Process Report

Fresh Fitness

Group 8:

Jaser Ghasemi - 267243
Yasin Issa Aden - 267276
Modaser Ghasemi - 267251

Supervisors:

Ole Ildsgaard Hougaard
Jakob Knop Rasmussen
Christian Flinker Sandbeck
Erland Kertil Larsen
Line LIndhardt Egsgaard
Jan Munch Pedersen

Software engineering

3rd semester

(Date)

Version: August, 2018
Template responsible: dans@via.dk

Table of content

1 Introduction

The group learnt from the previous SEP project that beginning late can have serious consequences for the project. So, in this semester these previous experiences should dictate how the next SEP project should be approached. First and foremost, the idea should be interesting and exciting for the group, secondly the project proposal should be something which is within the group's capabilities, not too hard and not too easy. Finally, the project should be well planned and structured following the scrum and agile approach. The group believed that following these guidelines would result in a presentable SEP project by the end of the semester.

The first phase (inception) had some tasks which needed to be done which were, project description, use cases, test cases and activity diagram. The next phase (analysis) was done by analysing the requirements. Here a domain model and analysis class diagram were made. The third phase (design), the group thought of the methods needed for the system. The design class diagram and sequence diagram were made during the third phase. The fourth phase (implementation), was to implement the system and lastly the system was tested.

All these phases follow the waterfall approach, conveniently the SCRUM sprint cover over everyone of the phases respectively. As mention before the SCRUM method should be used alongside the agile approach. For example, all the testing was not done at the end, instead one method or functionality was created and then tested shortly after. This was done continuously while coding. The group would continuously go back and forward changing and polishing previous code. Furthermore, new designs or effective design patterns came up despite already finishing the analysis and design phase. This meant that our diagrams needed to be continuously adjusted for the latest version of our program and in the end when all the diagrams were done, a comprehensive test of the entire system and its functionality was made.

2 Group Description

When SEP3 began, one of the first tasks was to organize a SEP group. The team you decide to form is extremely important for the project’s future, and a lot of variables was considered. After some discussion our group was finally formed. The group contained three students from Viborg, and because of an exception, this group was allowed to have three students with the same nationality. Based on SEP2 the group now had a different approach to this project, since a lot of problems occurred regarding our planning.

One of the obstacles for the group was the travel distance from Viborg and Horsens. The travel time would approximately take 2,5 hours each way. The group members would have to spend five to six hours just on travel time. This led to the group deciding to work from home most of the time, but this came with a cost. It was hard to get help because the group relied on using emails back and forth, in order to get help from the supervisors, and emails were not as efficient as being next to the supervisor in real life and get help directly.

Another important thing to mention is that the group members were relatively close friends, and this meant they were aware of each other’s strengths, weaknesses, and personalities, which was both helpful and negative because the group members would often be more laid back and sometimes could lack taking responsibility of own individual tasks. The group members decided to give each other deadlines for individual tasks, and if a deadline was overdue, the member would have to pay for dinner.

The same group was also used in SEP2, and back then a lot of challenges happened. First of all, due to a misunderstanding and miscommunication the group started to code the entire system in the last the 2 weeks of the project. This understandably led to a system that was unfinished, which affected the overall result and in the grading. Luckily, we passed and could take this as a lesson and experience for the new SEP3. In this SEP the group made it a priority to begin the coding in good time to tackle the inevitable challenges that will happen in that process.

Once the idea was approved by the supervisors, the group started to focus on the project by creating a schedule and organizing tasks. The group decided to create a group policy to abide by, in order to secure efficient teamwork. Once every member had signed the group policy it was decided that the group would meet up during Thursdays and work on the SEP project.

To be able to track the work done, the group structured the project in a SCRUM, with 5 sprints. First, the inception phase, where use cases were made, which later became the use case diagram. Secondly, the analysis phase where the classes necessary for the system were made, and afterwards the analysis class diagram was made. Thirdly, the design phase where the methods for these classes were decided. After the Design phase it was time to implement the system, and lastly, the system would be tested.

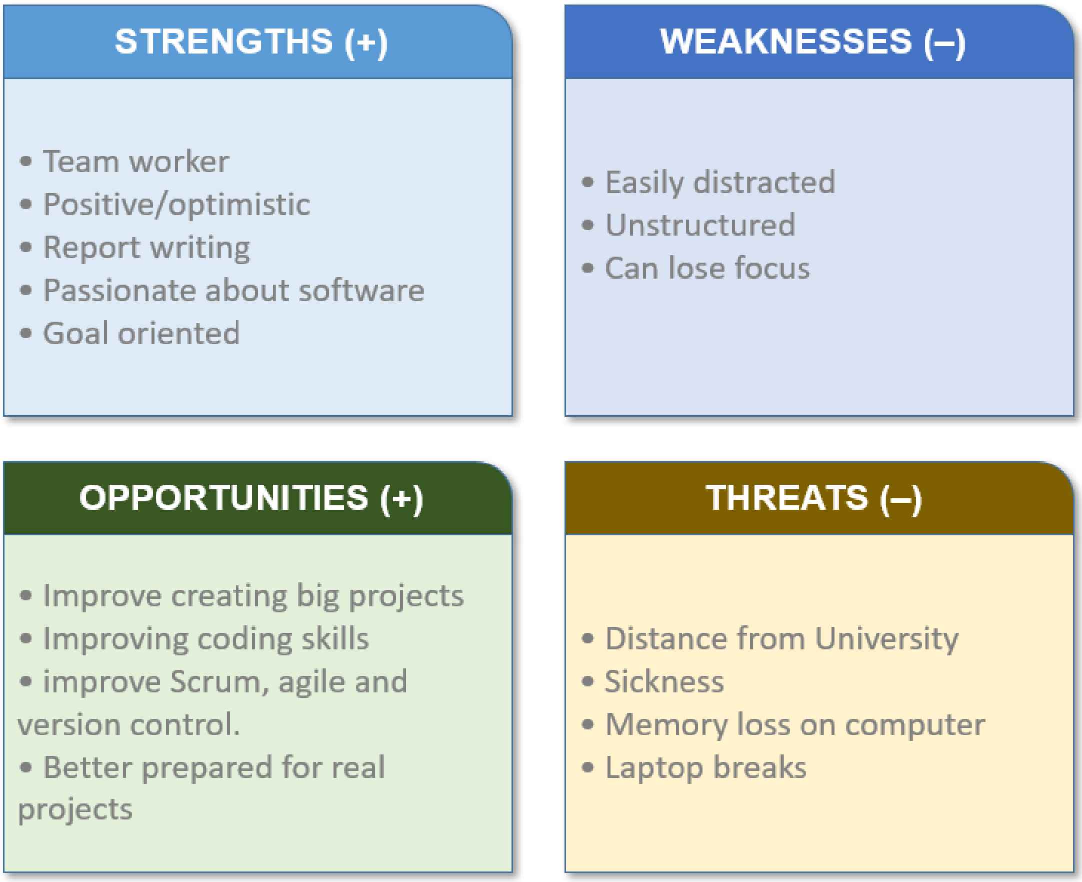
2.1 Member Introductions, SCRUM roles & SWOTS

Member Introduction – Modaser “Product Owner”

The first group member is Modaser Ghasemi, he is 21 years old and he is a third semester Software engineering student in VIA. Other than the previous SEP project from SEP1 and SEP2 he doesn’t have more experience with big projects.

SCRUM role & SWOT – Modaser

Product Owner owns the product backlog, the product owner decides the required functionalities and tasks for the project, and is responsible for providing requirements for the system.



STRENGTHS (+)

- Team worker
- Positive/optimistic
- Report writing
- Passionate about software
- Goal oriented

WEAKNESSES (–)

- Easily distracted
- Unstructured
- Can lose focus

OPPORTUNITIES (+)

- Improve creating big projects
- Improving coding skills
- improve Scrum, agile and version control.
- Better prepared for real projects

THREATS (–)

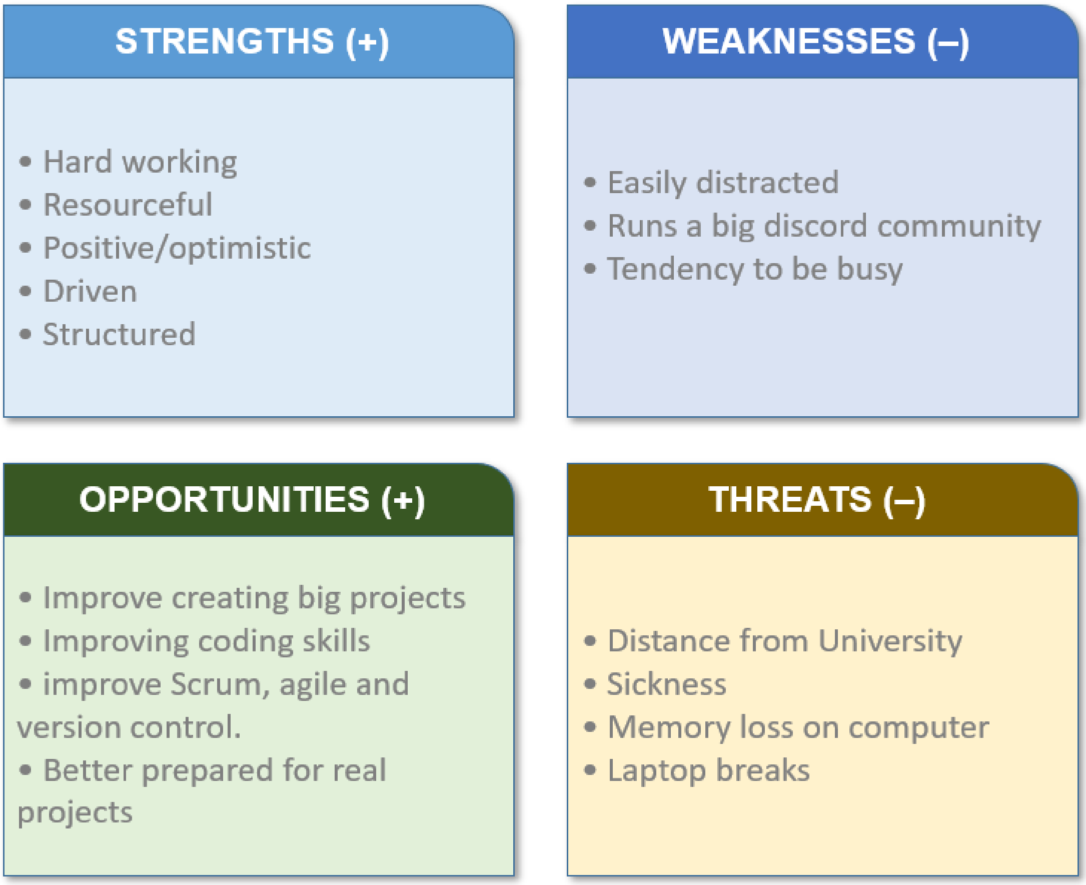
- Distance from University
- Sickness
- Memory loss on computer
- Laptop breaks

Member Introduction – Jaser "Product Owner"

The second group member is Jaser Ghasemi, he is 23 years old and he is a third semester Software engineering student in VIA. Other than the previous SEP project from SEP1 and SEP2 he doesn't have more experience with big projects. However, he has a little prior experience to creating some website-based projects.

SCRUM role & SWOT – Jaser

SCRUM Master is fully responsible for making sprint backlog and burndown chart, and to make sure that the group members are following and understanding SCRUM. It's also SCRUM Master's responsibility to organize meetings, both with group members and with supervisors.



STRENGTHS (+)

- Hard working
- Resourceful
- Positive/optimistic
- Driven
- Structured

WEAKNESSES (–)

- Easily distracted
- Runs a big discord community
- Tendency to be busy

OPPORTUNITIES (+)

- Improve creating big projects
- Improving coding skills
- improve Scrum, agile and version control.
- Better prepared for real projects

THREATS (–)

- Distance from University
- Sickness
- Memory loss on computer
- Laptop breaks

Member Introduction – Yasin "Developer"

The first group member is Modaser Ghasemi, he is 21 years old and he is a third semester Software engineering student in VIA. Other than the previous SEP project from SEP1 and SEP2 he doesn't have more experience with big projects.

SCRUM role & SWOT – Yasin

The development team is responsible for finishing the SCRUM tasks (building the product piece by piece). The development also creates estimates on every task required for the product.

STRENGTHS (+)

- Goal/result oriented
- Serious
- Team worker
- Determined
- Adaptable

WEAKNESSES (–)

- Unstructured
- Tendency to misunderstand things
- Easily stressed

OPPORTUNITIES (+)

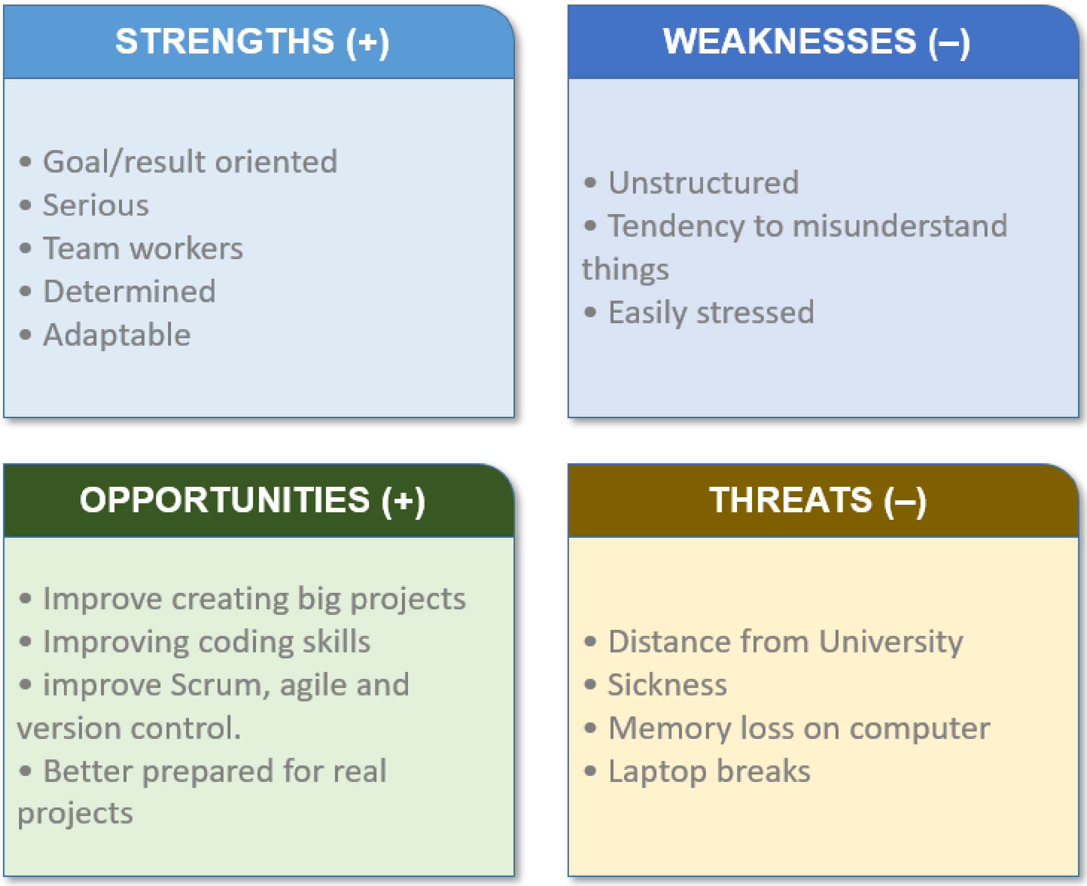
- Improve creating big projects
- Improving coding skills
- improve Scrum, agile and version control.
- Better prepared for real projects

THREATS (–)

- Distance from University
- Sickness
- Memory loss on computer
- Laptop breaks

2.2 Group SWOT Analysis

Being together, they thought that it would be the most convenient and efficient way of working together. Because they all live in Viborg, they could study together in Viborg in the library, so they did not have to travel so far to study. However, this also had its consequences, which was a limited help and too many obstacles. The group had to push each other. Furthermore, they also felt the huge impact of stress which hit them all since none of them had a lot of experience with coding, and they kept thinking of how to make everything work regarding course assignment and SEP. Furthermore, last SEP2 project didn't end well, so group used the previous experience to dictate this SEP's approach.



STRENGTHS (+)

- Goal/result oriented
- Serious
- Team workers
- Determined
- Adaptable

WEAKNESSES (–)

- Unstructured
- Tendency to misunderstand things
- Easily stressed

OPPORTUNITIES (+)

- Improve creating big projects
- Improving coding skills
- improve Scrum, agile and version control.
- Better prepared for real projects

THREATS (–)

- Distance from University
- Sickness
- Memory loss on computer
- Laptop breaks

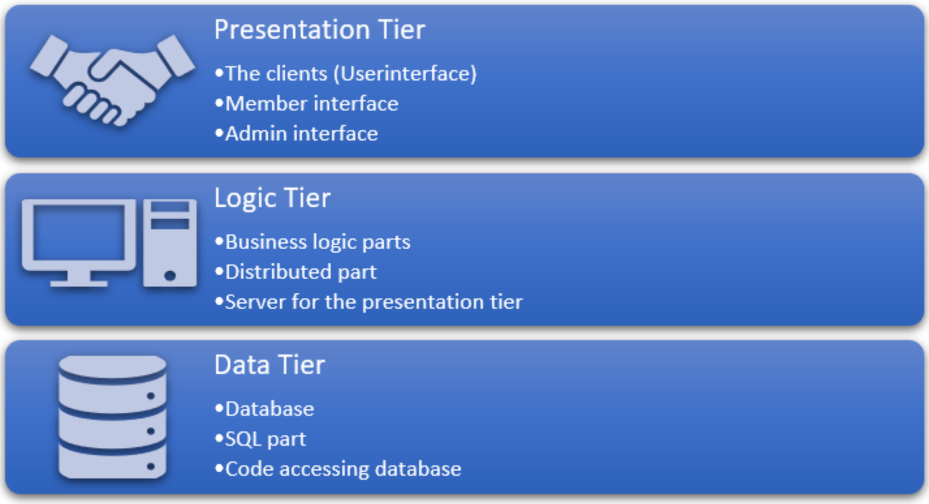
3 Project Initiation

The project initiation consists of three phases, group formation, project requirements and group tools. These phases are important to form an overview of the project initiation of group 8.

3.1 Project requirements & topic selection

Requirements for this project was to make a system consisting of 3-tier architecture and that is heterogeneous, meaning that it would be coded in different languages in this case Java and C#. Furthermore, consume and expose a web service, and a protocol should be designed and used for sockets. Lastly, a GUI should be made for each client.

When selecting a fitting topic that would encapsulate all the different topics, we got the idea for Fresh Fitness. The goal of this system is to create a three-tier distributed system with a center. The center should have different subscription types. A regular member, and a premium member that has different bonuses. The center's three-tier-architecture consist of:



Presentation Tier

- The clients (Userinterface)
- Member interface
- Admin interface

Logic Tier

- Business logic parts
- Distributed part
- Server for the presentation tier

Data Tier

- Database
- SQL part
- Code accessing database

3.2 Group formation

In this phase, the group was formed. The group formed was Jaser, Modaser, and Yasin. The group formation was based on the friendship and distance this these members had. This was convenient due to every member living in Viborg. When forming this group, the thought of being able to work together effectively was kept in mind. Furthermore, due to the distance, other class members were hesitant to work together with us. Another thought when forming this group was because of the good relationship we had. We all respect each other's idea and thoughts and were willing to listen to each other respectively. So, the threat of having internal group conflicts would be relatively low.

More knowledge about coding has been learned since last semester. Although we are still relying on the help from our teachers. We are also more experienced in how to handle projects compared to last semester, due to the challenges we faced. Overall, we as a team feel that we are better prepared to tackle this project, however the difficulty has increased a lot.

3.3 Group tools & planning tools

The tools needed to succeed in this project was given to us by the teachers. We had Astah for the making the diagrams, Eclipse and Visual Studio Code for the coding, and PGAdmin 4 for the database. Furthermore, we had a lot of PowerPoints, lessons and exercises from SDJ3, DNP, NES and SEP that we could use.

For communication we would meet up in the library of our home town or VIA. If we had to communicate from distance we all had each other's phone number. We would also use studienet/itslearning, Facebook or other social media.

For the planning tools we used SCRUM sprints to keep track of important dates for tasks. Furthermore, a calendar with important dates relating to the project. We also use studienet to plan supervisor meetings and would write all of them down in our logbook. As mentioned we had a logbook to keep track of the process and accomplishments alongside SCRUM. For the most part our planning was successful, but occasionally we would face obstacles such as sickness or ending up stuck on a task. Sometimes, we would end up not finishing all planned tasks. To overcome these obstacles, we would find alternative solutions or change plans.

4 Project Description

In the project description an overview of the problem was expected. This was done in the project description we have as an appendix. For the first phase an analysis of the problem should be made. The next phase is to define the problem. This was done with a purpose and a problem statement.

4.1 Purpose

A fitness-based registrations system where the users are able to keep track off and join different workout sessions depending on their subscription.

4.2 Problem Statement

The overall challenge is to create a system where users can have their own profiles, join workout sessions and keep track off upcoming sessions. The admin should be able to manage all this. This requires a lot of data to be transferred, stored and shared. The system will also be a distributed heterogeneous system. The system's components will be located on different networked computers, which then can communicate and coordinate their actions by passing messages. Furthermore, a 3-tier-architecture system needs to be implemented, which will use different languages.

Questions to be answered are the following:

1. How will the system be designed?
2. How will the system, which are coded in different languages be able to communicate?
3. How will the system be maintainable?

4.2.1 Delimitations

- The users will not be able to look at other users' profiles(privacy)

4.3 Project goal

When coming up with the goal of the system it was important that it was achievable, but at the same time challenging for us to overcome. For that reason, we made sure that every goal we set for ourselves had elements of stuff we already were taught at VIA. This was the thought process we had while deciding the goal for our project. Another important factor in deciding the goal was ourselves and our personal experiences and skills since it would not make sense for us to create an overly complicated system when we have no experience and are only a group of three individuals.

In the process, a lot of small goals were made leading up to the final product. SCRUM was used here separating the full project into smaller achievable tasks. SCRUM also helped us to narrow down what we would focus on and to make sure that we would do everything we could to achieve it. Of course, our SCRUM process was not perfect, and some tasks were not completed, but for the most part, our goals were reached.

5 Project Execution

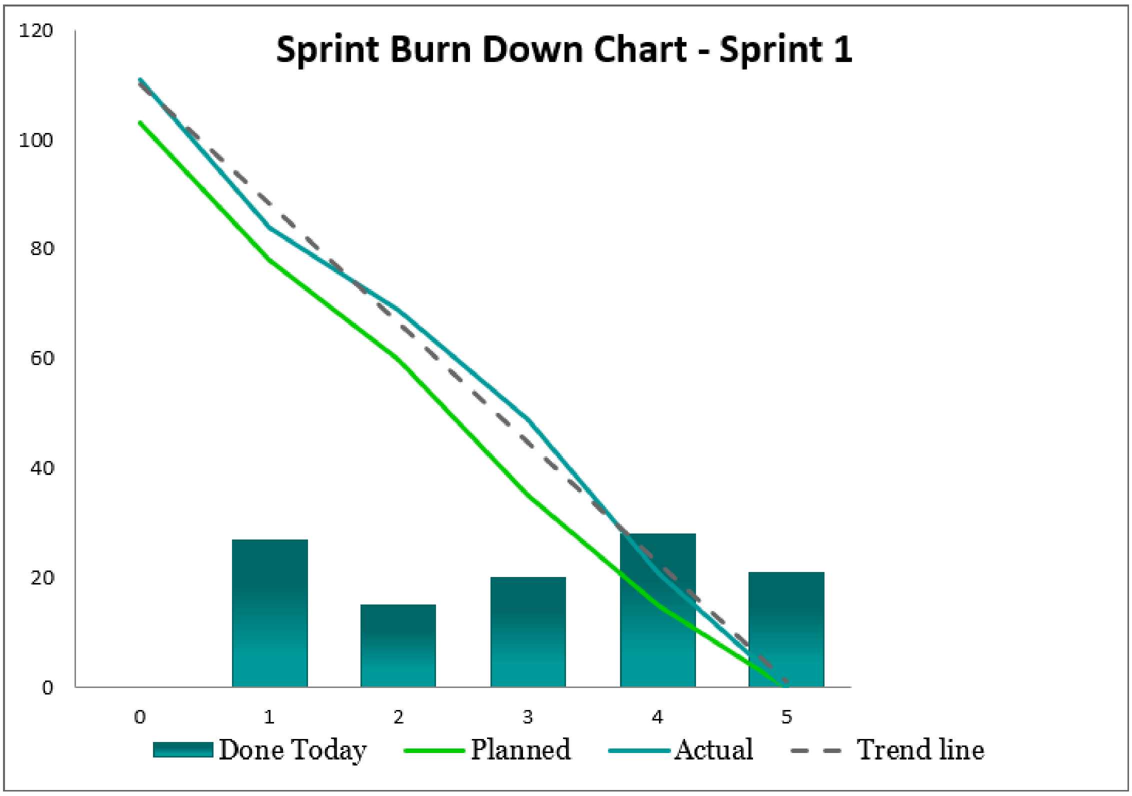
5.1 Method – SCRUM

To follow up on the project, the group decided to use SCRUM. This was done by separating the number of tasks into five sprints, that had different purposes. These sprints had some specific estimated hours, and the way the overall number of hours was calculated for the sprints was in the following way:

- Every student is supposed to spend 275 hours on the SEP 3 – 10 ECTS.
- Three students = 825 total hours.
- Project start 19/08 | Inception start 02/09 – Inception ends 25/10 | SEP ends 19/12
- 8 SEP days, eight hours for each SEP day.
- Project start till Inception ends, there was 8 SEP days (8*8*3 = 192) total.
- 825 – 192 = 633 total hours for all sprints combined.
- 633 / 5 because there will be five sprints = roughly 127 hours for each sprint.
- Each sprint will be five days, a total of 25 days.

Sprints	Task	Estimated hours
Sprint 1	Task 1 – Design/Implementation of model classes Task 2 – Second version of model classes and architectures design Task 3 – Design/implementation and of Database Task 4 – Continues work on database and testing queries Task 5 – Go back to working on model	Task 1 – 25 hours Task 2 – 18 hours Task 3 – 25 hours Task 4 – 20 hours Task 5 – 15 hours Total = 103 hours
Sprint 2	Task 1 – Continued work on java architectures Task 2 – Setting up server with REST API Task 3 – Setting up routes for the REST API and working on controllers Task 4 – Setting up routes for the REST API and working on controllers Task 5 – Testing and debugging	Task 1 – 25 hours Task 2 – 25 hours Task 3 – 25 hours Task 4 – 25 hours Task 5 – 15 hours Total = 115 hours
Sprint 3	Task 1 – Designing the GUI/client for C# and connect it to Task 2 – Begin implementation of GUI/client Task 3 – Continued work on implementation of GUI/client Task 4 – Testing and debugging Task 5 – Continued work on implementation of GUI/client	Task 1 – 30 hours Task 2 – 25 hours Task 3 – 25 hours Task 4 – 15 hours Task 5 – 25 hours Total = 120 hours
Sprint 4	Task 1 – Continued work on implementation of GUI/client Task 2 – Testing and debugging Task 3 – Continued work on implementation of GUI/client Task 4 – Testing and debugging Task 5 – Continued work on implementation of GUI/client	Task 1 – 25 hours Task 2 – 15 hours Task 3 – 25 hours Task 4 – 15 hours Task 5 – 25 hours Total = 105 hours
Sprint 5	Task 1 – Continued work on implementation of GUI/client Task 2 – Finishing with the last functionalities Task 3 – Testing Task 4 – Documentation of code and commenting Task 5 – Testing and polishing last bits	Task 1 – 30 hours Task 2 – 25 hours Task 3 – 25 hours Task 4 – 25 hours Task 5 – 15 hours Total = 120 hours

5.2 Sprints – Burn down charts



29/10 – We began designing and implementing the model classes in our domain model. The goal of this task was design and implement classes that would be central for the entire systems business logic.

30/10 – The architectures suited for the system was discussed. We also realized some mistakes in our previous domain model and came up with new designs choices. The goal of this task was to understand the architectures we would have to use in order to design our system. This was based on SEP meeting we had with one of our supervisors.

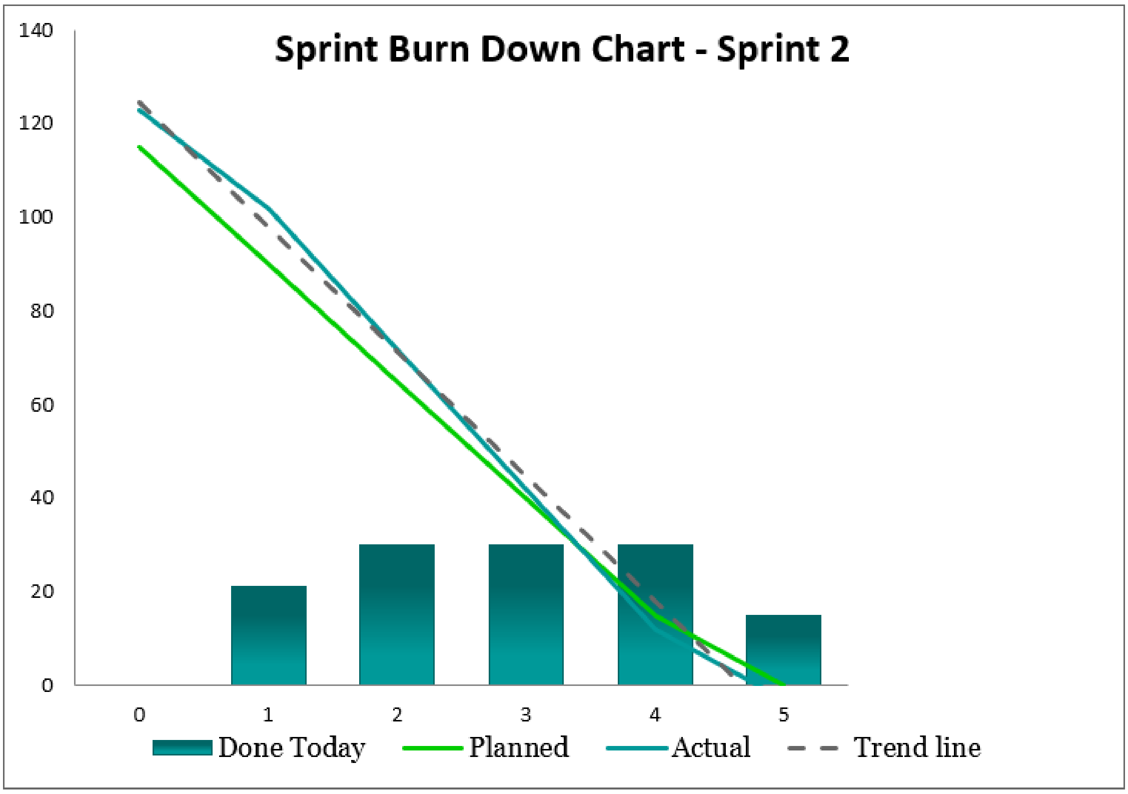
31/10 – Once the domain model was made, we also needed to make our ER-diagram this was done in this sprint. The goal of this task was to come up with a strong design for our database without using redundancy. Secondly, we began implementing it in our database.

01/11 – We continued to work on the 2nd version of the data modelling and then the domain model was made after realizing mistakes in the 1st version. We also implemented the database and tested queries that is relevant for our system.

02/11 – Implementation of the database. We went back and changed our java model classes to match with our database. The goal of this task was to sync the data from the database to the business logic.

Group retrospective

In this sprint we focused on making the fundamentals design and implementation for our domain model and data model. The first step was to create a good domain model that will encapsulate the core functionalities and business logic that our system would have. Secondly a good database that would have the important data for the system was made. We went back and forward to update our ER model and domain model. Overall this sprint went smooth without any major difficulties.



07/11 – Continued to work on java architectures with the new ideas in mind. We discussed how to make the system distributed with our supervisors. We came to the conclusion that we needed to make 1 system for each city and somehow connect them. A suggestion was p2p.

08/11 – In this day we did some research about Rest API and how to set one up for our system. Finally, we decided to go with spark framework. We also used the hikariaCP to connect our system to the database and fetch the data from our database. The goal of this task was to successfully setup a Rest API for our system and fulfill the requirements.

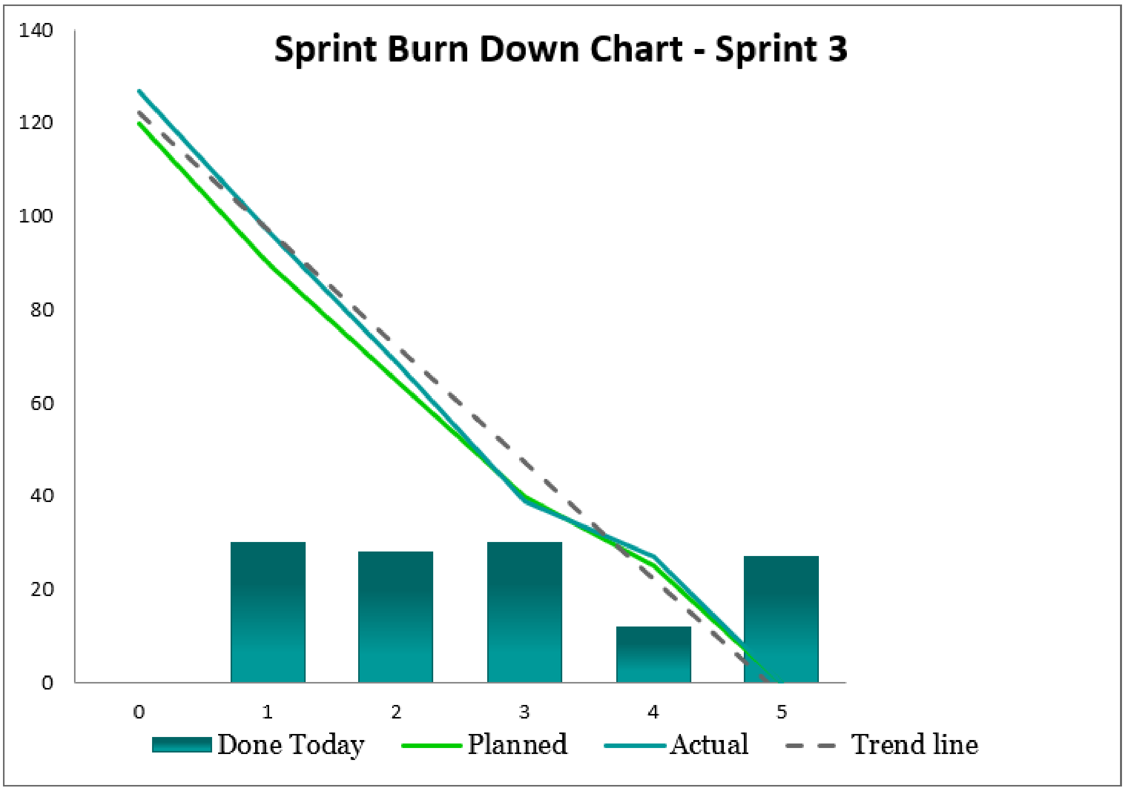
09/11 – In this day we made some routes for our API and we also started working on Controllers for our login in functionality. The goal of this task was to make some routes for our API that we could later use.

12/11 – In this day we continued to make some routes for our API. Overall, we continued the same work from the last task. We finish the login controller using the spark framework, but the GUI was still not implemented yet.

13/11 – This day was spent testing and debugging for errors in our system. We setup some break point to identify the errors and continued to fix them.

Group retrospective

In this sprint the goal was to setup the REST API, and start make some routes in order to fulfill one the requirements, which was to use webservices. Since this was our first time setting up an API a lot of time was spent researching the most efficient way to accomplish this. We stumbled across the Spark framework, which was convenient for us. Secondly, we also used the HikariCP to connect to our database and it worked well with this framework. Lastly, we worked on some routes for our API that we could use. Overall this was one complicated sprint, but we eventually reached our goals.



22/11 – In this sprint we started working on designing the GUI/client for our system this was done in C# and the goal was to connect our client to the API. We visualized different ways the system would look and started to work on it.

23/11 – Began implementation of GUI/client. Since C# was a new language for us the GUI programming was difficult and a lot of time researching, and studying was put into it. Luckily it had resemblance to java. And it also helped that a drag and drop functionality was available. The login functionality was finished in this day.

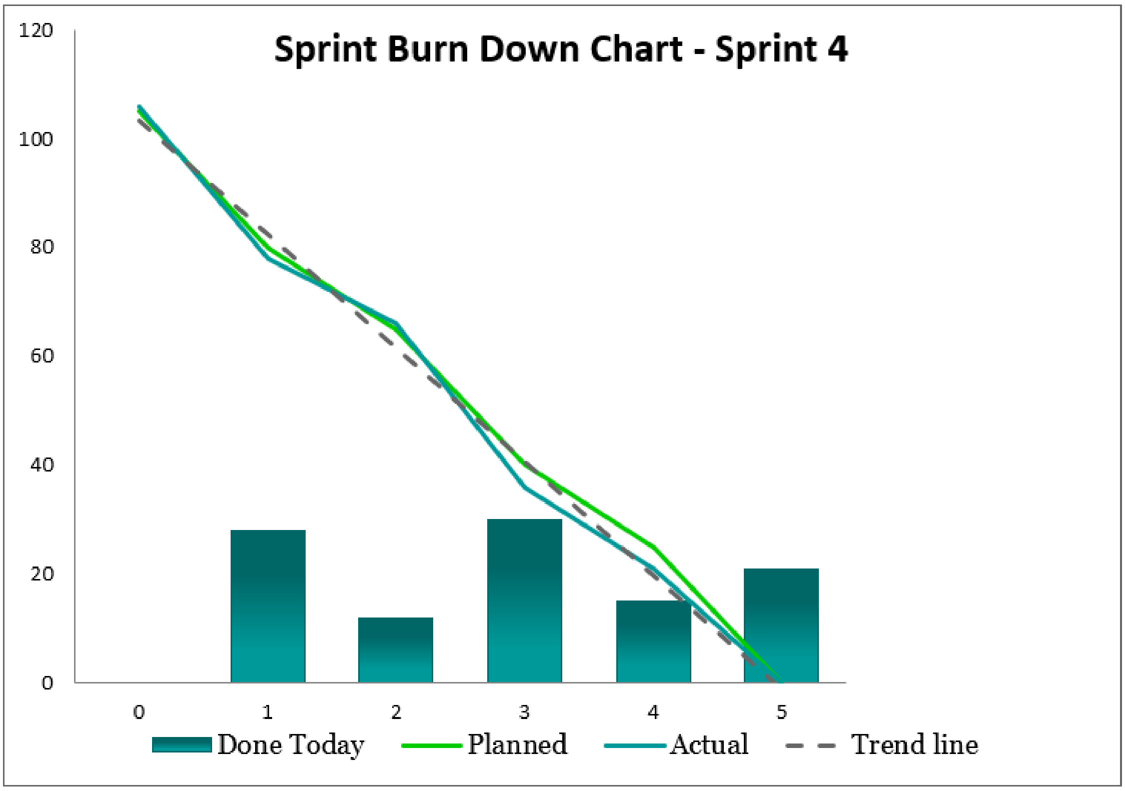
26/11 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to add some sort of functionality to our GUI. The first functional aspect was a login functionality. We setup some tabs and started to working on the first tab, which was the user profile.

27/11 – This day was spent testing and debugging for error in our system. We setup some break point to identify the errors and continued to fix them. The first tab for users was finish in this day.

28/11 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to continue adding functionality to our GUI. We began working on the second tab, which the workout tab.

Group retrospective

In this sprint the goal was to make the GUI for our system. In order to do that one of the requirements was that we should be coding using C# to make the system heterogenous. To accomplish this client should have some sort of the connection with the system in Java, therefore and we used our API so that the client could request data from the server and the server coded in java could response. This was done using routes and controllers. This was the most complicated aspect of system, requiring us to spend a lot of time making the GUI work. In conclusion we reached our goal and fulfilled our tasks, however a lot of worked remained in order for us to meet all the software's requirements.



03/12 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to continue adding functionalities to our GUI. We continued working the workout tab.

04/12 – We set this day aside to be spent on testing and debugging for errors in our system, but we also continued to code functionality going back and forward. The goal of this sprint was to identify errors and try to fix them. Initially we wanted the info from the tab in a text box, but we later realized that it would be easier to simply use table. We went with this design and finished the workout tab.

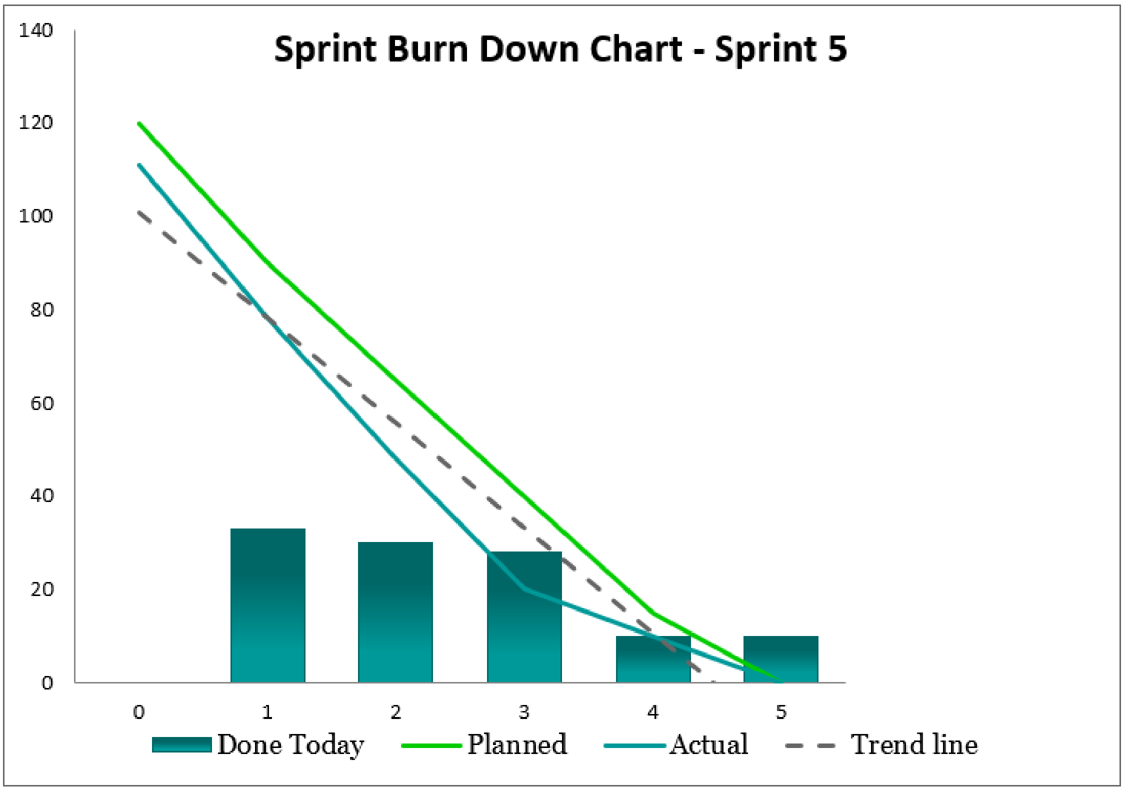
05/12 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to continue adding functionalities functionality to our GUI. We began working the activity and finished it.

06/12 – We set this day aside to be spent on testing and debugging for errors in our system, but we also continued to code functionality going back and forward. The goal of this sprint was to identify errors and try to fix them. In this sprint we started working on the admin interface, so we began working the tab for managing users.

07/12 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to continue adding functionalities functionality to our GUI. We continued working on the tab for managing users.

Group retrospective

This sprint was mainly set aside to complete the necessary functionalities for the GUI. In other the words the majority of the sprint was spent making controllers, routes and then establishing the functionality to the GUI. Furthermore, a lot testing and debugging was done, since we ran into errors/problems along the way that needed to be fixed. Overall, this was a difficult sprint for us, and we felt a lot of pressure however we were able to overcome it.



10/12 – In this sprint we continued to work on implementation of GUI/client. The goal in this sprint was to continue adding functionalities functionality to our GUI. We finish the manage users tab.

11/12 – In this sprint our focus was to be finished with our last functionalities for the GUI. We had some errors, but otherwise most of the requirements were done. We began working on the manage activities tab and finished it.

12/12– The goal of this day document all our code and make sure the it was commented and easy to understand. This accomplished before time.

13/12 – The goal of this sprint was polished, in other clean our as much as possible from irredundant code that isn't used.

14/12 – The goal of this sprint was polished, in other clean our as much as possible from irredundant code that isn't used. Report writing.

Group retrospective

This sprint was mainly spent working on finishing our system whether we could fulfill all the requirements or not. We had two errors that we struggled with. One was that we couldn’t add an activity and the other was that we couldn’t delete an existing activity. So, we focused on finishing these errors. Secondly, we also spent time documenting our code and commenting it. Overall this sprint went well since we finished all of requirements.

6 Scrum Reflections

Modaser Ghasemi ***(Product owner)***

As a product owner I had the responsibility to choose which functionality was required. That was a hard task because I did my best to get everyone’s opinion in on it. And decide from there if a functionality should be implemented in our system. Another part was that I also was responsible for the requirements for the system. I needed to be realistic by setting our goal. Since the three of us lacked experience with code. So, in this way I had to prioritize the task from high, medium and low.

Jaser Ghasemi ***(Scrum Master)***

I personally felt there was a lot of pressure on me when it came to keeping the guys motivated and not distracted. We knew each other, and we had a close friendship, so it was easy to get distracted, but despite that I decided to enforce our rules and act upon my role as a Scrum Master. My goal was to make sure that my group was focused on

the most important features for our system (highest priority tasks), at first this was forgotten, but later we decided to start and prioritize the tasks.

I was also responsible to make sure that the daily scrum meetings were scheduled, and I gave the group members individual tasks when they had completed their previous tasks, this was obviously done during the daily scrums. My responsibility wasn't only scheduling meetings, but also making sure that everything we did was written down, so we could write it into our reports at a later stage. This helped us a lot because it gave us a better overview of what's there and what's not.

Yasin Issa Aden ***(Development Team)***

As one of the developers I was responsible for making sure that the team would be able to do the task in the sprint and it would be completed on time. However, this was far from possible all the time. Despite every member being responsible for working on the sprint backlog I was mainly responsible for continuously holding the daily meetings with the SCRUM Master(Jaser) where we would discuss the progression of the sprint, what we completed and any obstacles that we faced.

As the developer I was also responsible for estimating how the long sprints would take, and inform the product owner with realistic estimates for the task. I also work closely with the SCRUM master with organizing the task and informing him about realistic estimates for the sprints.

7 Supervision

The supervisor meetings went very smoothly for the most part. All the group members were present for most of the meetings. The supervisors were very helpful for the progression of the project. They were always willing to sit down with us to help. The replies from the emails were also relatively fast, and it was very easy to setup meetings with them. Our group needed mostly needed help with design and analysis and the coding was mostly done by ourselves.

Supervisor meetings with Ole

04/10 – We discussed the possible architectures that we use for our system. Mainly, how to achieve a 3-tier-architecture and to make our system distributed. Ole gave a nice overview and illustrated on the black how the system could be designed.

05/12 – Consulted Ole with our current progress and what we need so far. We got advised to mainly for focus and adding functionality for our system, so we had something to present.

Supervisor meetings with Christian

04/10 – We discussed the possible architectures that we use for our system. Mainly, how to achieve a heterogenous system. Christian suggested that we make the client in our language and the server in another. For example, the client could be code in C#.

05/12 – Consulted Christian with our current progress and what we need so far. We got advised to mainly for focus and adding functionality for our system, so we had something to present.

8 Conclusions

In conclusion the process of SCRUM itself went well, since we met all our task and was able to complete them. This however took a loot of hard work and we felt we were under a lot of pressure. We took our experience from the last SEP and used to learn from our mistakes and hence we started the implementation phase as fast as possible. We spent every sprint on primarily implementation.

Some of the things that went well in our SCRUM was writing the reports alongside each phase. So, the analysis part of the report was written while working on the analysis and so on. Nevertheless, that meant that the reports would have to be updated when our system changed. Overall this SEP went well in our expectations, but we can always improve for the next time. For example, our estimations for each task wasn't always right. Some days we were done before and other days it took more time. Overall, we can draw a lot of lessons and experience for future projects, because of our experience with SCRUM.