3D Motor Expolorer

Firas & Slim

June 28, 2020

Contents

List of Figures

List of Tables

Chapter 1: Presentation

1.1 Introduction

The following work is part of our graduation project for the applied bachelor degree in Information System Development at XTech . This internship tasked us with the realization of a web VR application named : Motor Explorer . This Chapter will contain the hosting company introduction followed by primary studies we ran before getting to work on the project , leading into detailed explanation of our solution and eventually we will talk about the methodology used to realize this project .

1.2 Hosting organization Presentation

Figure 1.1: XTECH Logo

xTECH is an is an up-and-coming tech company based between Berlin and Tunis, developing web applications and cloud solutions. They help connect talented developers with innovative clients in order to create high quality solutions.

1.3 Preliminary studies

In this section we will discuss the origins of webVR and the pros and cons about webVR in general.

1.3.1 WEB VR

1.3.1.1 Definition and history

WebVR was an experimental JavaScript API that allowed web applications to interact with VR devices like the “HTC Vive” , “Ouclus rift” , “Google cardboard” and such . It was first conceived in early spring 2014 by “Vladimir Vukićević” from “Mozilla” and on March 2016 “Google Chrome” and the 2 “Mozilla VR” team announced the version 1.0 which by April 2017 was upgraded to version 1.1 and by than companies like Microsoft joined in on the project and are actively collaborating on the 2.0 version [2]

1.3.1.2 Goals

The API was designed with these goals in mind:

- It works with all OS and devices
- webVR development is cheaper than native app for a single device
- A webVR App can be easily integrated into companies
- Can be used without installation
- WebVR Apps are updated instantly and no need to go though approval process

1.3.1.3 Pros and cons

Pros	Cons
• It works with all OS and devices • webVR development is cheaper than native app for a single device • A webVR App can be easily integrated into companies • Can be used without installation • WebVR Apps are updated instantly and no need to go though approval process	• Does not allow intense or complex 3D rendering • It is difficult to navigate the app without the use of provided VR controllers • Currently , webVR apps are still not as popular as native VR apps • Requires more technical skills (Javascript in particular) experience than regular native VR apps

Table 1.1: Pros and Cons of webVR

1.3.1.4 Conclusion

The trend is currently noticeably towards the WebVR app when it comes to developing enterprise VR apps. This development is correct in that it puts more emphasis on the price / performance ratio and less on the graphics quality. 3 WebVR apps are easy to integrate into existing IT structures and reduce manual adjustment work for new end devices.

1.4 Project goals

The project's goal is to help design a virtual reality environment which a student or a worker can interact with on a physical level and understand the mechanics behind the work they are tasked to do, it is meant to enhance the level of understanding by providing a close to reality experience .

1.5 Proposed solution

Instead of using written documentation or 2D platforms, the user will have an actual interface in which he can interact with and get a better understanding and feeling to the experiment he is conducting and enhance the learning capacities of students and give workers a better environment to work in Therefor we will create a 3D simulator with sounds and visual 3D objects that will keep the user entertained during his immersive experience while conducting the task needed to be done and such solution will contain :

- Development of a 3D environment
- Development of 3D motors
- Development of a 3D Dynamometer
- Development of a interactive Control-Pannel
- Development of real-life 3D animations

1.6 Related Works

1.6.1 Motor explorer (original application)

Motor explorer is the original application from which we have chosen to migrate from a 2D platform to a 3D efficient platform for multiple reasons in 4 which we will mention below the different aspects from migrating from a simple web solution to a complicated web VR environment in which the user will have a more understanding of the purpose of the application. Motor explorer is used to measure torque in a simulated experience the 2D interface doesn't offer much details toward this suggested test and this application is used for only educational purposes as there would be no professional aspect of the application

.

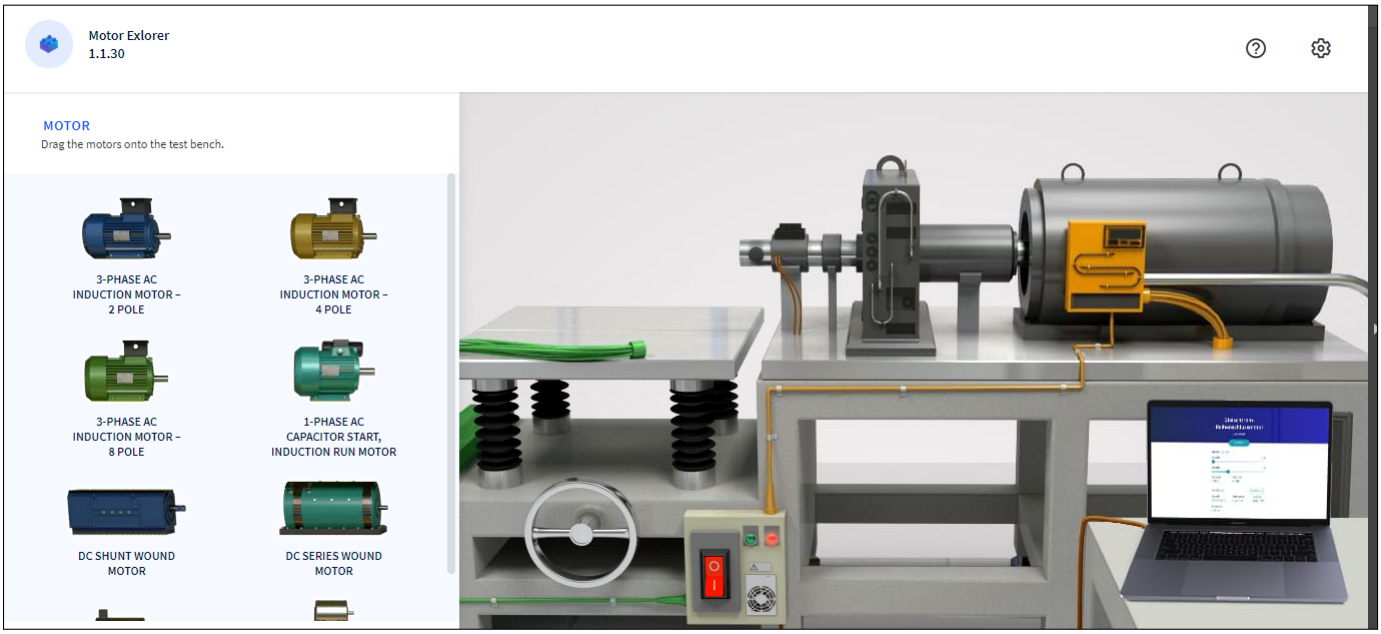

Figure 1.2: 2D Motor Explorer

As you see from the previous screenshot it's the simple interface of the motor explorer original application

Motor explorer 2D	Motor Explorer web VR
Simple 2D interface	3D detailed interface
Only drag-drop interactions	User can interact in anyway with the interface
No real-life experience can be felt during the tests	Tests are conducted in a real-life like experience
Only for educational purposes	Can also be applied in professional use

Table 1.2: Comparison between the 2d app and the 3d app

1.6.2 ESTPE VR

ESTPE VR is designed for universities, training centers and organizations providing professional development for the personnel at electric high-voltage substations. The main goal of this work is to improve the quality of training in electrical engineering via implementation into the educational process a Virtual Reality simulator developed specifically for the power industry of which we mention

- Transformer Oil Sampling In Virtual Reality

The participants are able to perform field-job and familiarize themselves with a transformer oil sampling procedure.

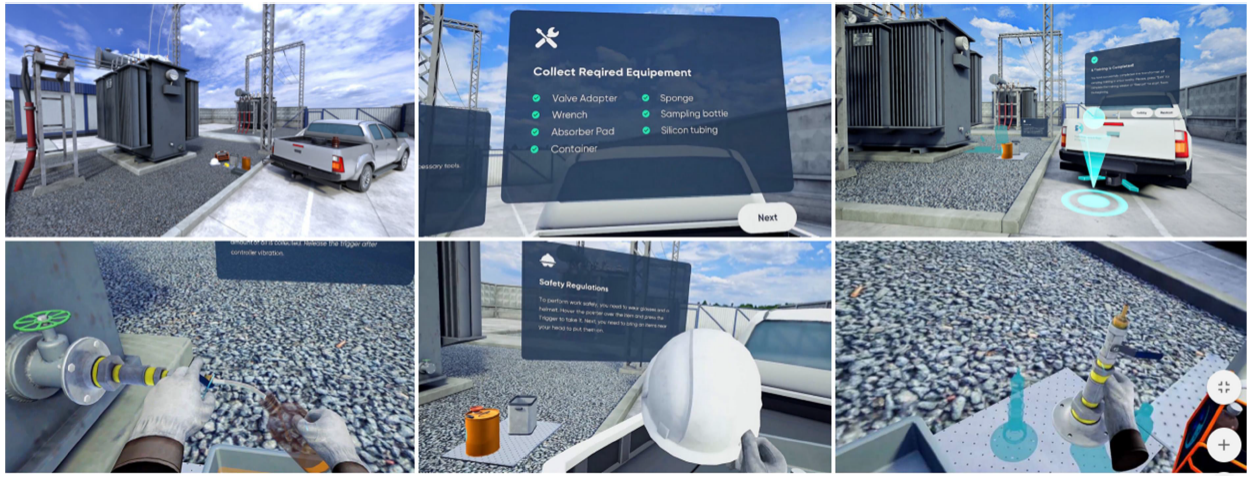

Figure 1.3: Transformer Oil Sampling UI

- Occupational Safety And Health Training for Electricians

This training is dedicated to maintaining safety at the workplace.

Figure 1.4: Occupational Safety And Health Training for Electricians UI

Comparison between apps

	ESTPE-VR	Motor Explorer
Implements VR	YES	YES
Difficulty	Hard to use	Easy to navigate
Required equipment	Requires multiple VR gadgets	Can be used without Headset
Control system	Very hard to manipulate	Easy to navigate
Targeted audience	Only workers	Both Students and workers

Table 1.3: Comparison Comparison between our app and an existing app

1.7 Development Method

The process of organizing a project and managing to deliver it in the right time is a stressful and hard to manage process, you could encounter technical problems that needs solving or conflict in the team

responsible for the development of the project.

Problems like these can be solved by using the agile methodology which we will apply during the work on this project .

1.7.1 Agile approach

Agile consists of breaking the project into several stages which makes the technical work more efficient and less sophisticated it also it involves the client or in this case Product Owner with certain reviews at the end of each stage in a way if a change is to bed made or an update is required it no longer requires to start from scratch but to work on the stage in which the problem has been encountered.

1.7.2 Scrum Methodology

The Scrum Methodology is the most common agile approach in which it involves around three roles :

- *The product owner*

The Product owner is responsible behind organizing the requirements that the project must meet, he also decides which functionalities get to be prioritized during each stage of the development.

- *The Team*

The team is basically the members responsible of turning the project from idea to an actual functioning product; they must respect the product backlog which is set by the product owner. The team's size must be as small as possible to avoid conflicts and contradictory work ideas due to the different roles.

- *The Scrum Master*

The Scrum Master is basically the SCRUM itself, he must master it in all different aspects and in a proper way. He ensures that the team is implementing SCRUM in its proper ways in final words his job consists of organizing the perfect flow between the team, their tasks, and the project manager.

Sprint: The following is a diagram that will help understand that each Stage we refer to as “Sprint”, throughout the workflow each sprint has its own updated backlog, review and goal.

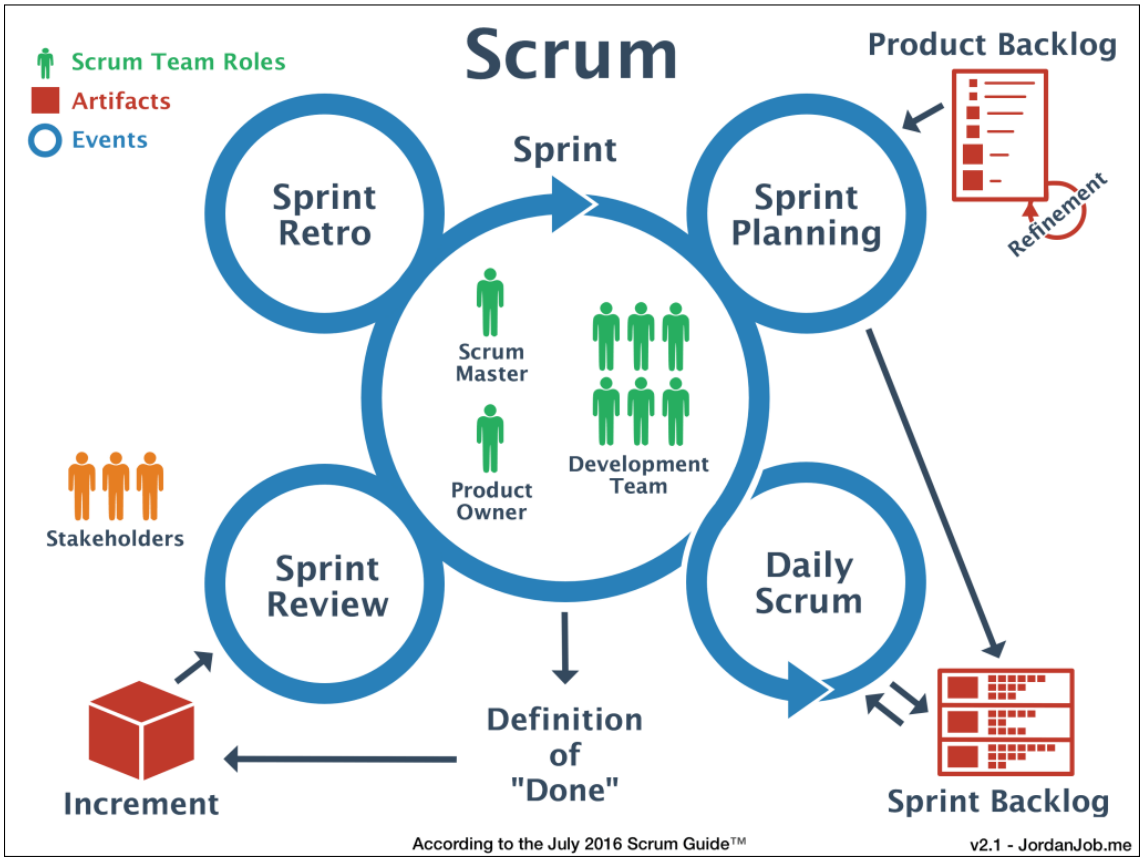

Figure 1.5: Scrum explanation

To summarize, SCRUM’s bread and butter is composed of four keywords:

- Sprint Planning:

  Setting the different goals that the team believes is able to achieve during the set period of a sprint.

- Daily SCRUM

The daily SCRUM is a basically a small quick meeting which contains three parts which are what was achieved the previous day, goals set for the current day and discuss work blockers which are preventing certain progress.

- Sprint Review

The sprint review is conducted at the end of each sprint, in which the they approve certain completion of set tasks by consulting the backlog and also provide a small demonstration of each completed task.

- Sprint Review

The sprint retrospective is a meeting which the SCRUM team conducts to discuss certain changes that emerge during the work; it also helps improving the product itself and discusses certain updates to be made.

1.8 Conclusion

This chapter contained information about the hosting company XTECH. It also provided a general definition of the project by discussing the main goal of the proposed solution, ending it by choosing the development methodology to follow throughout the realization of such project. In the next chapter we will provide our analysis conducted prior to the development.

Chapter 2: Requirement analysis and specification

2.1 Introduction

In this chapter we will discuss some of the analysis conducted to deduce the requirements and specification conerning the following project. We will lead by giving a brief definition to our SCRUM roles and actors, and then we will follow by requirements identification into product backlog and we will close by showing the sprint planning and some general modelisation .

2.2 SCRUM Roles

The scrum team is made of a product owner , the development team and a scrum master. You will notice in our case that both the scrum master and product owner roles are occupied by the same person

- Product owner / Scrum Master: Mr. Zied ben haj salah . He is responsible for making a decent presentation about the product characteristics and functionalities he also is responsible for the approval of the product development. His tasks as a SCRUM master are basically a supervision of the work progress, managing the team activities and organizing the meetings.
- The TEAM: Slim Bardaoui and Firas Bouadila they are in charge of delivering each sprint and the technical work as in the development.

Role	Description
Product Owner	Zied bel haj salah
Scrum Master	Zied bel haj salah
Team	Firas bouadila , Slim Bardaoui

Table 2.1: Scrum team

2.3 Actors identification

In general cases an actor is the targeted individual or system to make use of the application or the solution services. It conducts the general operations on the application and in our case the actor would be:

 User	Since our proposed solution doesn't require any complicated interaction we will only have a single user who will make use of the immersive experience in the VR environment his tasks are : • Pick the required motor • Initiate the test • Set the correct values needed • Take notes of the obtained results

Table 2.2: Actors identification

2.4 Requirements Identification

In this section we will take a look at how we can transform the basic goals of the project into a set of functions which can be developed. After having identified the main actor behind our application, we will talk about functional and non- Role Description Product Owner Zied bel haj salah Scrum Master Zied bel haj salah Team Firas bouadila , Slim Bardaoui functional requirements followed by a use case diagram explaining the global actions of our product.

2.4.1 Functional Requirements

- Must provide accurate calculations
- Enable user to have detailed information about each motor
- Must provide a guided interface

2.4.2 Non functional requirements

- The interface should be user friendly
- The graphics must be smooth
- Provide animations
- Provide inetractive noises

2.5 Product backlog

User story	Priority	Estimation(days)
I as a user would like to see a control panel	1	1
I as a user would like to set the load	2	3
I as a user would like to set the speed	3	3
I as a user would like to set the voltage	4	3
I as a user would like to turn the test on and off	5	2
I as a user would like to have a results menu	6	2
I as a user would like to have a motor menu	7	2
I as a user would like to see a dynamometer	8	2
I as a user would like to see a work-bench	9	2
I as a user would like to interact with the work-bench	10	8
I as a user would like to see the motors running	11	2
I as a user would like to hear sounds indicating motor status	11	2
I as a user would like to have accurate calculations for AC motors	12	5
I as a user would like to have accurate calculations for DC motors	12	5
I as a user would like to see a loader at the start of the app	13	2
I as a user would like to have panels containing information about the motors	14	4
I as a user would like to have a panel containing information about the dynamometer	15	4
I as a user would like to have an instructions panel	16	4

Table 2.3: product backlog

2.6 Sprint Planning

Using the previous backlog , and after conducting a workflow discussion we have decided on having 4 sprints each of 2 weeks duration and their tasks repartition is decided as this:

Figure 2.1: Sprint planning

Sprint name
Sprint 1 : First UI
Sprint 2 : Into the GUI
Sprint 3 : Animations and calculations
Sprint 4 : Optimization and final touch

Table 2.4: Sprints name

2.7 Task Management

We used Trello board to manage our tasks.

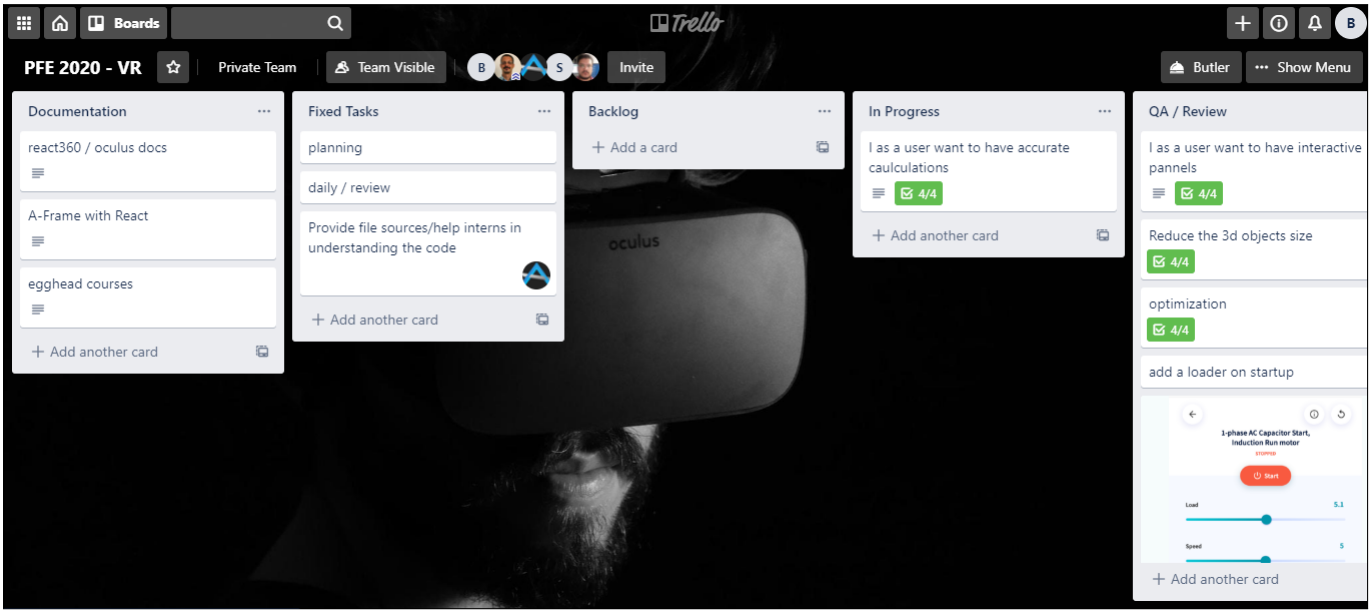

Figure 2.2: Trello

2.8 Functional Requirements Modeling

2.8.1 Use case

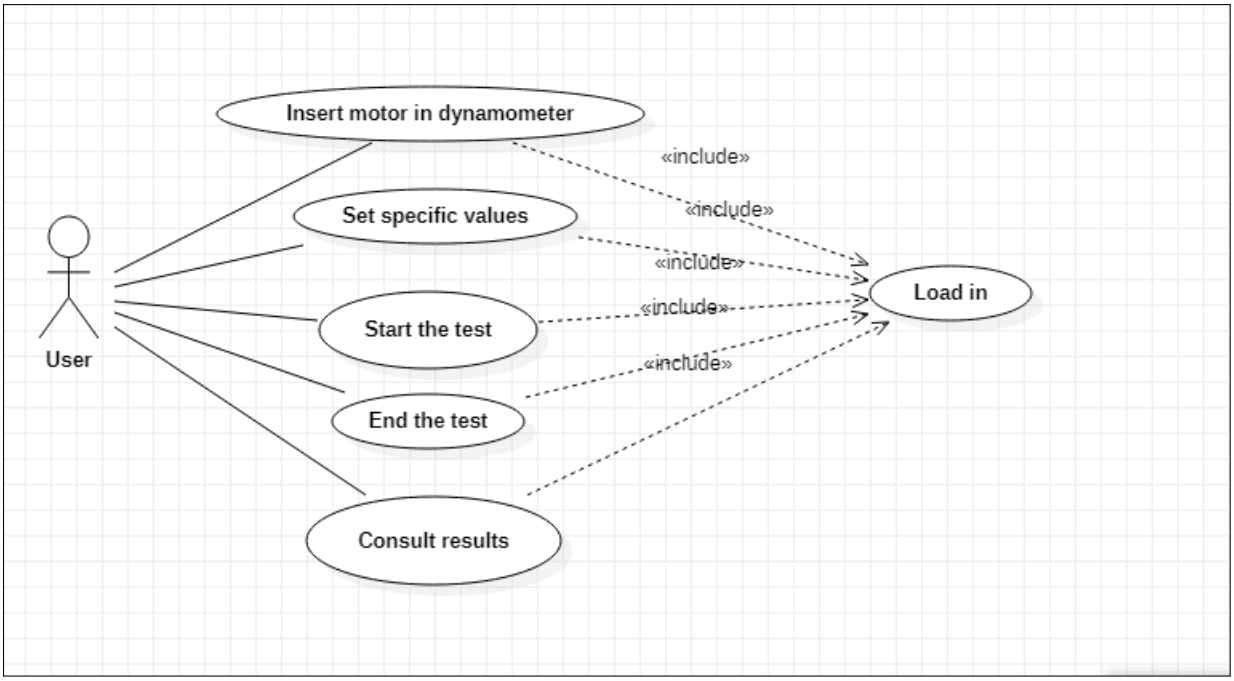

Figure 2.3: global use case

2.9 Conclusion

This chapter explained the work-flow and the approaches we used to manage the work using an accurate planning and estimate the time it's gonna take to deliver the project we also defined the functional and non-functional aspects of the application and in the next chapter we will start the actual work on the project .

Chapter 3: Project Initialization

3.1 Introduction

This chapter will desribe the architecture used for the development of the application and than we will discuss the materials used to relaise the project.

3.2 Solution Architecture

//PLACE HOLDER FOR THE SOLUTION ARCHITECTURE

3.3 Technologies used

In this section, we will present the different technologies used to develop our application. The following figure summarizes the development environment used.

3.3.1 Hardware environment

The applications were coded on two Asus laptop computers with the following features:

- Intel(R) core processor ™ i7-7500U CPU @ 2.70 GHZ 2.90 GHZ.

- A RAM of 8 GB.

- A 1 Terabyte hard disk drive

and tested with oculus quest.

3.3.2 Used paltforms

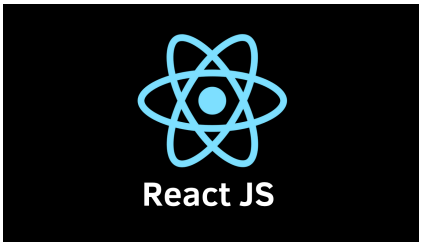

Figure 3.1: React Logo

React is an open-source JavaScript library for building user interfaces. React makes it painless to create interactive UIs using what is known as declarative views it makes the code easier to go through and debug . To summerise , React update and render each component you use while dynamically managing their own state making it fast in terms of performance and able to build complex UIs.

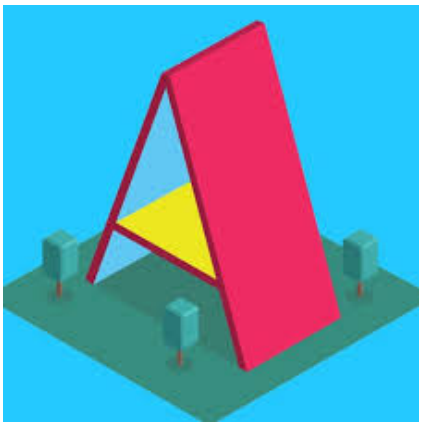

Figure 3.2: A-frame Logo

A-Frame is a web framework for building virtual reality experiences. Since A-Frame is built on top of the DOM, web libraries such as React, Vue.js, Angular, Ember.js, d3.js are able to sit cleanly on top of A-Frame.

A-Frame is an entity-component-system (ECS) framework exposed through HTML. ECS is a pattern used in game development that favors composability over inheritance, which is more naturally suited to 3D scenes where objects are built of complex appearance, behavior, and functionality. In A-Frame, HTML attributes map to components which are composable modules that are plugged into entity to attach appearance, behavior, and functionality.

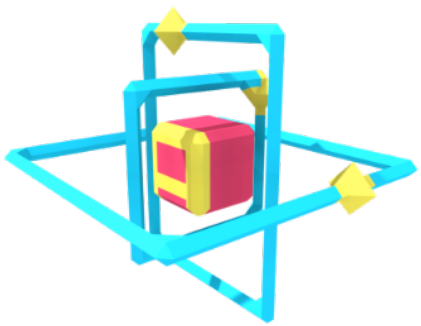

Figure 3.3: Aframe-react Logo

Released on the same day as A-Frame, **aframe-react** is a very thin layer on top of A-Frame to bridge with React. aframe-react passes React props to directly A-Frame using refs and .setAttribute(), bypassing the DOM. This works since A-Frame's .se-

tAttribute()s are able to take non-string data such as objects, arrays, or elements and synchronously modify underlying 3D scene graph.

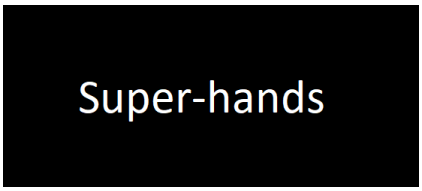

Figure 3.4: super-hands Logo

super-hands adds natural, intuitive interactions with tracked controller, touch, or mouse input in A-Frame. The goal of super-hands is to make it easy to handle user input in Web VR by providing a high-level API that is consistent across all devices. Instead of dealing directly with controller button events, raycasters, and collision detection components, you setup your scene and components instead to respond to 'gestures' like hovering and grabbing.

3.3.3 Tools used

3.3.3.1 Coding tools

Figure 3.5: visual studio code Logo

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop. Since in our case we manipulate multiple Languages and frameworks it helps conduct all the work in one place and avoid confusion and blockers and its extensions packs allow for multiple customisation options that allow the user to have a smooth work-flow.

3.3.3.2 3D designing tools

Figure 3.6: 3D builder Logo

3D builder is a Microsoft provided application that allows for viewing , creating and editing 3D objects with a variety of powerfull tools . It also allows to turn simple images into their 3D model with the option to manipulate the content as pleased and it also allows you to build from scratch your desired 3D objects.

3.3.3.3 Versioning tools

Figure 3.7: GitLab Logo

"GitLab is a complete open-source DevOps platform, delivered as a single application, fundamentally changing the way Development, Security, and Ops teams collaborate and build software. From idea to production, GitLab helps teams improve cycle time from weeks to minutes, reduce development process costs and decrease time to market while increasing developer productivity. GitLab helps teams design, develop and securely manage code and project data from a single distributed version control system to enable rapid iteration and delivery of business value. GitLab repositories provide a scalable, single source of truth for collaborating on projects and code which enables teams to be productive without disrupting their workflows.

3.3.3.4 Modeling tools

Figure 3.8: StarUml Logo

“StarUML is an open source software-modeling tool that supports UML (Unified Modeling Language). It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing generating code for multiple languages.”

3.3.3.5 Communication tools

Figure 3.9: Slack Logo

Slack is a proprietary business communication platform developed by American software company Slack Technologies. Slack offers many IRC-style features, including persistent chat rooms (channels) organized by topic, private groups, and direct messaging

.

3.4 Conclusion

This chapter contained the detailed information on different logical aspects of the development process explaining the tools used and the architecthre decided upon while also expanding reasons toward why we made certain choices as we move forward to the sprints section of this work .

Chapter 4: First Sprint

4.1 Introduction

After conducting the work needed in “Sprint 0” by planning the flow of our work and bringing focus to the different functionalities that we promised to deliver this chapter will contain all the different details about the first sprint.

It will contain the sprint backlog, the conception of this sprint and we will finish by multiple screenshots of the obtained results.

4.2 Sprint backlog

User story	Description	Priority	Estimation
I as a user would like to see a control panel	User gains access to a control panel that contains interactive buttons and sliders	1	1
I as a user would like to set the load	User gets to manipulate a slider that helps him set the value of the Load	2	3
I as a user would like to set the speed	User gets to manipulate a slider that helps him set the value of the Speed	3	3
I as a user would like to set the voltage	User gets to manipulate a slider that helps him set the value of the Voltage	4	3
I as a user would like to turn the test on and off	Add a visible on and off button for the user to initiate the testing	5	2
I as a user would like to have a result menu	On the control panel add a section of visible results for the test which will be updated in real time	6	2

Table 4.1: Sprint one backlog

4.3 Analysis and conception

The next section will contain detailed information on each functionality of the sprint.

4.3.1 Actors identification

The only actor is the user

4.3.2 Use case diagram

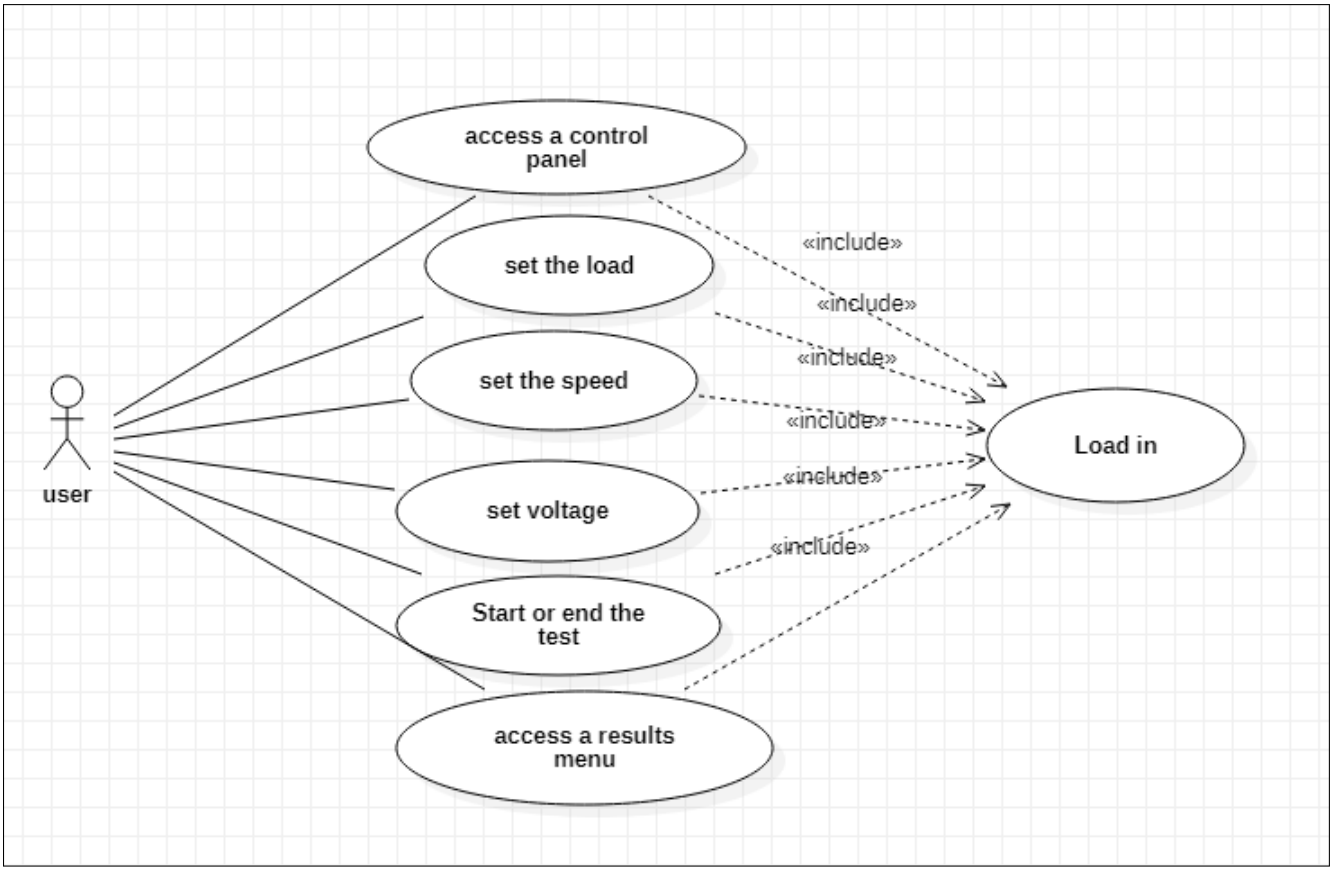

Figure 4.1: Use case diagram ”Sprint One”

4.3.2.1 Control panel

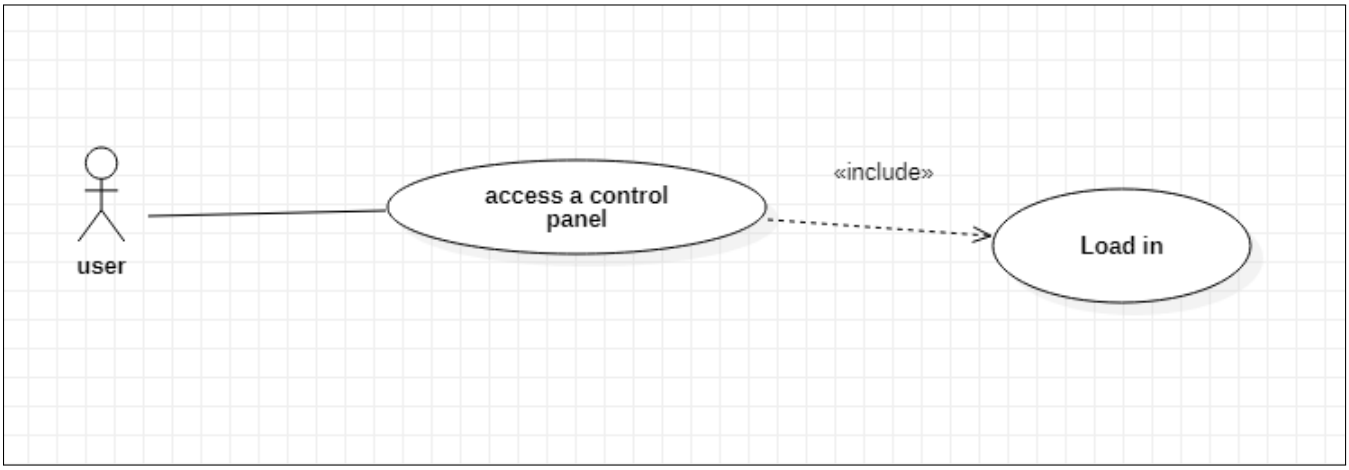

Figure 4.2: Use case diagram ”Control Panel”

Use case	Control Panel
Main actor	user
Pre-condition	Can’t be accessible until user places a motor
Post Condition	Shows the interactive sliders and buttons
Description	At first it will be a screen asking user to place a motor after user conducts the action the panel transforms into a multiple button / sliders composition revealing the controls to the user
Exception	The run button wont work until the user presses the on/off buttone

Table 4.2: Use case diagram ”Control Panel”

4.3.2.2 Load Setting

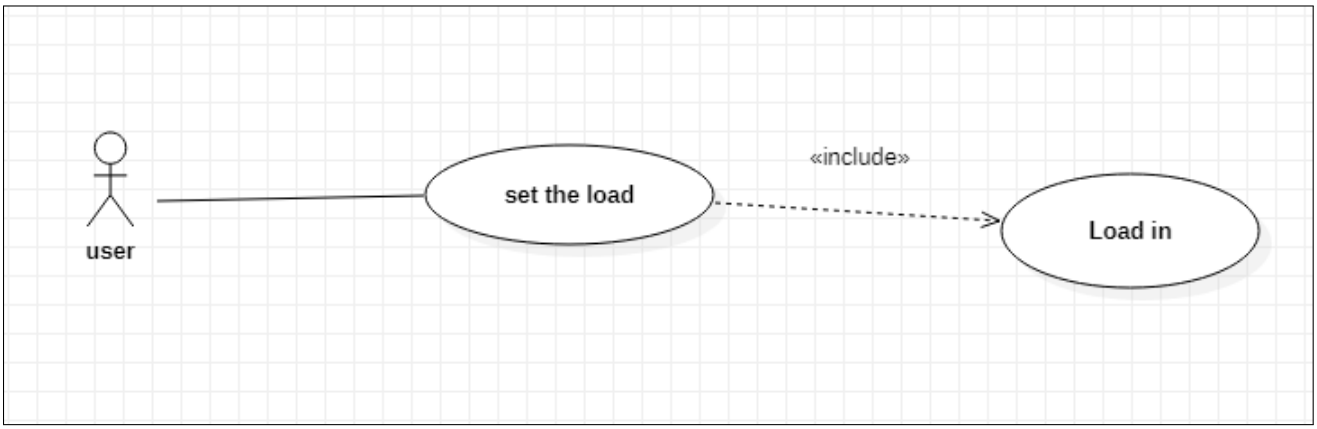

Figure 4.3: Use case diagram ”Load Setting”

Use case	Load setting
Main actor	user
Pre-condition	User must enable the control panel
Post Condition	Slider maximum and minimum value correspond to the motor inserted
Description	After placing a motor the load slider will allow user to set the value needed to conduct the test
Exception	

Table 4.3: Use case diagram ”Load Setting”

4.3.2.3 Voltage Setting

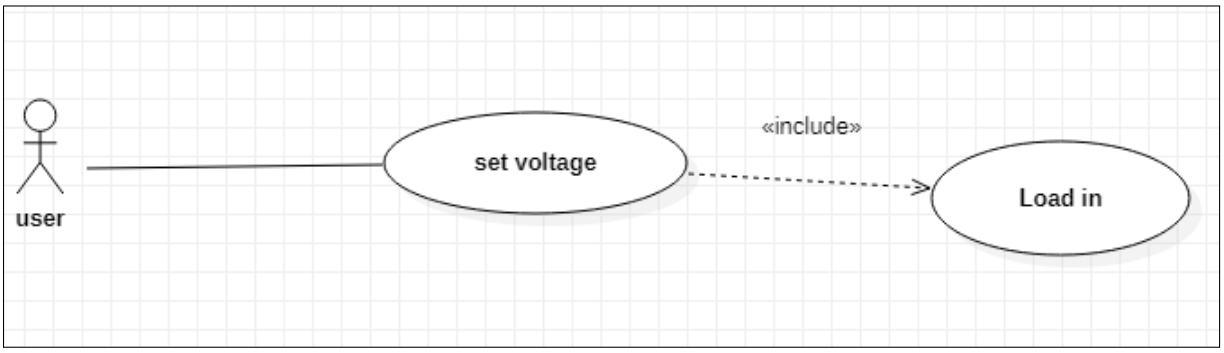

Figure 4.4: Use case diagram ”Voltage Setting”

Use case	Voltage setting
Main actor	user
Pre-condition	User must enable the control panel
Post Condition	Slider maximum and minimum value correspond to the motor inserted
Description	After placing a motor the load slider will allow user to set the value needed to conduct the test
Exception	

Table 4.4: Use case diagram ”Voltage Setting”

4.3.2.4 Load Setting

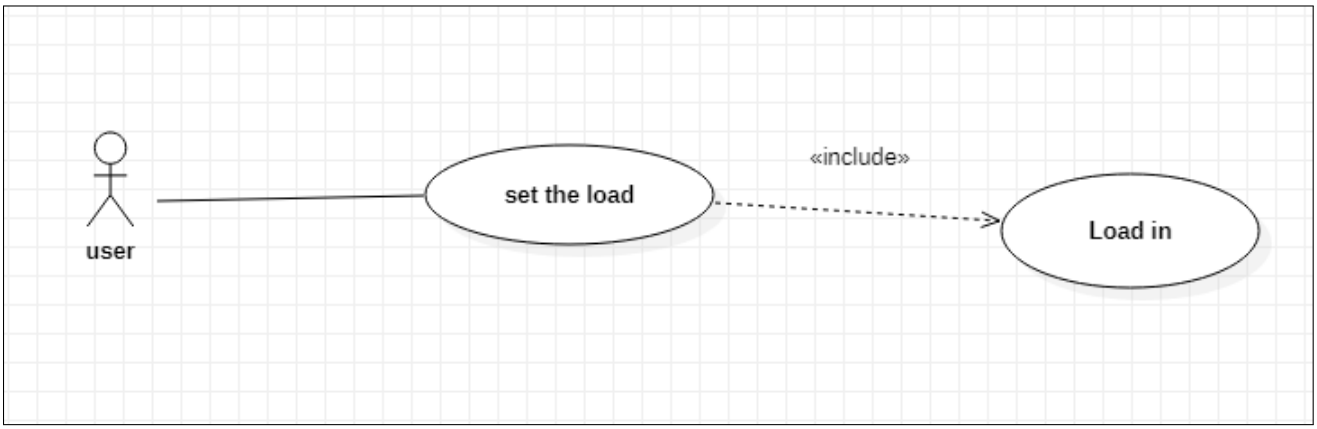

Figure 4.5: Use case diagram ”Load Setting”

Use case	Load setting
Main actor	user
Pre-condition	User must enable the control panel
Post Condition	Slider maximum and minimum value correspond to the motor inserted
Description	After placing a motor the voltage slider will allow user to set the value needed to conduct the test
Exception	

Table 4.5: Use case diagram ”Load Setting”

4.3.2.5 Speed Setting

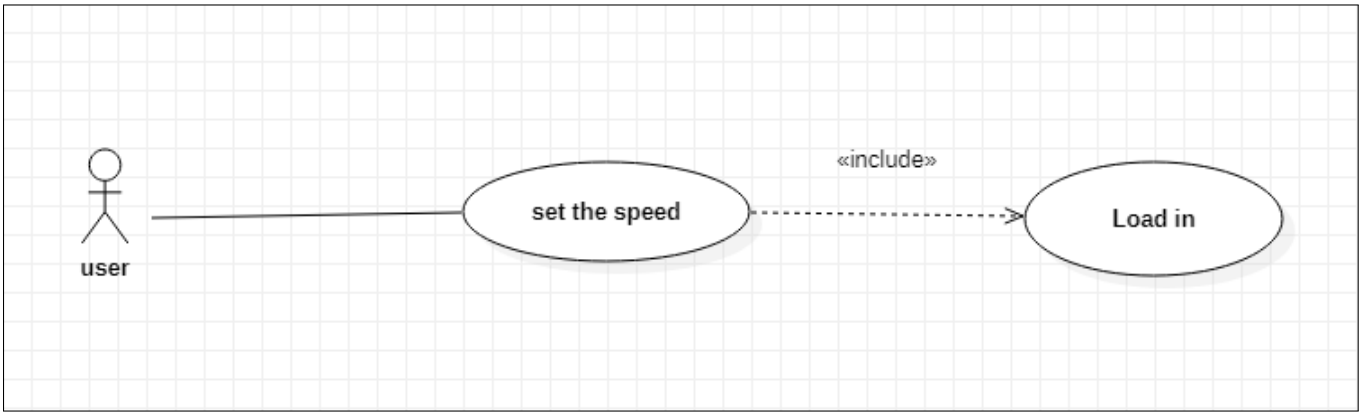

Figure 4.6: Use case diagram ”Speed Setting”

Use case	Speed setting
Main actor	user
Pre-condition	User must enable the control panel
Post Condition	Slider maximum and minimum value correspond to the motor inserted
Description	After placing a motor the speed slider will allow user to set the value needed to conduct the test
Exception	

Table 4.6: Use case diagram ”Speed Setting”

4.3.2.6 Start/End the test

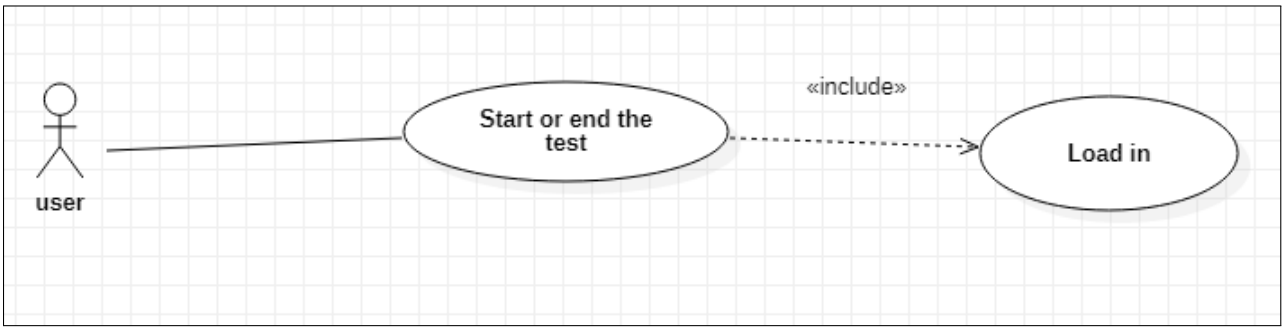

Figure 4.7: Use case diagram ”Start/End the test”

Use case	Start/End
Main actor	user
Pre-condition	User must enable the control panel and hit the on button
Post Condition	User can use the button again to end the current test
Description	This button allows for easy control to turn on and off the testing in hand
Exception	

Table 4.7: Use case diagram ”Start/End the test”

4.3.2.7 Results menu

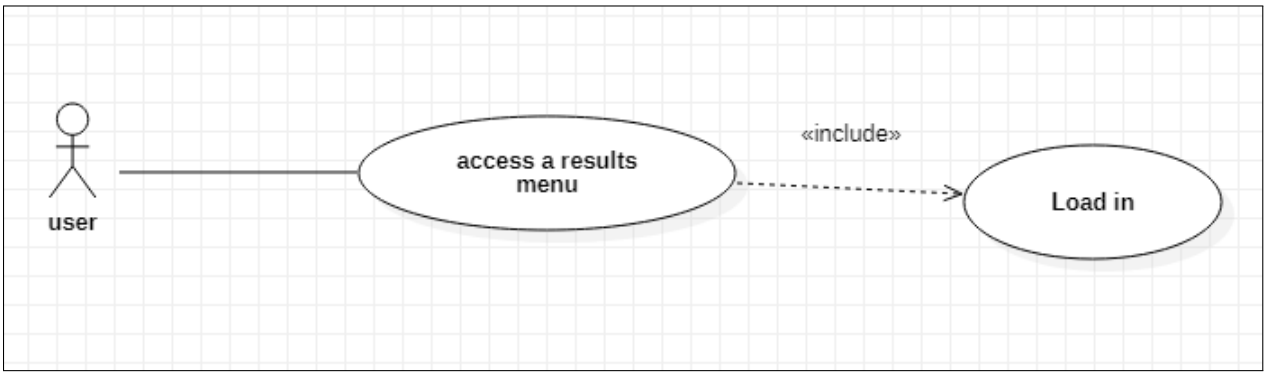

Figure 4.8: Use case diagram ”Results Menu”

Use case	Results menu
Main actor	user
Pre-condition	User must enable the control panel and conduct a test
Post Condition	Values on this menu will dynamically be updated during the test
Description	This menu is where the user can view the results , it is automatically updated during the test with the motor status and each value needed
Exception	

Table 4.8: Use case diagram ”Results menu”

4.4 Realization

4.5 Conclusion

This chapter contained the main content for the first sprint with detailed view on each functionality and concluding with a set of screenshots on the realized work as we push forward to the next sprint.

Chapter 5: Second Sprint

5.1 Introduction

After reviewing sprint 1 , we begin our second sprint. We lead by showing the sprint backlog into the analysis only to end by showing some of finished interfaces.

5.2 Sprint backlog

User story	Description	Priority	Estimation
I as a user would like to have a motor menu	User gets a visible motor shelf that contains all the motors	7	2
I as a user would like to see a dynamometer	User gets the dynamometer with the on/off switch button	8	2
I as a user would like to see a workbench	User obtains a visible workbench in which he places the motor	9	2
I as a user would like to interact with the work-bench	User has the drag/drop function to interact with the visual aspect of the application	10	8

Table 5.1: Sprint two backlog

5.3 Analysis and conception

the next section will contain the use case diagrams for each of the functionalities.

5.3.1 Actors identification

This sprint has one actor: User

5.3.2 Use case diagram

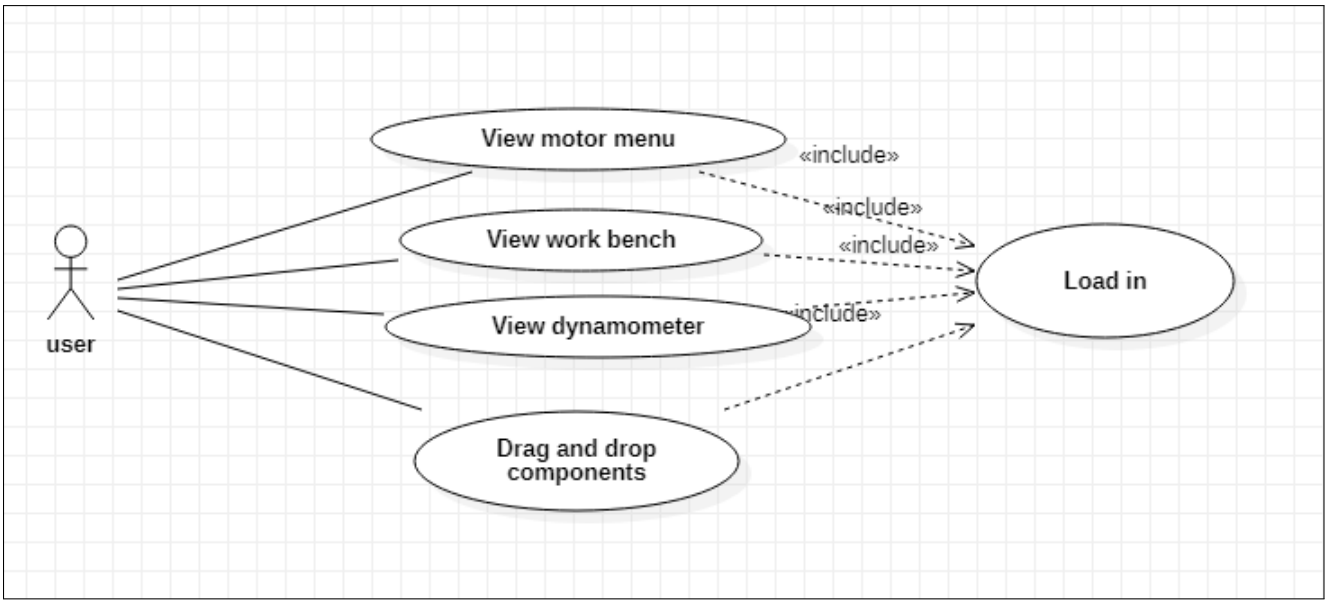

Figure 5.1: Use case diagram ”Sprint Two”

5.3.2.1 Motor menu

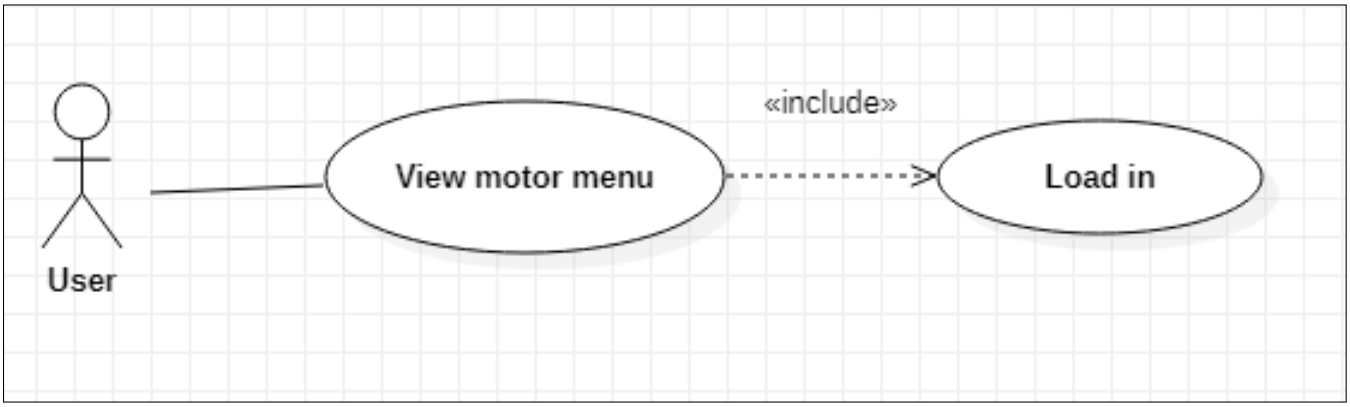

Figure 5.2: Use case diagram ”Motor Menu”

Use case	Motor menu
Main actor	user
Pre-condition	User is loaded-in
Post Condition	User can view the motor menu
Description	Shows a set of motors that the user can pick from to run the test
Exception	Only 1 motor can be picked at a time picking another one will result in placing the first motor to its original place

Table 5.2: Use case diagram ”Motor Menu”

5.3.2.2 Dynamometer

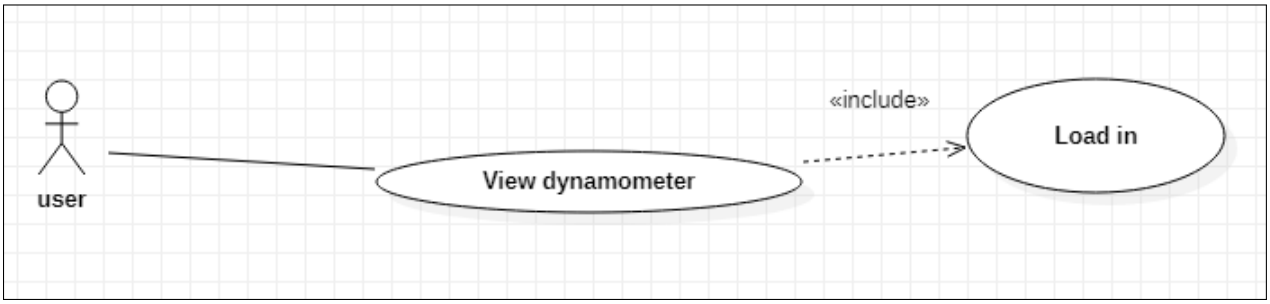

Figure 5.3: Use case diagram ”Dynamometer”

Use case	Dynamometer
Main actor	user
Pre-condition	User must be loaded-in
Post Condition	User can view the dynamometer
Description	The user sees a detailed dynamometer where the on/off switch can be pressed
Exception	

Table 5.3: Use case diagram ”Dynamometer”

5.3.2.3 Work-bench

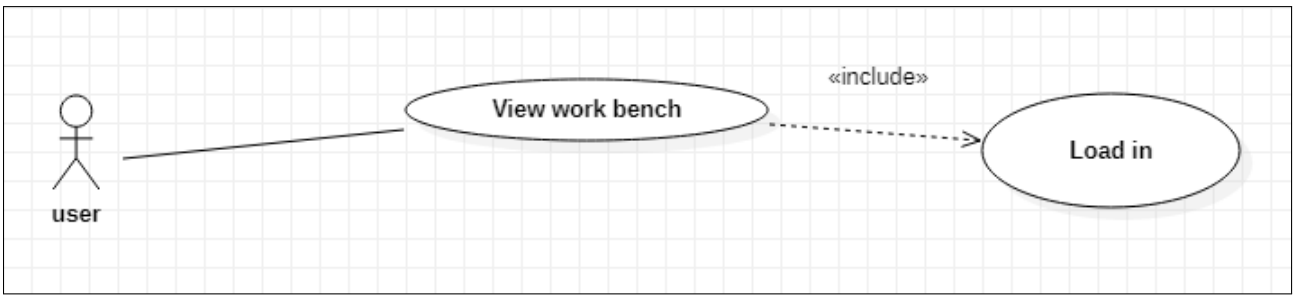

Figure 5.4: Use case diagram ”Work-bench”

Use case	Work-bench
Main actor	user
Pre-condition	User must be loaded-in
Post Condition	User can view the work-bench
Description	User will have access to the work-bench where he can place the motor to conduct the test pressed
Exception	Only 1 motor can be placed in the work-bench

Table 5.4: Use case diagram ”Work-bench”

5.3.2.4 Drag and Drop

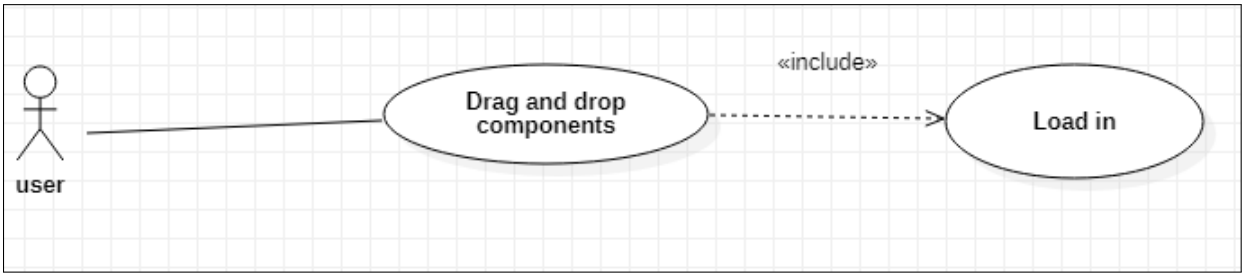

Figure 5.5: Use case diagram ”Drag and Drop”

Use case	Drag and Drop
Main actor	user
Pre-condition	User must be loaded-in
Post Condition	User can drag and drop motors onto the workbench
Description	User gets a set of visible hands where he can use to drag and drop components from the motor menu to the workbench and interact with the control panel
Exception	

Table 5.5: Use case diagram ”Drag and Drop”

5.4 Realization

5.5 Conclusion

This chapter explained the visual aspect of our application as we move toward the next sprint where the calculations and tests can be conducted with visible results.

Chapter 6: Third Sprint

6.1 Introduction

In the previous chapter we ended with making the visual aspect of our application , in this chapter we will discuss the required functionalities , by leading with the sprint backlog , into the analysis section and closing by the realization .

6.2 Sprint backlog

User story	Description	Priority	Estimation
I as a user would like to see the motors running	User will have rotating animation indicating that the test is being conducted	11	2
I as a user would like to hear sounds indicating motor status	User will have sounds indiciating that the test is being conducted	11	2
I as a user would like to have accurate calculations for AC motors	Values menu would be updated by the test results for AC motors calculations	12	5
I as a user would like to have accurate calculations for DC motors	Values menu would be updated by the test results for AC motors calculations	12	5

Table 6.1: Sprint three backlog

6.3 Analysis and conception

The next section contains the details and the use case diagrams on each of the user story motioned above

6.3.1 a. Actors identification

This sprint has one actor: User

6.3.2 Use case diagrams

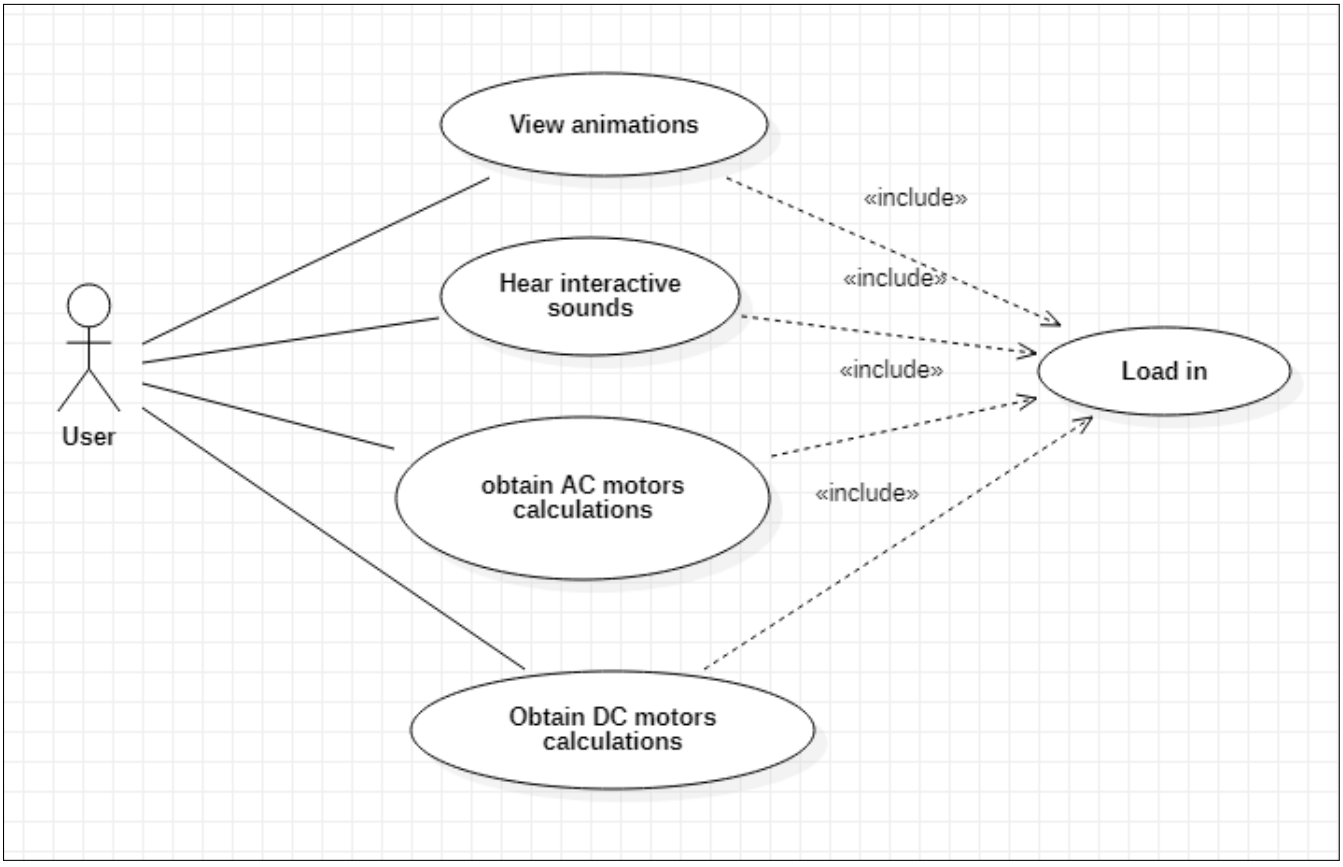

Figure 6.1: Use case diagram ”Sprint Three”

6.3.2.1 Animations

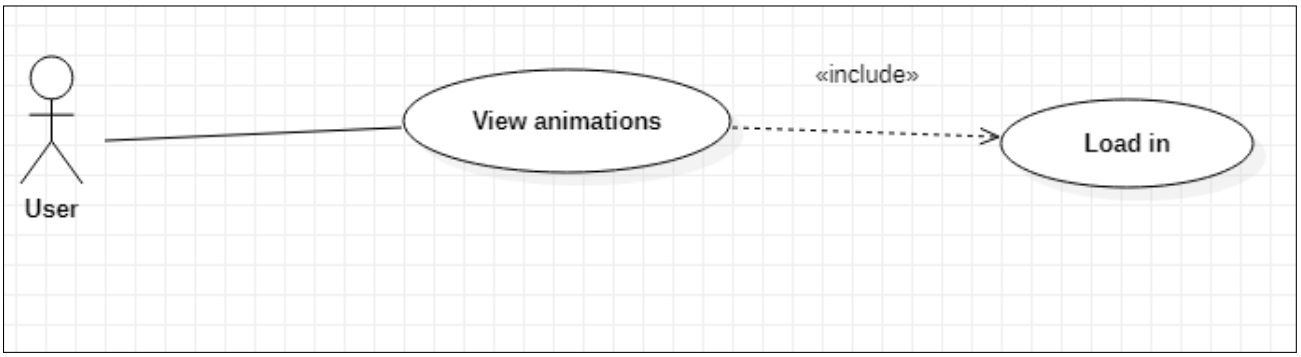

Figure 6.2: Use case diagram ”Animations”

Use case	Animations
Main actor	user
Pre-condition	User placed a motor in the workbench
Post Condition	Upon initiating the test , the motor will start rotating.
Description	This animation indicates the status of the test , if the motor is running or stopped.
Exception	Test must be started

Table 6.2: Use case diagram ”Work-bench”

6.3.2.2 Animations

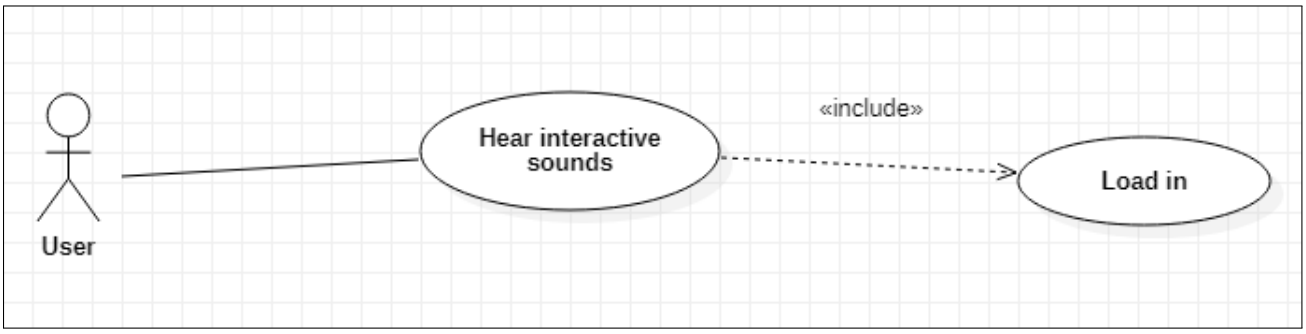

Figure 6.3: Use case diagram ”Sounds”

Use case	Sounds
Main actor	user
Pre-condition	User is loaded in the app
Post Condition	Every interaction with the application has its own sounds
Description	Each sound is set to indicate the different phases of the test or if the motors is being placed or not.
Exception	

Table 6.3: Use case diagram ”Sounds”

6.3.2.3 AC motors calculations

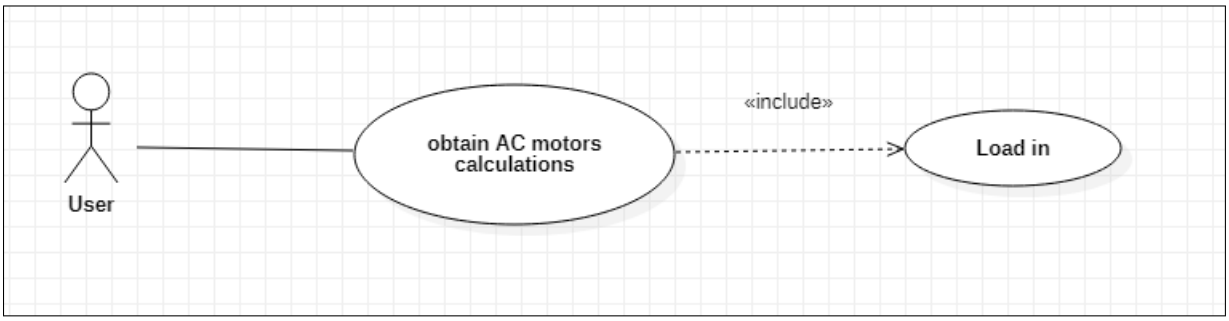

Figure 6.4: Use case diagram ”AC motors calculations”

Use case	AC calculations
Main actor	user
Pre-condition	User must place an AC type Motor
Post Condition	User must conduct a test sounds
Description	During and after the testing the value menu will be updated with the results needed.
Exception	

Table 6.4: Use case diagram ”AC motors calculations”

6.3.2.4 DC motors calculations

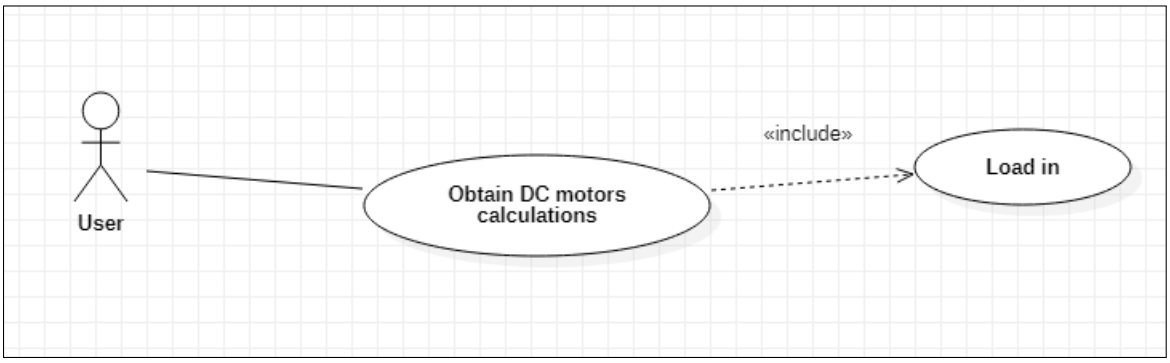

Figure 6.5: Use case diagram ”DC motors calculations”

Use case	DC calculations
Main actor	user
Pre-condition	User must place an DC type Motor
Post Condition	User must conduct a test sounds
Description	During and after the testing the value menu will be updated with the results needed.
Exception	

Table 6.5: Use case diagram ”DC motors calculations”

6.4 Realization

6.5 Conclusion

This chapter contained the basic functions that our application will be realizing which is the accurate calculations and after this comes the final sprint where it will contain all the optimization needed for the VR environment.

Chapter 7: Fourth Spring

7.1 Introduction

This chapter will conclude the development process of the application and will contain the final work that took place . As always we will lead by presenting the backlog , into the conception of the use diagrams and finish by a visual presentation of the product.

7.2 Sprint backlog

User story	Description	Priority	Estimation
I as a user would like to see a loader at the start of the app	User will have a loading screen when first loading the application	13	2
I as a user would like to have panels containing information about the motors	Upon hovering on a motor you will obtain detailed information about that motor	14	4
I as a user would like to have a panel containing information about the dynamometer	Upon hovering on the dynamometer you will obtain all the information on that device	15	4
I as a user would like to have an instructions panel	User has instructions to easily guide him throughout the application	16	4

7.3 Analysis and conception

The next part will have the detailed information on the sprint user stories.

7.3.1 Actors identification

the only actor in this sprint is the user.

7.3.2 Use case diagrams

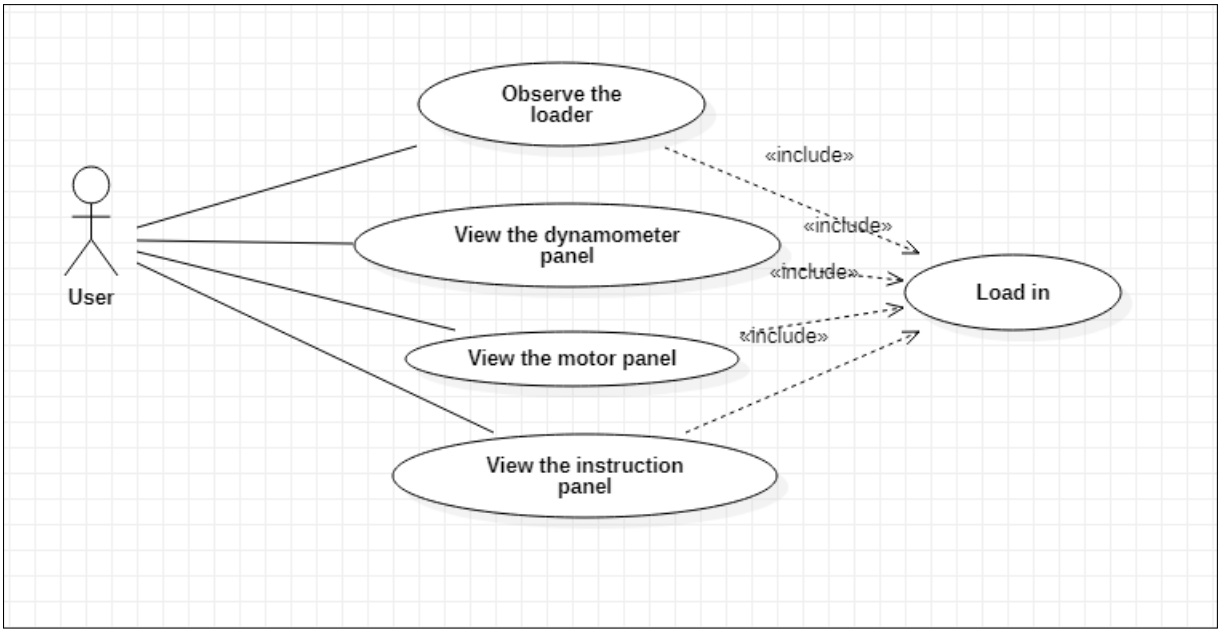

Figure 7.1: Use case diagram ”Sprint Four Use Case”

7.3.2.1 The loader

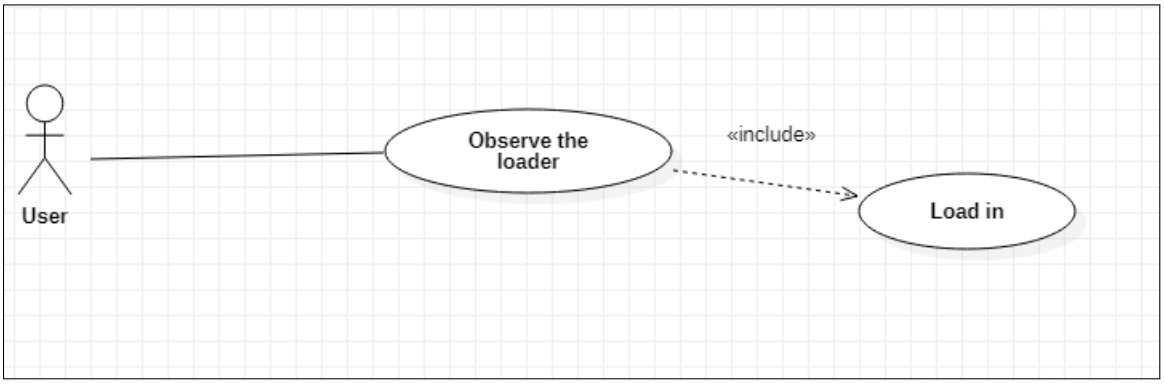

Figure 7.2: Use case diagram ”Sprint Four Use Case”

Use case	The loader
Main actor	user
Pre-condition	User launches the application
Post Condition	User sees loading progress on his screen
Description	Upon starting the application the user will get a loading screen showing him the progress of the loading
Exception	

Table 7.1: Use case diagram ”The Loader”

7.3.2.2 Motor panels

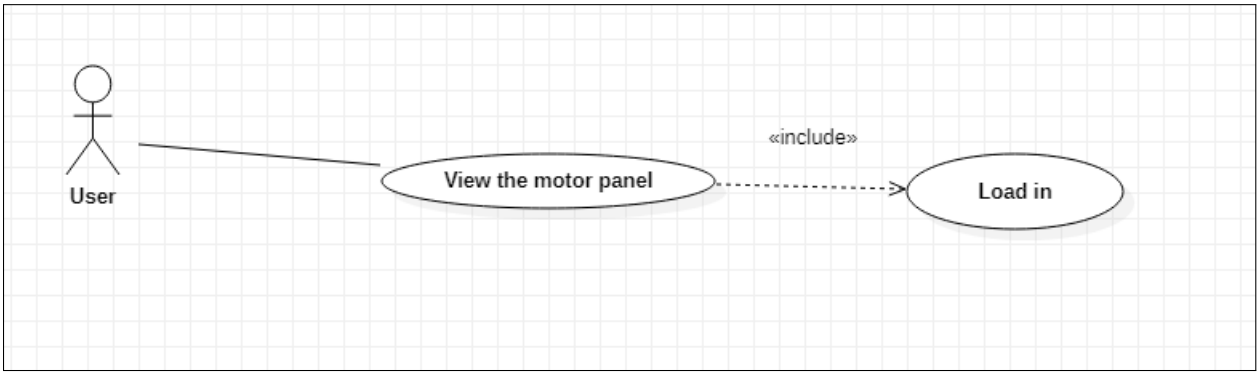

Figure 7.3: Use case diagram ”Motor Panels Use Case”

Use case	Motor panels
Main actor	user
Pre-condition	User launches the application
Post Condition	User can see the details on each motor
Description	Upon hovering on the motor user gets a visual readable text that shows him the details of that motor
Exception	

Table 7.2: Use case diagram ”Motor panels”

7.3.2.3 Dynamometer panel

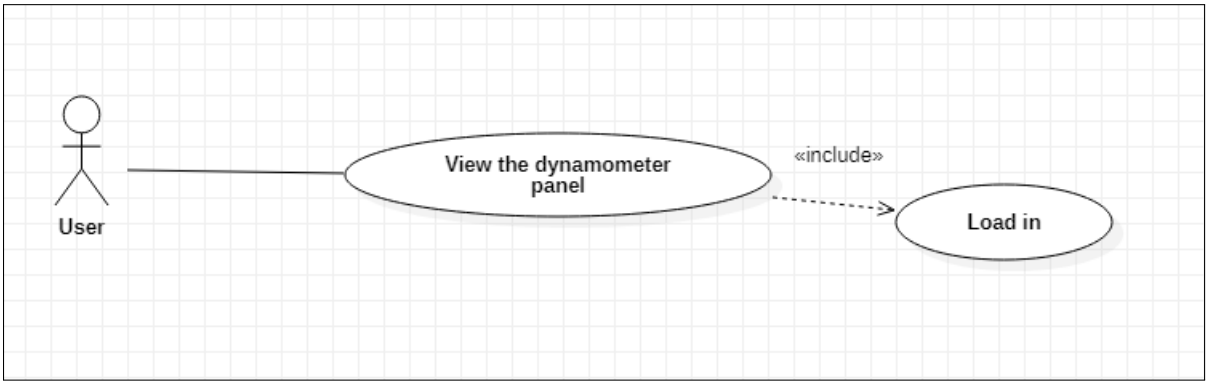

Figure 7.4: Use case diagram ”Motor Panels Use Case”

Use case	Dynamometer panel
Main actor	user
Pre-condition	User launches the application
Post Condition	User can read the information about dynamomemter
Description	Upon hovering on the dynamometer the user will obtain detailed description on what is the dynamometer .
Exception	

Table 7.3: Use case diagram ”Motor panels”

7.3.2.4 Dynamometer panel

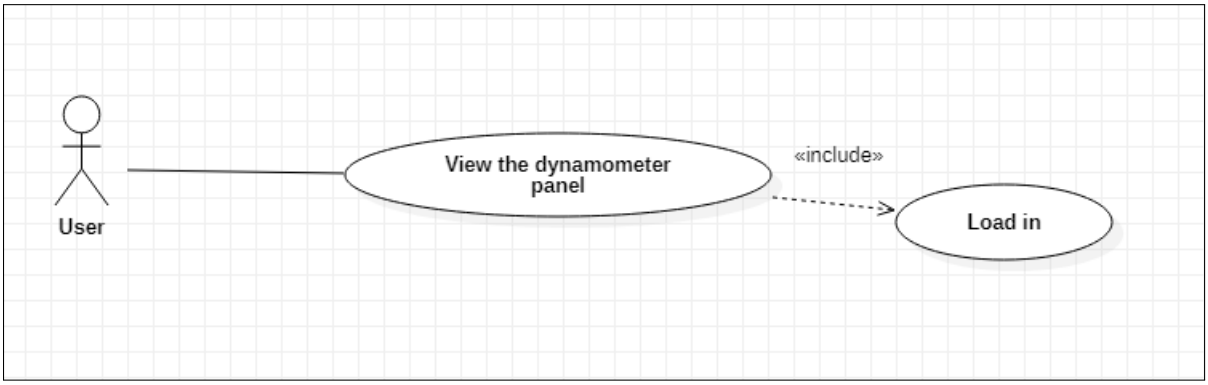

Figure 7.5: Use case diagram ”Motor Panels Use Case”

Use case	Dynamometer panel
Main actor	user
Pre-condition	User launches the application
Post Condition	User can read the information about dynamomemter
Description	Upon hovering on the dynamometer the user will obtain detailed description on what is the dynamometer .
Exception	

Table 7.4: Use case diagram ”Dynamometer panels”

7.4 Realization

7.5 Conclusion

Bibliography

[1] Clark Lambot, *A Brief History of VR*, https://veer.tv/blog/a-brief-history-of-vr/, 8 Nov , 2018.

[2] Clark Lambot, *A Brief History of VR*, https://veer.tv/blog/a-brief-history-of-vr/, 8 Nov , 2018.

[3] Leslie Lamport, *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.