Design Report

Smart electric vehicle charging user interface

Authors:

Aydan Allahverdiyeva, s1933906

Alex Olteanu, s1990675

Alexandra Iosif, s2189496

Ivo Broekhof, s2185970

Supervisor:

Marco Gerards

Course code: 202001048

November 3, 2021

University of Twente

Contents

Chapter 1

Introduction

Innovation and sustainability, which highlight the motto of the University of Twente: “High tech, human touch”, are two substantial key players in the continuous development of the campus. Members of the university’s community, both students and employers, bring a major contribution through their personal needs and choices.

Over the last years, more and more people have opted for electric cars, thus new settings of electric vehicle (EV) charging stations have become essential to encourage EV growth and make it possible at UT. So far, the cars have been charged using energy from the grid. This would be unsustainable in the future considering the increase of the consumption required. To address this, the adoption of solar panels as the main provider of charging energy is needed.

Meanwhile, charging ports deliver alternating current (AC) charging through a range varying between 3.3kW – 22kW. Therefore, the necessary power for supplying a parking lot with a considerable amount of cars charging simultaneously would require a very high-cost infrastructure. A system providing smart scheduling for the cars could offer a better power distribution through the charging period and solve any overloading issues.

In this regard, a new solar carport has been recently built near the Paviljoen building. This construction comes with nine EV chargers. The purpose of this setup is to make use of the Energy Management System (EMS) such that it synchronizes the EV charging to the solar panel assembly and a battery pack. The EMS manages the power supply of all chargers from the carport. Four of these EV charging points will be devoted to research started by the Computer Architecture for Embedded Systems (CAES) group of the University of Twente. This research is focused on smart scheduling so that the charging sessions will prioritize solar energy over grid energy.

In order to achieve this, a user interface is needed for the drivers to input their preferences regarding their charging session. In the following chapters, the system development will be analyzed thoroughly.

Chapter 2

Domain Analysis

In this chapter, the process of identifying the domain of the system is discussed. The view of the client and users on a system is issued from an ‘outside’ perspective, such that the existing problem is explained in more detail. By understanding the domain, the development will proceed smoothly and it aids the planning for future development.

2.1 Problem statement

In order to assure a good functionality of the EMS and guarantee end-user satisfaction, user inputs, specifically the estimated departure and arrival time, desired energy, and charging mode are key factors in achieving them. These inputs are called the charging preferences (e.g. ‘charge only sustainable energy at rate X’ or ‘charge a minimum amount of Y before time Z’). These preferences are used as input for the algorithms that are used in the EMS and always result in the desired charging profile which is represented by the graph that shows how much power/energy the vehicle receives over time in multiple discrete time intervals.

The ultimate goal was to provide the end-user intuitive upfront insight into the way the EMS charges the vehicle. It is expected that this form of transparent communication helps in convincing people to choose more sustainable options while sacrificing little to none of their charging flexibility available from the EMS. For instance, these methods could estimate the amount of CO_2 reduction compared to normal charging.

Our task was to design and develop novel methods to interact with the end-user (the EV driver) and analyze what options work better compared to others. For this, a web application designed for smartphones was required to be developed. Once an EV driver plugs in at the station, the driver should be able to effortlessly indicate their preferences via the webapp. This web application should show a small set of options (e.g. via sliders and easily interchangeable for research purposes) and show the customer in a comprehensive and visually appealing way how their vehicle is going to be charged (e.g. via a graph).

In this project, our team has implemented this user interface where electric vehicle (EV) drivers can indicate their preferences. An application programming interface (API) would facilitate the communication between the EMS and the outside world. Unfortunately this has not been provided at the time of system implementation, thus dummy data has been used instead. If the API for the EMS were available, the app would be able to send data to the local EMS, retrieve up-to-date information about the carport, calculate the desired charging profile based on the user input, and show this directly to the EV driver on their smartphone. In order for the UI to work as expected, we have also needed to interact and make use of the smart charging algorithms already developed. After the charging session, the user should be able to give feedback about his or her experience.

2.2 Client, Users, and interested Parties

Our clients were dr. ir. Gerwin Hoogsteen and ir. Bart Nijenhuis, who are both part of the CAES research group. At the same time, these clients have multiple partnerships, one of which is AmperaPark which tries to research intelligent power grids for parking lots, which is represented in this project by Richard Kokhuis.

2.3 Software Environment

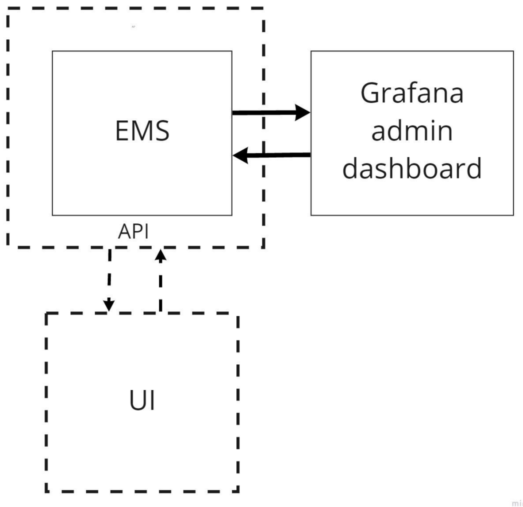


Figure 1 - EMS software environment

Before our project started, the EMS was a standalone server that could not communicate with other web applications through an application programming interface (API), but it could present the energy output through an admin page. Ideally it should be able to send and receive data to the UI we are building and our clients informed us that an API would be developed in the future. So we only needed to make mock requests from the UI side. Further sections discuss more about the software technologies and the architecture that we have used to develop this user interface.

The clients also use a Profile Steering algorithm (Gerards et al., 2015), which is a heuristic that seeks an optimal scheduling for a charging session such that the change in power output, or intake, stays as low as it can. Because it is a heuristic algorithm, it will not find the optimal solution but instead it will find a solution that gets very close to it. This algorithm has been implemented by our team and integrated into the system.

Apart from this, we have also implemented a naive planning algorithm which does not consider the strain on the grid it causes but simply seeks to finish the charging task as soon as it can.

2.4 Existing solutions

We first had researched and gotten inspiration from the way Tesla is doing scheduled charging (Klender, 2019) for their cars. Because a lot of Tesla car users also have a home charger they can also specify when they want to depart at such a time that their system knows when to charge to reduce energy cost, which corresponds with the grid peak time as well. Because of the similar process and elegant UI we managed to get the best of these UI and adapt it for our use case and design ideas.

2.5 Conclusion

In conclusion, we had gotten a clear picture of the requirements given by our clients and the third party company interested in this project, while at the same time we knew how the current software environment behaves and we researched the already existing solutions to get inspiration from them.

Chapter 3

System Proposal

This chapter describes key moments and phases of the negotiation process with the clients. The first section briefly mentions the goal and motivation behind the project. Later sections chronologically represent the project proposal at different phases: requirement proposal, a mock-up proposal, a system introduction, a usability testing phase, and a final presentation of the system to the client. The last section represents the meeting-by-meeting progress of the entire project.

3.1 Goal and Motivation

Since the final product implemented during this project will be used in our clients' research, it was important for our group to not only shape the basic requirements but also to understand how the profile steering works and what purposes it serves. Thus, we read background information on profile steering algorithms, as well as browsed the existing solution of EV charging apps. The meetings with our clients were scheduled weekly. During each meeting, we discussed the progress, got the work approved or denied, and implemented the given feedback before the next meeting.

3.2 Requirements Proposal

Since the software is at its earlier stage of development there was no base to start from. There was no established API or any specifications for the software. The clients had some strong ideas about what the web application should do, however, the project specification lacked the details. This gave us a lot of freedom in what we could implement and how to design the system, yet was quite challenging, since there were a lot of design and system choices to be discussed with clients, thus shaping Minimum Viable Product (MVP) took much longer than anticipated.

The main challenge of MVP was to balance between the software satisfying the goal of the research as well as the user-friendliness of the web application. Clients requested for the application to be suitable for research purposes, as well as, to serve a strong base for its growth into the startup. For each feature mentioned by the clients, our group proposed several possible solutions. While brainstorming ideas, the clients were mostly focused on the goals of the research. Therefore our group has decided to act as "defendants" of the users and see the application from their side. This helped us to successfully find the optimal solution for each requirement.

3.3 Mock-ups Proposal

In order to provide a better understanding of the MVP to the clients, the Lo-Fi (low-fidelity) and Hi-Fi (high-fidelity) mockups were provided.

3.3.1 Lo-fi

Lo-Fi represented the functional side of the MVP, as well as the key points of future design. In this phase of the design, we discussed the requirements of the products with clients and provided several solutions, as well as thought about possible side-effects of each approach. For example, one of the questions that arose during the implementation of the Lo-Fi was that users should be given an opportunity to pause and resume the charging session. This was the type of question that the clients did not have the answer to right away. The possible side-effect of allowing users to resume charging sessions would be drastic changes in price, CO2 emission, and charging schedule in general. So, together with the clients, it was decided to support only two states of the charging session: start and abort. In case, user accidentally aborts the session, they will always be warned with a pop-up that the charging session will be aborted. The final version of the Lo-Fi could be found in Appendix A.

3.3.2 Hi-fi

After the Lo-Fi design was approved, our team began to prototype the design of the future application. Same as with Lo-Fi, our team provided the clients with several solutions for UI to choose from and explained the consequences of each implementation.

The Figma framework allowed us to provide our clients with a prototype of the application at different steps of the design, where clients could click on buttons, see the changes and get the overall impression about the system's look & feel. This gave room for our clients to make some remarks and add additional components to the pages before the start of the implementation phase. The final version of the Lo-Fi could be found in Appendix B.

3.4 System Introduction

The system was introduced to clients halfway through the implementation phase. Our team has managed to successfully implement the graph component, which was highly important for our clients, so they have presented the fully functional graph with live updates. The clients were also introduced to the structure of the project's repository that ensures the modularity of the web application.

3.5 Usability Tests

Three stages of User Acceptance Testing (UAT):

1. Poll during peer review session - during the design phase of the project
2. Meeting with clients - during the design phase of the project
3. Prospective users UAT - during the implementation phase of the project

The testing during the design phase focused on styling of the UI: the poll tested light mode versus dark mode of the application, dropdowns versus slider etc. Some of the students during the lecture also shared more insight on the application, provided more concrete examples of what they would change, and shared their overall impression of the system. We then discussed the results with our clients and compromised poll results with clients' views of the application.

The "Meeting with clients" stage was initially planned in the format of a "live" demo (not exactly a live demo as users would play with a prototype not actually implemented version of the app). We invited our clients to ask more of their colleagues to join the session. So, during the meeting, we had a chance to present our prototype not only to our clients but also to a representative of the company that the web application is going to be supported by as well as one of the scientists developing the algorithm. Unfortunately, there was a bit of miscommunication during the meeting, which resulted in guests thinking of the demo as not something to interact with, but instead they shared their opinion about the application. However, our group decided to take that to our advantage and ask guests more questions on the functionality of the application. This resulted in highly valuable feedback, which we directly integrated into the system.

Prospective users UAT was the final stage of the application testing. We asked our clients to provide us with three participants, with whom we can conduct a live demo. Three members of our group were also responsible to demo the application with one participant each. Having two test groups provided more variety to our test users, thus making their feedback less biased. At this time, we provided both groups with clear instructions on what to do. We asked users to play around with the application at first, and after that, we asked them some questions and gave them a few tasks to test the user-friendliness of the application. Thinking aloud of the users was highly welcomed.

3.6 Proposal Presentation

At the end of the project, the system was presented to the members of the CAES research group and our clients. During the presentation the implementation process, full system overview and its added value to the current charging system, as well as, further support and enhancements were discussed. Demo of the web application was also held at the end of the presentation.

3.7 Results of the meetings

Since the design phase of the project followed the Waterfall project management model, the feedback received during every meeting with the clients was directly implemented for review at the next meeting. This ensured the efficiency of our work so that no time would be wasted on features that would shift the focus of the web application. This style of management was also useful during usability tests since after each stage of the UAT the changes were directly implemented to see how users would react in the next stages.

Chapter 4

Requirements and Project Management

In this chapter, we will discuss the way that we planned our project and discovered the requirements for our product.

4.1 Waterfall Project Management

For this project, we decided to follow the waterfall model where we develop the product from requirements to prototypes to actual implementation of the product, after which we use the corresponding testing methods (unit testing, integration testing, system testing, user acceptance testing etc.).

We decided to use the waterfall model instead of an agile development structure because we are developing the product from scratch and we wanted to put most of our time into design and prototyping instead of reimplementing part of the product multiple times.

During the implementation phase of the project, however, it did start to look like agile development with us taking issues from our GitHub board in a similar fashion to taking user stories from a backlog; nonetheless, it still functioned as a single sprint.

To prevent different components in development from crashing each other and to make sure our main branch stays stable, we used feature branching to isolate our concurrent developments and merge these branches into the main branch once the components were done so the rest of the branches could take the new component in if they needed it.

This approach has gone with few and minor problems, although a bit more overlap between the phases could have made the process a bit more efficient as the changes every iteration became less and less. This way there would be a bit more redoing in the later phases as the earlier ones change, but starting earlier on the later phases could be done quite stably with the parts that were deemed good so the finer details of that could be worked out together with larger-scale requirements in other parts of the product.

4.2 Stakeholder Requirements

To find out the requirements from our clients, we met frequently with them and presented our progress to get feedback from them and help discover their requirements for the product. As we discussed with them, we started making UML diagrams, sketching and prototyping and showing this to our clients for feedback until we reached a set of requirements for a Minimal Viable Product (MVP) as well as extra requirements for if we had extra time or the future of the product.

Chapter 5

Global and Architectural Design

This chapter discusses the global and architectural design choices. Each design choice is thoroughly explained and justified. The chapter also provides an overview of the entire system and its functionalities.

5.1 Global Design Choices

The web application is designed to help reduce the use of the grid energy on charging stations on campus. The algorithm that runs on the web application allows to schedule the charging sessions using as much solar power as possible. This prevents the grid system from overheating. Since the web application will be used for research purposes at the beginning, it was important to maintain flexibility and modularity of its design. This led to discussions with clients regarding the key pillars of the design.

5.1.1 Key Design Pillars

The key pillars of the project were the satisfaction of research purposes and support usability and user-friendliness of the application. Fortunately, we were able to establish weekly meetings with our clients, which allowed us to quite closely match their expectations. The testing was agreed to be conducted closely with users, which is why UAT consisted of three steps. We chose our tools and frameworks to match our needs, when it came to building prototypes and actual implementation. The other important factor into development of the system was its modularity. We made sure that it will be relatively easy for our clients to add/remove/overwrite components, so they can quickly develop the application further.

5.1.2 Established Work Process

It was important to establish the work process of the application before starting any design or prototypes. We had a series of discussions about activity diagrams with the clients, to understand better their expectations for the system. Final activity diagrams for user arrival and user departure processes are described in Figures 2 and 3.

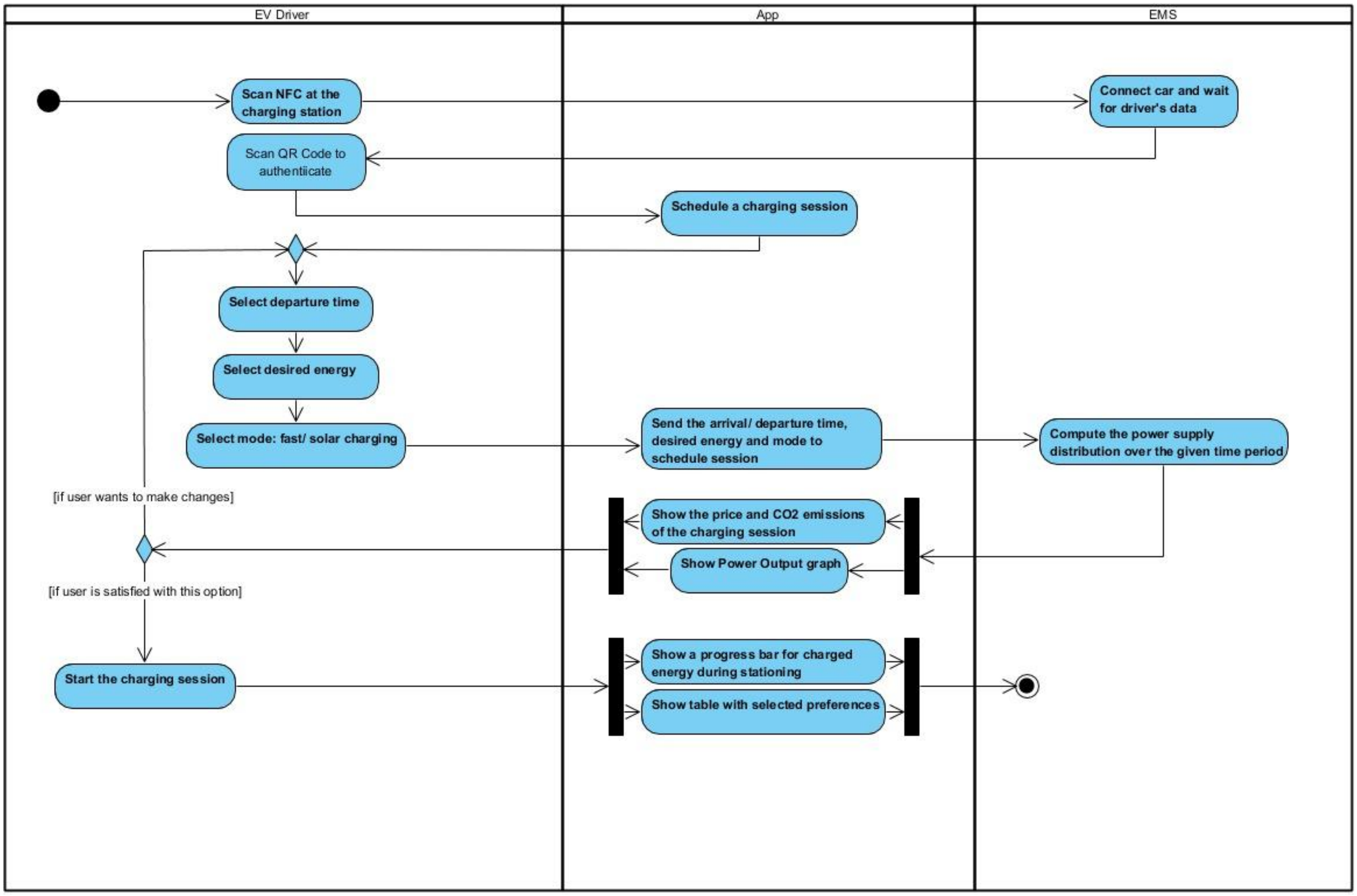


Figure 2. Activity diagram for user arrival.

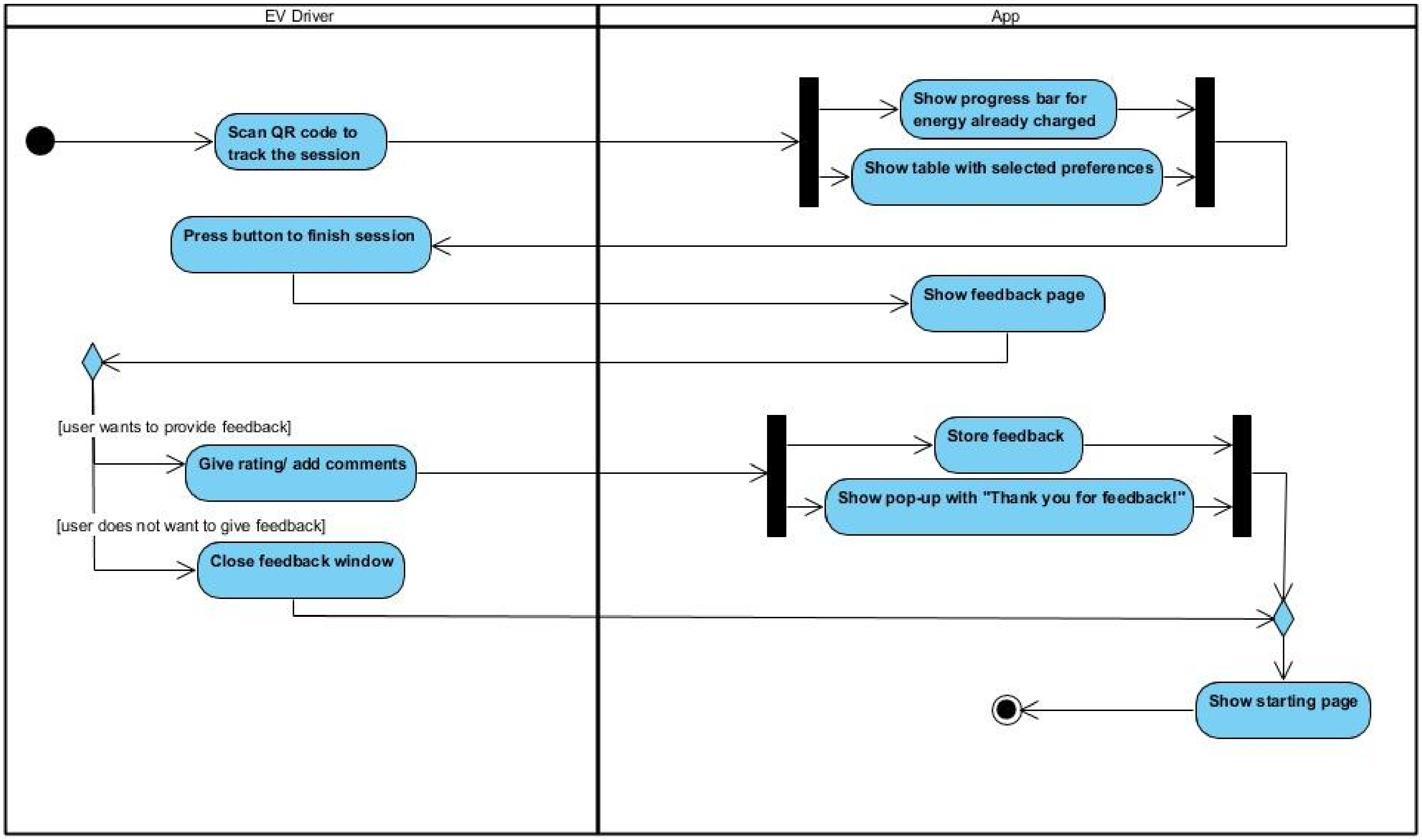


Figure 3. Activity diagram for user departure.

5.2 Preliminary Design Choices

During our first consultations with clients, we got an idea of what kind of web application they would like to have. Thus, we started to brainstorm possible preliminary design choices that would satisfy the web application's requirements. It was important to establish programming language, frameworks and libraries before the start of the actual implementation of the product.

5.2.1 Programming Language

The system is written using three languages: Javascript, Typescript, and CSS (or SCSS extension for CSS to be precise). Javascript is the language that can be used for both backend and frontend. Typescript extends the Javascript by adding types to constants, function parameters, and the function returns. Even though the UI can be written using Javascript only, adding Typescript ensures avoiding errors that could lead to the failure of the application during runtime.

CSS is used for styling the UI. SCSS is a more expensive version of CSS, that results in fewer code lines, thus reducing load time.

5.2.2 Frameworks and Libraries

Please refer to Table 1 for a brief overview of frameworks/libraries used in the system, as well as their contribution during the entire process of the project.

Framework/Library	Usage
Miro	Building Lo-Fi
Figma	Building Hi-Fi
React (library for Javascript)	Functional components to build UI
React Hook Form	Hook for functional components to handle the state of the form
Reactstrap	React Bootstrap 4 library
Material UI (MUI)	React Component Library
TypeDoc	Documentation generator (aka JavaDoc)
Typescript	Static typing for Javascript
Firebase	Deploy UI
Cypress	Unit & System Testing

Table 1. Frameworks/Libraries used at different stages of the project: design, implementation, testing

Miro is one of the most popular online tools for building Lo-Fi and Hi-Fi designs. While the Lo-Fi was built successfully, the possibilities to build the Hi-fi prototype were quite limiting. That is why the Hi-Fi prototype was implemented in Figma, which is a framework that specializes in building high fidelity designs only.

After multiple meetings with clients, it was established that the current version of the web application should be designed for smartphones only. However, our clients also shared the desire to implement the Admin interface and functionality in the future. That is why React library was an optimal solution for implementing the UI. React allows users to build responsive web pages suitable for all types of smartphone screen sizes, however, it also leaves the door open to implement Admin roles in the future for the desktop version of the app. In addition, React is a modern library that is used widely and is quite flexible, when it comes to integrating additional frameworks.

Firebase was chosen to deploy and host the UI. Mainly, the decision was based on the fact that Firebase is relatively easy to set up and can be exploited right away.

Cypress is one of the most popular frameworks to test React projects. In the beginning, our team was in between the Selenium testing framework and Cypress, however, we chose Cypress, since its setup is much easier compared to Selenium, as well as it executed in real-time, providing immediate visual feedback (Cypress, 2021; Selenium, 2021).

5.2.3 Architectural Design Choices

React library supports both functional and class components. Currently, there is a minimal difference between the two, however, functional components are much more flexible and easier to manage when it comes to supporting the state of the application. That is why our architecture consists of the functional components only.

The format for the URL links was also established. We agreed not to use underscores in the links for the users, since those might be very easy to miss. Thus, the structure is defined as follows: chargeview.com/${user_id}/${page_name}. Please refer to Table 2 for an overview of all page links:

Page	URL
Start page	chargeview.com/${user_id}/
Schedule page	chargeview.com/${user_id}/schedule
Charging session page	chargeview.com/${user_id}/session
Feedback page	chargeview.com/${user_id}/feedback

Table 2. Pages and URL overview.

Even though we understand that the current format allows users much easier access to the pages of other users, we still find this solution quite optimal. As has already been mentioned before, the current version of the application will be used for research purposes only and it is important to make the link as easier for users to type in as possible. In order to offer security to users, their user IDs would be established randomly in the system. This will lower the chances of users guessing each other's IDs. In addition, those URLs can later be used to connect the application to the API.

5.4 System Overview

Our system consists of several components identified and described in the remaining section. In order to enhance the user experience, an analogy with the sense of getting involved in a race has been made for the look of several elements, such that the user will go through the following sequence when shifting to the next component of the system: “Ready?” – “Set” – “Go” – “Stop” – “Finish”. Please refer to Appendix G for the final design of pages.

5.4.1 Start Page

The user is already identified and logged into the system when they scan the QR code. Thus, a welcoming message containing the user’s name is displayed at the top of the starting page. It is followed by an image of a generic car in order to make the user interface more appealing. Then the text field “Ready?” should serve as a stimulus for the user to press the “Set” button and go to the schedule page.

5.4.2 Schedule Page

This page is meant to set up the charging session, while it also outlines users’ options and their outcomes. The component covers four pillars in scheduling. Firstly, the user is required to select their estimated departure time. Secondly, the amount of energy desired has to be chosen using a slider. To ease this process for the user, automatic conversion between km and kWh is displayed right below. The next step is to pick the charging mode. The system comes with two possibilities: fast charging and solar power. The user needs to scroll down in order to see the power output graph. This is meant to provide direct feedback based on the chosen mode, specifically showing both the solar power over time and the actual power rate scheduled over the charging period. Meanwhile, the user can always see the charging price and the future CO2 emissions considering their preferences. When the user finishes the charging plan, they can “Go” to the next page.

5.4.3 Session Page

This component serves as a tracker of the charging session. The user is led to this page after the schedule is set. It can handle three states: when the session is in progress, finished, or aborted. In all cases, a progress bar is shown with the percentage of the desired consumption already charged; plus a table including the predicted finish time, the charged amount of energy in both kWh and km, and the mode previously selected. During the session, the departure time is displayed at the top of the page. The user can stop charging any time, a confirmation pop-up will appear, then a message with the abortion time will replace the top of the screen. Otherwise, when the session has ended, the user will know the exact time and can safely press “Finish”.

5.4.4 Feedback Page

This component aims at improving the user experience in future development by getting direct feedback from the user at the end of each charging session. They have a 0-to-5 rating option and can mark one of the four pre-set comments, namely: “I love it!”, “Easy to use”, “Annoying”, “Too much graphics”. Additionally, if none of the aforementioned statements applies, any thoughts can be placed in the text box provided. After submitting, a pop-up thanking the user will get visible. The page is not designed to put pressure on its users, thus they can always close it or submit an empty form and be redirected to the start page.

Chapter 6

Detailed Design

This chapter provides a detailed description of the system. The system's functionalities are discussed and design decisions of those are justified.

6.1 System description

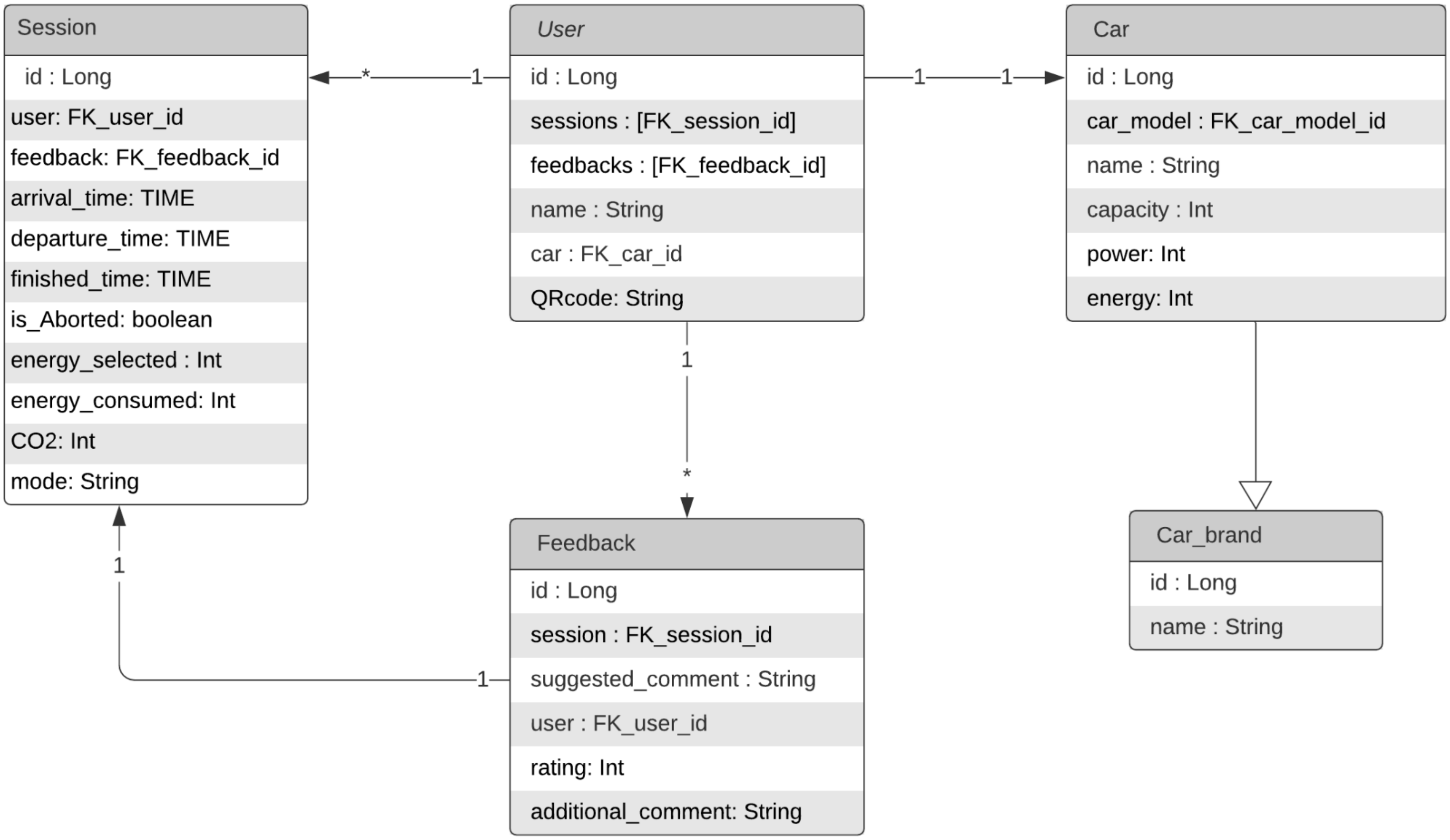


Figure 4. Class diagram of the entire system.

The Class Diagram above represents how the application stores the data. Each class has a unique ID and may interact with one or more other classes. The diagram is designed in a way that it only stores the information that is actually needed for the MVP product.

6.1.1 User Class

The center of the diagram is the User class. So all relationships between classes in the diagram are given from the user perspective. In order to greet the user, their name is stored as String. Since the current version of the web application will be used for research purposes only and participants will be known beforehand, each user is already assigned a unique QRcode (link) to access the web application. Each user will be also registered with one car and will have a foreign key (a reference to the Car class instance) stored in the user class instance as well. This ensures that the User and Car class has a one-to-one relationship, meaning that one user may have only one car and vice versa. Each user may have one or more charging sessions, so foreign keys of the user’s sessions will be stored as an array. Likewise, the feedback that is provided by the user.

6.1.2 Feedback Class

Aside from the unique ID, the instance of the Feedback class may also be identified by the tuple of user and session IDs. This is due to the fact that one feedback can be shared by one user and only about one session. So the Feedback class has a one-to-one relationship with both Session and User classes. The Feedback component itself is quite comprehensive, as was requested by the clients. So ideally, the user would provide a rating for the session (from 1 to 5), would choose one of the suggested comments, and also add any additional remarks. However, all the feedback fields are optional.

6.1.3 Session Class

The application supports three states for the charging session:
- “Charging” state - the session has successfully started and is still in progress
- “Finished” state - the session has finished/completed at 100%
- “Aborted” state - the user aborted the session before it finished (completed below 100%)

Thus, the Session class supports necessary information to maintain its three states, such as:
- “Arrival time” - the time when the user started the session
- “Departure time” - expected departure time filled in by the user
- “Finished time” - the time when the charging stopped (either completed or aborted)
- “is_Aborted” - boolean indication of the aborted session
- “Selected energy” - the desired amount of energy consumed by the car
- “Consumed energy” - the actual amount of energy consumed by the car (car may consume less than desired in case user aborts the session earlier than expected)
- “CO2” - number of CO2 emissions produced by the car
- “Mode” - mode of the session (fast charging mode vs solar power mode)

In addition, the Session class also stores foreign keys of the user and their feedback on the session. Together, these parameters serve as a base for our clients for further analytics and improvement of the algorithm.

6.1.4 Car and Car Brand Classes

The Car and Car Brand classes have an inheritance relationship within each other. For example, Tesla is a Car Brand, which includes cars such as Tesla Model S, Tesla Model Y, etc. Each car then will have different characteristics such as energy consumption, maximum power, and capacity. Both classes included unique identifiers and names stored as strings. The Car class would then inherit any possible fields from the Car Brand Class.

At the current stage of the web application having only one of the classes (either Car or Car Brand) would be enough, since the number of users will be limited to the number of participants for our client's research. However, as the number of users increases, it will be more efficient to store some properties of the cars of the same brand and with same characteristics in the "Car Brand" class, which will be extended by more specific "Car" class.

6.2 Design Choices

Several design choices have been made in consultation with the clients. This ensured that the optimal solutions were implemented given the functional requirements of the system.

6.2.1 System Look & Feel

The system had no starting base, which gave us the freedom to be creative with the Look & Feel of the web application. The core principle of the design decision made was to maintain user-friendliness and intuition behind the UI. For that, we browsed the already existing UIs for electric vehicles (EV). The main inspirations for the design were the Tesla application (Tesla, 2021) and the design published on dribble.com. Thus, the design was enriched with gradient colors, subtle borders and corners, soft colors, and shadows.

According to Rose M. Rider (2010), the green color represents nature and trees, while deep green tones are associated with wealth and status. So the green color was chosen to be incorporated into the design. There were two possible schemes for the design: light mode and dark mode. Due to time constraints, the choice between the two modes was made. Both modes were presented during the design phase as prototypes, where clients preferred the light mode more.

6.2.2 User Login & Registration

There were several discussions on how to register and login the users. The main challenge for the implementation of the registration procedure was the absence of API. Another doubt was that is the registration procedure even necessary since the web application will be used for research purposes only with already established participants. So together with the clients, it was decided to directly register known users into the system and provide them with unique QR codes, so that they could easily access the web application with personalized data on it.

6.2.3 Session States

It was important to establish possible states for the charging session during the design phase of the project. Should the system allow users to abort charging? Should the system allow users to put the charging on pause and then resume it whenever? Would the paused charging session be treated as a separate session or a continuation of the previous one? Those were all the questions that were heavily discussed with the clients. It was decided to allow users to abort charging if they would like to leave earlier, for example, however, the session should never be paused/resumed as solar power, as well as the price for the charging might change over time.

6.2.4 Pop-up Alert

Another scenario that should have been taken into consideration was when the users accidentally abort the sessions. For that, it was decided to implement a pop-up alert that would warn the user that the charging would be aborted and ask users to confirm their choice once more.

6.2.5 Graph

The Graph is the core of the application, as it represents the direct live update of the algorithm. There were several options on how the graph should be represented:

- Bar Graph - shows how much energy has been charged by the car at the moment
- Driving Range Graph - shows how many kilometers a car can drive at the moment
- Power Output Graph - shows the solar energy available at the time, as well as, predicted charging session

All three graphs would change as soon as, user's preferences change. However, it was decided to go with Power Output Graph, since it was more understandable for the users in terms of how their input influences the charging session.

6.2.6 Display of price and CO2

During the earlier stages of the design phase, the price and CO_2 emissions were displayed together with the graph at the end of the page. However, during the meeting with our clients' guests in UAT, the guests advised us to always display the price and number of CO_2 emissions. That would not only make the necessary information for users more accessible but also encourage them to look up statistics, such as the graph. So it was decided to move price and CO_2 emissions values into a fixed bar at the top of the page so that those would always be displayed.

6.2.7 Energy Slider

The clients were not sure whether it would be better to ask users to fill in the desired energy amount in km or in kWh. Thus, in the beginning, our team proposed to implement a converter from kWh to km and vice versa, so that the users would be able to type in values in any format they prefer. After multiple stages of UAT, it was decided to implement a single slider and the converter displayed right below it. Sliding the desired value instead of typing it relieves the user from using the application with both hands, as well as saves screen space instead of displaying the keyboard.

6.2.8 Time Interval

Since users need to input their expected departure time for the algorithm to run, the Time Component was necessary to add. Our team proposed using the Android clock component (Tesla, 2021), due to the ability to insert precise time using one hand only. However, the clients explained that having precise time up to the minute is not important for the algorithm to work, and they would rather prefer users to enter their departure time in 15 minutes range (e.g. instead of 16:04 users would input either 16:00 or 16:15 as their departure time). For that we have decided to implement a custom Time widget, that would be operated with up/down buttons. This way the component is quite intuitive to operate with.

6.2.9 Automatic scroll to Graph

The Schedule page is the most content-loaded page of the web application. It displays the 3 necessary steps to establish the charging session and shows the graph at the end of the page. For readability reasons, the graph was designed to take the major part of the screen. In addition, to fit in all the required components scrolling was enabled on that page only. However, since the other pages of the web app do not offer to scroll, it might not be obvious for some users to understand that the page can be scrolled down to the graph. For this reason, we agreed on automatically scrolling the user’s content on the graph, once the user selected the mode of charging session.

6.2.10 Progress Bar Display

The challenging part of the progress bar was to display the percentage sign of the charging session. It could have been quite ambiguous for the user since the user might think that the percentage sign refers to the battery of the car itself, rather than the progress of the charging session. For that reason, it was decided to display the percentage sign right next to the progress bar, instead of the center of the page, as was planned initially. Please refer to Figure 5.

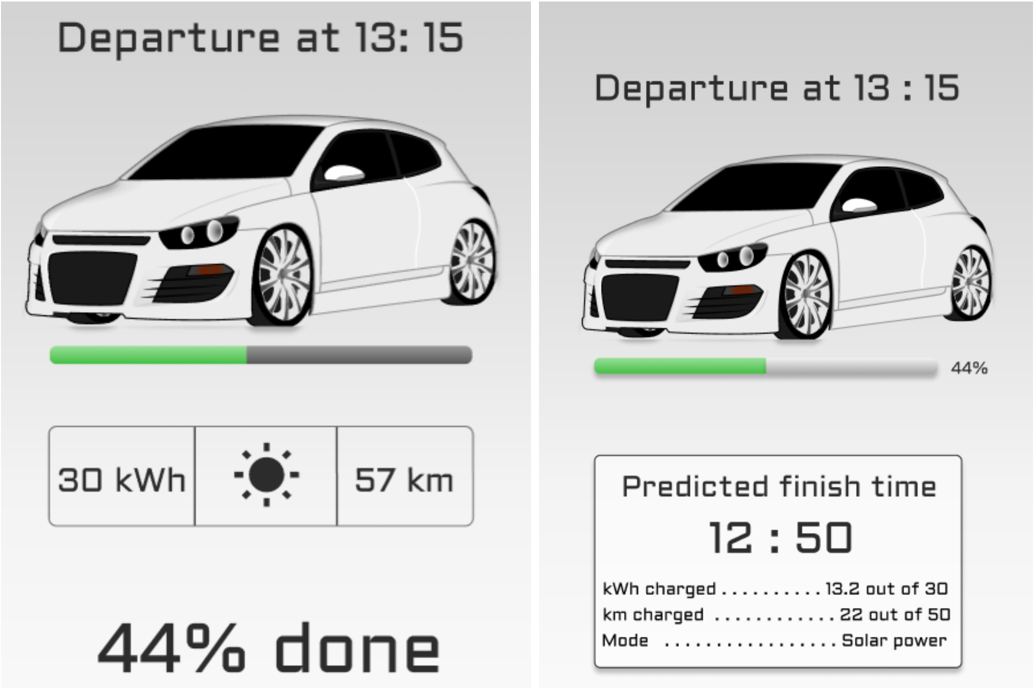


Figure 5. Design changes in Session Page. Before (left picture) and After (right picture).

6.2.11 Table overview Display

In the first version of Hi-Fi, our team offered clients to display chosen preferences for the charging session on the Session page. However, in order to provide users with more insight on the charging, as well as, it was decided to show the users complete table in “amount of amount” format. Please refer to Figure 5.

6.2.12 Feedback

Since the current version of the application will serve as a base for a future project and will undergo a lot of user testing and focus groups, it was important to have a comprehensive Feedback system integrated into the app. That is why the Feedback system we developed consists of three steps:

1. Rating from 1 to 5 (basic star rating, but instead of star emoji, we are using lighting emoji)
2. Choose one of the four recommended comments
3. Add additional remarks and suggestions in the open comment section

Ideally, we would like users to follow all three steps for the feedback, however, we also allow users to close the feedback session or follow any combination of the three steps.

Chapter 7

Test Plan

This chapter aims to offer a plan for all the necessary testing that will be performed before and after the development of the software system.

7.1 Unit Testing

In order to minimize the risks of users finding bugs in the UI, it was necessary to properly test each unit of our software. We had three main types of components in React that needed to be tested. Styled components, which are default HTML parts (e.g. button or checkbox) with additional styling. It was also important to test the widgets, since they were more complex in terms of functionality. In addition, four pages of the application needed to be tested as well.

7.1.1 Styled

For Button, Checkbox and TextField were individually checked to ensure that their styles would not break the responsiveness of the entire app.

7.1.2 Widgets

The widgets are the more complex components in terms of both functionality and nested classes. The functionality of the widgets was tested via “console.log()” (a function that writes a message to log on the debugging console in the “Chrome inspect” tool). For example, Schedule and Feedback pages both store the entered information in the “form” object, so console.log() was used to print the form object and check if the widgets update it accordingly. For some of the widgets cypress tests were written as well.

Later paragraphs provide an overview of all widgets of the web application, and what our group focused on during the testing.

7.1.2.1 Graph

The most complex part of the entire web application is the Graph component, which means that it can also generate the most amount of errors. It can for example break the responsiveness or not generate the expected output based on the two different modes we have. So these things needed to be rigorously tested.

7.1.2.2 ModeSelector

The ModeSelector had to be individually tested to make sure the expected button color is applied once it is clicked. However, more importantly was the system testing in which the interaction of the ModeSelector with Graph components was tested.

7.1.2.3 EnergySelector

The EnergySelector is also error-prone. So it was needed to be properly tested with the default conversion values between kWh and km, such that it would uniformly offer the right increasing values.

7.1.2.4 Time selector

The time selector's functionality is to change the time state according to the up or down change pressed by the user. Thus, we made sure it would only display the hours in 24 hour format and the minutes could never pass 59, while keeping the cycling behaviour. At the same time we assured that the state it's kept among the next pages.

7.1.2.5 Navbar

The Navbar offers two important pieces of information to the user. One is the price and the other is the CO2 emissions. We need to make sure both get updated accordingly and that they are visible to the user, while also making sure during the system testing that the state of the two pieces of information is kept correctly.

7.1.2.6 ProgressBar

It was necessary to simulate time passing and check that the progress bar updates smoothly and according to the rate the update was set to.

7.1.2.7 Comments

It was important to make sure that users are allowed to select only one of the four suggested comments. The unit tests for individual comments from the Styled components tested all the important functionality. However, testing those as a group was necessary to make sure they offer the desired output.

7.1.2.8 Stats

The Stats component was tested to provide the accurate state of charging offered by the rest of the components

7.1.2.9 Rating

We needed to assure that the desired behaviour of the Rating component was achieved. For example, it is only to select the rating progressively, and selecting the first and last 'bolt' for offering a 2 star rating would not be permitted. It was also important to ensure that the color of the selected 'bolt' is changing to green on click.

7.1.3 Pages

For each page, we checked the responsiveness of the elements manually by inspecting the page with different screen sizes, making sure that it still looks acceptable and the elements adapt in size/shape properly.

7.2 System testing

In order to provide good user experience, we needed to test the entire usage of the app. We split this up per page, with some scenarios concerning the transition between the pages too. For each of the pages cypress tests were written based on the following scenarios.

7.2.1 Start page

The only possible interaction between user and Start page is when the user clicks the “Start” button. The expected behaviour is that the user is transferred to the Schedule page.

7.2.2 Schedule page

The schedule page is the most complex page and where the most user activity happens. We had to test that it would be responsive for most device pixel ratios and that it would have the correct state given the right input. The emissions and price should update accordingly based. Finally, we assured that the selected mode would output the desired graph change. Table 3 shows the test scenarios and expected results.

Test Scenario	Expected Behaviour
User sets their preferences	The price and emissions are displayed, as well as a preview of the session in the graph.
User presses “Go” after setting their preferences	Page redirects to the session page, preferences are sent to EMS to start charging.
User presses “Go” without setting energy (energy is at 0 kWh)	An alert pops up and the preferences are not submitted.
User does not select charging mode	Graph, price, emissions do not show up.
User presses “Go” without selecting a charging mode	Same as above; alert pops up and preferences are not submitted.

Table 3. Test Scenarios and Expected Behaviour for the Schedule page.

7.2.3 Charging Session

The charging session page offers up-to-date information of the current charging session to the user, thus it is crucial to assure responsiveness and immediate update of the progress bar and statistics table. Additionally, the user should be able to abort a charging session or to avoid it if the “Stop” button has been carelessly pressed. In Table 4, the test scenarios performed and the expected behaviour are intended to ensure the desirable functionality of the session page.

Test Scenario	Expected Behaviour
User follows the progress of the charging session	The progress bar and statistics update continuously during the charging period.
User wants to abort session	Pop-up asking for user confirmation is displayed, along with two buttons “Yes” and “Cancel”, without interrupting the charging session.
User confirms the abortion, presses the “Yes” button	The abortion time is shown on top of the session page; the progress bar stops and its filled part becomes red; “Stop” button becomes “Finish”.
User cancels the abortion, presses the “Cancel” button	The user is returned to their initial charging session page.
User lets the charging session to end	The finish charging time is displayed, replacing the departure time; “Departure at” becomes “Finished at”; “Stop” button becomes “Finish”.
User finishes the charging session	The Feedback page shows up when pressing the “Finish” button.

Table 4. Test Scenarios and Expected Behaviour for the Session component.

7.2.4 Feedback page

The Feedback page is responsible for handling user’s feedback. Users are transferred to the Feedback page after each charging session and encouraged to share their experience with the application. The page also was tested on its responsiveness. Table 5 shows the tested scenarios and their expected behaviours.

Test Scenario	Expected Behaviour
User submits feedback	“Thank you for your feedback” alert pops up; form is saved; the user is transferred to the start page after

	timeout.
User exits	“Please rate us next time” alert pops up; no form is saved; the user is transferred to the start page after timeout.
User fills in the form but exits	“Please rate us next time” alert pops up; no form is saved; the user is transferred to the start page after timeout.
User submits empty form	“Please rate us next time” alert pops up; no form is saved; the user is transferred to the start page after timeout.

Table 5. Test Scenarios and Expected Behaviour for the Feedback component.

7.3 User acceptance testing

As mentioned in section 3.5, three stages of user acceptance testing have been considered: the poll conducted during the peer review session and the meeting with clients during the design phase of the project, together with prospective users UAT in the implementation phase. Each stage is further discussed in the remaining chapter.

7.3.1 Poll during peer review session

In the second peer review session, a poll has been conducted among four groups from the Design Project Module. The poll aimed at testing two different styles, light mode or dark mode of the system, and dropdowns or scrolling in the schedule page. After all these four designs have been shown to the participants, they have received the following polling questions:

- Dropdowns or scrolling?
- Light or dark?

7.3.2 Meeting with clients

During this meeting, the clients together with two of their colleagues have tested the high-fidelity prototype. The participants have taken this testing phase more as an open discussion of the system, providing recommendations for further improvements and different designs for the graph. They have shown a good understanding of the user interface.

7.3.3 Prospective users UAT

During the final user acceptance testing, we have asked the users multiple questions that would offer us insight into the intuition behind the entire system and we have used the insights in the Future Planning section. In order to see the questions that have been asked, please go to Appendix E. The results of the tests can be found in Test Results.

7.4 Approvals

All the tests conducted were in agreement with our clients and more importantly the same way the system was designed for a certain persona, the same way the test suite is made also for this persona.

Chapter 8

Test Results

In this chapter, the results and final conclusions of the tests discussed in the previous chapter will be covered. They serve as an in-depth understanding of the system behaviour.

8.1 Unit Testing

The purpose of unit testing was to check the responsiveness of the components, as well as correct behaviour of the forms on Schedule and Feedback pages. As was already mentioned those were testing using “Chrome inspect” and “console.log” of the form object. After multiple iterations of testing, debugging and modifications, the pages provided the expected behaviour and were responsive.

8.2 System Testing

After going through the scenarios that we posed, we concluded that the application satisfied all expected outputs after fixing a handful of bugs. The testing was mainly done manually, by inspecting and interacting with the UI, and for a small part with Cypress.

8.3 User Acceptance Testing

8.3.1 Poll during peer review session

Considering the poll made during the peer review session in the design phase of the project, the participants have concluded that scrolling would be a better choice for the schedule page and the dark mode has been mostly preferred. The poll results can be found in Appendix F.

8.3.2 Meeting with clients

The valuable feedback from the participants of this meeting includes the integration of a statistics table instead of only displaying the progress bar on the charging session page. Scrolling has been preferred by this testing group as well, additionally they have suggested an automatic scrolling once the charging preferences are filled in. It has been decided to opt for it in the implementation phase. Finally, the light mode has been the final decision on the UI after discussing with the clients.

8.3.3 Prospective users UAT

The final user acceptance tests have been performed during the implementation phase of the project to two different focus groups of participants. Three members of the CAES group chosen by our clients have tested the system, while our team has decided on three students who do not have a background in smart charging. Further, each participant has concluded two scenarios as it is detailed in section System testing. In the remaining section, the feedback received from these groups will be analyzed, together with the similarities found within each group and the feedback differences between the categories.

Independent of group, the following aspects have been observed and concluded from the first question set, testing user's scrutiny and understanding of the components:

- The users have not paid attention to the price and CO2 emissions, elements of the navbar from the schedule page; however, the participants have tended to guess them or to mention their values found in the table from the session page, which have been hardcoded at that time, thus different.
- The predicted finish time has generally been known.
- At the end of the two scenarios, the participants seemed to understand the difference in price, CO2 emissions, and finish time between the modes, fast charging and solar power.
- The progress bar has been misleading, more than half of the participants could not remember it or considered the bar showing how much their car is charged, not the desired energy.
- Generally, the feedback page has been perceived as not being mandatory, some of them noticing the option of submitting an empty feedback or the closing button.
- When asked about their energy consumption in both kWh and km, the participants could remember maximum one of the values.
- If remembered, the users have been able to understand the difference between the times displayed on the session page, departure time and expected finish charging time.
- STOP and FINISH buttons have been unclear for some users, they could not notice the difference.

From the first group, a relevant knowledge has been expected in smart scheduling and the functionalities of the EMS in order to observe whether any obstacles are faced in understanding the graph from the Schedule page. It has been noticed that the CAES group had a better understanding of the graph and were able to intuit its axes than the student group.

Moving to the next phase of our feedback structure, the average ratings have been received from a scale from one to five, as they can be seen in Table 6. No difference in rating has been noticed between the two groups, thus the average results involve all participants.

			Question	Result
1.	How intuitive is the app?			
	a.		Overall structure	3
	b.		Schedule page	4
		i.	Selectors	4

		ii.	Graph	3
	c.	Session Page		5
		i.	Pop-up	5
		ii.	Table with stats	4
2.	How intuitive is the charging process?			
	a.	Charging until full		4
	b.	Aborting		4
3.	How useful is the information displayed on...?			
	a.	Graph		3
	b.	Table with stats		4
	c.	Conversion of kWh to km		4
	d.	Price and CO2 emissions		4
4.	How user-friendly/clear is the information displayed on...?			
	a.	Graph		2
	b.	Table with stats		4
	c.	Conversion of kWh to km		4
	d.	Price and CO2 emissions		2

Table 6. Average ratings from a one-to-five scale given by both focus groups

At the end, the feedback sessions have been finished with an open discussion. The first impressions have been definitely positive, the participants have appreciated the responsiveness of the system, the design of the components, and its simplicity. However, later the users encountered some problems and inconveniences with the web application. Please refer to Table 7 for the complete overview of the problems during UAT and their current and future solutions.

Problem	Solution
Users could not remember displayed Price and CO2 emissions	**Current solution:** The navbar has been made bigger

	Further solution: change the color of the navbar from white to green to accentuate it more
Users found it annoying to give feedback after every charging session	**Current solution:** for the research purposes feedback should be requested after each session, so there is no need to solve that **Future solution:** Once the application becomes live, it will be more user-friendly to ask for feedback only after an update of the system
Time selector component was not displayed properly on IPhone.	Current problem is that the inputs of the Time Selector component are disabled, since the component is treated as an independent widget. Safari browser in IOS applies special styling to disabled components (Android works fine) **Future solution:** Enable inputs and treat them as direct inputs of the form
Users found Energy selector confusing	**Future solution:** Add info icon with an explanation of what energy selector means
Users found Graph confusing	**Future solution:** Add info icon with an explanation of what graph means
Misleading Progress Bar (users could not understand if the progress bar was for the current session or car’s overall battery level)	**Future solution:** Add info icon with an explanation of what progress bar means
Incorrect display of the Departure Time	**Future solution:** Will be fixed as soon as the API will be connected, since the state of the application will be managed more efficient
Rating on the Feedback page has been working slowly	**Current solution:** The custom rating component was replaced by MUI Rating component, which is faster
When pressing “Stop” on the session page, a blank page has been unexpectedly displayed	**Future solution:** Will be fixed as soon as the API will be connected, since the state of the application will be managed more

	efficient
The “Submit” button on the feedback page has not responded after being pressed multiple times continuously	**Current solution:** The Feedback form was re-written completely supporting MUI elements, which made the form much faster and efficient
The error message on the schedule page has appeared multiple times when pressing “Go”, even if the desired energy and the charging mode have been selected before by the user	It turned out that the problem was related to a much larger issue as constant yet unstable re-renders of the entire Schedule page with all components (even the components that were not changed were still re-rendered). **Current solution:** The number of re-renders was reduced dramatically with the proper use of hooks. Thanks to that the Schedule page rarely re-renders entirely, while its separate components re-render only when updated.
Sometimes Feedback page was not displayed at all	**Current solution:** Once again the entire Feedback form was re-written using much faster components **Future solution:** Connect real API after proper testing
Time Selector component’s arrows were too small	**Future solution:** increase the size of the Time Selector
Progress bar halted at 89%	The problem arises since our group has used a mock-api to simulate a charging session (with real API users would have to wait until the car is charged in real time; could be hours). So there is no need to fix mock-API. **Future solution:** Connect real API after proper testing

Table 7. Problems depicted from UAT and their (possible) solutions. Rows marked green represent the problems that were resolved completely for the current version of the web application.

Further, the participants had additional suggestions:

- The elements of the navbar (price and CO_2 emissions) should be displayed bigger

- The option of a simple view in the session page, only with the progress bar, and another of an extended view, which would contain the table
- Replacing the feedback with a smiley-face rating
- The font should be bigger
- Replacing the graph with a pop-up notifying the users if they are eco-friendly
- More information for the progress bar

In conclusion, the general sentiment from the user acceptance tests has been that the user interface has the potential of bringing impact on driver’s decision when charging their car. Our team had the chance of receiving valuable insights for further development through users’ feedback and general observations of our team on system and users’ behaviour.

Chapter 9

Recommended Enhancements

In the following sections, the next phases in the system development will be discussed. It is worth mentioning that they are out of the scope for the current quartile. Our team does not take the responsibility of delivering the following features of the system. However, they have been agreed with the clients and are part of the project proposal in the bonus section.

9.1 Global and Personal User Statistics

The next phase of the system development should include a new page component showing statistics when the user finishes the charging session. These should refer to both individual and overall performance. Three types of data could be taken into account:

a. Average charging time: the user should be able to compare a certain charging time with the ones from their own past sessions and with the global charging average, considering all sessions.

b. Average energy consumption: in the same manner as above, the user should see his own average consumption and the general one.

c. Total CO2 reduced: the system should compute the CO2 reduction at each charging session and add it to the total amount for both individual and global profiles. This should serve as an incentive to opt for the solar power mode.

9.2 Connection to UT Login and Guest Users

The current stage of the system does not allow guest users to use it, each user has their credentials integrated into their unique QR code. However, ideally, all EV drivers should be able to use the system, regardless if they are employees, students, or other third parties involved with the university. On the other hand, unique generated QR codes may become a burden for the system with the user network expansion. Thus, shifting to a general QR code printed on each EV charging point to facilitate the connection to the mobile application would be desirable. After scanning the QR code, the user will have the possibility to log into the system using the UT account or continue as a guest user. The interface should be adapted to support guest users, who do not have an account registered in the system. In this case, the users will not benefit from the personal user statistics described above but they will be able to compare their current charging session with the global statistics. Further, the user's name found on the starting and feedback pages will be missed. The functionality and efficiency of the system will stay intact.

9.3 Interface with EMS

An interface with the Energy Management System would serve as an aid in handling the communication with its algorithm. The component would be designed to take as input the parameters for the EMS' algorithm. It would ease the work of our clients and the system development and improvement would become more efficient.

Chapter 10

Evaluation

In this chapter, we evaluate the workflow of the project and its outcome. Planning during this quartile, task division of responsibilities, team evaluation, and final results are discussed.

10.1 Planning

Our planning has been semi-flexible throughout the quartile, scheduling a meeting weekly with our clients but meeting among ourselves only whenever we felt it was necessary or practical to do so instead of regularly. Of course, the regular meetings with our clients also provided a regular moment for us to discuss any matters and we frequently met just before or after the meeting among ourselves.

As to keep some structure to our planning, we split our project into types of activities and distributed them over the weeks of the project in a Gantt chart (see Appendix C), and put all tasks that had to be done on our GitHub project board, assigning them to the member(s) working on it.

Inside the documents like this report, we used the Google Docs comment system to assign sections among ourselves.

10.2 Responsibilities

Within our team, the responsibilities have mainly been distributed evenly instead of per specialization. However, not every member of our team were as familiar with certain tools as others and this split made it natural to have the members more familiar with a tool to work with said tool more than the others, while the others took another task upon them in the meantime.

Task	Responsible person
Activity diagrams	Alexandra
Class diagram	Aydan and Ivo
Lo-Fi	Alex
Hi-Fi	Aydan and Alexandra
Start page	Alex
Schedule page + internal components	Ivo and Alex
Session page + internal components	Alexandra
Feedback page + internal components	Aydan
Graph component	Ivo
Global styles	Aydan
Profile Steering Algorithm	Ivo

Cypress setup	Aydan
Firebase setup	Alex
Peer review presentations	Aydan
Manual/Documentation, Deliverables, Testing	All

Table 8. Task division.

10.3 Team Evaluation

In the beginning of the project, one of our members was not able to attend the meetings physically and had to work online, which made it a bit more difficult but we managed to make it work despite this challenge.

10.4 Final Result

The project resulted in a User Interface that is ready to be connected with an API, deployed and further developed. In the process of designing and implementing this, we have discussed a lot with our clients, trying to give them an image of the user's preferences and requirements, and sought to make the UI as intuitive and easy to use as possible while still offering the feature set required by our clients. From the user acceptance testing we can conclude that we did quite a good job in this but there are several points that can still be improved upon, something that our clients will be able to experiment with.

10.5 Conclusion

Through our Waterfall model, we were able to design a new product and refine this to a working UI that pursues the goals of intuitivity. The finished product is also easy to develop further and easy to plug into the bigger system. The frequent requirement validation and the continuous updating of the prototypes was a lot of work but this paid off in the end, thus the more agile aspect of the project management proved useful too.

During the project, we gained valuable insight in the design and development of a product from scratch, in contrast to most projects we had during our study program that were more set in stone in terms of requirements. Not all of us had much experience with the tools and frameworks we used either but we learned quickly and were able to teach each other where needed. In addition to that, we got acquainted with the EMS and its workings behind the front-end and learned a lot about its workings.

Bibliography

1. Cypress. (2021). Why Cypress. Retrieved 10 25, 2021, from https://docs.cypress.io/guides/overview/why-cypress#In-a-nutshell

2. Dribble. (2020, 09 25). EV charging app for oneCharge. Dribble. Retrieved 10 25, 2021, from https://dribbble.com/shots/14266892-EV-charging-app-for-oneCharge/attachments/5916670?mode=media

3. Material Design. (n.d.). Time pickers. Retrieved 10 25, 2021, from https://material.io/components/time-pickers

4. Rider, R. M. (2010). Color Psychology and Graphic Design Applications. COLOR PSYCHOLOGY, 11. https://digitalcommons.liberty.edu/honors/111

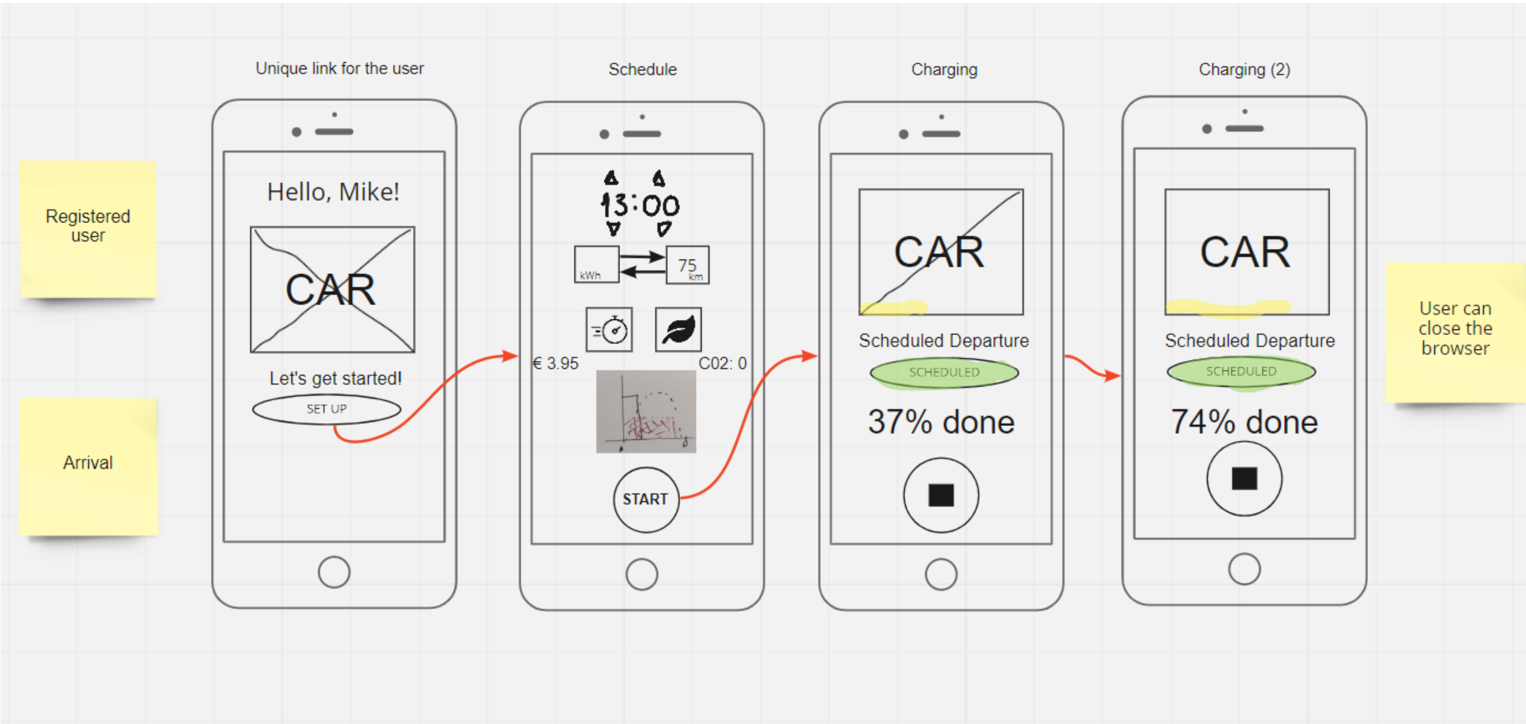
5. Selenium. (2021, October 19). Documentation. Retrieved 10 25, 2021, from https://www.selenium.dev/documentation/

6. Tesla. (2021). Support. Tesla App Support. https://www.tesla.com/en_EU/support/tesla-app

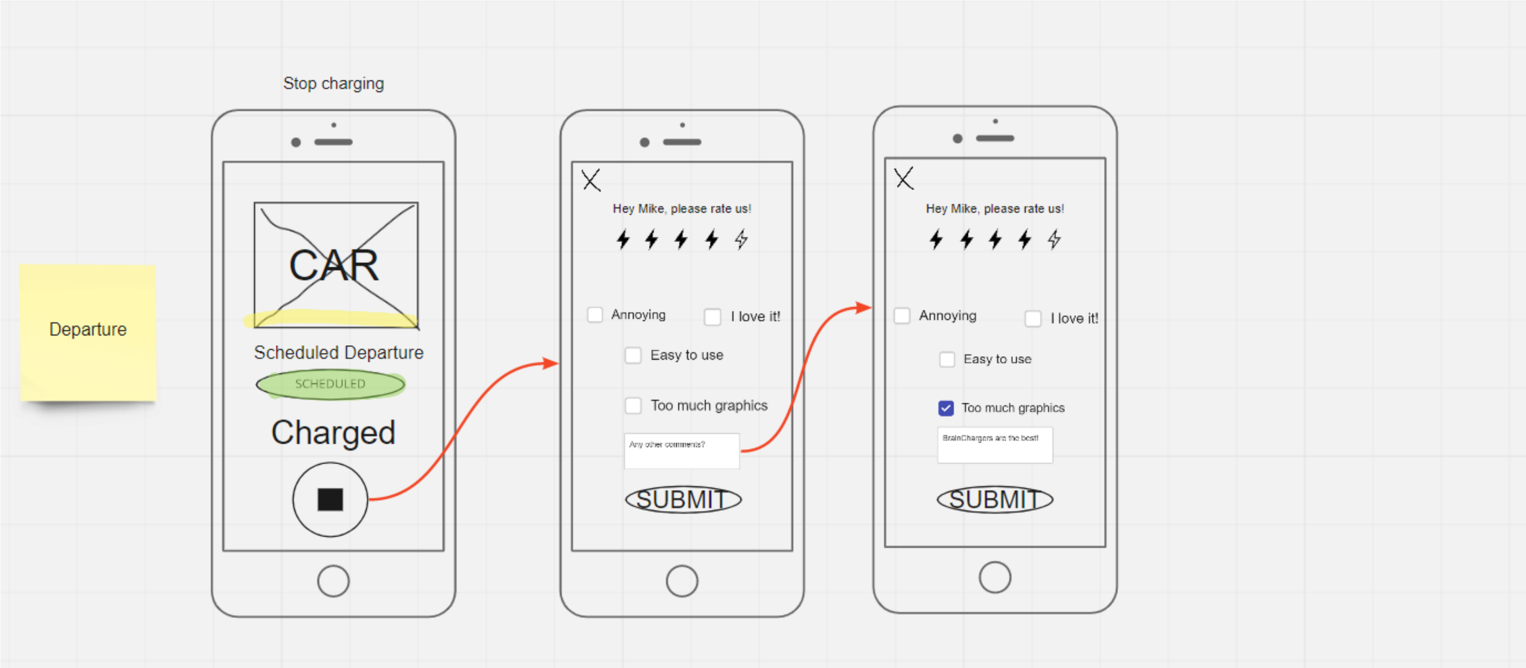
7. Klender, J. (2019, October 26). Tesla rolls out 'Scheduled Departure' feature to customize charging completion by time and day. Teslarati. Retrieved 10 25, 2021, from https://www.teslarati.com/tesla-scheduled-departure-customized-charging-explained/

Appendices

Appendix A. The final version of the Lo-Fi

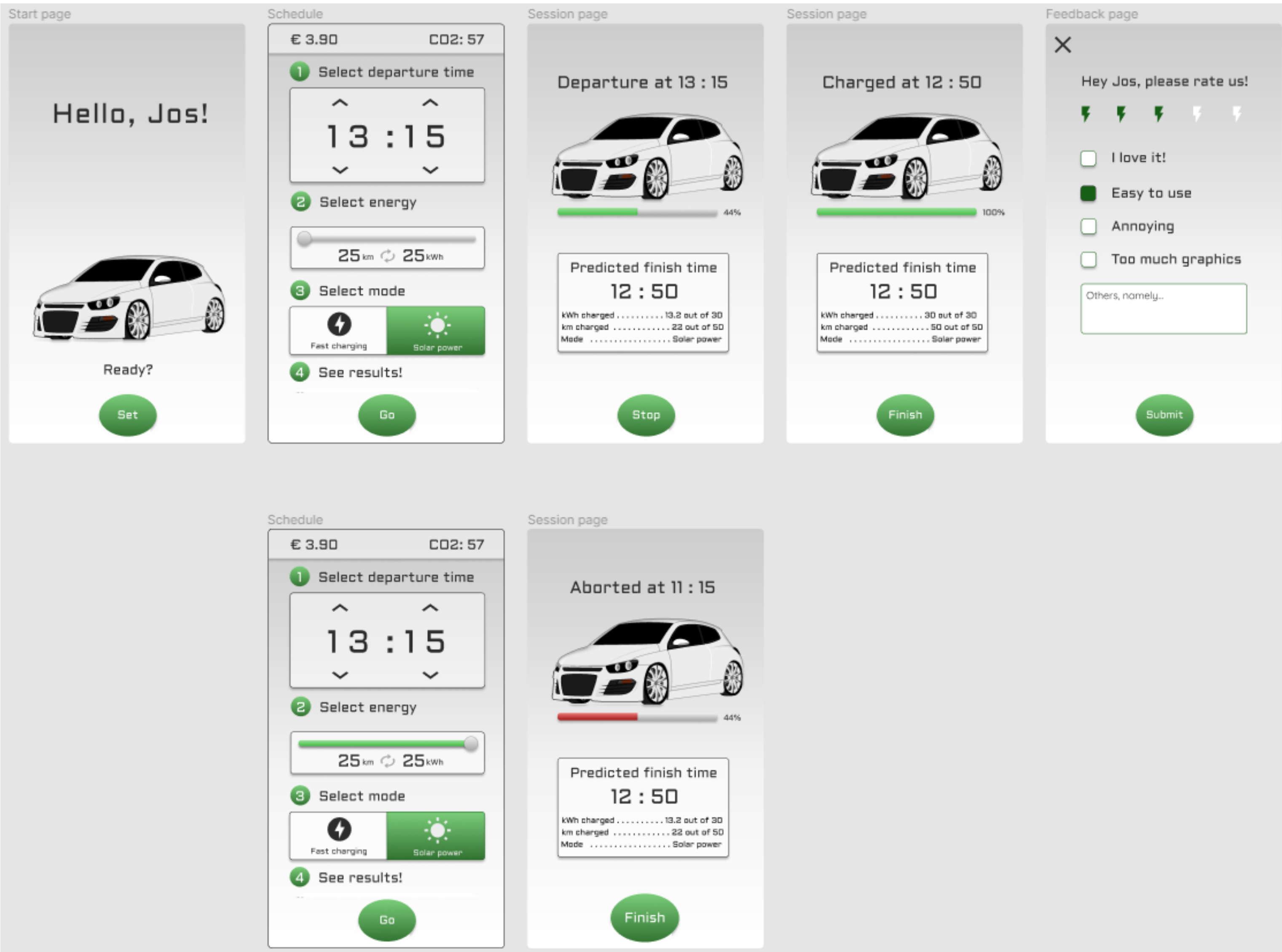


Workflow of the web application at the user arrival

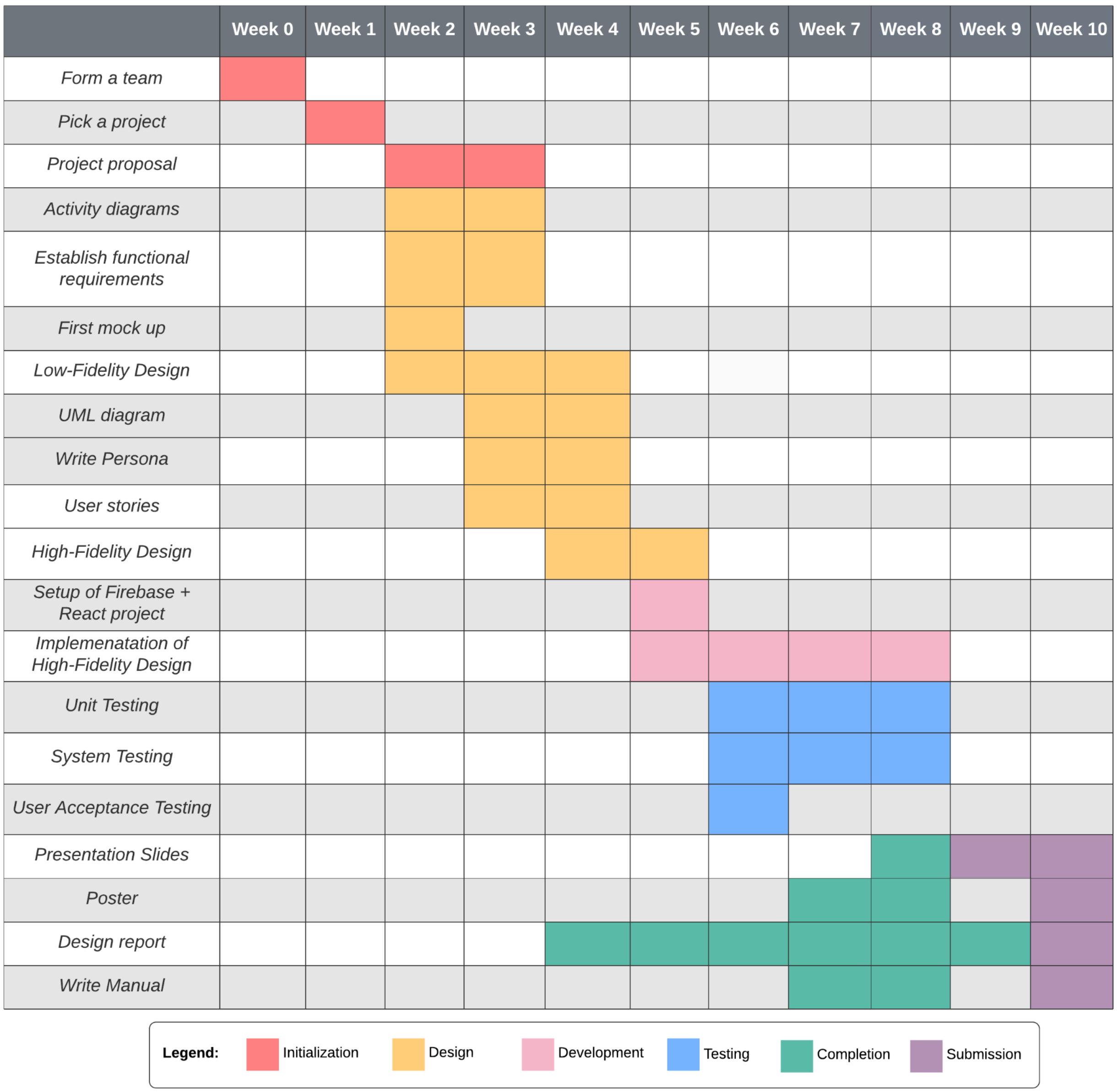


Workflow of the web application at the user departure

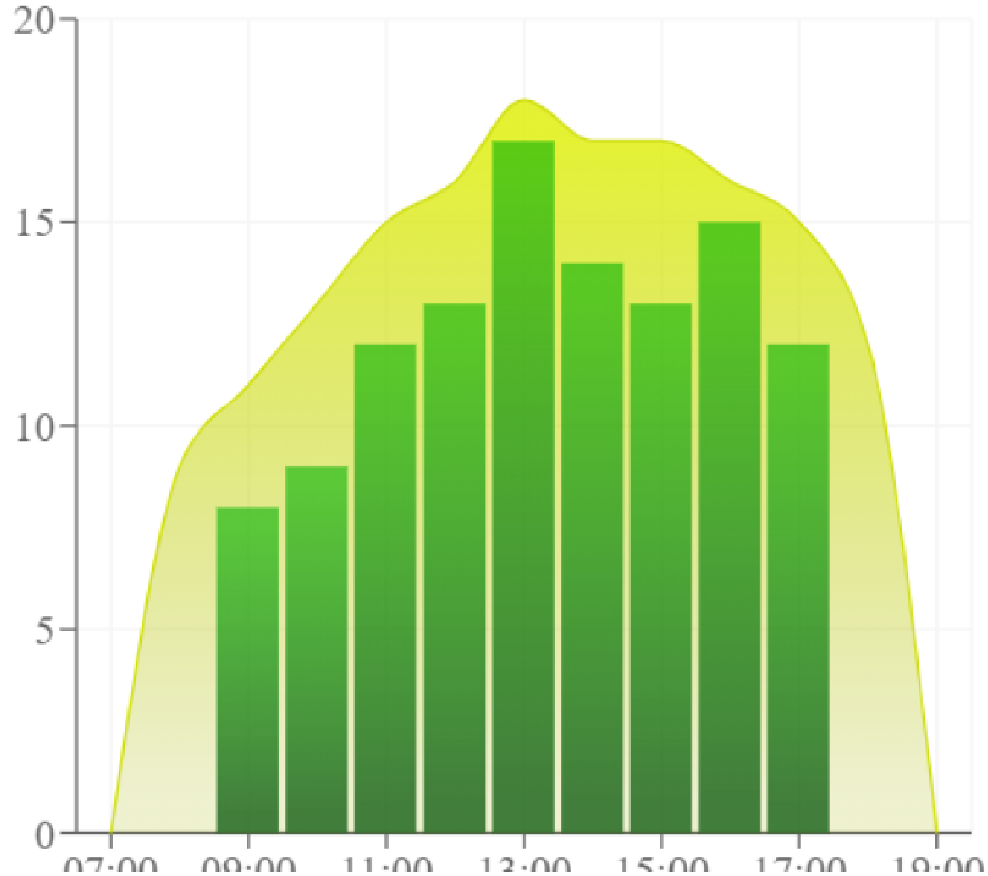
Appendix B. The final version of the Hi-Fi



Appendix C. Gantt chart of the project



Appendix D. Power Output Graph



Example of Power Output graph: yellow area - solar energy prediction; green bars - predicted charging schedule; power output over time (x and y axis respectively)

Appendix E. User testing questions

First words

We are going to give you a web app to test and two scenarios in which you will have to try to use it.

Scenarios

(do not tell which scenarios, just how many)

Let's presume you have an electrical vehicle and you want to charge it on campus. In order to accomplish this in a real situation you will be given a RFID card that needs to be scanned by the EV charger. After this you will have to scan a QR code, which is located on the back of the RFID card, with your phone.

I. Let's assume you do your work in the meantime an you come back when your car is fully charged
II. Now let's assume that you wanted to charge your car fully but you have to leave earlier than expected

Questions

(do not correct them unless it ruins the next questions)

1. What was the price of your session?
2. How much CO2 emissions were produced by your charging session?
3. What did the graph show you?
    a. What did the yellow area represent?
    b. What did the green bars represent?
4. What was the predicted finish time of your charging session?
5. Can you try to describe the difference in scheduling between 'fast charging' and 'solar power'? E.g. Which one finishes earlier; which one makes less CO2, is cheaper; etc.?
6. What did the green horizontal bar mean?
7. What can you do if you do not want to submit feedback?
8. Which questions of the feedback page were mandatory?
9. How much energy did you charge your car with?
    a. How far can you drive with this energy?
10. What is the difference between the two displayed times on the session page?
11. What can you do if you have to leave earlier than expected but your car is not completely charged?
    a. *In case they do not answer (pop-up is not triggered):* What happens when you press 'Stop'? And if you press 'Finish'?

b. *<u>In case they answer:</u>* if you accidentally pressed this button and don't want to stop the session, what can you do?
c. *<u>In case they stopped/aborted the session:</u>*
    i. How much was charged? (%) 89
    ii. How many kms were charged?
    How much kWh were charged?

Feedback questions (1 to 5 Scale)

1. On a scale from 1 to 5, how intuitive is the app?
    a. Overall structure
    b. Schedule page
        i. Selectors
        ii. Graph
    c. Session page
        i. Pop-up
        ii. Table with stats
2. On a scale from 1 to 5, how intuitive is the charging process?
    a. Charging until full
    b. aborting
3. On a scale from 1 to 5, how useful is the info displayed on..?
    a. The graph
    b. Table with stats
    c. Conversion of kWh to km
    d. Price and emissions
4. On a scale from 1 to 5, how user-friendly/clear is the info displayed on..?
    a. The graph
    b. Table with stats
    c. Conversion of kWh to km
    d. Price and emissions

Open discussions

1. Initial impressions and comments?
2. Was it easy to answer our questions during the scenarios?
3. Did anything frustrate/annoy you during the use of the app?
4. If you would change/add one thing in the app, what would it be?
5. Any additional features/components that you would like to see in a future version?
6. Any other comments? Closing remarks?

Appendix F. Poll Results

Polling Question
Dropdowns or scrolling?

1: 5 ||||||||| 38%
2: 8 |||||||||||||| 62%

Polling options
1: Dropdowns
2: scrolling

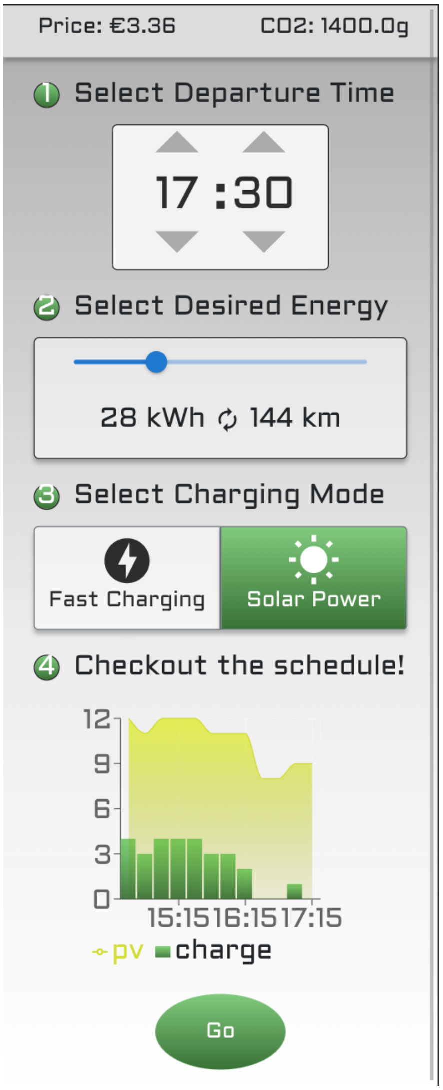
Polling Question
Light or dark?

1: 3 ||||| 20%
2: 12 ||||||||||||||||||| 80%

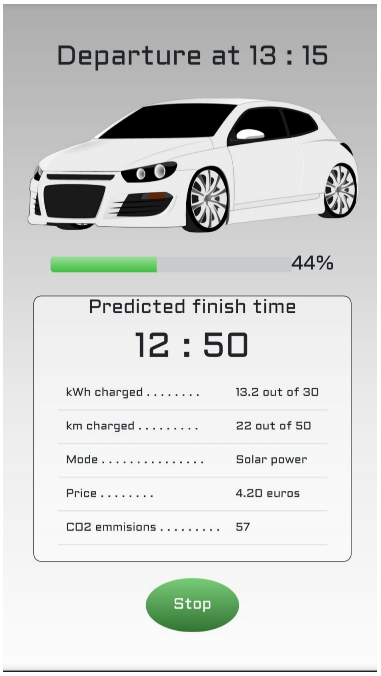
Polling options
1: Light
2: Drak

Appendix G. Final Product

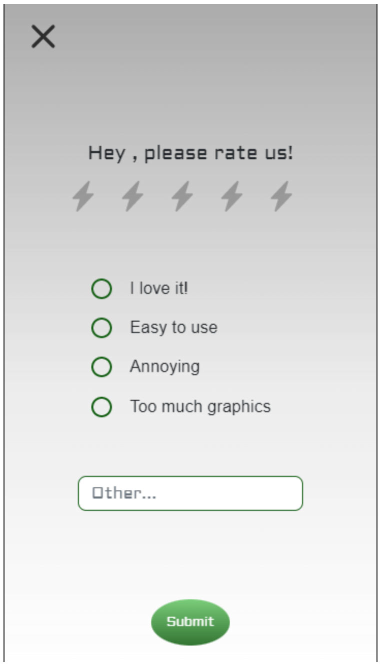
Start Page



Schedule Page



Session Page



Feedback Page