed to the other groups.

We also conducted an interview with *Birken*, this time with less focus on getting new features, but more on getting feedback on what has been done.

4. The Fourth Sprint – We informed the customers of the current state of the system. We also updated the documentation relating to the system and preparing the system to be worked on by students the next year.

In conclusion, we made several improvements to the product, and documented the unfinished tasks still left in the system. However, the product is still not ready to be deployed, but it has gotten considerably closer to being stable.

Preface

This paper is written by project group *SW612F17* from the Department of Computer Science at Aalborg University. The duration of this project is February 2017 to May 2017.

The theme of this semester is *Developing Complex Software Systems*. The focus of the project is to continue the development of the *Giraf* system.

We expect readers of this report to have a basic understanding of software engineering, as well as a basic understanding of programming and software development techniques.

We would like to give thanks to our supervisor *Giorgio Bacci* for giving us advice while writing this paper. We would also like to give thanks to the other groups working on *Giraf* for good teamwork while developing the system.

The Project is developed with three institutions focused on ASD, the institutions are the kindergarten *Birken*, the school *Egebakken* and the housing institution *Enterne*.

The citation method used in this report is the Harvard citation, with sources at the back of the report. Figures and sections are referred to by their number, which is identified by their chapter number.

Contents

Introduction	7
0 Initial Sprint	9
0.1 Methodology	9
0.1.1 Scrum	9
0.1.2 Scrum of Scrums	11
0.1.3 Customer Meetings & User Stories	11
0.1.4 Tools	12
0.2 Responsibility	12
0.3 Customer Contact	14
0.3.1 Preparation for the Meeting	14
0.3.2 Customer Meeting	14
0.3.3 Reflection on the Meeting	15
0.4 User-stories	15
0.4.1 Reflection on the Sprint	16
1 First Sprint	17
1.1 Objective	17
1.2 Status	18
1.3 User Stories	19

1.3.1	Stories From the Previous Sprint	19
1.3.2	Stories for the Next Sprint	19
1.4	Customer Contact	20
1.5	Conclusion	20
1.5.1	Reflection	20
2	Second Sprint	22
2.1	Purpose of the Sprint	22
2.2	Status	22
2.3	Customer Contact	23
2.3.1	Before the Meeting	23
2.3.2	Customer Meeting	24
2.3.3	Reflection on the Meeting	25
2.4	User Stories	26
2.4.1	Usage of Previous Stories	26
2.4.2	Stories for the Next Sprint	26
2.5	Conclusion	26
2.6	Reflection	27
3	Third Sprint	28
3.1	Purpose	28
3.2	Status	28
3.3	Customer Contact	29
3.3.1	Before the Meeting with Enterne	29
3.3.2	Enterne Customer Meeting	30
3.3.3	Reflection on the Meeting with Enterne	33
3.3.4	Before the Meeting with Birken	33

3.3.5	Customer Meeting with Birken	34
3.3.6	Reflection on the Meeting with Birken	36
3.4	User Stories	36
3.4.1	Usage of Previous Stories	36
3.4.2	Stories for the Next Sprint	37
3.5	Conclusion	38
3.6	Reflection	38
4	Fourth Sprint	39
4.1	Purpose	39
4.2	Status	39
4.3	Customer Contact	40
4.4	User Stories	40
4.5	Code	41
4.6	Conclusion	42
4.7	Reflection	42
5	Project Status	43
6	Wiki Page	44
6.1	Wiki Structure	44
6.1.1	Introduction	44
6.1.2	Wiki Page Setup	45
6.1.3	Structure Argumentation	46
6.2	Start-Up Guide	47
7	Collaboration: Between Product Owner and Week Planner	49
7.1	Collaboration	49

7.1.1	Customer Questions	49
7.1.2	Demonstrating the Week Planner Application	50
7.1.3	Smaller Interactions	50
7.2	State of Week Planner	50
7.2.1	Stability	51
7.2.2	Accessibility	51
7.2.3	Aesthetics	51
7.2.4	PDF	52
7.2.5	Tutorial	52
7.3	Suggestions for Future Development on Week Planner	52
8	Conclusion	54
8.1	Project Reflection	55
A	Questions for the Customers	ii
A.1	Questions Birken February 15	ii
A.2	Questions Egebakken	iii
A.3	Questions Enterne	vii
A.4	Questions Birken April	xi
B	User Stories	xiv
B.1	User Stories Birken 15/02-17	xiv
B.2	Egebakken User Stories 22/03-17	xv
B.3	Enterne User Stories 06/04-17	xvii
B.4	User Stories 01/05-17	xviii
C	Guides	xx
C.1	How a Guardian Uses the System	xx

C.2 How to be a Product Owner	xxvii
C.2.1 Contacting the Customers	xxvii
C.2.2 Holding the Meeting	xxvii
C.2.3 User Stories	xxviii

Introduction

There are many people in the world struggling with Autism Spectrum Disorder (ASD), and it is estimated that around 1% of newborns have it [1]. Social education workers specialising in autism, strive to help people with ASD to function as close to the social norms as possible, with various methods and tools designed specifically for being used by individuals suffering from this disorder. The goal of the Giraf project is to develop IT-tools to ease the everyday life of people living with these afflictions, reducing the workload of the social education workers.

We are assigned the task of working on a multi-group project, having us co-operate with other groups on a joint product. This means that we will work together with several other students from our year in multiple groups specialising in different tasks for this project.

This report focuses on aspects of being the product owners on the Giraf project, which means we handle the customers, showing the customers our progress, and providing feedback to the developers.

Since student groups initiated the project in previous years, there were initial tasks of familiarising ourselves with every aspect of the former work; thus we will start by focusing on getting to know the requirements and the customers' wishes.

Initially, we have begun by having a look at the Giraf product handed over to us from previous students, having a look at their suggestions on how to continue its development. There were many additions to Giraf which we could work on, and focus on a small package, containing the most requested features from the customers.

This decision was made immediately after because the customers would like to get something functional they could start using when we hand in the product. The project is intended to be made for three different institutions: *Enterne*, *Egebakken*, and *Birken*. The three of them each work with the same citizens at different stages of their lives. The citizens have difficulty adjusting to changes; therefore they could benefit from using the same system throughout their upbringing.

However, the institutions have slightly different requirements:

Enterne houses adult citizens, where each of them gets their own tablet.

Egebakken instead takes care of younger citizens, although each citizen will still have their own tablet.

Birken takes care of even younger citizens, but their citizens share only a couple of tablets.

The system is designed to help the citizens evolve their abilities as they grow up. This requires the system to be customisable to each citizen's needs, which requires that it is possible to change separate settings for each citizen. Some of the system's customisable options should be adjustable in Giraf's apps through features, involving a citizen in the decision-making on a day-to-day basis. An example could be if a guardian believes a citizen is ready for a larger overview of the daily tasks, as the citizens become more adept at handling and keeping track of multiple tasks.

Chapter 0

Initial Sprint

The initial Sprint covers the introductory period for the project, where we got our first impressions of the current system and make a decision about the work methods.

0.1 Methodology

It was decided to use the scrum method that a previous years' groups have described. At the end of the Sprint, we will evaluate if this method is useful, for this semester of the multi-project.

0.1.1 Scrum

Scrum is a software development framework that facilitates agile software development; it structures a project over a number of short sessions—called *Sprints*—which typically last between one and two weeks each. Every Sprint starts with a *Sprint Meeting*; the purpose of which is to discuss and plan the work over the course of the Sprint.

Scrum facilitates the scheduling and management of meetings, which aims to share knowledge and to find out what the members of the team have done since the last Scrum meeting.

Backlog

During the project, all tasks get sorted into different backlogs. There are two kinds of backlogs: a *Project Backlog*, and a *Sprint Backlog*.

Product Backlog Keeps track of everything we need to do over the course of the project. Figure 1 shows an example of the product backlog.



Figure 1: Product Backlog for the project.

Sprint Backlog Functions similarly to the project backlog, but spans over a single Sprint instead. When work on a task has begun, it is moved to the list of active tasks, and once finished; it gets moved to the list of finished tasks.

0.1.2 Scrum of Scrums

The Scrum of Scrums is the act of managing a very large group of people. Scrum of Scrums takes representatives from each smaller Scrum team and holds Scrum meetings with those. These representatives then relay the information gained to their own respective teams.

Weekly Scrum Meetings Each week a short meeting is arranged so that each group knows how much the other groups have progressed. At these meetings, there are discussions regarding what groups are actively working on, have almost finished, and problems the groups might have stumbled upon throughout the week.

Only a single active member of each group participates in these meetings.

Sprint Meetings When a Sprint ends, all groups get together to discuss which new tasks need to be done, along with a retrospective on how the Sprint went. All members of the groups are invited to attend these meetings, to ensure everyone can contribute.

0.1.3 Customer Meetings & User Stories

Customer contact primarily happens through meetings with the customers of Giraf from one of the different institutions. During these meetings; prototypes, desired functionality and features are discussed, in an attempt to get the product close to the customers' expectations. Questions from the Giraf teams will get formulated in a non-technical manner, due to the assumption that the customers lack technical knowledge. It is, therefore, our job as product owners to be the mediator between the developers and customers.

The User Stories are made based on the information gathered from the meetings with personnel at the various centres for citizens with autism. These stories get sorted by priority and relayed to the rest of the Giraf project teams during the monthly Sprint meetings. We formulate the user stories in the form “As a [type of user] I would like [specification]” [2]. This way of formulating the requirements puts them in a clear language that strips away the unnecessary details of the interview. This should make it clear for those reading it what the customer expects from the system.

User Stories help to organise which features the end-users require, in the actual software; enabling the creation of tasks for the development teams through these stories. The User Stories for this semester can be found in Appendix A

0.1.4 Tools

When this project got entrusted to us from the previous year, there were several tools which had already been set up and used for the project.

Phabricator An online platform designed for use in managing larger software projects. The most important of these, are task management, peer-review, Git integration, and the built-in wiki[3].

Git The version-control for the source code of Giraf. It allows for version control of the code, while also allowing for code branching. This allows for many developers to work on many parts of a system at once[4].

Android Studio Google's official Integrated Development Environment (IDE) for Android development, which is based on JetBrains' *IntelliJ IDEA* IDE. It contains several useful features for development, like debugging and Gradle-based development, and unlike IDEA, it is specifically designed to handle Android development[5].

Gradle A build tool used for managing software packages in *Android Studio*. It is responsible for fetching code from remote servers and keeping it up to date[6].

Jenkins A tool which is used for continuous integration and is capable of building the project and applying specification of tests. Once a change in the master branch happens Jenkins attempts to build and deploy a Android Package Kit (APK) file if it fails an email gets sent to the responsible group[7].

Docker A platform for creating small self-contained “containers”. The purpose of which are to have consistent environments to deploy software as not to run the risk of code working on one computer but not another[8].

0.2 Responsibility

There are several areas of responsibility for this project, each dealing with a different part which needs a responsible group, with many tasks and areas, then having a

responsible group means you can get experts on the specific topics with which you need help.

Scrum Has the task of managing all Scrum related works and organise the communication between the different groups.

Product Owner Handles the customer contact and generating user stories based on meetings with the customers.

Launcher Ensures the launcher works in an acceptable manner.

Repository Manages the code repository.

REST Client Handles the structures the REpresentational State Transfer (REST) Application Programming Interface (API) for client use.

Infrastructure Handles the database setup and the Artifactory REST API.

REST Server structures the REST API for server-side usage.

Our Responsibility

Our group will focus on the responsibility and tasks for the product owner group. We have the responsibility of relaying the customers' demands to the other groups, leading to a greater understanding of the product, like the ones showing it to the customers. The customers will be contacted doing the development of Giraf to ensure the final product meets the expectations of the customer. In addition, we will focus on managing and updating the project's wiki, with guides, documentation, and other information relevant to the project.

0.3 Customer Contact

Initial customer contact was done in this Sprint, to improve the understanding of the previous project handed over to our teams, who are responsible for developing the software.

0.3.1 Preparation for the Meeting

As product owners, we are in charge of the customer contact; therefore we spent time preparing for the first meeting. This preparation required us to create and gather questions—regarding the project—from other groups working with us.

With little knowledge of how Giraf functions and the complete state of every application, we decided to ask the customers about their experience with the version they got the previous year. Examples of this, are which of the system’s applications they deem most important, and how they would like to use Giraf. Since we cannot assume the customers have the sufficient background for understanding the technical terminology, we opted for rephrasing the questions for the customers. We worked on improving the questions we created, as well as those received from the other groups, reformulating them into something that could be more easily understood by the customers. Whenever we could not avoid the technical terms, we tried to explain their meaning and how they affected the system, by providing examples.

These questions were made to find out what they wanted the most—that we realistically could provide—by the end of the semester. Because the meeting started with a presentation from Birken, we decided to ask the unanswered questions afterwards if any remained. The complete list of questions prepared and discussed during the meeting (see Appendix A.1).

0.3.2 Customer Meeting

Two representatives from *Birken*, part of *Centre for Autism* in Vodskov, North Jutland, came to the University and held a presentation regarding the general concerns for the system. This meeting is the first with customers, and all teams are currently gathering information about the Giraf project; therefore everyone assigned to the Giraf project and was present at the meeting.

The two representatives from *Birken*, gave a general explanation of the centre they worked at, explaining what difficulties the residents face when it comes to using soft-

ware. The purpose of this meeting is to give a better understanding of the users, and further the Giraf development towards people with ASD.

One crucial part was how literal some of the children might be, for instance, if an image displays a red apple, then they believe it must be a red apple, or telling them not to bite sometimes also translates to not chewing food. Their linguistic development is tied to physical interaction, meaning that having the residents look at a tablet screen would limit their ability to mimic the movements of the caretakers.

0.3.3 Reflection on the Meeting

The meeting was structured primarily around their presentation, asking them questions as they went along. In the open question session, some of the questions—from the other groups—had to be rephrased to get rid of technical terms—for the guardians to understand them correctly. An example of this could be a group asking “how far into the system, they would like function to be”. This got rephrased as “the number of taps they would be required to perform an action in the week planner”. One thing was made clear in the presentation: they were more concerned with getting a functional system, as opposed to getting more features. They also emphasised that we should use the term “citizens” instead of children, since they will still use the system when they grow up, at which point “children” becomes demeaning.

The system has to be compatible with the Treatment and Education of Autistic and Related Communication-Handicapped Children (TEACCH) system, as the entire centre is set up around using that. This system gives each citizen a schedule, which is maintained by the guardians.

These schedules contain pictograms to show what each task is about, along with a title. Each pictogram is required to be easily changeable, as each citizen has individual requirements.

0.4 User-stories

Each of the requirements *Birken* gave, got formulated into user stories. These user stories were prioritised based on their needs; our group attempted to evaluate their needs based on the information and observations we get during meetings. As the guardians primarily requested stability, those features will have the highest priority. We also rated user stories, which would result in a citizen being unable to handle the stimuli from the system highly. The additional features they required will have second

highest priority after functionality, as these features are necessary for their usage of the system. Any quality-of-life improvements will get a lower priority.

The customer meeting resulted in the following user stories found in Appendix B.1.

0.4.1 Reflection on the Sprint

During this Sprint, unforeseen problems occurred, which resulted in missed deadlines and left us with incomplete work. To solve this, we will have to be more focused on completing tasks and on catching up on the work left over, while preparing for small missed deadlines in the next Sprint. The Sprint after that, however, should not have any missed deadlines

An area to be improved from the initial Sprint was the planning for the customer meetings. The questions for the customer meeting were made only by us, where the discussions for the questions was not done in person but on a text communication tool. This resulted in that not everyone had seen the questions, while also resulting in less feedback for the questions. We could improve upon this problem by planning a structure for the meetings and sticking to the planned structure.

Chapter 1

First Sprint

In the first Sprint—after the initial Sprint where we had researched the entrusted solution—we will work with the user-stories from the previous Sprint, and try to evaluate the user-stories with the other groups, who are working on Giraf. In this Sprint we will also try to set up a new customer meeting and develop new user-stories that will be used in the next Sprint.

1.1 Objective

The primary goal of this Sprint is to arrange another customer meeting, this time with the other institutions (See Introduction). The questions asked during the meeting will be formulated during the Sprint. Their experience with the app gets used to form the new user stories. The other groups primarily work with familiarising themselves with the system’s code, along with fixing its stability issues.

The secondary objective is to update *Phabricator*. This means all parts of the wiki will be verified and updated. A good amount of the information is still up to date, meaning the task is not as daunting as it first may seem.

Additionally, we have decided to focus on developing a “*small package solution*”, which focuses on actually getting a small subset of the system to work and be stable, rather than piling on more and more new features.

Any excess time will be spent on formulating the documentation, and developing user guides to help the customers. We will also allocate some time to familiarise ourselves with how to use the system.

1.2 Status

When the project got handed down to us, there were many features already implemented. However, many of the features have major bugs, which cause multiple crashes. The project also contains some design choices which result in the system being unresponsive or slow.

Examples of these design choices are:

- How the app downloads the entire server-side database on start-up.
- How the pictogram copying-system functions regarding copying different days into different weeks, which required a series of steps.
- How crowded the User Interface (UI) of the system is with buttons, which make it difficult to get a clear overview of the things.

These are some of the choices which can be made to speed up the internal tasks for the system. There are also undesired features left in the system, such as a login screen using Quick Response (QR) codes.

Some of the system's problems include:

- The week planner does not scale the UI buttons when the tablet get turned 90°.
- The games are dependent on the Giraf environment, making it necessary to run them through Giraf.
- The week planner does not have any means of adding new pictograms.
- The entire system has general performance issues, taking up to several seconds from tapping an icon to updating the contents on the screen.
- The week planner lacks several convenience features, such as template weeks.

In the initial Sprint, the wiki was reviewed to get an overview of which information needed to be updated and which information is still useful, these will then be updated during this Sprint.

1.3 User Stories

1.3.1 Stories From the Previous Sprint

We discussed the user stories gathered from the initial Sprint between the project groups, where the six highest prioritised stories out of fifteen, have been selected, and are formulated as follows:

1. As a citizen who uses the system, I want the system to be as stable as possible, as I am not able to handle crashes, due to my illness.
2. As a guardian, I want the ability to replace pictures and icons for the children in the system.
3. As a guardian, it is very important to be able to create a weekly schedule and manage the tasks the children can see at any given time.
4. As a user of the system, the response time is an important factor.
5. As a guardian, I would like the ability to access the schedules of the kids' tablets remotely.
6. As a user of the system, I'd like the interface to be simple, things should never be more than 2-3 taps away.

We prioritise the user stories and list them in order: 1 has the highest priority, while 6 has the lowest. The groups have decided that it is best to focus on software stability and to get everything to work, rather than adding additional features. The users of the system had also complained about the speed of the system, making that a priority. For these reasons, the six stories above were chosen as the highest priority tasks to be worked on within the Sprint. The complete list of user stories for the Sprint can be found in Section B.1.

1.3.2 Stories for the Next Sprint

Due to issues organising a meeting (see section 1.4) no additional user stories have been made for the next Sprint.

1.4 Customer Contact

During this Sprint, we wanted to gather information about the other institutions' thoughts on Giraf; where we wanted to talk about what they wanted, and their thoughts on the decided semester goal (see section 1.1).

We wrote an email, which would be sent to the contacts of each institution, asking for a meeting with all of them to address the general questions we had. The meeting got planned for the following week. However, as none of the customers responded to our mail, we tried to call the different contacts whom we had a phone number of and managed to get a hold of Egebakken. We arranged a meeting with them the following week and expected to get a hold of the other contacts in a similar way since we did not get any response on email. However, we learned that the customers we didn't get in contact with on the phone, actually showed up for the meeting as written in the mail; but since nobody had responded to our mail, we made other plans, and were not there to greet them.

From this experience, we have learned the importance of being precise in terms of communication, asking for a response on emails to verify details. Therefore, in the following emails sent to customers, we are more specific about the exact time and place. We will also formulate the emails into a question to request a reply if they read it; otherwise, we will contact them again. The future emails include contact information from at least two members of the group, the time and place of next meeting, and to ask if the time suits them.

The meeting with *Egebakken* is planned to happen during the second Sprint (see chapter 2), and the meeting happens at their facility.

1.5 Conclusion

Since the primary objective for our group on this Sprint, was to hold a meeting with the customer—which was unsuccessful—the objective was not fulfilled. Precautions have also been made to ensure proper contact with customers to ensure meetings in the next Sprint.

1.5.1 Reflection

As described in section 0.3, the initial attempt at contacting the customer was not successful, as no guardians replied to our email. However, despite not responding, the

institution *Enterne* still arrived. We have reasons to believe that this happened due to the way we formulated the email; we described in a way which made them think the meeting would happen whether or not they arrived. Future emails will have to be more explicit in asking for a response, although we believe that organising these meetings by calling them is preferable for now.

During the work process, we would sometimes forget to update the finished tasks set up in Phabricator. This is something several groups did, including our own. As our tasks are somewhat unsuited for reporting back, due to them not being critical for the system to function, it was not as severe of a problem. It was, however, still an area of improvement, not only for our group but also for several others. To improve, we will have to be more rigorous in the usage of Phabricator.

Other minor areas of improvement on the delegation of work. This includes who goes to which meeting, and who does what task. To improve upon this area, we will assign group members to the task at least one day before.

All in all, the areas of improvement for this Sprint is primarily about communication. Meaning we will work on properly communicating with other groups, and accurately expressing our intentions of work.

Chapter 2

Second Sprint

2.1 Purpose of the Sprint

The purpose of the Sprint is to address the concerns raised in the previous one. The groups decided it is important to increase stability of the libraries used, to increase the accessibility and usability of the applications, and to focus on fulfilling the first, fourth, and sixth user story (see section ??). For the groups focusing on the server and client libraries, the task is to begin the implementation of the features for the second, third, and fourth user story.

Our job is to meet the people at *Egebakken* and establish an interview with them. This interview has the purpose of gathering more information about what they expect from the system—such as the week planner, launcher, and the login system—concerning functionality so that we could generate more user stories for the developers to look see.

2.2 Status

A decent amount of stability upgrades is now in the system. Most of the crashes have been found and fixed, to the point of being able to use the system in an intended manner without experiencing any crashes.

Due to the previous problems pertaining concerning achieving a customer meeting, the user stories we wrote from the interview with *Birken* are still the ones used for this Sprint.

In the First Sprint, a calendar for the project was created, and uploaded to the wiki for the Scrum and Product Owner groups to use relating to the project work; primarily, meetings and deadlines.

2.3 Customer Contact

This Sprint we meet with the institution *Egebakken*, as discussed in section 1.4. We held the meeting at their place, as opposed to meeting at the University.

2.3.1 Before the Meeting

After getting a further look at Giraf—and our customer meeting with Birken—we found that the focus should be on the stability of small package (see section 1.1). Therefore, the primary purpose of this meeting will be to see how they want to use the application, as well as finding potential areas of improvement. This entails the difficulties they might have when they try to understand the program, if the program will fulfil the citizen's requirements, or if there are ease of use improvements to be made. We try to structure our questions for the customers, structuring them in the following categories: *Launcher*, *Login System*, *Week Planner*, and *Other*. This resulted in the questions which—along with the customer's answers—are in appendix A.2.

We plan to record the meeting. Because we are using the question structure described above, it is easier to find accurate information from meetings, since we do not have to listen through the whole recording to find the information on a topic. We also created a list of task to be handled during the meeting and assigned the responsibility of each of them to a group member. We came up with the following four areas:

- Asking the prepared questions and keeping the conversation on track.
- Showing the prototype on the tablets, with new features from the other groups.
- Ensuring we had the recording equipment in order and double checking that they answered all questions.
- Taking notes during the meeting, involving writing down what has been said, with emphasis on what is important. When we show the prototype, we note down if and where the user has difficulties.

The purpose of creating these tasks was to get a better plan of what happens during the meeting, and ensuring that we had someone assigned to each area.

2.3.2 Customer Meeting

We started the meeting by telling them what this year's Giraf-students have decided to focus on and why we chose this, as discussed in section 1.1. The customers were happy about this decision and said they just wanted something finished; leaving potential expansions for afterwards.

We showed the Giraf application at its current revision, and asked them what they liked and disliked about it. First, we showed them the Login menu, and they immediately told us that the QR Code login option was very annoying. They were unsure as to how they wanted it to be instead. The Launcher was "okay", but they wanted it to be more clear when they were logged in on a device through the use of pictures. The Week Planner was a little confusing, but that is mainly because it was very unstable, our central question here was whether the system should auto-rotate with the device, or only rotate when they tap an icon. The customers like how the planner turns a weekly view into a daily view when they rotate the tablet. However, the icons must not be half-visible on the screen when there are too many since the citizens are unable to handle partial pictures as their task. They said auto-rotate was intuitive enough, and there were no general comments on the UI itself. We then asked about the tutorial built into the Week Planner, they said it was helpful, but they would like the tutorial to be split into multiple videos, showing how to use the individual features.

During the questionnaire, we received general information about how their institution functioned. During the questionnaire they mentioned that it was important that we called the standard user a "citizen" and not "child". With the reasoning that Giraf will follow them throughout their life, and they should not think it is for children only. The replacement for the QR Code was also an important discussion, most of the citizens would not be able to log in using a password since many of them lack the linguistic skills required to interpret written text, let alone the ability to input it. The Guardians wanted an option of a hidden log-in for Guardians and a way to get past the log-in screen on the citizens' devices. One Guardian made the suggestion of holding the menu button for an extended period, to open a hidden login screen for Guardians, else there should just be pictures of the citizens on the institution. They would also like a gray-scale option, as some of the citizens cannot handle too many different colours. These changes should be synchronised to either individual or specific groups of citizens, depending on how the institution organises them.

The week planner needed a way for the Guardians to add their own pictograms. Several ease-of-use suggestions were also approved, such as copying days and weeks, and adding templates.

When the system would crash, it would need to jump back to the last stable point,

as opposed to closing the entire program. It was also required for it not to show any error message, as the citizens could not handle these.

Lastly, we took a picture of one of their week plans currently in use, and how they set it up. This should help us make the system feel familiar to the guardians. However, we lost the picture, so we can only explain how it looked. The week plan shows what they have planned every day, with Velcro Tape on the backside of the pictures, which allowed the guardians to make sudden changes. There are no time stamps on their plan; when asked, they told us that the plan works as a buffer, sometimes the citizens need a break or are unfocused. Then this allows them to handle that without moving tasks, which would confuse the citizens.

2.3.3 Reflection on the Meeting

We included only the most significant problems above, the full list user stories derived from the meeting can be found in the appendix B.

When we created the user stories after the meeting, we also discovered that we had not considered the role of a parent in the system, as they would like to check in on their offspring, we made the following questions.

- What should parents be able to see?
- Can parents interact with the system?
- Should parents have the same options as a guardian for their citizen?
- Should parents use the same schedule or have their own separate system for home?

However, before asking any of these questions, it would be necessary to consult the other groups to figure out how to best implement it in the system.

Structuring the questions into categories based on parts of the Giraf application should have been done more strictly. During the meeting, the discussion often ran further than intended, while the discussion often ignored the order of the questions. To prevent this, we should try and be more strict about taking one question at a time in the designated order. It could also be helpful to go through the list of questions with the customer before discussing them.

2.4 User Stories

2.4.1 Usage of Previous Stories

Due to there not being a meeting in the first Sprint, we made no additional user stories.

2.4.2 Stories for the Next Sprint

The user stories will primarily focus on the additional functionality which *Egebakken* required, meaning they get the highest priority. There is also a lot of ease-of-use features which the meeting gave rise to; however, these will get a lower priority than the functionality.

With these user stories, we have tried to structure the user stories differently. Due to how each group has a different area of responsibility, we have categorised the user stories based on what part of the application to which they belong. The reason for this is that it should make it easier to delegate the user stories to each different group, as each user story is for specific parts of the system.

The full list of created user stories is in B.2.

2.5 Conclusion

In this Sprint, we have held the intended meeting with *Egebakken*, where we have learned about some of the system requirements relating to their specific needs. This interview has also made us aware of other issues, which we will prepare to research in the next Sprint.

In the Sprint meeting, it has been decided to also include code review in the work process; so as to better let the teams understand each other's work, and get some additional validation on the code. However, we also decided that this was only meant for the other code-responsible groups, meaning this is not one of our tasks; however, we still have the option to do so, which means we can take on any code review tasks that time allows.

2.6 Reflection

Holding the meeting at *Egebakken* has seemed to be a positive experience, and is something we should try again. The planned structure of the meeting is very strict; however, the manner of conduction, was not. The questions would often veer off the outlined order, resulting in more work sorting the notes than necessary. However, it is only a minor difference in strictness, making it unnecessary to make changes in the structure; although, we should still be careful in case the difference becomes any larger.

Chapter 3

Third Sprint

3.1 Purpose

In collaboration with the other groups, we decided that for this Sprint, we have to focus on creating stability, accessibility, and usability of the current solutions, and expand on some user-stories. The product development will also focus on implementing functionality for the user-stories on the server.

Our plans for this Sprint is to arrange a new interview, this time with *Enterne*. The interview will mainly focus on validating the prioritisation of our user stories in section 2.4, while also expanding with some new user stories, including those which are considered parental involvement in the system. Furthermore, we plan on arranging an interview with *Birken*, that will focus on getting the UI tested with the social education workers, to see if the solution lives up to the customer's expectations.

3.2 Status

Some additional features have been finished. The Graphical User Interface (GUI) has been partial redesigned. Some new features—such as copying—have begun development.

The other groups also fixed a few major issues, an example of such an error was that one could get expired login-sessions back by changing the system time, thus creating a serious security breach. Additionally, they also fixed some dependency issues.

There were also several improvements to the back-end. Critical regions—which could

cause performance issues—were identified. The controllers for the REST API are also almost finished.

Phriction received updates containing a guide for creating tasks, the tasks created will be used to track development and potential problems. The guide will help next semester students who will be taking over the Giraf project, to figure out how to create the tasks for development. The current scrum group made most of the guide, who handed it to us with the purpose of uploading it to the *Phriction Wiki*.

In the second Sprint, some information was found, which was last updated two-to-four years ago. These were found and then split into tasks where each group worked on those relevant to their experience. Furthermore, the list of Sprints and customer meetings received an update for the semester.

3.3 Customer Contact

During this Sprint we have two customer meetings, the first one early in the Sprint with *Enterne*, the second one at the end of the Sprint with *Birken*, both meetings happened at their places.

3.3.1 Before the Meeting with Enterne

This meeting is our first with *Enterne*, which means that we want their view on many of the questions we had for *Egebakken*, as well as explaining the focus area of the entire development team. *Enterne* is a residential institution, and the citizens living there are adults; therefore some of the questions have been updated to suit them better. Due to the difference in their citizens, there is a difference in priorities. *Enterne*'s citizens are mainly young adults who live there, while *Egebakken*'s citizens are children who are there mostly during working hours. We made minor changes to some of the questions, and a few others added, we planned on handling the changes during the meeting, by the group member responsible for the conversation.

Before starting, we will go through the list of questions with the customer; this will let the customer know what we are going to talk about and hopefully keep the meeting on track. When the conversation moves to another question further down the list, we will stop that conversation, and return to the planned order of questions. The list of questions prepared for this meeting are in Appendix A.3.

3.3.2 Enterne Customer Meeting

We arrived at the institution and saw one of the halls their citizens lived in were split into living areas with four citizens in each. They told us that their apartments are private and that guardians do not enter without permission. They held the meeting in their common room, which is their location for evening activities such as movie night. The walls were very bare and white with important minimalist information—such as a schedule—and a huge clock they could read from the back of the room. They told us that the design choices were made to make the citizens feel more at ease in the room since many colours and posters—usually found in an institution’s common room—would confuse or distract them.

Once again we started as planned with a conversation about our goals for the Giraf project this year. The customer has been part of this project for many years and told her colleagues about it. The customer expressed that she would like to get something to show and start using to reflect the time spent on the project.

We then moved on to the prototype, but we forgot our tablet and therefore used theirs. They complained about the QR Code log-in and were happy to hear that we decided to get rid of it. Since we could not show the newest changes but only talk about what the final product will look like, we didn’t get much feedback and then moved onto the questionnaire.

Through the questions, we learned that the guardians at *Enterne* are more careful when handling the citizens. The mood of the citizens is necessary to monitor and share with the other guardians if a citizen is in a bad mood a guardian should not go into their room smiling and talking. This directly results in one significant difference, guardians need a fast way of either sharing knowledge, changing, or fixing the schedules accordingly. If the citizen highly disagrees with—or is in any way upset by—the plan, they can be prone to violence. Both children and young adults with ASD tend to react in similar ways, but getting hit by a child hurts much less than getting hit by an adult.

Parental involvement is not a priority for *Enterne*. The guardians at *Enterne* work on making the citizens more independent, which is something they believe that major parental involvement would impede.

A feature which they liked at *Egebakken* had almost no priority at *Enterne*, which was the ability to turn the system UI grey-scale. Citizens at *Enterne* are more capable when it comes to processing many colours, making grey-scale unnecessary.

A new feature they wanted was a different way of representing time. Some of the citizens require a graphical way of representing it, currently *Enterne* used a solution

called *Time Timers*[®] for this. A *Time Timer*[®] is a clock which shows time remaining in red. The hand on the clock colours the red area white as time passes to give a visual representation of how much time has elapsed[9].

There were, a lot of features which they considered to be good. Copying was one of them, although they wanted it to make sure the user would understand what they were copying over and if they were overwriting. They liked the print feature and the solution with error handling where the application goes back to the last stable position.

They requested a feature to easily share the mood of a citizen between guardians, like a note only visible to them, to get a warning before getting into a room with a citizen in a bad mood. If they needed to do something on a tablet in a citizen's apartment, it should happen quickly without problems.

At the end of the meeting, we asked if we could see an example of their current physical system and they showed us one for a citizen, see figure 3.1

	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
8.00	Medicin						
→ 10.00	Morgenmad og brusebad						
.00 til 10.30	Bage boller	Ramasjang	Ramasjang				
→ Aktivitet	Pause						
	Aktivitet: 10.45 svømming; Øster Vrå svømmehal	Aktivitet: Gåtur i skoven, Hente blad i byen, handle	Aktivitet: Gåtur i skoven, Hente blad i byen, handle	Aktivitet -Ridning: Skal ride kl. 10.30, Vestermølle Langholt	Aktivitet: Gåtur i skoven, Hente blad i byen, handle	Aktivitet: Gåtur i skoven, Hente blad i byen, handle	Aktivitet: Gåtur i skoven, Hente blad i byen, handle
ter endt aktivitet	Pause						
→ 14.00	Frokost						
→ 15.00	Pause						
→ 16.00	Ordne vasketøj						
→ Ca. 17.30	Pause						
Ca. 17.30	Aftensmad						
→ 19.00	Pause						
19.00	Røre dej til boller						
→ 19.20	Pause						
→	Nattøj						
20.00	Medicin						
→ 20.30	Tilbydes the, knækbrød eller risikiks i stuen	Slik og Cola	Tilbydes the, knækbrød eller risikiks i stuen				
→ 21.00	Pause						
21.00	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog	Afslapning i sengen under kugledyne. Med et blad eller en bog

Figure 3.1: How Enterne used their physical week planner.

On their week plan, we can see that each day is very similar with many repetitions and not much space for quick changes. At *Enterne*, the citizens are older and more capable when it comes to the week planner, so their day does not need to have the same amount of information. However, when they have breakfast, they get a list of choices they can combine depending on what is available. The colour scheme is the same as they use on the other institutions and the one designed in Giraf.

3.3.3 Reflection on the Meeting with Enterne

This was our second meeting at an institution, and we have noticed that even though they are working on the same citizens—albeit at different ages—their requirements and needs for the citizens at the various institutions differs more than anticipated. This means we have more to consider when making a design choice for the customers, which could then result in two solutions, potentially making us work on features that some of the institutions might not even want. Giraf is designed for the citizens to use throughout their lives, which means as they develop their abilities, Giraf should be able to reflect this.

After this meeting we arranged a meeting with *Birken*, to get a point-of-view from the final customer.

3.3.4 Before the Meeting with Birken

This is the second meeting with *Birken*, although it is the first time we arrive at their place. We did the meeting with *Birken* with the focus on verifying the new features within the system, which entail how they use the new features, and how much they like the new features and potential improvements to them. The week planner is the only part of these implementations; therefore that is what we will show them. The questions are in A.4.

We will first ask questions about the individual applications and then ask about parental involvement. The order of the applications is determined by order-of-usage, as when a user starts Giraf, they move from the launcher to the login system, onto the week planner.

We will use the same delegation of work as we have done in both *Egebakken* 2.3 and *Enterne* 3.3.

3.3.5 Customer Meeting with Birken

At the meeting, we talked with a single representative of *Birken*, Kristine.

We first showed her the current system. We started by going over the additional features put into the upper right menu button. Kristine liked these features; however, she said the icon for cancellation was covering too much of the pictograms. Midway through showing these features, the system crashed.

After showing these additional features, we started trying to add pictograms. However, at this point, a fatal error occurred in the application.

After demonstrating the features, we began the questionnaire. Many of the questions were there so as to verify that we were progressing in the right direction. This meant many of the questions were the same as in previous meetings. However, asking the questions still resulted in new information, as we saw just how different *Birken*'s requirements are, compared to the other institutions.

The main difference between *Birken* and the other institutions, was that their citizens did not have their own tablet. This means that citizens should be able to log in from any tablet. They also had some minor differences, such as not requiring gray-scale and requiring pictograms.

There were also some additional requirements. At *Birken* some citizens have a small note on their week plan detailing special requirements, some of which do not relate to the week plan at all. The rotation of the week planner also needed to be locked for some citizens, limiting how many pictograms they could see. We base these requirements on how *Birken* had their week plans.



Figure 3.2: An example of a week plan from *Birken*. Note the small notice in the top of the schedule.

3.3.6 Reflection on the Meeting with Birken

The crashes reflect poorly on our work because the system was more unstable than before. It shows that we did not test the system properly before using it. In any future projects where we contact the customer, we should be more careful demonstrating new features to ensure that crashes will not happen again.

Despite not expecting additional requirements for the system, we still found some. They would like an option of adding a note to every citizen only visible to guardians, such as “Bathroom before fruit and again before TAXI” (see figure 3.2). This is not a problem, but it still is a good idea to remember, so that we can ensure our work is flexible enough to handle smaller unsuspected additions.

3.4 User Stories

3.4.1 Usage of Previous Stories

During the sprint meeting the user stories were presented, where all groups discussed tasks based on the user stories.

The ones with the highest priority are the following:

- General
 - 1. As a user of the system, I would like the system to move back to the nearest stable state when the app crashes.
 - 2. As a guardian, I would like the option to toggle between grey-scale and fully coloured in the system for a citizen.
- Launcher
 - 1. As a guardian I would like the login button to let a citizen tap to log in as a citizen and hold to log in as a guardian, only providing login information at that point.
- Login System
 - 1. As a guardian, I would like not to have QR codes for login, but a regular password instead.

2. As a guardian, I would like the ability to set a password for specific users of the system.
3. As a guardian, I would like the system to automatically log me out of my account after a set time limit (approx. 20 min).

- Week Planner

1. As a guardian, I would like to add my own pictures as an addition to pictograms, as the citizen might need more specification than what a pictogram allows.
2. As a guardian I would like to copy an entire day and paste it to one of the other days in the week and modify it, to ease the creation of schedules.
3. As a guardian, I would like to print a list of an entire day for a citizen.
4. As a guardian, I would like to have a pre-made template schedule I can modify or not use at my own accord.
5. As a citizen, I need the option of having timestamps at/by the pictograms.
6. As a guardian, I would like the ability to mark multiple elements from several days and copy them to a single day.
7. As a guardian, I would like to be warned if I am about to overwrite a complete day by using copy/paste.

The categorisation is done to ease the task of delegating work between groups.

The stories from the previous Sprints still has a higher priority than these. The priorities are based on essential functionality, while quality-of-life features get a lower priority. The necessary features base themselves on requirements the citizens have, which we did not consider. For instance, we did not expect colours to be an issue, but for some citizens, colours give an information overload. The quality-of-life features are rated based on how much the guardians want each feature (see section 1.3 for how we determine this).

3.4.2 Stories for the Next Sprint

We made only a few user stories, as many of the requested features are the same. However, many of these have their priority changed, so as to also fit with *Enternes* perspective. The priority change still uses the same methodology as the last Sprint; essential features first, ease-of-use features second.

We also planned a second meeting, one with the purpose of verifying the system. This will be with *Birken* at their institution. Here the new features of the system would be showcased and verified.

The user stories will not follow the structure used for the user stories made in 2.4, due to the dislike of the structure expressed by the other groups.

Both lists of created user stories are in B.3 and B.4.

3.5 Conclusion

In this Sprint a customer meeting was held. The meeting helped ensure our user stories had the proper prioritisation, while also helping us create more user stories.

We held another meeting at *Birken* with the purpose of verifying the system. This meeting, however, took place right at the end of the Sprint, so any work done on it will be written into the next chapter.

3.6 Reflection

Just as in the previous Sprint, the meeting structure ended up being looser than expected. However, it was much more loose than the previous Sprint, therefore requiring some work. Later questions should, therefore, try to be looser; attempting to get into some of the points instead of being a single list of questions.

The categorised way of having user stories makes them harder to delegate appropriately. This categorisation was originally meant to make it easier to delegate, as each category would correspond to only a few groups' work. However, the categorisation instead made it harder to properly prioritise each story, thus making it more difficult to delegate the work. Because of this, we scrapped that method of organising user stories.

Chapter 4

Fourth Sprint

4.1 Purpose

As this is the last Sprint, no one will work on any new features.

Instead, work will instead focus on finishing and implementing all features currently in progress. The largest required implementation will be swapping the previous REST library out with a new one. All groups will work on implementing the new REST API, including both the scrum group and our group. Therefore there is a code section in this Sprint.

Every group will work on preparing the project for delivery to other semesters' groups. This includes creating additional documentation for the project, which consists of updating older parts of the documentation. The new system has had so many changes that it will be necessary to discard many of the old code sections.

4.2 Status

This section will describe the status of the product and the updates made to the wiki through the third and fourth Sprint, as the fourth Sprint is the final one.

In the third Sprint, we researched whether or not it was possible to restructure the wiki to make it easier to use. After researching the Phabricator documentation, we came to the conclusion that the requested structure would make it harder to understand the wiki. The request was for the document hierarchy to be structured with the user guides at the bottom instead of mixed in with the rest of the wiki.

The server that the project ran on, crashed late in the third Sprint. The problem occurred because the server ran out of memory. The university did not supply any additional memory to the server.

4.3 Customer Contact

As this is the last Sprint, we will not hold a meeting with the customer for two reasons. We generate user stories after these meetings for the next Sprint and use them to create tasks for that Sprint. The current coding tasks are about implementing the REST API; this means there is nothing new for the customers to try. We will, however, inform the customers of the current progress of the system, thank them for their help and tell them that we hope they would be helping again next year. To do this, we sent a short email to all the customers involved with Giraf.

4.4 User Stories

In this Sprint we do not have excessive time for any new features to be implemented and will focus on implementing current features. Therefore the stories are not discussed with the other groups, as there is a collective agreement there should be no time to work on them. Despite that, we still made them so as to benefit the next semesters' project. We will only show the new user stories, as there have been no major changes to the user stories themselves, nor their prioritisation.

- As a guardian, I want it to be clear if I'm logged in on a tablet.
- As a guardian, I want it to be clear if I have logged out successfully from a tablet.
- As a guardian, I need the option to have special individual notes on each citizens' schedule.
- As a guardian, I need to be able to lock the rotation of the week planner for specific citizens.
- As a guardian, I want to decide on which side of a pictogram the text is when making the Portable Document Format (PDF).

The guardians have some troubles understanding whether or not they are logged in. This is rated a serious problem, as it might result in security problems.

Individual notes became a new requirement, as often each citizen has unique requirements which the guardians write on their week plan. Due to how all the functionality of their week plans needs to be in Giraf, it must include these notes.

Some of the citizens also require that they should only see one day at a time. The easiest way to do this is to lock the rotation of the tablet so that they can only access the day planner.

The print layout is different than what they used at *Birken*, and since the others might do it differently, they need to modify it to suit their needs.

Every group will work on preparing the project for delivery to other semesters' groups. This includes creating additional documentation for the project, which consists of updating older parts of the documentation. The new system has had so many changes that it will be necessary to discard many of the old sections.

4.5 Code

In this Sprint we have been assigned a coding task. The part of Giraf which handles pictogram-searching needs the reconfigured REST library implemented. This will make it support asynchronous searching, meaning the application will be less likely to freeze when searching for pictograms. It will also result in removing some functionality, as the pictograms no longer support tags.

This required an update of all libraries, therefore the task was split into many smaller ones and assigned to other groups. We were assigned this task, as it does not require much knowledge of the structure of the system compared to the other tasks. We will also collaborate with SW610F17 as they are the ones who are going to use the code, adding any additional missing details to it.

The primary change was to create the asynchronous search of pictograms. This was done by using the *Volley* library, commonly used for fetching data from a server. *Volley* contains several response methods—methods which will be called at specific points in the data fetch—the response we will override will load the pictograms into the user interface.

There were also some minor changes done, which included importing libraries and deleting unused methods.

4.6 Conclusion

The focus of this Sprint was the documentation. This resulted in two guides which will help new developers of this system. The first guide is a tutorial for the Guardians on how to use the system, while the second lists good practices for being the product owners of the Giraf project.

The customers were also informed of the status of the system. The communication was not done in person, but rather over email. All parties involved in Giraf project received the mail, as opposed to sending it only to those involved with the project.

Lastly, there had been some work done on the code. However, there were several difficulties with implementing the REST library. These difficulties ended up complicating the implementation of other features. However, it has still been improved, making it easier for other people to continue development on it.

4.7 Reflection

The REST API's model layer does not match with the Kubernet's model—Used for automatically building large projects[10]—, resulting in the calls from the REST API to Kubernet using different naming conventions; this can be solved by a revamp of the model layer. This happened due to erroneous communication between the teams working on the model layers and mismatches between the two layers results in unhandled exceptions. Due to our main topic being communication and administration of users, we should not have taken the task of changing code structure and instead worked on tasks such as, code review or error fixing.

Chapter 5

Project Status

In this chapter, a final status of the project will be described and include the topics of completed tasks from the fourth Sprint.

In the fourth Sprint, it was decided to restructure the wiki, which we did in collaboration with the scrum group. We did this to increase the ease of accessing knowledge for new developers on the Giraf project, and to reduce the time needed to understand the structure of the system. This is discussed in chapter 6

Furthermore, we went through all meeting notes and other material we got from all the customer meetings this semester, with the purpose of generating a guide for how a user uses the system. However, due to the solution not being deployable at the time, we could only produce a guide for how a guardian uses the system (seen in Appendix C.1). This guide has been uploaded to the wiki for others to find it.

We also developed a guide detailing our suggestions for how to handle being the product owners (see Appendix C.2) for a multi-project. We did this as the information in the previous groups' guide did not align with our experiences. The guide is meant to give introductory tips and tricks to the group entrusted to be the product owner for upcoming years. The guide will also cover how to formulate contact with the customers, how to hold the customer meetings, and how to generate the user stories.

Chapter 6

Wiki Page

This chapter will describe a restructuring of the internal source of information regarding the Giraf development and has been created in collaboration with group SW611F17.

6.1 Wiki Structure

6.1.1 Introduction

The wiki—called “Phriction” in Phabricator—provides a place for storing information, which is deemed necessary to share across departments and semesters. The wiki provides easy access to shared information for all developers throughout the project and can be adapted as needed. In its current state, it is adapted to be useful for our semester. However, we believe that the current format is not as useful for the developers that will have the project entrusted in the future. We would, therefore, like to change it.

To make this change, the product owner team and the Scrum team have started collaborating on re-structuring the wiki. This collaboration seems suitable, as both groups are involved in the project management, and therefore have some insight into the communication and the flow of information across the teams. The product owners have previously been involved in the adaptation and maintenance of the wiki, making them the experts on the field. The Scrum team has been involved with management, in both task-generation and project-administration via the Phabricator tool. As such, we deem that the collaboration between our groups is natural on this task.

We plan on performing our collaboration by dedicating one member of each team to this task, as it is not very large in scale and ought to not require more manpower than this. We will first discuss the issues of the wiki, by comparing what we need from it at the beginning of the project with what is currently in it. This way, we can hopefully reach a version of the wiki, which will be beneficial to the next semester's project. Once a plan is established for re-developing the wiki, we will bring it up on the following Scrum meeting to allow the developers to give feedback on the plan and bring new suggestions forward. We cannot be certain that we have covered the needs of all developers, that is why we consider it important to ask their opinion before starting the actual development. After preliminary brainstorming and reviewing with the other teams, we will work on implementing the new structure of the wiki to fit the vision.

6.1.2 Wiki Page Setup

- Table of contents
 - A general table of contents for the wiki page.
- Introduction to Giraf
- Guides
 - Startup Guide

Project Management

- * How to Scrum
- * Task Creation
- * Phabricator guide
- * Google play-store

Development

- * General Development Guide
- * Graddle Guide
- * End-point Naming Convention
- * Library Guide
- * Android Development Guide

Server-side

- * Server Migration

- * MySQL Guide
- * URL Guide
- * ASP.net Exceptions
- * REST Guide
- * Kubernet Guide
- * Docker Guide
- * Server Connection Guide
- * Server Back-up Guide

- Goals of Giraf

This will describe the goals for the Giraf project.

- Terminology

This will describe the terminology used at the institutions that is collaborating with the Giraf project.

6.1.3 Structure Argumentation

The introduction should serve the future developers to understand the Giraf project. Not in technical terms but in the conceptual construction of it, such that they understand what role Gradle plays in the development, without having to read the technical specifications for Gradle. We have chosen to provide such an introduction based on our own experiences at the start of this project, where it was difficult to establish a good overview of the project as a whole, beyond the general goal of the product. We expect that a well-written introduction will help future developers quickly ascertain which department they might be most suited to begin within and, consequently, which guides they might benefit from reading the most.

We have experienced that a good split between the groups' work areas, is into three distinct departments; one for project management, one for client development and one for server development. The experience of splitting into departments has given a clear structure of which areas of information were lacking, and thus we have been able to produce guides to help the transition between the past and future semesters. These departments ought to aid the development teams in quickly accessing the information required for their area of interest. We hope that the future semesters will read the guides, as those should be very specific and informative.

When reading this wiki, the developers will be presented with the goal of the Giraf project. However, we believe that it might be worthwhile to dedicate some space on

the front page to keep this goal static, such that they can refer to it whenever they need to include it in their reports.

Some special terminology has been used in the Giraf project and to provide some consistency across the semesters; we have chosen to include a table listing and explaining these terms for the benefit of the future developers.

The next section is a guide that is expected to be among the first things the new developers for Giraf will be reading, which will result in a change of reading perspective as the target for the text change.

6.2 Start-Up Guide

Welcome to the Giraf project. We are hereby entrusting the project to you, and have produced a few guides to help you get started. By the time you read this, we are not expecting you to have chosen an area of focus.

We strongly recommend to use the agile development method Scrum as the primary framework for the project, as we—in our project—have had good experiences using it. We recommend that you form teams in the size of the project groups you have formed this semester, to divide the workload in a manageable way.

For each team, we recommend to have one area of focus (client, server, management, etc.) but assist groups working in other areas whenever able. The Scrum framework is concerned with frequent meetings; we also recommend that you set up a shared *Google Calendar* to administrate these meetings. Your supervisors and Ulrik might also be interested in having such a calendar available. We have followed the following structure:

Administration

- Scrum Masters
- Product Owners & Wiki-Update

Development

- Launcher Application
- Week-Planner Application

Server

- REST Client

- Server Infrastructure
- REST Server

The teams have worked primarily with the above topics, but most teams have had other tasks assigned to them, that are not specifically within their own area of focus. You should not expect to be solely dedicated to one area throughout the project.

The following list of guides will be presented by the associated department, in the order that we would recommend that you read it to understand the Giraf project properly. One guide is not associated with a specific department; we recommend that all read this guide (hyperlink guardian guide) to gain an understanding of how the product is being used by our clients. It is relevant for all departments to understand this.

Project Management

- How to Scrum
- Phabricator guide
- Task Creation
- Google App-store

Development

- General Development Guide
- Library Guide
- Android Development Guide
- Graddle Guide
- End-point Naming Convention

Server-side

- REST Guide
- Kubernet Guide
- Docker Guide
- ASP.net Exceptions
- URL Guide
- Server Connection Guide
- MySQL Guide
- Server Back-up Guide
- Server Migration

Chapter 7

Collaboration: Between Product Owner and Week Planner

This chapter is written collaboratively between SW610f17 and SW612f17, with a focus on the current state of the Week Planner application. Its purpose is to provide future developers proper knowledge, to start working on the next iteration of the Week Planner application. It will begin by presenting some central interactions between the group, followed by the final state of Week Planner at the end of the collaboration, and what the customers think about it.

Communication with customers happens through meetings, where they get to see the product, communicate their desires, and comment on flaws to the *Product Owner* group. The meetings with customers provide information on the customers' thoughts about both potential and actual design choices.

7.1 Collaboration

During the semester we have had a few meetings, where sw610f17 and sw612f17 planned in which parts of the week planner the customer should be questioned.

7.1.1 Customer Questions

We planned the questions for the customers about the week planner application together. At the meeting, we were going through the application to find out what would be relevant to talk about with the customers. This was done to get a combination of

what the customers wanted and what was feasible for the developers to engineer. The customers do not know how much time a task requires, therefore being able to provide the customers with that information helps the customers prioritising which tasks to handle first.

7.1.2 Demonstrating the Week Planner Application

This meeting was set up as a prerequisite for a customer meeting where they would see the changes made to the Week Planner application. This meeting is important because sw612f17 have no prior knowledge about the Week Planner application, nor any of the changes made to it. The demonstration covers everything about what the application can do, and how they navigate around it as well as what has changed and why.

At the institution *Egebakken*, sw612f17 presented the application to the customers. They are happy with the new features but less than satisfied with the stability as the application crashed multiple times. Back in the group room sw612f17 tries to show and replicate the crash but without luck. The crash is subsequently marked as *not reproducible* and closed as we expect it has something to do with server communication and will replace it with the REST API.

7.1.3 Smaller Interactions

Throughout the semester sw610f17 have contact with sw612f17 about all the small things that comes up as the Week Planner application is being developed. This is mostly small questions about changes to UI or how the customers would like something to be done.

7.2 State of Week Planner

The Week Planner is more stable when using existing functionality, but the new features such as the REST API makes it more unstable, as there was not enough time to test.

The customers have highly requested a minimum viable product to be released soon, as seven years without a running solution is a long time.

7.2.1 Stability

The old Week Planner application, before the implementation of the REST API, is stable in the group rooms, while it is very unstable when tested at the customer's institution. This issue has been difficult to solve because we have not been able to replicate the error in the group rooms.

The application is still unstable with the REST API, though we expect to solve a group of exceptions, such as those explained in 7.1.2 with it, as offline no longer is an option.

As such the Week Planner application currently is unstable and still crashes quite often. This is a problem because the primary request from the customer is the stability of the program.

7.2.2 Accessibility

The PO group sw612f17 held meetings with the customers, about the problems they had using the interface, in regards to what happens and finding the feature they want to use. The current version is designed as a combination of customer requirements such as fewer clicks to access features, new features they would like, naming and reordering of features in the system. To clean up the user interface, and make it consistent in vertical and horizontal mode, a drop-down menu is introduced to replace the buttons. As a consequence of this, some functionality, such as copying days to other schedules, is removed because it is not very accessible.

The customers like the changes to the system and find it easier to use, the drop down menu makes sense, and they see what they expect where they expect it. However, they want the english terminology to be swapped with Danish terminology and replace words such as “overwrite” with “erstat”.

7.2.3 Aesthetics

Some of the customers show a desire for grayscale, as some citizens have difficulties handling the stimulation from many colors. Grayscale support is available, but it only supports the **activities** in the app, not *dialogs*.

The red crosses on pictograms used to cancel tasks are made smaller, to avoid them blocking the entire pictogram and furthermore make them aesthetically pleasant.

The customers are satisfied with the design because it is set up to resemble their paper version of a schedule. However, they still need more functional features to evaluate the product's design fully.

7.2.4 PDF

A request from the customers is a function to print the Week Planner application's schedule, to use it in situations where tablets might not be viable or accessible. In its stable state, it can create pdf files and open them in a pdf viewer on the tablet. It can show the order of pictograms and their names, and also show choices in a manageable way. A choice is a special pictogram that when tapped opens a dialog with multiple pictograms. The citizen can choose one of the pictograms, and it will replace the choice-pictogram in the user's current week plan. It is unknown if the implementation works since Pictosearch does not work, so we cannot add any pictograms to a weekday, and thus not create a pdf from it.

The customers like the implementation shown to them. Since it is a backup solution, they prefer us to work on getting a functional product before implementing more features.

7.2.5 Tutorial

There is a tutorial in the Week Planner application that can guide the user through using the features in Week Planner. Changes to the user interface result in the tutorial being out of date.

The customers want the app to be easy to use, and quick to learn. Therefore a well-written guide can save a lot of time and avoid confusion for the customers. The tutorial is not updated this semester because the functionality of the applications needs to work before a guide makes sense.

7.3 Suggestions for Future Development on Week Planner

We have the following suggestions for future development, based on the state of the app and user stories. The tasks are in order of importance according to an estimated done by the associated groups.

1. Get the Week Planner application into a stable state with the new REST API so that it doesn't crash or freeze.
2. Developing a fully functioning custom templates for guardians to use for citizens
3. Copy & paste capability to clone day schedule into a citizen's plan.
4. Search functionality for schedules and citizens instead of just listing them all as in the current version of the application.
5. The guide should be updated to include the new features and get rid of the old ones.

We suggest that when the application teams work together with the *Product Owner* group, they should start by drawing up its design. The expected result of this will be a lower amount of changes for the final version, and a productive discussion on how a feature should look and function, giving a better end product.

Chapter 8

Conclusion

At the end of the final Sprint, we evaluated the goals of this project; we then reflected on what could be done to finish the uncompleted features. Some goals had not been achieved, due to miscommunication between the semester's groups. Some of the unfinished features were the interaction between the model layer of the REST API and the *Kubernetes* server model layer, as there are some mismatches there. We recommend clearing these mismatches as soon as possible.

We have learned from this experience how important communication between teams can be, especially when developing a solution where teams are working on parts of the solution that are dependent on each other. We recommend that future developers on *Giraf*, talk about the models first, and design the model layers together before implementing different models for the different parts.

The launcher for Giraf is close to a deployable state. One cannot open apps inside it, but it does not crash, and it no longer uses QR Codes.

The week planner is also almost finished. It is currently not possible to test if the week plan is pushable to the server via a POST request. It also received some additional features, such as copying and printing.

The REST library still has a decent amount of problems. There are various crashes still happening, where only some have known causes. It also has a decent amount of optimisation left, as all requests end up returning too much data at once. It does however accurately update the system whenever a user's information changes.

The server which hosts the project has had some changes. There was a bug in the university's system, but a workaround was applied to fix this. The server currently hosts *Artifactory*, and *Kubernetes*—a program for assisting deployment of projects.

Overall the project did not end up being ready for delivery. However, as many parts of the system do not require much work to be completed; so we can say, it has come closer to a finished state.

8.1 Project Reflection

In this project, we have had an exceptional amount of customer contact compared to the previous years. From this experience, we have learned about questioning the customers. Here, we learned the importance of keeping a stricter structure when questioning. It is important to enforce this structure during the meeting; otherwise, the questionee will have the option of controlling the structure of the meeting.

Furthermore, we learned the importance of proper formulation from our questioning. After unintentionally misleading the customers in the first sprint, we have become more aware of how to precisely formulate written contact to both the customers and supervisor. This has helped us improve our linguistic skills, especially when making our questions more accessible to people with different viewpoints.

During the fourth Sprint, the coding task required more experience with the code for the system than we had. We should probably not have had the coding task, but since we did not have a costumer meeting during the fourth Sprint, we took it to get some non product owner tasks done this semester. In future large projects, we should be prepared to handle other people's work and collaborate on tasks. It would also be effective to communicate more with the other groups on a daily basis to monitor progress and work in relation to what the customers want.

Bibliography

- [1] CDC. Autism spectrum disorder (asd). website, 2012. URL <https://shop.lego.com/en-US/NXT-Intelligent-Brick-9841>. last seen: 2017.4.18.
- [2] Mountain Goat Software. User stores. website, 2017. URL <https://www.mountaingoatsoftware.com/agile/user-stories>. last seen: 2017.5.20.
- [3] Inc. Phacility. A complete software development platform. website, 2017. URL <https://www.phacility.com/>. last seen: 2017.5.24.
- [4] GIT Community. Git-local-branching-on-the-cheap. website, 2017. URL <https://git-scm.com/>. last seen: 2017.5.24.
- [5] Android Studio. Everything you need to build on android. website, 2017. URL <https://developer.android.com/studio/features.html>. last seen: 2017.5.24.
- [6] Gradle Inc. Accelerate developer productivity. website, 2017. URL <https://gradle.org/>. last seen: 2017.5.24.
- [7] R. Tyler Croy. Jenkins. website, 2017. URL <https://jenkins.io/>. last seen: 2017.5.24.
- [8] Docker Inc. Docker. website, 2017. URL <https://www.docker.com/>. last seen: 2017.5.25.
- [9] Time-Timers. How much longer? website, 2017. URL <https://www.timetimer.com/>. last seen: 2017.5.18.
- [10] The Kubernetes Authors. Kubernetes. website, 2017. URL <https://kubernetes.io/>. last seen: 2017.5.24.

Glossary

ASD Autism Spectrum Disorder

ADHD Attention Deficit Hyperactive Disorder

ADD Attention Deficit Disorder

TEACCH Treatment and Education of Autistic and Related Communication-Handicapped Children

API Application Programming Interface

APK Android Package Kit

IDE Integrated Development Environment

UI User Interface

GUI Graphical User Interface

QR Quick Response

PDF Portable Document Format

REST REpresentational State Transfer

Giraf The Graphical Interface Resource for Autistic Folk

Appendix

Appendix A

Questions for the Customers

All answers to questions are in italic.

A.1 Questions Birken February 15

Hvilke apps har tidligere studerende præsenteret?

Hvordan vil I rangere de forskellige apps efter hvor ønsket de er disse i forhold til behjælpelighed i dagligdagen?

Hvilke af disse har de tidligere studerende præsenteret som færdige?

Med henblik på at finpudse og færdiggøre nogle apps som er klar til at blive taget i brug i dagligdagen, vil vi gerne høre hvilke apps der ville kunne hjælpe jer bedst?

I forhold til dette vil vi gerne høre om hvor vigtigt det ville være at lave apps'ne således at de kan køre på iPad?

Vil I foretrække at kunne bruge "spillene" uden adgang til internettet, hvis dette sker på bekostning af at det ikke vil være muligt at finde informationen der ellers kan tilgås?

Og vil I gerne kunne bruge dem uden at skulle have Giraf-app'en åben?

Hvor vigtigt er det at app'en er hurtig?

Dette er primært i henhold til børnene.

Da det er muligt at styre om skærmen skal kunne gå i sort, når man ikke er trykker

på den i en periode, er der så nogle tidspunkter i brugen af uge planlægnings app'en at dette kan være et problem?

Kan det være hensigtsmæssigt at der indføres at skærmen slukker selvom at apps er åbne hvis tabletten ikke bruges?

Når I vil planlægge en uge hvilke informationer vil I normalt benytte?

Og hvilke vil børnene være med til selv at udvælge?

Har I eventuelt et eksempel I kan vise os?

A.2 Questions Egebakken

Launcher

- Skal launcheren kunne skifte fra pædagog til barn til pædagog til anden pædagog?

– Hvordan ønsker I at det fungerer?

* For at undgå login.

Skal ikke være sin egen knap.

Forslag fra pædagog emil var at man muligvis kunne holde menu knappen nede i 5-10 sekunder

- Er der flere pædagoger per barn?

Ja.

- Modificering af antal apps på skærmen og farveskema?

– På individniveau eller på grupperingsniveau?

– Er der brug for styring på grupperingsniveau?

Der er huse, stuer, klynger, klasser, etc.

Farveændring skal være muligt på individ niveau

– *Skal dog også eksistere pakkeløsninger.*

– *Samt måde at slå farver til og fra.*

Antal apps der er i menuen er ikke nødvendig at ændre.

Det ville være dejligt, men det har lav prioritet.

Loginsystem

- Hvem skal bruge det pædagog eller barn?

Begge skal kunne logge ind.

- Kodeord på pædagog- eller institutionsniveau?

Evt kode på grupperingsniveau.

Der skal være en klar separering mellem pædagog og forældre.

Institutionsniveau var en mulighed for pædagogerne, men de var lidt usikre på det.

Hvis det skulle være på institutionsniveau, så skal man have mulighed for at vælge hvilken pædagog man er.

- Hvor lang tid kræves en sessions login?

Eventuelt kan institutionsleder afvise en enhed adgang grundet forsvundet enhed?

Dette regner de ikke med er nødvendigt.

Børnene skal aldrig logges af, men administratorer kan smides af efter 15-20 minutter, som minimum.

- Skal kodeord overhovedet bruges?

Vi vil anbefale at det bruges for at undgå forkerte personer får adgang til forkerte handlinger.

Ja.

Ugeplaner

- Brugergrænsefladetilpasning (UI Customisation)?

- Sort/Hvid?

Skal kunne ændre alle de individuelle farver.

Hav også defaults ligesom der snakkes om i launcher.

- Nummerering på pictogrammer?

Synlige tal/bogstaver ud for tingene på dagsplanen.

Giver mening for arbejdssystemer, men ikke for tidsplanen.

For tidsplanen giver det bedre mening med klokkeslæt.

- Hvordan skal simple copy/paste mellem dage fungere?

- Valgtilstand hvor man kan lave en action.
- Fra barn til barn på tværs af skemaer?
- Kun override eller kun tilføje?
- Synliggørelse af ændringer?

Det røde kryds gør det for svært at se den aktivitet det fjernede.

Gør det måske tyndere, eller lettere gennemsigtigt.

- Dette er noget de meget gerne vil have.

Vil gerne have at det er enkelte dage som kopieres.

Det kan så bagefter justere det kopierede.

Skal også kunne markere flere elementer til kopiering.

Hvis man kopiere over en dag så skal den overskrive, der skal dog gives en advarsel.

Skal gøre valgpunkter tydeligere.

- Reserveret felter hvor børn kan være med til at vælge?

De bruge et lykkehjul til at symbolisere det.

Der ville være lidt problemer med om de vælger ting, som andre har taget, så de er lidt usikre på om systemet skal håndtere det.

- Flip i ugeplanen?

- Manuel knap eller intern enhedsrotationssensor?

Fint at den bruger enhedsrotationssensoren, den skal dog skalere bedre.

- Modificering af overview med antallet af pictogrammer per dag?

- Lodret (enkelt dags)?

Skal i det mindste ikke fylde så meget.

- Vandret (ugeplan)?

Ingen modifikationer nødvendigt.
- Skal der være grundskema over andre skemaer?
 - Printfunktion: Skal skemaet kunne printes?

Ja.
 - Access rights: Hvem skal have adgang til skemaet?

Borgeren, forældrene og pædagogerne.

Specifik tablet til hver borger burde sørge for at andre ikke har tilgang til andres skema.
 - Skabeloner?

Lav prioritet, men ville være dejligt.

Der skal helst eksistere defaults på forhånd.
- Hjælpeguide
 - Hvor god er den?

Den er fin, en YouTube guide ville også være en god løsning.
- Personlige ugeplaner, eventuelt periodeskemaer?

Video af brug af fysisk pictogramskema.

Andre ting

- Hvis en af apps'ne crasher, skal den så give en fejlbesked og gå et step op i systemet (fra weekplanner til launcher)

Skal den sige den går et step tilbage, eller prøve at genstarte?

Et trin tilbage ville være dejligt.
- Hvordan de ønsker at identificere pictogrammer i ugeplaner: navne, numre, anden form for identifikation?

Egne fotos skal kunne tagges med søgeord.

Borgernes billeder skal kunne individualiseres.
- Pædagogerne går senest hjem kl 17:00

A.3 Questions Enterne

Launcher

- Skal launcheren kunne skifte fra pædagog til barn til pædagog til anden pædagog?

– Hvordan ønsker I at det fungerer?

* For at undgå login.

Skal primært kunne skifte fra pædagog til barn.

- Er der flere pædagoger per barn?

To pædagoger for hver fire borgere (gruppe/klynge), men der er altid kun en enkelt tablet til rådighed.

- Modificering af antal apps på skærmen og farveskema?

– På individniveau eller på grupperingsniveau?

– Er der brug for styring på grupperingsniveau?

Farver, er ikke noget som brugere selv skal kunne pille ved. Det er bedst hvis det er ensartet alle steder.

Loginsystem

- Hvem skal bruge det pædagog eller barn?

Begge skal kunne logge ind.

- Kodeord på pædagog- eller institutionsniveau?

Personalet skal bruge loginsystemet, for nogle borgere kan hverken bogstaver eller tal, eller har motorik nok til QR-koder.

Det ville være smart at pædagogerne har en kode, og at brugerne ikke skal bruge en. Nogle borgere har regnet ud at man kan ændre på skemaer.

Koden skal være på gruppeniveau, ikke personale- eller institutionsniveau.

Borgerne kan måske ikke tænke sig til at danne en kode i hovedet, før så at skulle bruge den. Så det er ikke ideelt at en borger skal taste en kode som fx er bestående af billeder.

- Hvor lang tid kræves en sessions login?

Automatisk logout for pædagoger ville være rart. 5-10 min ville være fint.

- Skal kodeord overhovedet bruges?

Ja.

- Andre ting

Et backup system vil være godt til at kunne smide den samme bruger op på en ny tablet, da der er en ret stor tabsprocent.

Borgerne skal ikke være tilgængelige på hinandens systemer. En pædagog skal kunne styre borgertablets, men ikke andet på tværs.

Forældre bruger sjældent pictogrammerne, men det ville være rart at de ikke "tog hjem uden førehunden".

Synlige tal/bogstaver ud for tingene på dagsplanen.

Det at rode med borgernes tablet skaber en masse usikkerhed, så det skal helst være at pædagogerne kun skal redigere på deres egen skærm.

De er i tvivl om hvor vidt det skjulte login er en god idé. Det skal være meget hurtigt og praktisk, da det er forstyrrende for dem at rode med deres "førehund" (tablet). Det skal helst bruges til akutte situationer.

Ugeplaner

- Brugergrænsefladetilpasning (UI Customisation)?

Det at skjule andres ugeplaner er ikke et behov, da ugeplaner allerede er synlige på væggen.

Sort/Hvid er ikke nødvendigt. Og numerering er også ligegyldigt.

Borgere som bruger klokkeslet er ikke noget der findes hos Enterne.

En time-timer til at illustrere hvor meget tid der skal gå.

Nøjagtige klokkeslet kan være et problem da nogle borgere vil være nøjagtige helt ned til sekundet.

Nogle borgere har slet ikke et begreb om tid.

- Hvordan skal simple copy/paste mellem dage fungere?

Funktionaliteten med flytning/kopiering er logisk, men i stedet for "overskriv" skal derstå "erstat".

Det ville give mening at kunne kopiere dagsplaner på tværs af borgere.

I nogle tilfælde vil det at overskrive være rart, og i andre ville det være rart hvis tingene stod samtidigt.

Det skal være tydeligt hvad der bliver overskrevet, når der bliver overskrevet.

- Reserveret felter hvor børn kan være med til at vælge?

Målet er at gøre borgerne selvstendige, så det at de skal kunne træffe valg selv, er rigtig vigtigt. Det at trykke på en aktivitet skal bringe muligheden for at vælge. Forskellige borgere bruger forskellige ikoner til at indikere at der er mulighed for et valg.

- Flip i ugeplanen?

- Manuel knap eller intern enhedsrotationssensor?

Det at rotere skærmen for at skifte mellem dag- og ugeplan er perfekt.

- Modificering af overview med antallet af pictogrammer per dag?

Der skal være mulighed for at vælge hvordan pictogrammer skal vises på skærmen. Nogle kan ikke overskue mere end fx tre pictogrammer, nogle vil gerne have hele overblikket. Nogle borgere har forskellige behov alt efter dag.

- Scrolling med finger vil være bedst. Bekræftelse er et centralet tema her.

- Skal der være grundskema over andre skemaer?

Grundskema vil ikke være brugbart for Enterne. Det at kopiere hele uger ad gangen vil ikke blive brugt, da de har behov for at være meget fleksible. Uforudsigelighed dikterer meget hvordan skemaet kommer til at se ud.

Hvis det er for avanceret at kopiere dagsplaner på tværs af borgere, vil de hellere bare gøre det flere gange.

- Printfunktion: Skal skemaet kunne printes?

Printfunktion af dage eller opgaver vil være rart i nødstilfælde, eller i tilfælde hvor en tablet er upraktisk.

- Access rights: Hvem skal have adgang til skemaet?

Borgeren, forældrene og pædagogerne.

Specifik tablet til hver borger burde sørge for at andre ikke har tilgang til andres skema.

- Skabeloner?
 - Lav prioritet, men ville være dejligt.*
 - Der skal helst eksistere defaults på forhånd.*
- Hjelpeguide
 - Hvor god er den?
 - Den er fin, en YouTube guide ville også være en god løsning.*
- Personlige ugeplaner, eventuelt periodeskemaer?
 - Video af brug af fysisk pictogramskema.*
- Andet
 - Ugeplanerne er blevet rigtig gode.*
 - Identifikation af pictogrammer skal helst ske på skrift, helst med et navn. Det skal være nemt og hurtigt at finde frem.*
 - Når et pictogram trykkes på skal det kunne blive highlighted, aflyst, eller færdiggjort.*
 - Borgerne skal egentlig kun kunne interagere med pictogrammer i form at *vælgetravler*, som tillader borgere at kommunikere via billeder, da ikke alle borgerne er i stand til at tale eller har et ordforråd til at kunne udtrykke sig.*

Forældre/Værgé

- Hvad skal en forælder kunne se med deres konto (account/user)?
 - Forældrene skal kun have samme rettigheder som en borgers mht skemaerne, men de må godt skulle være i stand til at kunne tilføje pictogrammer, men dette er ikke så vigtigt.*
- Hvad skal de kunne redigere i?
 - Der bliver ikke stillet nogen krav til forældrene med hensyn til deres børn, så det kommer an på de enkelte hvor vidt de overhovedet kommer til at bruge ugeplanen.*
- Skal de kunne komme med forslag?
 - Man kunne lige så godt bare oprette et selvstændigt system hos forældrene som er separat resten. På den måde er systemet isoleret fra Enterne.*

Andre ting

- Hvis en af apps'ne crasher, skal den så give en fejlbesked og gå et step op i systemet (fra weekplanner til launcher)

Det smarteste vil være at have at fejlene ikke er synlige, og at systemet bare går tilbage til det sidste stabile stadie. Systemet skal være driftsikkert og nemt at benytte.

- Andet

Identifikation af pictogrammer skal helst ske på skrift, helst med et navn. Det skal være nemt og hurtigt at finde frem.

Når et pictogram trykkes på skal det kunne blive highlighted, aflyst, eller færdiggjort.

*Borgerne skal egentlig kun kunne interagere med pictogrammer i form at *vælgetavler*, som tillader borgere at kommunikere via billeder, da ikke alle borgerne er i stand til at tale eller har et ordforråd til at kunne udtrykke sig.*

- Andet

Der vil kun være en enkelt tablet til rådighed for pædagogen, og en tablet for hver borgers.

De vil gerne have en tablet som kan styre systemet.

Skal ikke bruge for lang tid hos en borgers for at sætte hvad der skal ske op.

Templates for dagsplaner kan kaldes "Færdige strips".

A.4 Questions Birken April

Vore Beslutninger Siden Sidste Møde

1. Som ønsket har vi fokuséret på en mindre løsning.
2. Vi arbejder på højtryk for at gøre den færdig og få den ud til jer.
3. Suggestions?

Prototype - Anders

1. Ny Menuknap
2. Select
 - Flytte elementer
 - Aflys (ved kryds)
 - Slet
 - Gem
3. Print til PDF

Vores Spørgsmål

- **Launcher**

1. Hvordan ønsker I at logge ind på Giraf, skal der kunne skiftes mellem borgere og pædagoger? Hvordan?
2. Hvor mange pædagoger er der tilknyttet hver borger?
3. Modificering af hvad I kan se i launcheren så som farver, mængden af apps, og lign.?
4. Hvordan grupperingsstyrer I borgere?
5. Andre ting?

- **Loginsystem**

1. Skal en tablet/iPad automatisk logge et barn på, og ønsker I at logge ind på deres tablet/iPad og lave ændringer?
2. Skal hver pædagog have et kodeord eller skal det være på institutionsniveau?
3. Hvor lang tid kræver en login session?
4. Andre ting?

- **Ugeplaner**

1. Ønsker I et felt hvor børn selv kan være med til at vælge?
2. Virker skærm-flip intuitivt?
3. Skal der kunne begrænses hvor mange pictogrammer der vises ad gangen?

4. Ønsker I et grundskema for kan smides på en uge?
5. Må der gerne vises halve pictogrammer?
6. Er det okay at der kun kan printes en dag af gangen?
7. Andre ting?

- **Forældre/Værge**

1. Hvad skal forældre kunne gøre?
2. Skal de kunne redigere?
3. Skal de kunne komme med forslag?
4. Andre ting?

- **Andet**

1. Hvis en af siderne crasher, skal den give en fejlbesked eller gå tilbage til før fejlen opstod?

Appendix B

User Stories

B.1 User Stories Birken 15/02-17

The User Stories created after the customer meeting on the 15th of February, listed after assigned priority.

1. As one of the children who use the system, i want the system to be as stable as possible, as my mental problems require that of the tools which i use.
2. As a pedagogue I want the ability to replace pictures and icons for the children in the system.

Different kids have different ability to interpret abstract concepts. Some require specific colours, shapes or objects to be present, and some still can't do with drawings, but rather need literal photos. For this to work, there should be a way to replace the pictograms in the system, for each separate kid.

3. As a pedagogue, it is very important to be able to create a weekly schedule and manage the tasks the children can see at any given time.
4. As a user of the system, the response time is an important factor.
5. As a pedagogue I would like the ability to access the schedules of the kids' tablets remotely.
6. As a user of the system, I'd like the interface to be simple, things should never be more than 2-3 taps away.

7. As a pedagogue, I would like the option of adding placeholder tasks, which can then be changed to something else later. This enables the pedagogues to add tasks to a kid's schedule later on.
8. As a pedagogue, I would like to schedule a day by copying another day.
9. As a pedagogue, I would like to schedule a day using templates.
10. As a pedagogue, I would like to have certain keywords which i can use to search for pictures by category.
11. As a user, I would like to be made visually aware of any potential changes made in the schedule.
12. As a user, I would like to have a way of visualising the order of the tasks, i.e. explicitly numbering the tasks, because some kids prefer explicit ordering.
13. As a legal guardian, i would like the option to add notes about the individual children, which will then be available to the pedagogues.
14. As a pedagogue, i would like the option to add notes about the individual children, which will then be available to other pedagogues.
15. As a pedagogue and or legal guardian I would like the more complicated tasks to expand into sub-tasks when pressed.

B.2 Egebakken User Stories 22/03-17

General

1. As a user of the system, I would like the system to move back to the nearest stable state when the app crashes.
2. As a guardian I would like the option to toggle between grayscale and fully colored in the system for a citizen.
3. As a user of the system I would like the option to toggle between grayscale and fully colored in the system.
4. As a guardian I would like a video I can use for more advanced tasks such as creating schedules etc.

Launcher

1. As a guardian I would like the login button to let a citizen tap to log in as a citizen, and hold to login as a guardian, only providing login information at that point.
2. As a guardian I would like to be able to push a template setup onto a group of citizens for the launcher.
3. As a guardian I would like the option of adding an app to citizens both as groups, individually or marking a list of citizens.

Login System

1. As a guardian I would like not to have a Quick-Response (QR) codes for login, but a regular password instead.
2. As a guardian I would like the ability to set a password for specific users of the system.
3. As a guardian I would like the system to automatically log me out of my account after a set time limit (approx. 20 min).
4. As a citizen I would like the system to not to log me out automatically.
5. As a guardian I would like the option to make a specific device always log into a citizen's account without password.
6. As an institute, we would like to be able to log in as the institute as a whole.

Week Planner

1. As a guardian I would like to add my own pictures as an addition to pictograms, as the citizen might need more specification than what a pictogram allows.
2. As a guardian I would like to copy an entire day and paste it to one of the other days in the week and modify it, to ease the creation of schedules.
3. As a guardian I would like to print a list of an entire day for a citizen.
4. As a guardian I would like to have a premade template schedule I can modify or not use at my own accord.

5. As a citizen I need the option of having timestamps at/by the pictograms.
6. As a guardian I would like the ability to mark multiple elements from several days and copy them to a single day.
7. As a guardian I would like to be warned if I am about to overwrite a complete day by using copy/paste.
8. As an institution I would like only, the guardians, the parents and the citizen itself to have access to the specific citizen's schedule.
9. As a guardian I believe the red X to mark tasks as cancelled/deleted can be made thinner to avoid shadowing the pictogram.
10. As a guardian I would like the system to scale from week to day when I rotate the unit.
11. As a guardian I would like the UI elements to scale icons when rotating the tablet.
12. As a citizen I would like the pictograms on the day planner to be scrollable when exceeding page length.

B.3 Enterne User Stories 06/04-17

1. As a guardian, I would like the ability to quickly and seamlessly login to a citizens tablet and make changes.
2. As a guardian, I would like to be able to adjust the time it takes for the system to automatically log me out.
3. As a citizen, I would like to be able to quickly get a replacement tablet with the same settings and data as the previous one.
4. As a citizen, I would like not to see the other citizens' accounts from my account.
5. As a guardian, I would like to see and manage the citizens' accounts in the group I manage.
6. As a citizen, I never want to see half a picture on the system, either the whole picture should be visible or none of it.
7. As a guardian, I would like the system to use the word "erstat" instead of "overskriv".

8. As a citizen, I would like a time-timer illustration of how long a task should take.
9. As a guardian, I would like to be able to see timings of task for a citizen for schedulability.
10. As a citizen, I would like not to see the time a task is scheduled.
11. As a guardian, I would like the ability of setting finished “strips” onto the planner, like washing hands can be expanded to show the amount steps included.
12. As a guardian, I would like the ability to cancel a task and decide if the cancellation should be visible to the citizen.
13. As a guardian, I would like it to be clear what is being overwritten.
14. As a user I would like to set the amount of pictograms shown on a day-to-day basis.
15. As a guardian, I would like the following options on a pictogram: Cancelled, Highlighted and Finished.
16. As a parent, I would like the ability to view my citizens schedule, while also having the option of adding pictograms.

B.4 User Stories 01/05-17

1. As a user, I need the system to function without crashing.
2. As a guardian, I want it to be clear if I'm logged in on a tablet.
3. As a guardian, I want it to be clear if I have logged out successfully from a tablet.
4. As a citizen, I need to be able to clearly distinguish between active and complete activities, either by toning the colour down, or marking it with a green check mark.
5. As a guardian, I need the option to have special individual notes on each citizens' schedule.
6. As a citizen, I need the system to never show half of a pictogram.
7. As a citizen, I want to be able to log in by tapping on a picture of myself, potentially labelled with my name.

8. As a guardian, I need the option to choose how many days each citizen is capable of seeing.
9. As a guardian, I want template schedules I can use for multiple citizens.
10. As a guardian, I need to be able to lock the rotation of the week planner for specific citizens.
11. As a guardian, I want a hidden login option to be able to change from a citizen to a guardian.
12. As a guardian, I want to decide on which side of a pictogram the text is when making the PDF.
13. As a parent, I would like to be able to see my child's day.
14. As a parent, I want to be able to create and manage a separate daily schedule for my child at home.

Appendix C

Guides

This chapter contains two guides, one which shows how a guardian uses the Giraf system, to help future developers, and a guide designed to help future Product Owners getting started on Giraf. One can also find these guides on the wiki.

C.1 How a Guardian Uses the System

Everything written in **bold** is functionality that is missing.

The current login screen is being circumvented.

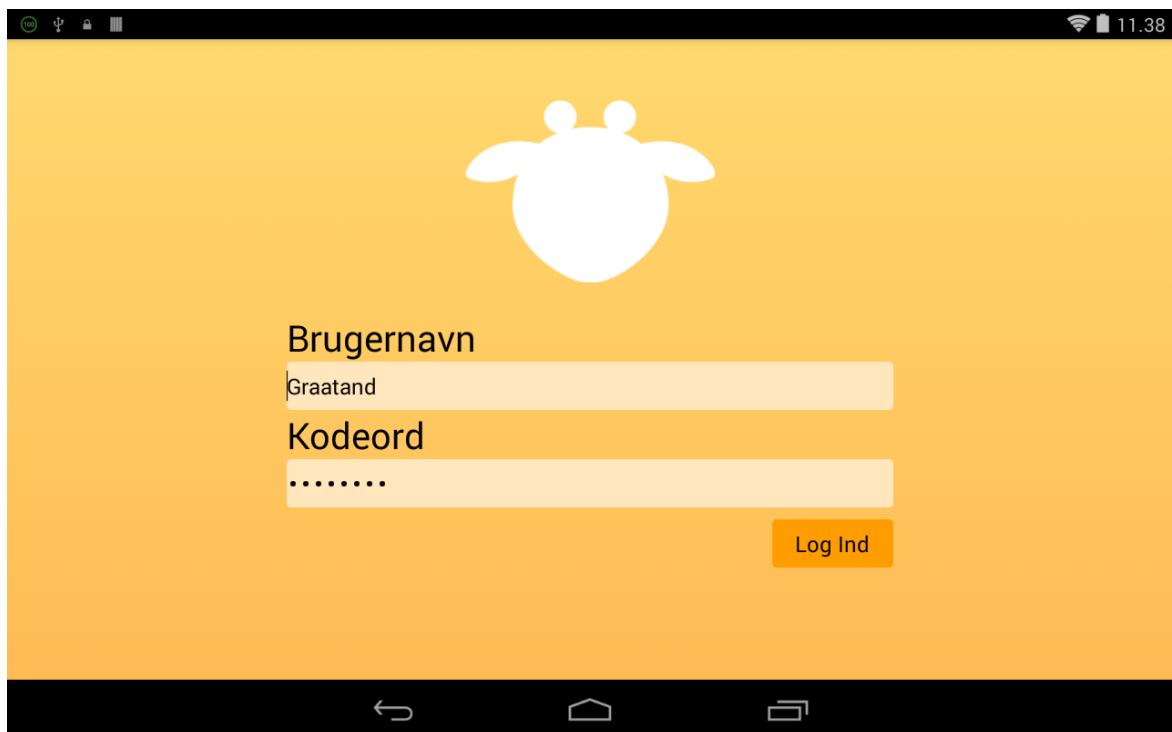


Figure C.1: Login Screen of Giraf.

This brings us to the initial screen of the week planner. From here one presses the button to choose the citizen whom they make changes to.

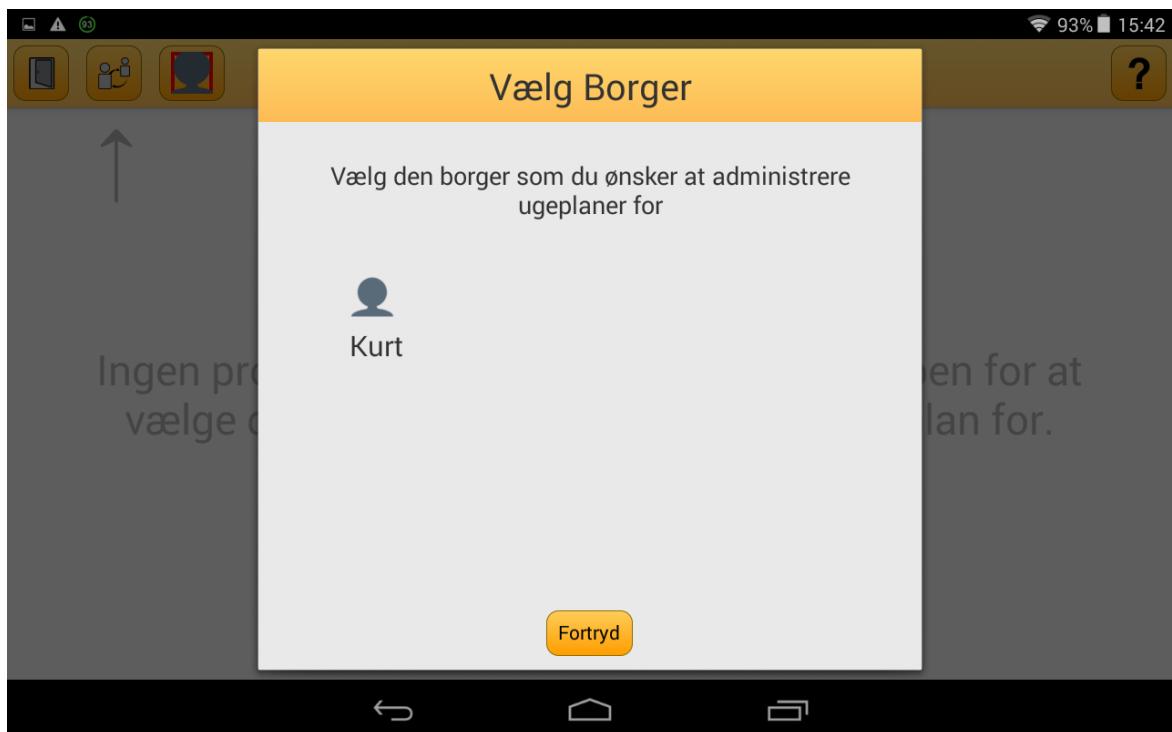


Figure C.2: The menu for choosing a citizen.

There should be a way of choosing groups of citizen

After choosing a citizen, one can now see all their schedules. From here, one can copy schedules over to other citizens. To do that, one presses the *copy* button and then select the citizen who will receive it.

Insert picture of the selection screen for schedules

Now we will open a schedule by clicking it, which brings us to the week planner menu.



Figure C.3: The Weekplanner of Giraf.

A few institutions have requested the ability to add notes concerning individual citizens. Low priority

Some citizens should only see a limited amount of pictograms for each day.
Some also should be limited in how many days they can see

If one turns the screen, it will change from showing a week to a day, while also changing how many pictograms are shown.

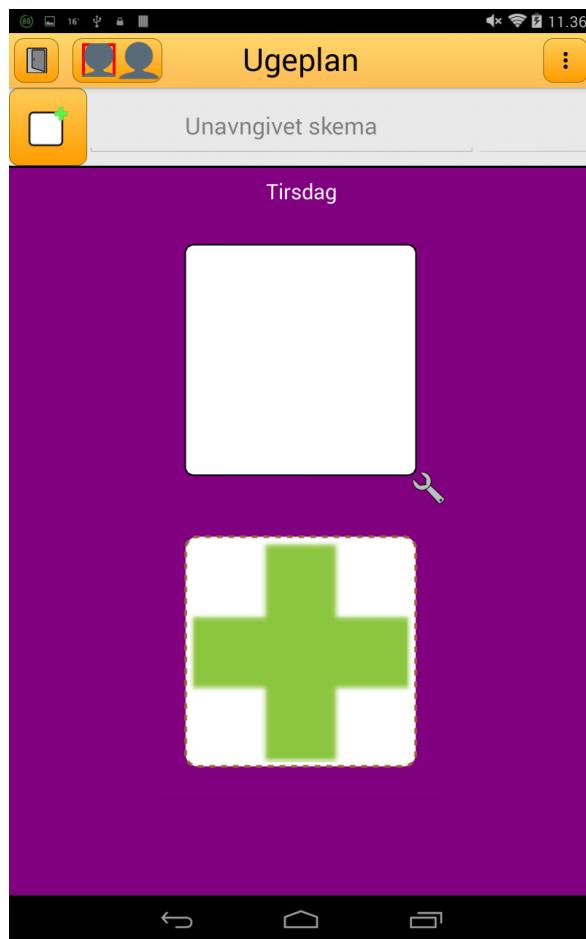


Figure C.4: The Dayplanner of Giraf.

In the upper right menu, one can choose to print the week plan. It does this by converting one single day to a PDF file.

In the upper right menu, we can also choose to mark things. This can also be activated by holding a finger on a pictogram. In marking mode one can copy, move, and cancel the events.

To add a new pictogram, press either a pictogram or the green “+” icon. This opens up the menu for adding pictograms.

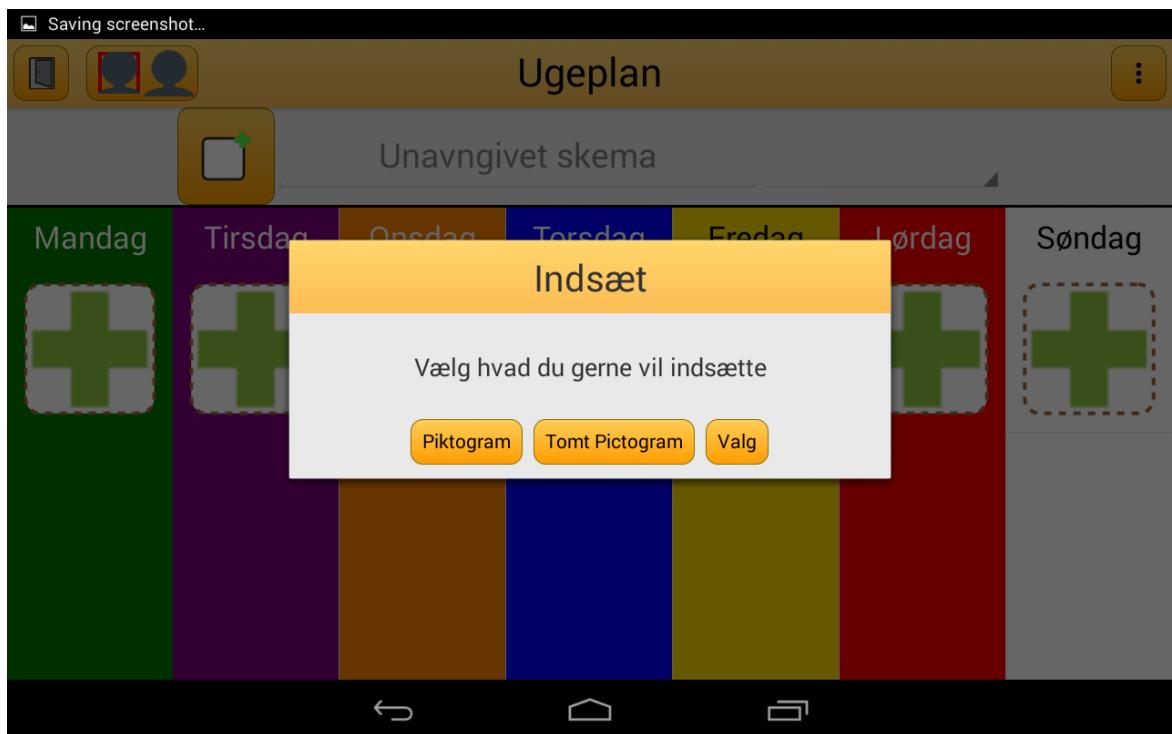


Figure C.5: The Picto Select Menu.

Enterne requested a way to add a timer for each event, which should work in the same manner as a time-timer

To add an additional pictogram press the green “+” icon here. This will open the pictosearcher.

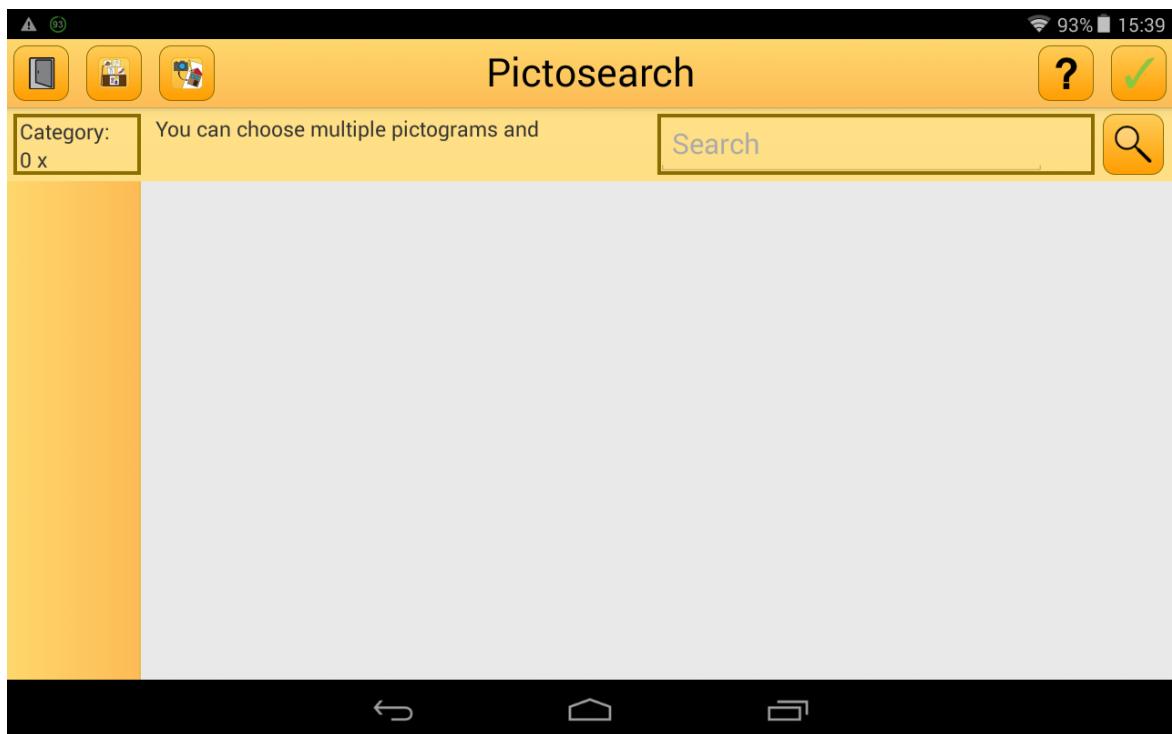


Figure C.6: The Pictosearcher.

One can add several pictograms to a single event in the same manner one add a pictogram. In these cases, the citizen will have the option to choose themselves which of the events they will participate in.

C.2 How to be a Product Owner

This guide explains how to be the Product Owners of a multi-project, and what we found to be the best practices.

C.2.1 Contacting the Customers

The most important part of contacting the customers is aware of the formulation when communicating with the customer. It is especially important to make clear whether or not you are arranging a meeting with them or inviting them to an already predetermined meeting. This is usually best done by asking the customers which time would be best for them, or by requesting some form of written feedback.

The method of contacting the customers can either be done by sending an email or by calling them. Email is the most formal method; however, it is also the one most susceptible to misunderstandings, caused by a poorly formulated email. There is no guarantee that the guardians read or respond to the emails. Calling them usually limits the influence of the problems which can happen with email. It is, however, non-formal and you have no guarantee that the time will be remembered. We suggest primarily using emails.

When asking for a meeting, you need to mention what you plan to do at the meeting. You do not need to explain it in depth, a simple remark stating what the meeting will be about, for example:

The meeting will focus on some new features which have been implemented, and we would like some feedback on the design and use of the application.

The date of the meeting should preferably be planned a week in advance; it is not a hard requirement but is considered polite. Make sure that the customer is required to respond. They also need the choice of cancelling or moving the meeting. The meetings can either be held at the University or their institution; we preferred holding them at their location. This takes less time out of their schedule, and can, therefore, more easily be arranged.

C.2.2 Holding the Meeting

Before the meeting, make sure to have planned what to do during the meeting. This not only includes having prepared questions but also knowing who does what. A good

delegation of work could be: one person takes notes, one person is responsible for the technology and structure, one primary speaker, and one backup speaker. However, also be prepared to ask questions outside of those previously made.

During the meeting, be aware of the structure. Depending on who is holding the meeting, one might either loosen or tighten the structure. The way we did, was to first make a demonstration of the system and then ask questions afterwards. Some of the demonstrations were filmed, and every meeting was recorded, so that everything was documented. The video also helps in documenting how the customer uses the system, which could—for instance—reveal if they had trouble using a menu button.

After the meetings, verify all the notes taken and be prepared to add additional information; for instance, how well the system was navigated or something similar. Always reflect on how the meeting went, whether you should change your methodology, what went well, etc.

C.2.3 User Stories

User Stories is a short story which describes a single feature that the customer wants for the system. They are told from the perspective of the customer.

The user stories should primarily be made from the data gathered at customer meetings. They should be accurate in describing what changes need to be done, such as whether or not there is a need for a new feature or modifications to an existing one. They should also only refer to a single feature, any which refer to several features should be split into smaller user stories.

We recommend that the user stories will be written as:

As a user/citizen/guardian/developer I want _____ so that _____.

All user stories should be prioritised, as this ensures better delegation of work. One should not consider the difficulty in implementing the features, only how much the customer needs them. The prioritisation should also consider the importance of each application, for instance, the week planner is more important than the voice game.

While the user stories are made by the Product Owners, they should still be verified by the other groups. The best time to do this would be at the Sprint meetings. The verification will focus on the prioritisation of the user stories; they should not make changes to the stories. If the other groups are not satisfied with the user stories, then they should suggest it outside the meeting, where it can then be discussed.