Blekinge Institute of Technology

Institution of Computer Science

PA1414

Individual Report

Ali Reza Nazari
2022-10-27

Table of Contents

Project Overview

How I started the project and why

In this section om going to explain how I started the project and the reason behind doing so. I started the project by reviewing the course material, reading the task description and requirements, analyzing the requirements, creating a questionnaire that included all the questions regarding the requirements which were ambiguous, establishing a contact portal with the customer, updating the requirements list with the new information that I got from the customer, researching the ambiguous concepts, and creating a product backlog.

Reviewing the course material was vital because I needed to brush my knowledge and make sure that I understood, was ready, and had all the necessary theoretical tools to get started with the project.

Reading the task description and requirements was important because it gave me an overview of what the task was about and which requirements it had. Moreover, analyzing the requirements was also important because it is important to make sure that the customer and I were on the same page.

After reading and analyzing the task description, created a questionnaire for the customer to get clarification on the unclear requirements. Later, I established a communication portal with the customer through email to both send the questionnaire and have a communication portal for the future.

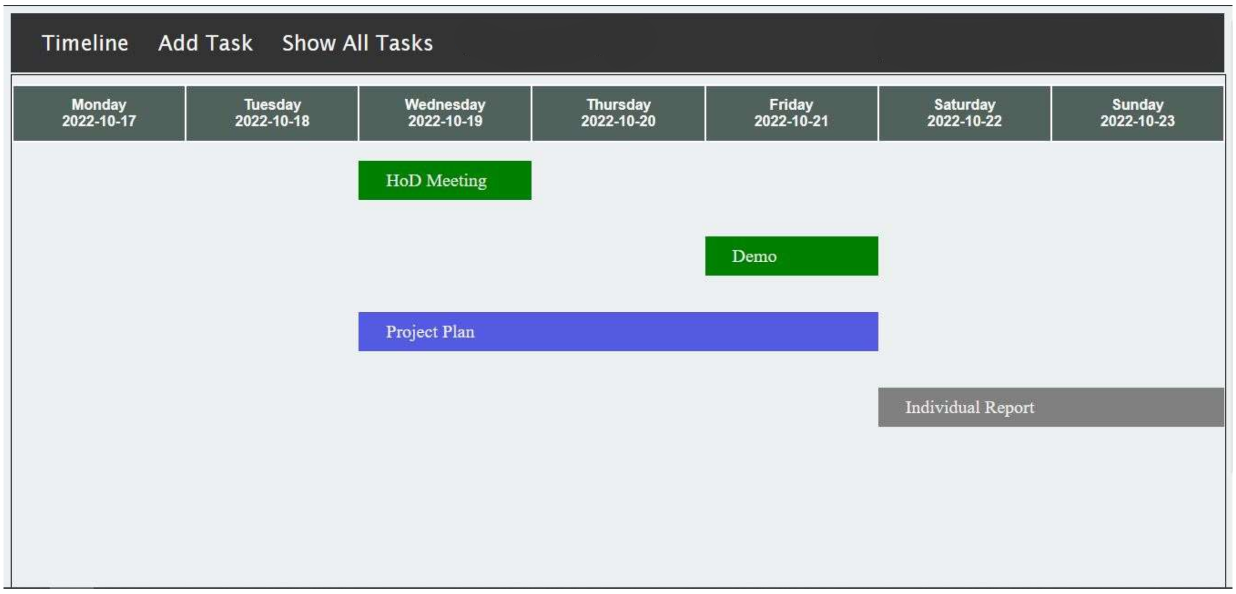
Furthermore, I created a revised version of the task description and requirements based on the answers to the questionnaire that I got from the customer. I did so because it was important to make sure that the customer and I were on the same page. By being on the same page, I could develop a product that the customer wanted.

At this point, I had reviewed all the course material and had built an understanding of what the task was about and what the customer wanted, but when it came to my knowledge, I still had not a firm grasp on the concepts of how the whole project planning, management, and execution worked. To fix that, I started researching on google to make sure that I knew what I needed to do and most importantly what I was doing. I started my project this way because it is important to develop what the customer wants; having happy customers leads to new/more opportunities including new customers, a good reputation, and so on.

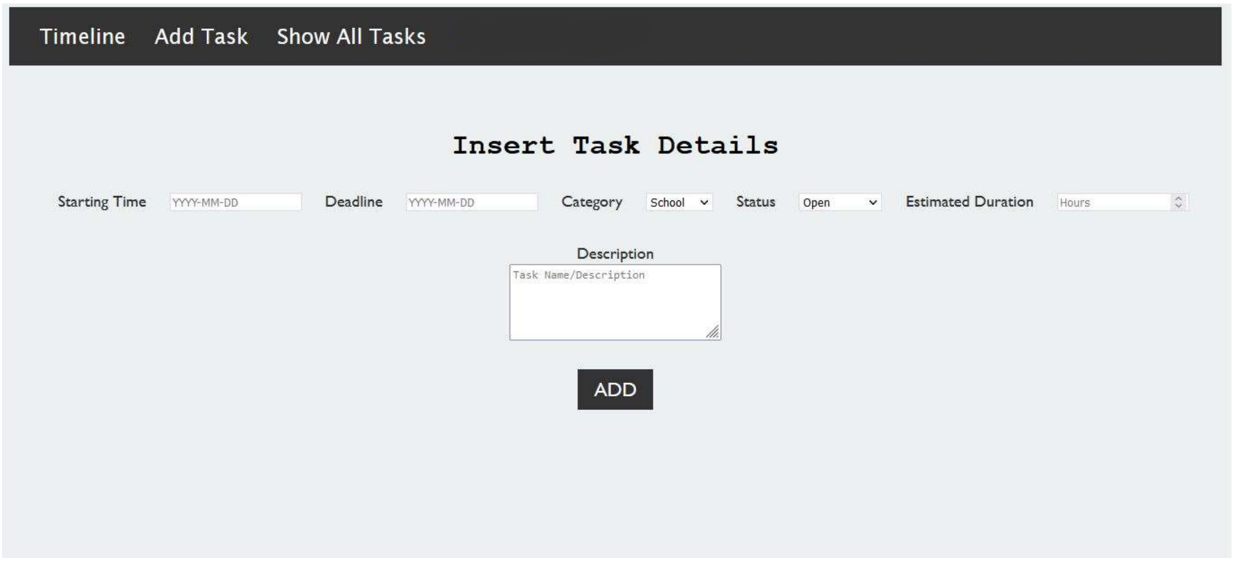
Task Overview

In this section, I am going to write explain what the customer wanted. The customer wanted/ordered a web application that helps people to schedule and estimate the time of completion for a specific task. A user would enter the task details, and the software will estimate when the task can be completed based on the user capacity, tasks in the pipeline, and task duration. The application should also visualize the tasks in the pipeline. Moreover, the software should display all the tasks in a timeline, show a list of all the tasks, and have a search function. After reading the task description and requirements, I pictured the application to look something like the pictures below.

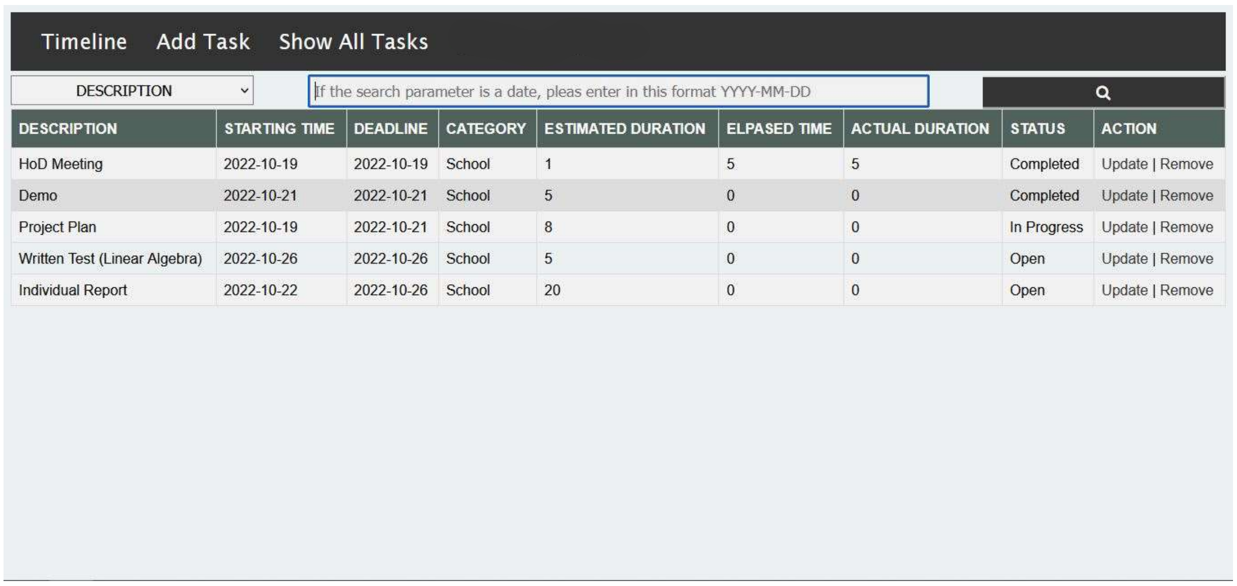
Home Page/Timeline



Add a Task



Show All Tasks



Timeline Add Task Show All Tasks

DESCRIPTION | If the search parameter is a date, pleas enter in this format YYYY-MM-DD

DESCRIPTION	STARTING TIME	DEADLINE	CATEGORY	ESTIMATED DURATION	ELPASED TIME	ACTUAL DURATION	STATUS	ACTION
HoD Meeting	2022-10-19	2022-10-19	School	1	5	5	Completed	Update | Remove
Demo	2022-10-21	2022-10-21	School	5	0	0	Completed	Update | Remove
Project Plan	2022-10-19	2022-10-21	School	8	0	0	In Progress	Update | Remove
Written Test (Linear Algebra)	2022-10-26	2022-10-26	School	5	0	0	Open	Update | Remove
Individual Report	2022-10-22	2022-10-26	School	20	0	0	Open	Update | Remove

Requirement Handling

In this section, I am going explain the actions I took to refactor the requirement list and list down which requirements did the customer have. First, I started to analyze each of the requirements that the customer had published on Canvas. During the analysis phase, I checked/evaluated the following aspects of each requirement:

1. Is the requirement concise; do I understand what it implies?
2. Is there any ambiguity in the phrasing of the requirements; does it make sense?
3. Can it be implemented during the project's timeframe?

Secondly, I created a questionnaire with all the questions that I had regarding the requirements after analyzing them. I proceeded to send the questionnaire to the customer and waited for his response. After receiving all the answers, I updated the existing requirement list which looks like the following:

1. Product Requirements

RQ1: A user should be able to schedule a task.

RQ2: The task can be started and completed (scheduled) based on user capacity, current tasks, and task duration. The user enters the capacity (unit of measurement: hours).

RQ3: A user should be able to see the tasks as a timeline (unit of measurement: week).

RQ4: A user should be able to read the list of all tasks and their current state (Open, in progress, done).

RQ5: A user should be able to search current task lists by all fields in RQ8 to access tasks quickly.

RQ6: A user should be able to keep track of the elapsed (unit of measurement: hours) task time.

RQ7: A user should be able to schedule tasks in parallel.

2. Data Requirements

RQ8: A task should contain the following fields (more fields can be added)

Field Name	Datatype
Id	int

Description	string
Category	string (select from list)
Starting time	Date Time
Deadline	Date Time
Estimated duration	int
Actual duration	int

3. Performance Requirements

RQ9: The software should process requests within no more than 2 seconds.
RQ10: The software should run on a laptop with an Intel Core i7 processor (or equivalent) and 16GB of RAM.

5. Bonus (optional):

RQ11: Export the tasks as an excel sheet

RQ12: Export the tasks as a PDF file.

Choice of Technical Solution

In this section, I am going to write to explain if I have used a certain tech stack and why I chose them. My choice of technical solution/ tech stack was the following:
Tech Stack, I had worked with before:

- Maria DB
- Express.js
- Node.js
- Ejs
- HTML
- CSS
- Npm
- Cygwin
- Nodemon

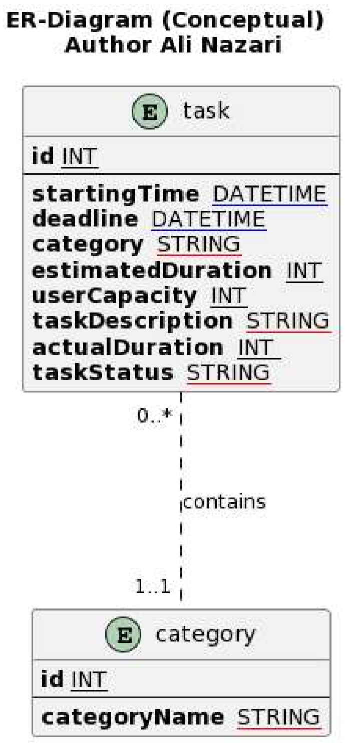
Tech Stack, I had not worked with before:

- Morgan

I chose Morgan for testing the response time of the web application. The main reason behind choosing Morgan over other Node.js libraries was its simplicity to use, it its ease of implementation.
Since I had experience with developing a web application from a previous course, called databases and web technologies, I chose to work with the tech stack I had learned during that course. The main reason behind choosing the tech stack above was to play it safe since the focus of the course was on learning the concepts of agile project planning, management, and execution. To make a long story short, I choose the tech stack above because I had already worked with them, and I wanted to focus my effort on learning how to run a software development project and the time frame of the project was limited.

Design

I used the entity relationship diagram below for structuring the different database tables for the web application.
It consists of two tables, a task, and a category.



Coding

I used a JavaScript back-end framework called Node.js with UTF-8 encoding and 4 spaces indentation. When it comes to the code standards and practices, I followed the following:

1. Component-based Project Structure.
    a. I divided my code base into smaller components to ensure that the code base is scalable, maintainable, and clean.
2. Use a D-facto standard
    a. I used it for checking possible code errors and fixing code style, not only to identify nitty-gritty spacing issues.
3. CamelCase naming standard for naming the variables

Testing

I only conducted manual testing and test-driven development. Under development, I used to test each line so that it functions as.

- Unit Testing
- Component Testing
- Integration Testing

My goal for unit testing was to isolate my code and test it to determine if it works as intended. In other words, I conducted unit testing to detect early flaws in code which may be more difficult to find in later testing stages.

Moreover, my goal in component testing was to find bugs at a very early stage; to make sure that a group of code works well as a component. For example, a function that retrieves and categorizes all the different task details from a user input consists of many smaller parts such as function definition, local variables, for loops, conditional statements, and so on. To ensure that the function works as intended we need component testing.

My goal in integration testing was to make sure that different components work and cooperate as a group to perform a certain task such as displaying all the tasks in a timeline that has many components such as information retrieval from the database, sending the retrieved data to the right route, and so on.

Quality of the product

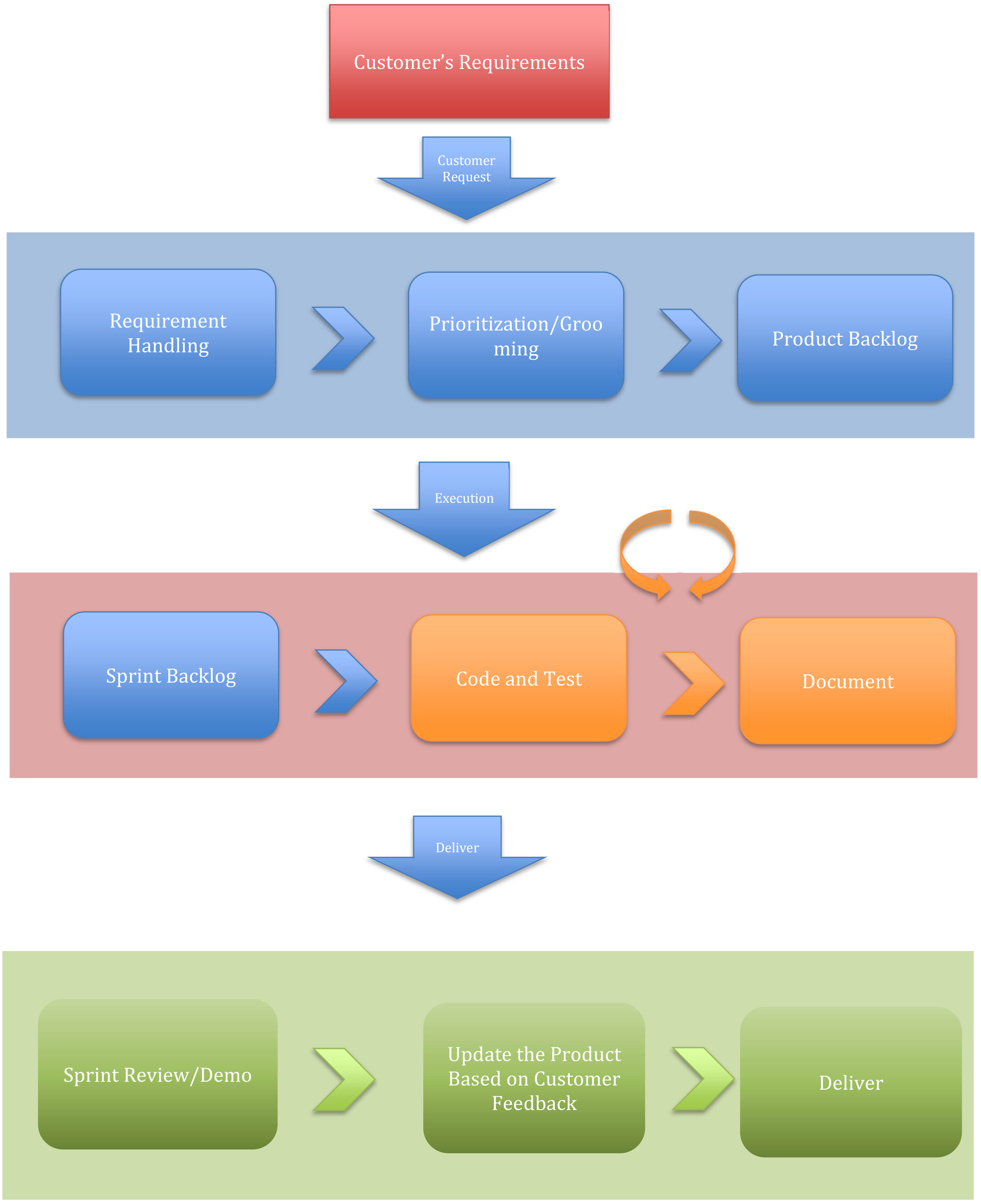
Based on my experience in web development, the quality of my product is quite good. When it comes to the user interface, it has an appealing design, and is user friendly; the user can get started without needing to go through a long manual, high performance; it processes the requests in under 1 second.
I will not have any problems if I get asked to show my product to you; I would show it by recording a video or deploying it to a server or showing it in person.

Customer Relation

The relationship between the customer and me was professional and good. It was good and professional because we had established a communication portal via email and most importantly, I had made sure that the customer felt heard and that his feedbacks were valuable to the development process.

Process

To complete the project, I followed the process below.



Challenges

During the project’s run, I faced many challenges each week. For the sake of this report, I am going to list them down below and I am also going to explain what I did to tackle them.

Week 36

Establishing a good contact portal and getting the customer involved in the project was an inevitable challenge. To solve this, I kept my emails short, and concise and not overwhelmed the customer with many questions.

Week 37 and 38

Being consistent with the sprint plan and doing deep work was one of the challenges ahead during these weeks because I was usually wasting many hours just by getting distracted from what I should be doing such as going to YouTube to watch a lecture, but I ended up clicking on random videos which were not work-related. To tackle this, I utilized a study with-me video which was four hours long on YouTube. The video helped me not to work until I was burned out and to perform deep work.

Week 39

Showing and implementing the task in a 30 days timeline was one of the biggest challenges during week 39 week since I did not have the technical expertise. To tackle it, I tried to rely on external research such as google to find the necessary help and improve my technical expertise.

Week 40

Assuring that the application will meet the performance requirements was one of the challenges in the upcoming sprint 3. To tackle it, I relied on external resources such as Google and YouTube to broaden my technical horizon. Moreover, implementing the feature where the app estimates a starting time for a given task was also challenging as well. To overcome it, I tried to discuss it with my classmates and find out how I could implement it as well.

Progress Evaluation

In this section, I am going to write about how far I completed the project. I expected to implement 10 of the 12 requirements posed by the customer. Among those requirements the two below were optional.

- Exporting the tasks list as a pdf
- Exporting the tasks as an excel sheet

Based on my expectation, I can argue that I have implemented the core functionality and features of the web application which implies that I reached my goals.

To make a long story short, my project’s result was in line with what the customer had asked for.

Reflections

In this section, I am going to look over what I would have done differently if I started over. Since my expertise was limited to front-end web development, I wrote all my front-end code in vanilla JavaScript, basic HTML, and CSS. It led to one major problem meaning implementing a timeline became a nightmare in a way that I had two write 1578 lines of code to just implement a 7 days timeline whereas my classmates had implemented the same functionality by writing less than 100 lines of code.

If started over, I would have learned and used a front-end framework such as React.js, divided the code base into even smaller chunks, not forgotten to do a risk analysis, and met face-to-face with the customer to discuss the requirements rather than communicating via email.

Learning and using a front-end framework would have saved me a lot of time and would have given me the ability to design an even better user interface that was more interactive, interesting, and added functionality. While watching my classmates who had used frameworks present their demo, I noticed that they had implemented far more functionalities than me. For example, one of my classmates had implemented a login functionality that made his app more personalized, and another classmate had implemented a feature where you could see all the task's information if you hovered over it.

Towards the end of sprint 3 i.e., the last sprint, I noticed that scaling/adding/implementing new features was a little harder and confusing especially in the database. For example, while performing unit testing, I had to scroll over hundreds of lines of code to get to the bottom of the page where I had my new function. Considering that, I would have divided my source code into even smaller modules. By doing so, I would have made my code and product more scalable and scalable.

When it was time to write a project plan, I came across a section which was about risk analysis. If started over, I would have made sure to do a risk analysis before starting the project. By doing so, I would have made sure that I was ready for unexpected events which could have affected the project negatively. I was lucky that all went well during this project, but I will not rely on luck rather than being prepared.

Lastly, I had a face-to-face meeting with the customer during the last week of sprint 2. During that meeting, I got the most valuable feedback on the requirements for refactoring which I could not have gotten over email. If I started over, I would set up a meeting with the customer and seek answers to all the questions that I have. By doing so, I would make my work so much easier and more effective because I would have known what the customer wants from the start; it would have resulted in less deviation from what the customer wanted and what I implemented.