Process Report

Bachelor Project – Electricity Recording Assistant

ERA

Electricity Recording Assistant

Supervisors:

Poul Væggemose

Students:

Gais El-AAsi - 279910

Marcel Valentijn Daniel Notenboom - 279963

Number of characters incl. space, own figures and tables (800 char/figure) and footnotes: 32.164 characters

Software Engineering Bachelor Project - VIA University College Campus Horsens

17th of December 2021

TABLE OF CONTENTS

List of figures

Introduction

The purpose of this paper is to describe the process of completing the bachelor project for the Software Engineering program. It covers the timeline from the project initialization and to the finish of the project, including learnings for the creators.

The project started in the 6th semester with the formation of the group and definition of the group contract that should govern through the entire project period. Following, was the definition of the project idea, that was elaborated into the project description.

Starting September 2021, the more substantial work on the project began, that went through the 7th semester. This is the period that created most of the artifacts, including project execution, time allocation and planning, obstacles, learnings and supervision.

The paper elaborates on the software development frameworks used and how they assisted us in completing the project's requirements. A chapter dedicated to what went *wrong* is present as there are many things to learn from, that will assist in our future career as software engineers.

Next, we will present some of our reflections of the project. It will include both reflections as a group, towards the project, as well as individual reflections. Additionally, a presentation of how the supervision helped and assisted us throughout the project execution.

Finally, a presentation of the learnings gained throughout the project will be summarized.

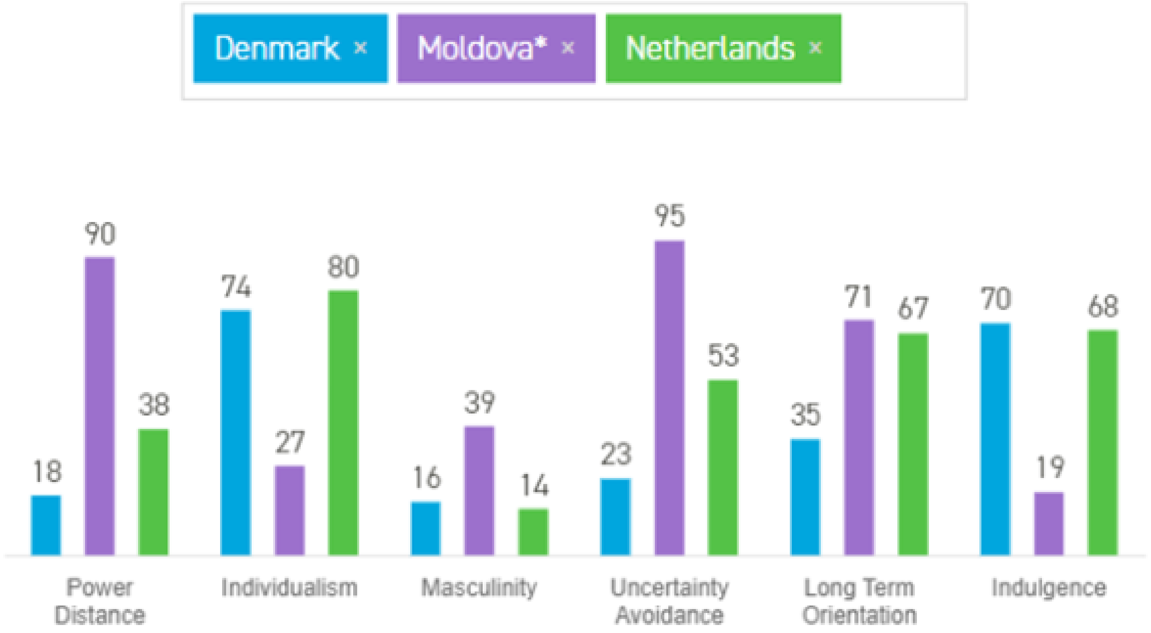
Group Description

Hofstede's Cultural Differences

To understand better how the group is going to perform, a comparison of the differences in the culture will be done. The group is formed of the members – Moldavian and Dutch (Hofstede Insights, n.d.).

At the same time both members of the group have been living in Denmark for some time (5 and 6 years) and both members were greatly influenced by the Danish culture, therefore we included it along with the specified above cultures, the Danish one, which will act as a normalizer.

Figure 1 - Hofstede Cultural Differences



Power Distance

There is a noticeable difference from Romanian (90) and Dutch (38). Even if considering the Danish culture influence, when analyzing the Power Distance dimension for the group's members, it can be clearly seen that in the Romanian culture there is a greater acceptance of the fact that the power is distributed in-equally. To combat that, the Section 3 of the Group Contract addresses the power distribution in the group.

Individualism

Even if there is a considerable difference between Dutch (80) and Romaninan (27), when adding the Danish (74) culture as a normalizer, the difference decreases. When looking from the project's perspective, there is a risk of some members working individually, and not developing the project as a group. To prevent that, timely meetings will be scheduled to ensure a proper collaboration between the members, and eliminate the risk of members working alone.

Masculinity

When analyzing this dimension, it can be seen that the difference between cultures decreases. With Dutch being 14 and Romanian 42. Even if this dimension still represents an important factor in creating a balance in the group, a great difference does not directly affect the quality of the work, as long as both sides strive towards the common good of the project. Both sides of the dimension are important and diversification of the group can be transformed in an advantage if correctly harvested.

E-Stimate Personal Profile

The members of the groups had the next results for the e-stimate (E-Stimate, n.d.) personal profile:

- Gais El-Aasi – predominantly blue, followed by red and green, with no yellow;
- Marcel Notenboom – perfect balance between blue and red, with no green and yellow;

As can be observed the group member’s personal profiles are no extremely diversified, which results in the group needing to be more aware about certain aspects during the development of the project. Both members have strong orientation towards organization and initiative, with little dominance of creativity and mediation.

This can be a perfect formula for constant conflicts without resolution. But given the previous experiences (working on semester project for the past 6 semesters) and very similar work style will alienate some of the potential issues. In addition, having a set of clearly defined rules – Group Contract – and unspoken rules – created during previous experiences – will ensure that all the conflicts can be resolved. Even so, some attention will be required in the creativity dimension of the project to ensure that the project is innovative enough and that the members do not get lost in technical details. In this sense the supervision is a perfect addition to keep the balance.

Group Contract

A group contract, that was refined over the years of working together was created that shall govern to provide a common ground and set of rules for the group’s members. It does not focus on limiting or imposing obstacles in how the members work together but rather focuses on facilitating team-work and a good work environment. The group contract can be found in Appendix A.

Project Experience

From the first semester, both members have been in the same group – with other members on and off. In this sense, it means that we both have experience in working with each other. This allows for aligning when it comes to the expectations for this project, quality and style of work, needed effort and dedication. In addition, having previous work experience build trust between the members, which allows us to easier *count* on each other and trust that we are acting in good faith and towards the success of the project.

Project initiation

The project was initialized during the 6th semester starting with group formation. Given a good collaboration in the previous semester, the decision of working together was quite easy. After some deliberation whether to bring some potential members in, it was decided that having a greater man power and a more diversified group could work towards the benefit of the project.

Unfortunately, it was not the case, as there was no alignment between other potential members and groups' expectation for the effort and quality of the outcome, so the decision was to maintain the current group's members.

Once settled on the definition of the group, we starting out some potential ideas for the project. Each member submitted sever potential ideas. After a session of brainstorming potential spinoffs of the ideas and weighting pros and cons for each idea the decision was made to continue with the current idea. Some of the pros included:

- Good mix of areas that we have some experience in;
- At the same time, it pushed us out of the comfort zone with some more challenging technical details;
- Having a hardware attached to the project which we both are enthusiastic about;
- Potential for future development from project to product.

The last step was defining some tools that will assist us in collaboration and work. These tools were chosen to ensure that, even if we have to resort to work remotely, we can maintain the pace of the project:

- Git – version control to allow easier code sharing and version control;
- GitHub – remote repository that we can both access via internet;
- Discord – communication tool;
- Google Drive – for storing document and project related files. It allows for real-time co-writing, changes tracking and version control;
- Google Sheets – tool used for planning and organization;

These tools were sufficient to allow us having a good team-work and collaboration on the project during the entire period.

Project Description

Project Description was the phase of turning our idea into something more tangible and substantial. We made some formal definition of what we are interested in, what we are trying to solve and how we are going to do it. Even if we both had a good understanding on what the idea was, putting it on paper in a more formal way, helped us to understand better the idea as well as some of the initial obstacles. In addition, it helped us to align so that we are sure that we have the same understanding.

Another important consideration was having a formal description of the problem that we are trying to solve, as that was our guiding principle throughout the project. Moreover, we needed to make sure that we are aware on what we cannot achieve and make sure that we delimited those items, so that we had a clear image of the scope of the project.

The next step was to define a timeline and put some milestones that would allow us to measure our progress and to make sure that we stay on the track in how we spend our resources. Having a timeline of the project, allowed as to define a more detailed plan when the project execution started still maintaining the initial milestones that were defined.

As a last step, we have looked at some potential risks that could already at that phase be identified. Even if it is impossible to predict all the risks, identifying at least the major ones, by defining some of the identifiers that can be watched to see if the risk is bound to happen, helped us in making sure that we can avoid unwanted obstacles. In addition, having a contingency plan, in the case that a risk would happen gave us a reassurance that we are in control of the situation.

All in all, the project description was a great tool to make sure that we are both align and answer some important questions that are essential to the project’s success:

- What it is that we want to do?
- Why are we doing this?
- How we are going to do it?
- What is the plan, timewise?
- What potential risks are we going to encounter, and how are we going to overcome them?

Having answered the questions above, we were ready to start the actual project process. The *Project Description* can be found in Appendix B.

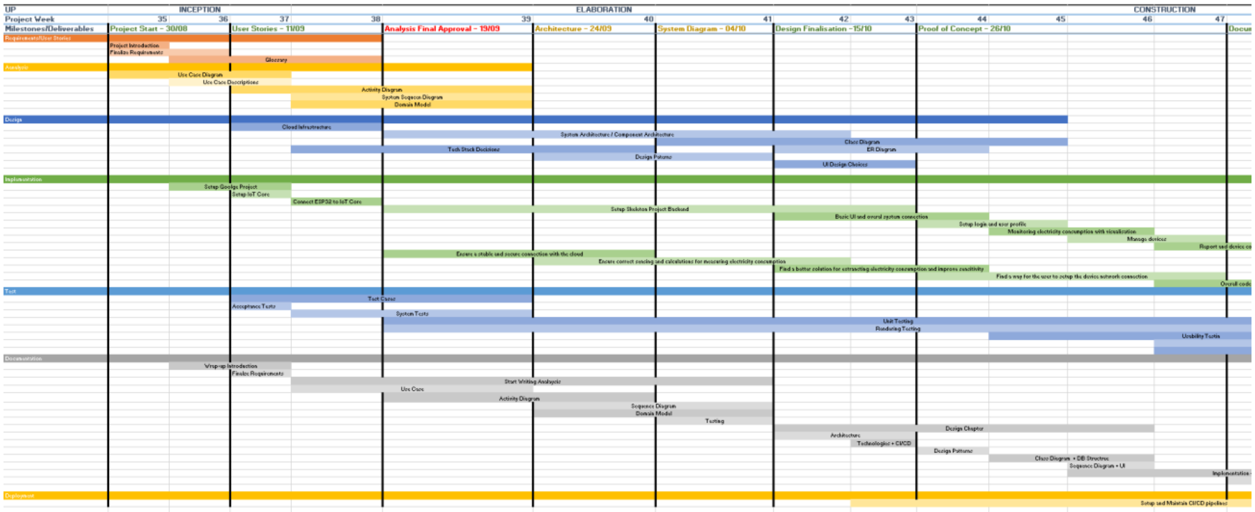
Project Execution

The project execution constitutes the period from September 2021 until the end of the project on 17th of December same year. It included the main work on the project, starting with the Analysis, Design, Implementation, Testing and Deployment. Of course, the process did not go in a straight line from A to be, but rather in iterations since we were using an agile software development process. Each iteration being a small traversing of each phase.

In our execution we used a few tools to assist us with the development process. Each tool has had a great importance and impact on how the project was unveiling.

Unified Process

The unified process has helped us make sure that we focus on the right things in a structured way. Even if at any time, we were working on all the phases, having the UP framework helped us ensure that the majority of the effort is spent doing the phase's appropriate tasks.



The figure above is intended to showcase how UP was used from a bird's view, the actual file is included in the Appendix C and can be drilled down to observe the details.

But as can be seen, each column of the file represents a week, and each row color represents a phase in the development process:

- Orange – Requirements/User Stories
- Yellow (1) – Analysis
- Blue (1) – Design

- Green – Implementation
- Blue (2) – Testing
- Gray – Documentation
- Yellow (2) – Deployment

Each phase than consisted of several *swim lanes* that contained a large task, that was further broke down internally.

On the top, along with each column that represents a week, there are the dates for each milestone that were separated with the vertical bolded lines as well as the UP-model phases.

Based on the phase that we were in, we tried to focus on the appropriate swim lanes:

- Inception – Requirements and Analysis
- Elaboration – Design
- Construction – Implementation and Testing
- Transition – Documentation

In this way we were sure that our software development process had a logical flow and that each topic is covered accordingly.

Kanban Software Development

Kanban Framework helped us stay agile and deliver continuously. We have worked to get a CI/CD pipeline as soon as possible together with a proof of concept, which allowed us afterwards, only to work on new features and enable our setup for continuous delivery. In that sense, we were not focusing on an *iteration* but tried to deliver items in a continuous manner. The helper document for utilizing Kanban can be found in Appendix E.

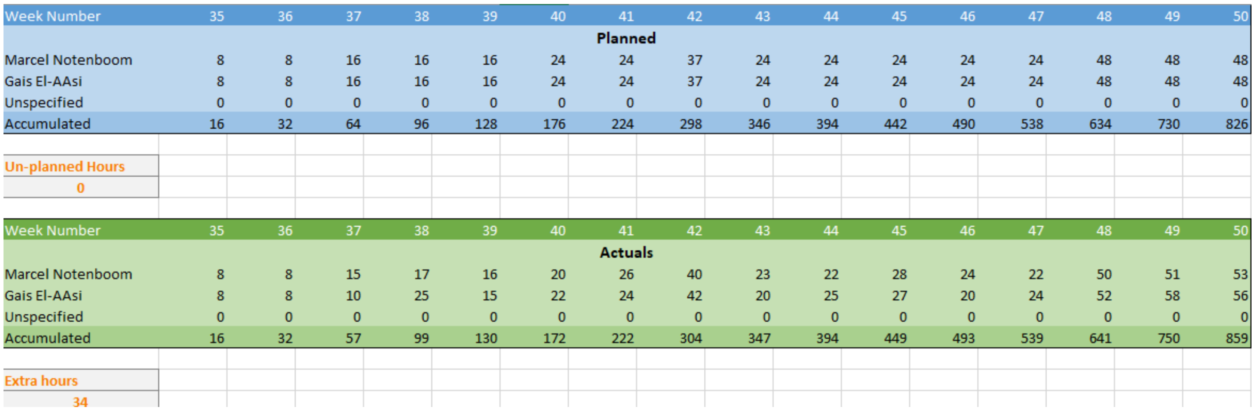
With that approach a few obstacles existed that needed to be taken care of:

- How do we ensure that we stay on track? – for that the milestones were constantly checked and we worked hard to ensure only small deviation that can be caught up;
- How do we ensure that the team is aligned? – timely status meetings – at least two times a week first only the developers and the second one with the supervisor, allowed us to make sure that we collaborate and are aware where the project is at;

- Work in progress – principal was used, that did not allow us to work on more than 2 swim lanes at the same time, to ensure that we do not get swamped with work in progress without finalizing it;
- Every time a swim lane was finished a review of the work was made to ensure that it is of appropriate quality;

To ensure that we are aware of the resources (time) available in the project a project allocation was compiled and maintained to make sure that we stay on track with the resources burned for the project.

Figure 2 - Project Allocation



Week Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
							Planned									
Marcel Notenboom	8	8	16	16	16	24	24	37	24	24	24	24	24	48	48	48
Gais El-AAsi	8	8	16	16	16	24	24	37	24	24	24	24	24	48	48	48
Unspecified	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accumulated	16	32	64	96	128	176	224	298	346	394	442	490	538	634	730	826

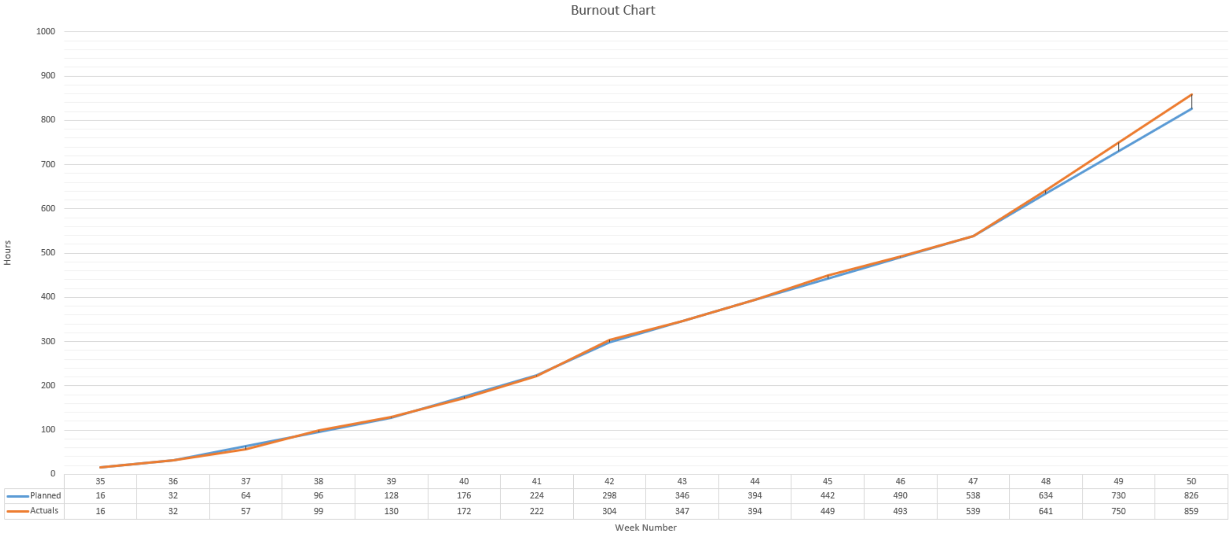
Un-planned Hours
0

Week Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
							Actuals									
Marcel Notenboom	8	8	15	17	16	20	26	40	23	22	28	24	22	50	51	53
Gais El-AAsi	8	8	10	25	15	22	24	42	20	25	27	20	24	52	58	56
Unspecified	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accumulated	16	32	57	99	130	172	222	304	347	394	449	493	539	641	750	859

Extra hours
34

And a visual representation of the time spent on the project.

Figure 3 - Burnout Chart



Overall, we maintained the number of hours allocated to the project as planned with a slight ramp up at the end of the project, due to less time needed to be spent on other courses.

Project Obstacles

During the project execution we encountered a few obstacles that had an impact on the development process. Each of these obstacles had to be overcome in a way or another to ensure a successful completion of the project.

Hardware

ESP32 is an experimental development board, which means that it is not production ready an might come with some hidden issues. In our case we had some issues with the board not sensing correctly the raw data sent by the ACS712. The initial assumption was that the issue occurred because of the current divider circuit, as the ACS712 outputs 5V but the ESP32 can only receive 3.6V, unfortunately after debugging and double checking the issue was not with that.

The second assumption was that initially we opted for a 20A sensor that are not as sensitive as a 5A sensor, combined that with the splitting of the current we assumed that it just removed too much information from the raw data. After replacing the 20A with a 5A, the issue still persisted.

A third try to solve the problem was to use a *SCTT013* as a dedicated current sensor coupled with the *ZMPT101B* as a dedicated voltage sensor. Unfortunately, even in that setup the readings were not of any quality.

After troubleshooting with an Arduino counterpart, the issue was not present, which meant that the issue occurs only with the ESP32 even if running the same code. After some research of the forums that are dedicated to troubleshooting the exact model that we used in the project, the version that we had was known for having issues with the Analog to Digital converter port.

The solution was to utilize an Arduino Uno as the measuring unit for the ACS712 and then to pass the data from the Arduino to the ESP32 via a *UART* connection.

Cloud SQL Access

Google Cloud SQL has a very restrictive access policy. It is based on a list of whitelisted IP address, and a set of credentials for the database. While working in locally, it was not a big issue as it was possible to whitelist our IP address that would allow the application to login using the give credentials.

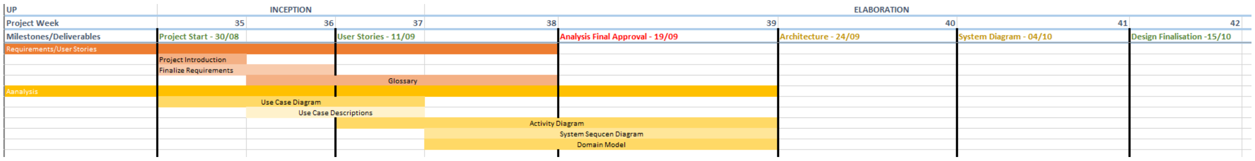
Unfortunately, when deployed on an App Engine service, there are multiple instances that can be scaled out or down, in that case it was cumbersome (and hardly achievable) to white list the IP address every time a new instance was spined out.

The solution was to utilize an API called *Cloud SQL Proxy* offered also by Google, that allowed instead of connecting directly to the database, to connect to a proxy that then allowed the connection to the database. The benefits of this setup were that it was possible to provide access via the *Identity Access Management* portal for the App Engine without compromising the security of the database.

Deadline

During the project execution a few deadlines were missed due to an optimistic approximation. Because one missed deadline acts as a domino chain, causing the other to be delayed, missing one deadline caused a negative on the following deadlines. It caused us to shift the focus on ensuring that we can catch up with the deadline, and it took two more deadlines until we were able to be back on track.

The first deadline to be missed, was the approval of the Analysis, which we spent a larger number of resources than expected. Even so, we needed to make sure that we have finalized the Analysis as good as possible as it was a core foundation of the project. This caused the Architecture and System Diagram to be delayed, only to catch up on the Design phase.



All in all, even if we had some missed deadline, we were able to re-group and focus our resources in such a way that allowed us to catch up with other deadlines.

Project Critique

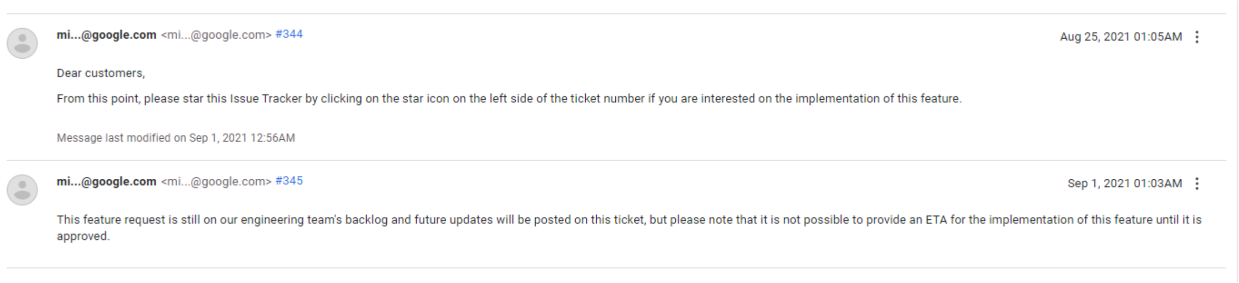
This chapter will include some of the critiques that we have towards the project. It is our own feedback on what could have been handled in a better way. We have realized these facts during the execution of the project, but were unable to correct without affecting the project in a drastic way.

Timeseries Database

When the decisions for the persistent storages were made, the initial decision was to use PostgreSQL for its robustness combined with *Timescale DB* an extension of the PostgreSQL that allows for indexing time series data. By doing so it heavily increases database performance, which has a great impact on the user's experience. Unfortunately, after setting up the database and everything around the Google

Cloud Platform we realized that Cloud SQL – the service for holding the PostgreSQL database – does not implement the *Timescale DB* extension. This makes our queries quite slow which affects the overall performance of the system. As can be seen the last update is on 1st of September when their team still did not approve the progress of the extension.

Figure 4 - Google Issue Tracker - Timescale DB



mi...@google.com <mi...@google.com> #344 Aug 25, 2021 01:05AM

Dear customers,

From this point, please star this Issue Tracker by clicking on the star icon on the left side of the ticket number if you are interested on the implementation of this feature.

Message last modified on Sep 1, 2021 12:56AM

mi...@google.com <mi...@google.com> #345 Sep 1, 2021 01:03AM

This feature request is still on our engineering team's backlog and future updates will be posted on this ticket, but please note that it is not possible to provide an ETA for the implementation of this feature until it is approved.

To handle this, we would have needed to either change the type of the database to a *Wide Column* database that offers a much better support for timeseries data usually used for IoT data. Another option was to change to another database provider that offers possibility for *Timescale DB* – like Azure. Or define our own database setup in a virtual machine within the same cloud provider.

SignalR Technology

The initial idea of presenting live data for the user was using a *pull-and-push* method. In that setup the backend would continuously pull sensor data directly from the *Pub/Sub* and push it to frontend using the *SignalR* technology. In some sense building a continuous *conveyor* like belt, where data was processed as it passed through the system. Unfortunately, both the continuous pull technology from *Google's Pub/Sub* service, and the *SignalR* proved to be unstable and with a lot of overhead.

Because of that, valuable resources were spent trying to make *it work* at the end going with a simpler and more stable method of *push-and-pull*. The data was pushed to the backend as it arrived in the *Pub/Sub* via a function, where the backend could distribute load, process data and upload it to the database. At the same time, the frontend would pull data, on a user-defined rate getting the latest data available in the database. This resulted in a slight delay in data-to-user time, but in a much stable and balanced system.

Nevertheless, going initially for this setup would have saved us resources that could have been invested in adding more features to the system. While it is always good to check and try new technologies and approaches, we knew from the beginning that our approach was a bit experimental, even if it sounded *cool* on paper.

Personal Reflections

In this chapter the project participants will elaborate on their personal reflections towards the project.

Marcel Valentijn Daniel Notenboom

The bachelor project gave me a very free feeling because we did not have any limitation on what we wanted to do as long as we could get it approved. The fact that the bachelor group was made of people with the same aspirations with a good working and problem-solving relationship was done deliberately. The decision was made to avoid adding stress and conflict to a project and ensure a smooth communication in the current covid affected climate.

Luckily, we came to a project idea quite fast and it seemed as we were in line with the expectation of what we wanted the end product to entail. During the project, we had a good division of labor where we both did the section that we were passionate about and had some knowledge about. As well contributed together to the common part.

Because I could focus on the part of the project that I preferred, I was motivated to keep overcome more difficult obstacles. Even though we were separated in the parts of responsibility and workload, having often status and alignment meetings, helped us stay on track with the project as an overall. Moreover, being able to communicate freely, ask for advice, brainstorm ideas, and perform *rubber duck* exercises made the project being even more enjoyable.

Because of the solid requirements to make the different parts work together we had little to no issues when integrating it in the whole. It was a great idea to work on the communication proof of concept quite early on, which allowed me to provide Gais with a testing device which provided dummy data to the service for testing and development applications. At the same time, it ensured that we had the main infrastructure setup and could focus only on adding new features.

Even though I had some serious issues with the different level of voltage of the components that turned out to be a tough nut to crack, I did find a solution after deliberation with Gais, which we were both happy with. I would have liked a longer development time for this project. It would have allowed me to go from a rough working prototype to a production prototype. The main areas that I was eager to improve were: making a custom circuit board in a smaller form factor and crafting a proper casing to provide more safety since we were working with high voltage.

The supervisor teacher provided valuable feedback on the areas we could improve. He was always interested and eager to hear about the improvements that we have made.

Gais El-AAsi

The bachelor project was a great experience. The single most important factor that made it so, was having a great and dependable team member. Having trust at the base of our collaboration allowed me to be able to focus my energy on the project, as opposed to, trying to *fix* unnecessary and pity conflicts. Even if we lacked manpower, having a good synergy and energy, allowed us to bring the project to a good stage, whilst improving our soft and hard skills.

The second factor, was having the freedom to both pick the project idea as well the means to achieve it – starting from technology and ending with the software development process. Even if the freedom was a reinvigorating feeling (as compared to previous projects) it came at a cost – decisions.

Each time I needed to pick something over something else, it was a decision that we needed to be conscious about. I think that this was a great experience as it allowed me to explore different options and self-argue for which would be better and why. Trying to answer those question, offered valuable knowledge that is highly appreciated in the industry. Additionally, it made the project seem more personal, rather than just a *school* project.

Another essential factor was supervision. Our supervisor was close to the project, offering guidance and helping us to have a better direction of where we should go. At the same time, he offered us the freedom in making our choices – as long as we could argue for them.

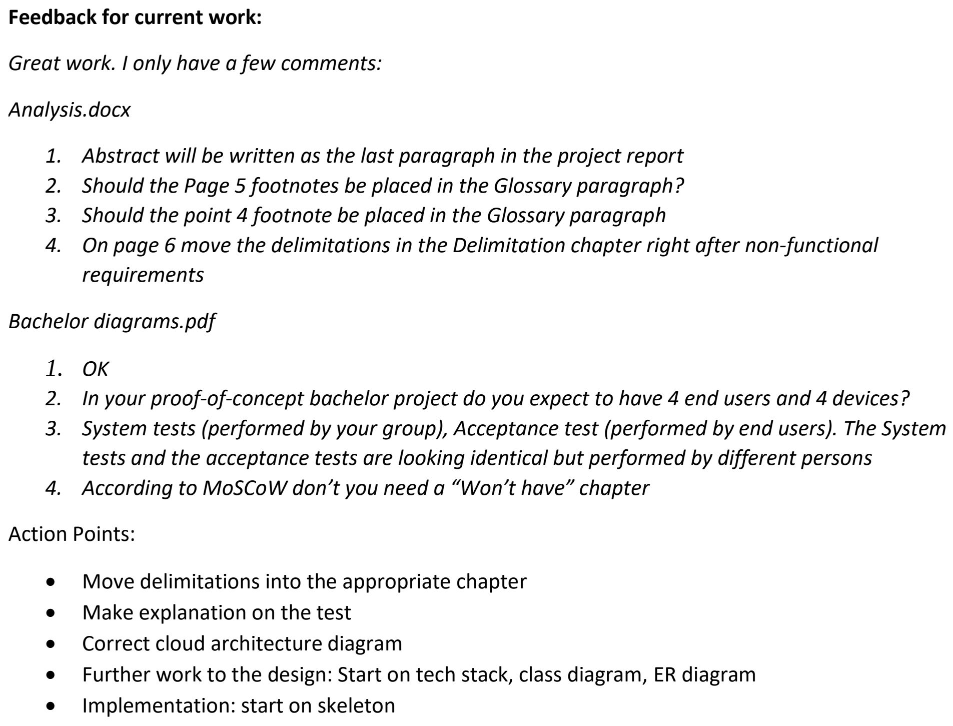
All in all, I consider this project to be an important milestone in both my personal and technical development.

Supervision

Throughout the project our supervisor has greatly helped us shape this project. He was always quick to offer feedback – both positive and negative – to ensure that the project stays on track and has good progress. He kept us motivated, and challenged us to push as hard as we can, whilst making sure that we have a good state of mind.

The supervision was done both through meetings and written feedback. During the project period, we have had a total of 12 meetings. In these meetings we discussed the current progress on the project and the future steps, resulting in action points for the project. Many times, rather than directly suggesting what we need to do, he helped us discover the direction ourselves through good questions and discussions. All the meeting notes can be found in Appendix D.

Figure 5 - Example of meeting notes



Feedback for current work:

Great work. I only have a few comments:

Analysis.docx

1. *Abstract will be written as the last paragraph in the project report*
2. *Should the Page 5 footnotes be placed in the Glossary paragraph?*
3. *Should the point 4 footnote be placed in the Glossary paragraph*
4. *On page 6 move the delimitations in the Delimitation chapter right after non-functional requirements*

Bachelor diagrams.pdf

1. *OK*
2. *In your proof-of-concept bachelor project do you expect to have 4 end users and 4 devices?*
3. *System tests (performed by your group), Acceptance test (performed by end users). The System tests and the acceptance tests are looking identical but performed by different persons*
4. *According to MoSCoW don't you need a "Won't have" chapter*

Action Points:

- Move delimitations into the appropriate chapter
- Make explanation on the test
- Correct cloud architecture diagram
- Further work to the design: Start on tech stack, class diagram, ER diagram
- Implementation: start on skeleton

To conclude, we are very grateful for all the support and help that resulted from working together with our supervisor. The project would not have been even close to the current form if it was not for all the guidance and feedback that we have had during the supervision sessions.

Conclusions

The overall process of this project has greatly contributed to our professional development. Even if there were some obstacles during the development of the project, the process was overall smooth, with a steady and constant progress. Each step/phase of the project has helped us get closer to its current shape and offered great lessons to be learned.

The group synergy was great, both of us contributed with their best efforts to the progress of the project, while maintaining a good collaboration environment that encouraged knowledge sharing and team-work. The supervision helped us stay on track and make sure that we are in control of the situation on every level.

At the end of this project, we have learned a few lessons that can be summed up in a set of learnings that we will try to keep in mind in our future projects:

- Choosing the team members makes a great difference, it is important to make sure that the members are aligned on the expectation and on the level of effort one is willing to put in. Do not create teams based on friendship and/or first impression. While having a good *chemistry* is important, there are more important factors to consider;
- Always document Minutes of Meetings, it will provide valuable lessons and make it easier to conclude and extract learning from them. Do not rely on memory during the meetings, while it is important to stay focused during a meeting, which sometimes can be an impediment on taking notes, overall having written notes has more benefits in long term;
- Spent the time on the analysis and design, while many people (including us) tend to go as soon as possible for the implementation, having a good analysis and design of the project, and keeping them in the back of the mind will offer a solid foundation and common understanding on what we are trying to do;
- Start early and put the maximum amount of effort in getting the initial skeleton/proof of concept/infrastructure as soon as possible. After having a good foundation with the analysis and design, the next big step is to prioritize the requirements, extract those that are at the core of the solution and work really hard on them. Having the basic infrastructure in place, is one of the most important steps in the implementation part;
- Understand that this is a team effort. Having a good collaboration and knowledge/problem sharing benefits greatly both on the individual level as well as group level.

- Do not be afraid of critic, both to receive and give. It is important to allow a free, un-obstructed communication within the team, including valuable feedback that only helps the project get to a better level;

Sources of information

E-Stimate. (n.d.). *About e-stimate*. Retrieved from E-Stimate: https://www.e-stimate.dk/en/moed-teamet-bag-e-stimate/

Hofstede Insights. (n.d.). *Our Models*. Retrieved from Hofstede Insigts: https://hi.hofstede-insights.com/models

Appendices

Appendix A – Group Contract

Appendix B – Project Description

Appendix C – Planning

Appendix D – Meeting Notes

Appendix E – Kanban Cheat Sheet