POILTECNICO

MILANO 1863

e-MALL

e-Mobility for All

RASD

Requirement Analysis and Specification Document

Sara Limooee, 10886949
Parham Ebadi, 10870289

Fall 2022

Table of Contents

1. INTRODUCTION **1**

1.1. PURPOSE 1
1.1.1. Goals *2*
1.2. SCOPE 3
1.2.1. World phenomena *4*
1.2.2. Shared phenomena *4*
1.3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS 5
1.3.1. Definitions *5*
1.3.2. Acronyms *5*
1.3.3. Abbreviations *6*
1.4. REVISION HISTORY 6
1.5. REFERENCE DOCUMENTS 6
1.6. DOCUMENT STRUCTURE 6

2. OVERALL DESCRIPTION **7**

2.1 PRODUCT SERVICE 7
2.1.1. Scenarios *7*
2.1.2 CLASS DIAGRAM *9*
2.1.3. State charts *10*
2.2. Product Functions *11*
2.3. USER CHARACTERISTICS 12
2.4. Assumptions, Dependencies and Constraints *13*

3. SPECIFIC REQUIREMENTS **14**

3.1. EXTERNAL INTERFACES REQUIREMENTS 14
3.1.1. User Interfaces *14*
3.1.2. HARDWARE INTERFACES *17*
3.1.3 Software Interfaces *17*
3.1.4 Communication Interfaces *17*
3.2. FUNCTIONAL REQUIREMENTS *18*
3.2.1. Use Case Diagram *20*
3.2.2. Use Cases *21*
3.2.3. Sequence Diagrams *28*
3.2.4. MAPPING ON REQUIREMENTS 34
3.3. PERFORMANCE REQUIREMENTS 38
3.4. DESIGN CONSTRAINTS 38
3.4.1. Standards Compliance *38*
3.4.2. HARDWARE LIMITATIONS 38
3.4.3. ANY OTHER CONSTRAINTS 38
3.5. SOFTWARE SYSTEM ATTRIBUTES 39
3.5.1. RELIABILITY 39
3.5.2. Availability *39*
3.5.3. SECURITY 39
3.5.4. MAINTAINABILITY 39
3.5.5. PORTABILITY 39

4. FORMAL ANALYSIS **40**

4.1. CODE 42
4.2. Result *49*
4.3. GENERATED INSTANCES 49

5. EFFORT SPENT **50**

6. REFERENCES **51**

1. Introduction

1.1. Purpose

In recent years, due to the al warming and the increase in the number of pollution in the air produced by fuel vehicles because of using petrol, gasoline etc, there is a huge need for models of cars which must be less disastrous for the environment. Hybrid electric vehicles (HEVs) and electric vehicles (EVs) use less or even no fuels so they have much less effect on the environment. However, these kinds of cars need to be charged whenever their battery is low. So, there must be some Charging Point Stations so that the EV drivers can charge their car, thus generating new business opportunities such as EV charging business which lets the EV drivers understand what options they have.

These Charging Point Stations are handled by Charging Point Operators (CPOs). While, each CPO has its own IT infrastructure administrated through the so-called Charge Point Management System (CPMS), they can decide to work with different Distribution System Operators (DSOs) which produce and provides energy to charging point stations.
The idea is to develop an electric Mobility Service Provider(eMSP) so that users who are EV drivers can easily make a decision among various charging points and book a place for charging their electric vehicles based on some factors e.g., distance, price, etc.

This document focuses on the Requirements Analysis and Specification Document (RASD) of the system and describes the main goals, the domain assumptions, the scenarios which may happen, the uses cases, the list of functional and non-functional requirements which system should fulfill and finally the diagrams to visualize the interactions between components and performance of the system.

1.1.1. Goals

Goals	Descriptions
G1	eMSP allows EV-drivers to search for charging stations nearby, their cost, and any special offer.
G2	eMSP allows EV-drivers to book a charge in a specific charging station 15 minutes before connection.
G3	eMSP allows EV-drivers to sort the available charging point stations based on the selected filter.
G4	eMSP allows EV-drivers to rate the charging station they received the service from.
G5	eMSP allows EV-drivers to add credit to the wallet inside their account for paying later.
G6	eMSP allows EV drivers to pay for the obtained service either using internal wallet or by credit card.
G7	CPMS allows CPOs to know about the amount of energy used by each vehicle.
G8	CPMS allows CPOs to know the “external” status OF charging stations, e.g. location of available charging sockets, type, cost
G9	CPMS allows CPOs to monitor the charging process to infer when the battery is full
G10	CPMS allows CPOs to know the “internal” status of a charging station
G11	CPMS allows CPOs to know the current price of energy acquired from the DSOs

1.2. Scope

In order to help EV-drivers to find nearby charging stations for their electric car and know about any special offer that either the eMSP or the CPMS (means any offer or suggestions that the CPOs give to the customers), The application has the following parts:

1. EV-driver login to the eMSP system and inserts data about his/her car. According to this information, he can search for charging stations by various filters e.g., distance, price, type of sockets, etc. He can also pay for the obtained service through the interface provided by the eMSP.
2. CPOs log in to the CMPS system to control the general internal and external status of their charging stations, the current price of energy obtained from each DSO that the charging station is working with and to control some other functionalities that can be also controlled by humans manually.
3. eMSP system interacts with the CPMS system to start the charging process, shows the charging process to the end user (EV-driver), giving information about the available charging stations and free sockets of any type of socket (slow, fast, rapid) and their prices and generally any information that must be changed between eMSP and CPMS.

EV-drivers might receive some special offers either from eMSP system for using the application or from CPMSs. Moreover, some suggestions can be given to EV-drivers about when to charge their car based on the previous charging and the distance that the car has passed (this information can be obtained by an API between the CPMS and the navigation system of the car in order to estimate how much distance a car can go at most).

Since this project has been done by a group of two students, the last point mentioned above, the suggestions given by CPMS to the EV-drivers have not been considered in the project.

1.2.1. World Phenomena

World Phenomena	Descriptions
WP1	The user finds out that his/her car needs charging
WP2	The user drives to the station
WP3	User plug in the car
WP4	The user shows the QR code of booking confirmation to the station

1.2.2. Shared Phenomena

Shared Phenomena	Descriptions	Control
SP1	User searches for nearby charging stations	World
SP2	eMPS shows the nearby stations by default	System
SP3	User sort the available stations	World
SP4	eMPS shows the occupied and free sockets of that station (type, cost, etc.)	System
SP5	eMPS shows the amount of time for the first occupied socket to be freed	System
SP6	The user books one of the free sockets	World
SP7	eMPS provides a receipt (QR code) for booking confirmation	System
SP8	eMPS shows the remaining time of charging to the user	System
SP9	Users can monitor the charging process of his/her car	World
SP10	eMPS notifies the user when the charging process is finished	System
SP11	User pay for the bill	World

1.3. Definitions, Acronyms and Abbreviations

1.3.1. Definitions

Definitions	Descriptions
External Status	number of charging sockets available, their type, cost, and estimated amount of time until the first occupied socket is freed
Internal Status	amount of energy available in its batteries, number of vehicles being charged, amount of power absorbed, and remaining time of the charge of each vehicle
Notification	A message shown to the user by the system when he/she must be notified about something (ex: when a new offer is available, or when the charging process starts or finishes)
API	Stands for Application Programming Interface. APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

1.3.2. Acronyms

Acronyms	Descriptions
eMPS	e-Mobility Service Provider
CPO	Charging Point Operator
CPMS	Charge Point Management System
DSO	Distribution System Operators
API	Application Programming Interface

1.3.3. Abbreviations

Abbreviations	Descriptions
G	Goal
WP	World Phenomena
SP	System Phenomena
D	Domain Assumption
R	Requirement

1.4. Revision History

Version	Date	Modifications
1.0	17/12/2022	First version
2.0	22/12/2022	Final version Added: • Alloy results • Diagrams • User interface mockups • effort spent Updated table of contents and refernces

1.5. Reference Documents

- Specification Document: "Assignment RDD AY 2022-2023_v2.pdf"
- Course slides
- "OCPI-2.2.1.pdf"

1.6. Document Structure

- Section 1

Contains an overview of the purpose of the project and defines the scope of the system. Describe the specifications such as the definitions, acronyms, abbreviations, revision history, and references. As well as introducing the goals, world and shared phenomena of the application.

- Section 2

Defines the main scenarios and explains the main features in the application by class diagram and state charts. Explaining actors of the system in user characteristics. The subsection product function defines the functionalities of the application. Finally, the domain assumptions are defined.

- Section 3

The main part of the project introduces interface requirements such as user interface, hardware interface, software interface, and communication interfaces. Presenting the functional requirements that are shown by use case diagrams and sequence diagrams. In the end, the mapping of requirements to use cases is written.

- Section 4

Analyzing the system using Alloy Language and brief comments for clarifying the Alloy codes.

- Section 5

Amount of time spent by each member of the group for the project.

- Section 6

Contains the references used to make the project done.

2. Overall Description

2.1. Product Perspective

2.1.1. Scenarios

1- Book a charge:
Mike is an employee in a big city. Every day he needs to use his electric car to go to his office. he has heard about the new system called eMSP in his town. The new startup provides fantastic offers to customers so He was motivated to use this service to charge his car. He registered himself in the application as a customer. He searches for nearby stations and filters the results by price and distance. The eMSP also shows him the specific types of sockets available in each station. He books a charging socket in the specific charging stations. After the booking process is completed, the eMSP provides an electronic receipt containing a QR code to the user, which is required to start the charging process.

2- Charging process:

Monica is an old customer of the eMSP system. One day when she feels that her car needs to be charged, she logged in to her account as a customer and has reserved a place in a charging station near her house. When she arrives there, she parked her car in the reserved area for her and plugged in her car, and uses the digital QR code provided after booking, to start the charging process. The CPMS (Charge Point Management System) automatically scans the QR code and validates her personal and car information and begins to charge the car. The CPMS shows the charging process and the remaining time for the charging to be finished.

3- Payment Process:
David is an EV driver. One day he attended one of the charging stations to charge his electronic car. After the charging process of his car, Based on the type of socket and the current price of energy that was used for charging his car, a bill is issued for him by eMSP. So, In order to reach that, first he logs in to his personal account and after the charging process is finished, he will be notified by the eMSP, so He can choose whether to pay it with cash or by credit card. Then he chooses to use his credit card to pay the bill so he clicks on the credit card icon and pays the bill. After ending the payment process, the receipt of payment is sent to him. He can download the file of that receipt via the share button.
Also, after each charge, the system will give some tokens(the amount of tokens is in relation to the amount of money that is paid every time), which can be stored in the virtual wallet and be used as a discount for the next time of charging. They can be also used as a discount in some other markets, hotels, restaurants, etc. which have a contract with our system.

4- Making decisions about the way of using energy:
Mattia is a manager of a charging station in a crowded city. As there is hard competition among all CPOs in the city, organizing the price and the energy consumption of the whole station is an important matter in order to increase the efficiency of the station and to get a higher rate from all consumers. Alex is a manager in one DSO, an external 3rd party company that provides energy to the charging stations. There is an interaction between CPOs and DSOs throw specific API. As usual of every day, Mattia comes to his office and logs in to his specific account which is designed for CPOs managers. On the main page, there are available functions for him. So, he can enter the station's status page, where he can monitor the situation of his station and then make decisions about whether to use batteries or get energy from DSO or mix of energy and when to store energy in the batteries of the station.

5- Monitoring the external status

Sarah is the head manager of one of the CPMSs in a big city. One of her responsibilities is to provide information about the charging station that is working with her company. In order to achieve that goal, she has to log in to her personal account which is designed for CPMSs managers. On the main page, there is a button named (charging station status), she clicks it and transferred to the page containing a list of stations. After clicking one station, she will be transferred to the next page. On the next page, to view the external details, she clicks on the (External status) button. The system has provided different facilities for them to check the external status of a station which contains different things such as charging sockets and their types (fast, slow, rapid). She can easily determine the types of sockets available in the specific CPO. Moreover, she can enter each CPO section and specify the remaining time for each socket to be freed. All this information is being shown to users through CPOs by eMSP.

6- Monitor the internal process:

Alice as an internal manager of CPMS has a lot of duties in the company. One of her main duties is updating the internal status of the CPO who is working with her company. Every day she should log in to her personal area in the application. And then there are some available functions for her to do. She first enters the charging station's status and after choosing one station, she will be able to see the general information of that specific station. To visit the internal status of that station, she clicks on the (Internal status) button. On the next page, there are two buttons named (charging vehicles, view battery energies), she clicks on charging vehicles to monitor the number of cars being charged in the station. Furthermore, she is able to check the amount of power absorbed by each car in the station and the remaining time of charging each of them. And also she has to check the status of the batteries, so she enters the view battery energies page in which she is able to check the energy status of the batteries. She should update this information and send them to CPO.

7- Rate process

Daniel is an old user of eMSP and prefers to use it to charge his electronic car. One day when he is driving to his work, he feels that his car needs to be charged. So, he login into his personal account on eMSP application. After entering the main page, he searches for the nearest station and goes there for charging. After the payment process is finished, he is notified to rate the CPO that he has used. When he clicks on the rating button he is transferred to the next page where he can enter his comment about the service he received, also he can tick (Anonymous) button. In the end, he clicks on the (submit) button and is notified that his comment is posted. Moreover, he is able to see his previous comments on other stations by clicking on the (comment) button and he is able to edit them.

2.1.2. Class Diagram

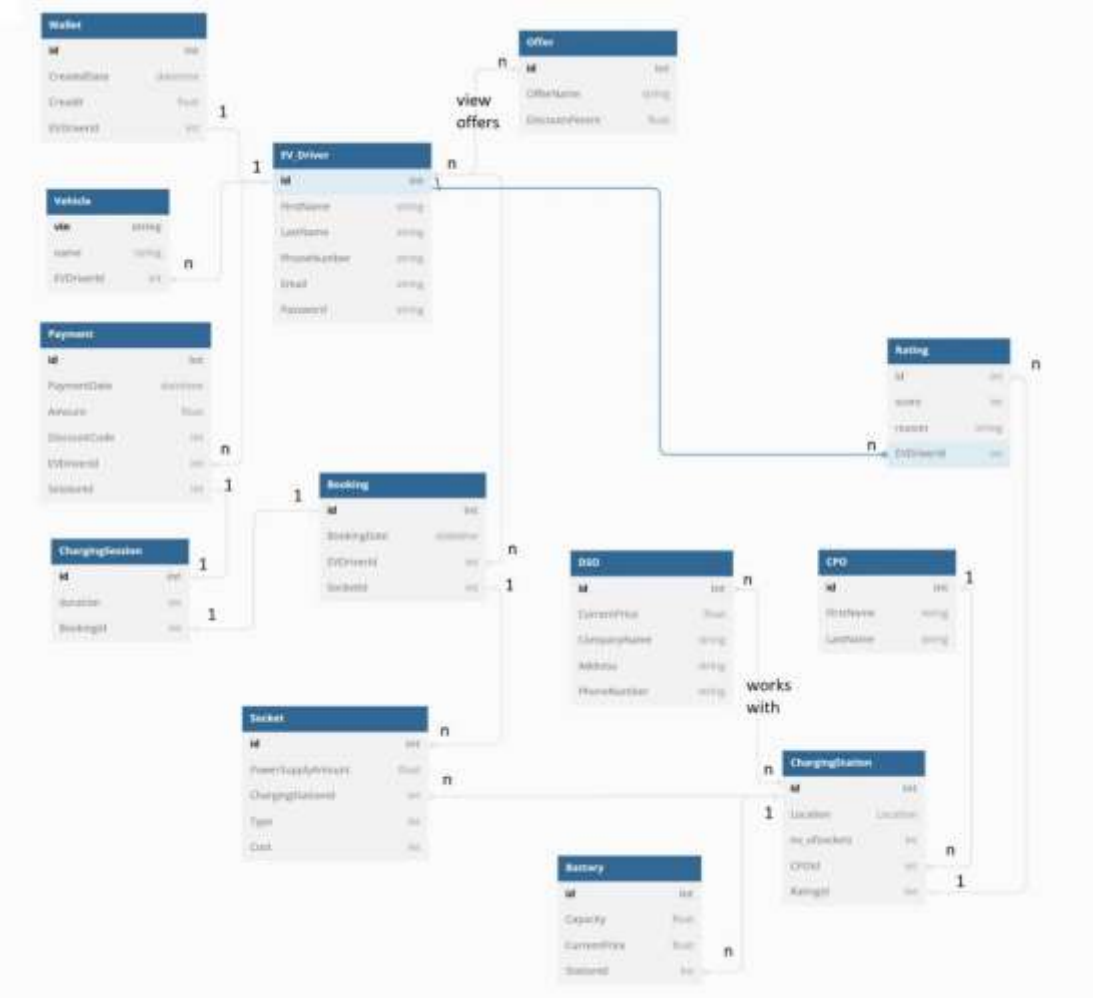

Figure 1: Class diagram

2.1.3. State Charts

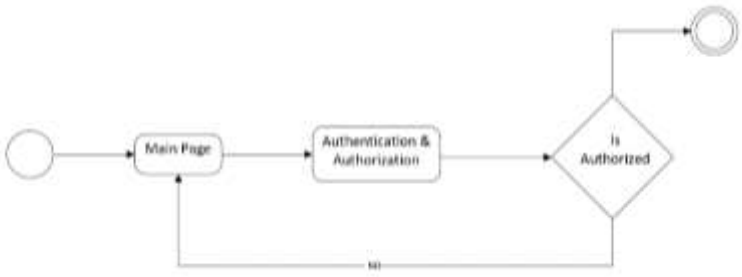

Figure 2: Authentication

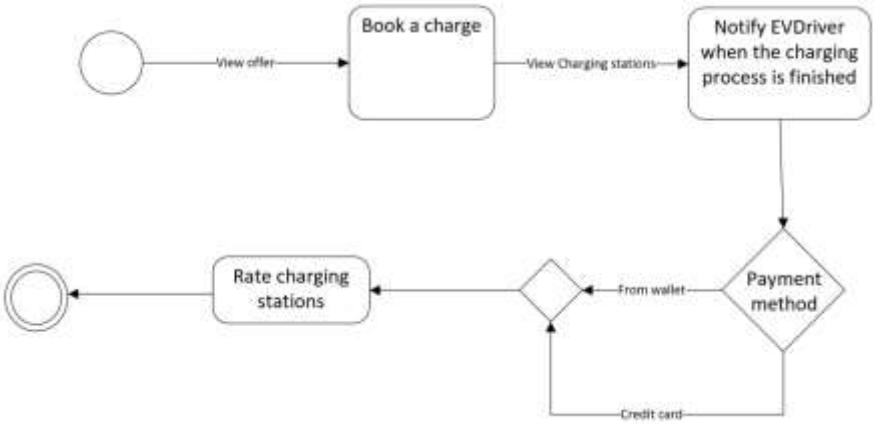

Figure 3: Book (EV-driver can view offers and then book a place in a charging station. After charging process is finished, the eMSP sends a notification to the EV-driver mobile phone to redirect him/her to payment page. He/she can pay for the obtained service either through wallet or credit card. Finally, he/she can rate the charging stations.)

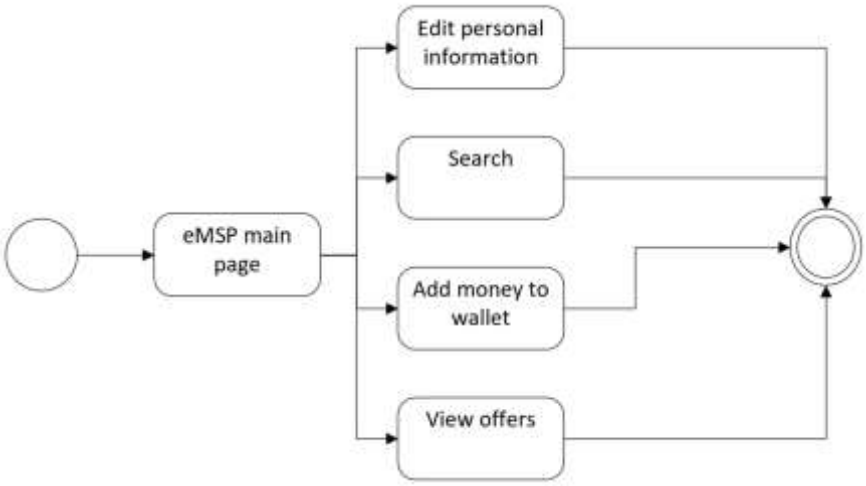

Figure 4: Add money to wallet

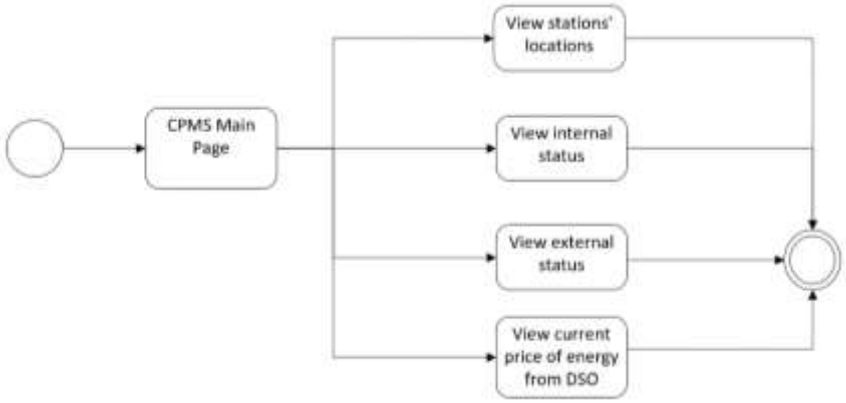

Figure 5: CPO

2.2. Product Functions

2.2.1. Search

Searching is a very important functionality of eMSP system which is also the main goal of the system so that EV-drivers can easily find the best charging station that is suitable for them. They should first register and log into the application and press the Search button. There they can view the charging stations by either list or their location on the map. They also have the option to filter the charging stations based on various factors, their rate, distance from their e-vehicle and lowest or highest price. They can also view some more details of each charging station by clicking on each of them.

2.2.2. Book

Booking a place in one of the charging stations is another important function for the EV-drivers. They are able to charge a place 15 minutes before the connection. They also have the option to choose between rapid, fast, or slow sockets which are different in price.

2.2.3. Start Charging

This is a CPMS system function that is run by the system automatically after EV-driver connects his/her car to the charge and the CPMS scans the booking QR-code verification. CPMS system decides the amount the power to use for charging the vehicle, from which DSO to acquires energy, and the charging method (only from internal batteries, only from DSO energy source, or a mixture of batteries and DSO energy source).

2.2.4. Pay

After the charging process is finished, the user (EV-driver) is redirected to a payment page where he/she can choose whether to pay using the internal wallet or pay with a credit card by entering card information.

2.2.5. Rate

After payment is done, the user can rate the service he/she received so that both charging point operators and eMSP can improve their services. Users can give a score and provide a reason for the score he/she gave.

2.2.6. View internal/external status

One of the main functionalities of CPMS system for CPOs is to provide an interface for them so they can monitor and view the internal/external status and locations of charging stations. They can also search for charging stations through the map and click on any of the stations to view more details of that specific station.

2.2.7. Add new offer

CPOs can provide different offers to the EV-drivers for different events. They can add a new offer through their CPMS system. Then a message is sent to all eMSPs that the CPO is working with that a new offer has been added and that they can show it to their users (EV-drivers). They can Insert a name and a discount percent (the amount of the offer) and a code so that EV-drivers can use it in the payment section after viewing the offer code the in View Offer Page of the eMSP application.

2.3. User Characteristics

- EV drivers:

  A person who registers in the eMSP system as a user and has an electric car. They can search for charging stations and any available charge based on different factors e.g., distance, price, and type of sockets. They can also book a charge in a specific charging station 15 minutes before connection and pay for the obtained service through the app after the charging process is finished.

- CPO:

  A person who can control the external and internal status of charging stations through CPMS system and can control some functionalities of the CPMS system. CPOs are operators of the CMPS system who works for the owner company of charging stations and they do not need to register in the system. The only action for authorization is the login.

2.4. Assumptions, Dependencies and Constraints

Identifier	Descriptions
D1	The voltage of charging is compatible with the user's car model.
D2	EV driver inserts the car specifications correctly
D3	The charging stations always cam have batteries and/or are supported by DSOs
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct.
D5	Energy supply from Energy Company(DSO) is continuous and without problems
D6	The Current price of energy provided by DSO is correct and updated instantly
D7	EV driver pays the bill first and then plugs out his/her car
D8	The user must be able to attend the station in 15 min after booking a charging space
D9	The comments and rates which are given by every EV-driver for each station are correct
D10	Each EV-driver uses each station's facilities correctly and after finishing the charging process, the EV-driver put the hose back in its space correctly

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

The following mockups show the different main parts of both eMSP and CPMS systems interacting with EV-drivers and CPOs respectively. Complete sections of the systems will be mentioned and discussed in Design Document.
Both systems have mobile applications so that the related user (EV-driver and CPO) can use it.

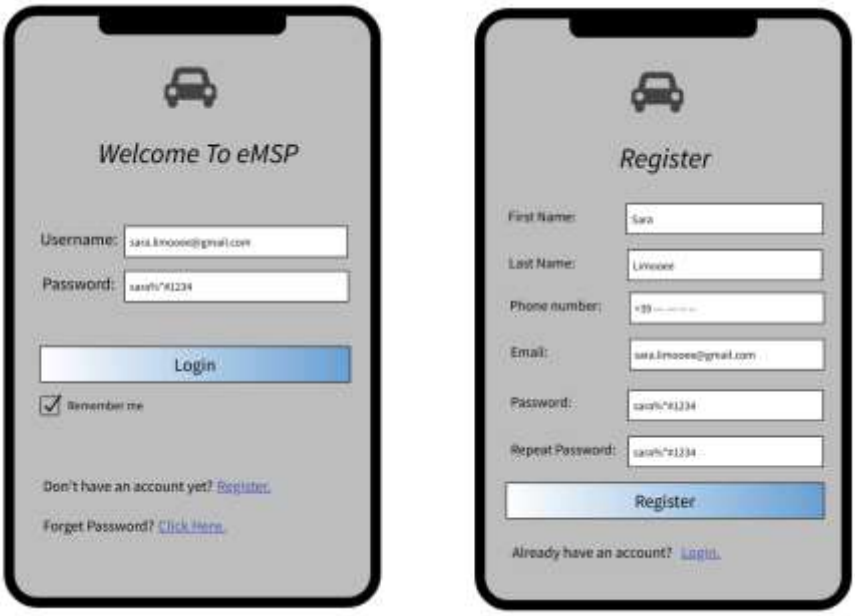

Figure 6-Mockup(eMSP): Login - Register

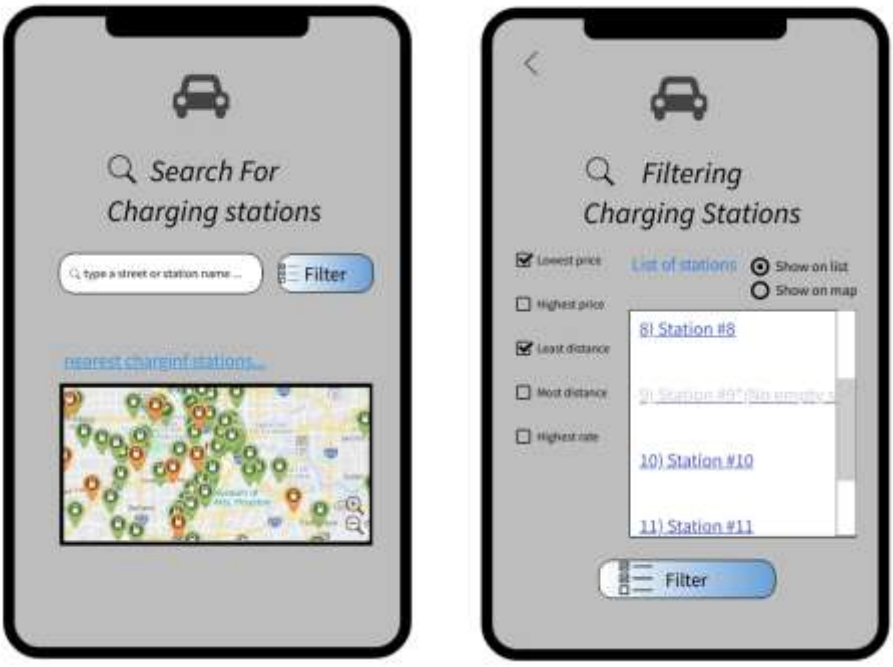

Figure 7-Mockup(eMSP): Search - Filter stations

Figure 8-Mockup(eMSP): Booking receipt - Book

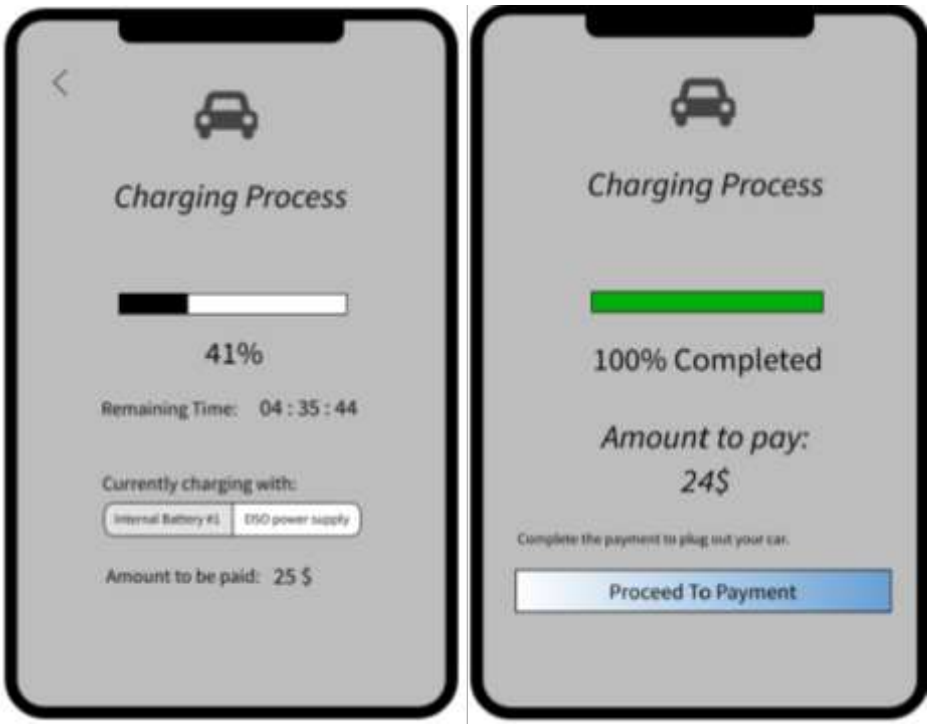

Figure 9-Mockup(eMSP): Charging process - Charging process completed

Figure 10-Mockup(CPMS): CPMS system – Charging stations status

Figure 11-Mockup(CPMS): Internal and External status

3.1.2. Hardware Interfaces

The hardware that both systems eMSP and CPMS need are:

- EV-drivers have to use a smartphone or web browser where they can search for charging stations and book a charge and make payments.
- We also need some scanners at the charging stations so that the charging process start after the device scans the user booking confirmation.

3.1.3. Software Interfaces

The application uses an external interface:

- MAP API for user and CPO to view the locations
- API for connecting banks' systems

3.1.4. Communication Interfaces

- Communication with DSOs:

CPMSs must be able to communicate with DSOs to get the current price of energy in order to calculate the amount that the EV-driver should pay for the obtained service.

- Communication with CPMSs:

eMSP must be able to communicate with CPMSs to get any required information such as the location of charging stations, the charging process, the amount that must be paid by the EV-driver and etc.

3.2. Functional Requirements

Requirement	Descriptions
R1	The eMSP must allow an unregistered user to register
R2	The eMSP must allow the user to contact the supporting team in case of having a problem with registering or logging the process
R3	The eMSP must send an error if the user inserts an incorrect form of needed blanks or if he/she is already registered
R4	The eMSP must send an email to the user's email address in order to verify him/her
R5	The eMSP must allow a logged-out user to log in again
R6	The eMSP must allow users to add a virtual wallet into his/her personal portal in order to enable them to easily pay via that in the next payments
R7	The eMSP shall allow users to pay either with their virtual wallet or with another card
R8	The eMSP shall allow a registered user to enter the specifications of their cars
R9	The eMSP must show the nearest charging stations in their near specific area
R10	The eMSP must show the related stations to the user according to the user's car specifications
R11	The eMSP must allow the user to view other driver's comments on one station and its rate
R12	The eMSP must show the types of each space of the station
R13	The eMSP must show the free spaces of each station to the user
R14	The eMSP must allow users to sort the shown charging stations by different categories and according to the user's car's specification
R15	The eMSP must show the 15 min countdown after the booking process
R16	The eMSP must send a receipt containing a QR code to the user after he/she books a charge space
R17	The eMSP must provide the direction to the specific charge station via the map
R18	When the user plugs in his/her car, The CPMS must scan the QR code provided by the user to verify him/her
R19	The CPMS must start the charging process after the verification
R20	The eMSP must alert the user that the charging process is started by sending a notification
R21	The user must be able to see the remaining time of his/her car to be fully charged processed by CPMS through eMSP

R22	The user must be able to see the current price of energy provided by CPMS
R23	The eMSP must allow the user to know that his/her car is being charged by the internal batteries of the station or from DSO
R24	The CPMS must end the charging process when the car is fully charged
R25	The eMSP must send a notification to the user that the charging process is finished
R26	The eMSP must allow users to plug out his/her car whenever they wanted
R27	The CPMS must calculate the price of charging and show it to the user via eMSP
R28	The eMSP must allow the user to use his/her tokens which were taken before, for payment
R29	The eMSP must allow the user to use different bank payment gateways to pay the bill
R30	The CPMS must calculate the amount of token acquired by the user after paying the bill according to the amount of the bill and transfer it to his/her personal account
R31	The eMSP must send a notification to the user that the payment process was successful
R32	The eMSP must allow the user to download or print the payment receipt
R33	After finishing payment, the eMSP must allow the user to leave a comment about the station and rate it
R34	The eMSP must allow the user to edit, remove his/her written comments and ratings
R35	The eMSP must save the user's comment in his/her personal portal
R36	The eMSP must store the history of previous stations used by the user and the bills
R37	The CPMS must calculate the value of available tokens and shows the facility which can be used by those tokens
R38	The CPMS must allow logged out CPO to LOGIN
R39	The CPMS must allow CPO to LOGIN
R40	The CPMS Allow CPO to monitor external status (type (fast, rapid, slow), available sockets, estimated amount of time until the first socket of that type is freed)
R41	The CPMS allows CPO to monitor the internal status(for each charging vehicle, amount of power absorbed and time left to the end of the charge)
R42	The CPMS allows CPO to monitor the status of the batteries
R43	The CPMS allows CPO to view the current price of the energy through DSOs

3.2.1. Use Case Diagrams

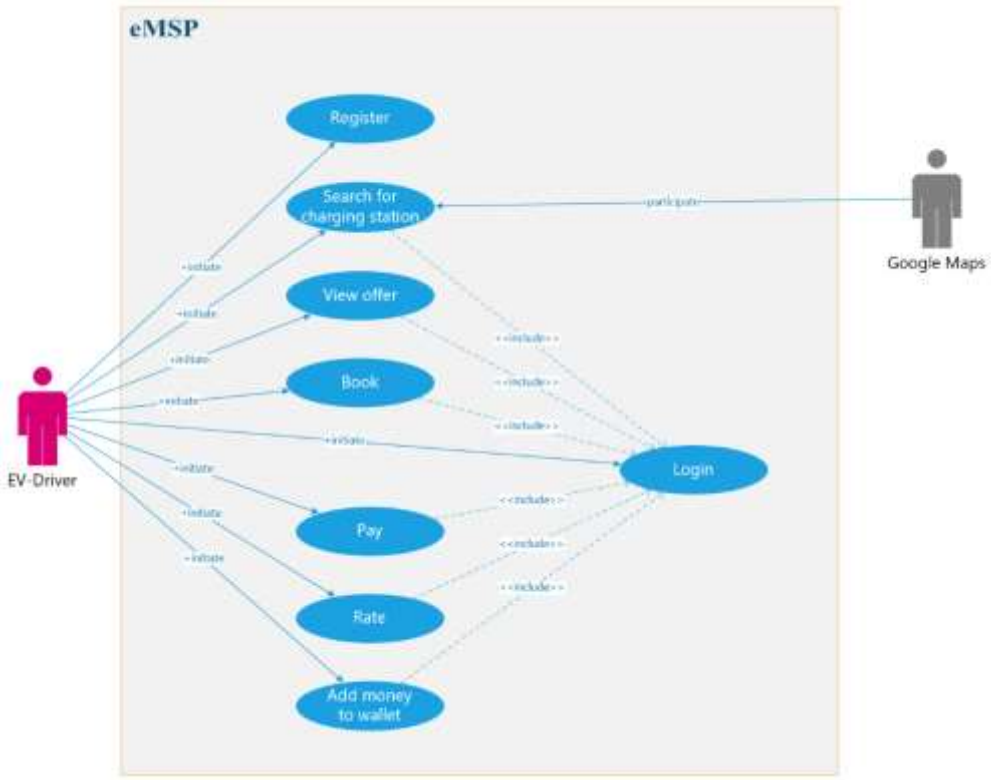

Figure 12- Use case Diagrams: EV-driver

Figure 13- Use case Diagrams: Unregistered EV-driver

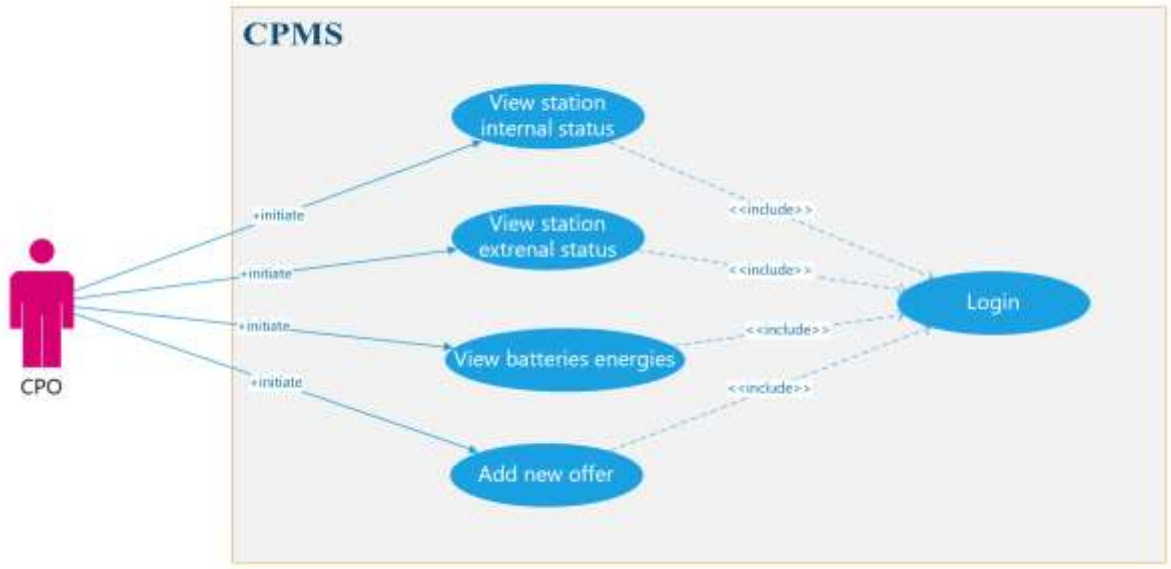

Figure 14- Use case Diagrams: CPMS

3.2.2. Use Cases

Name	**Register**
Actor	EV-driver
Entry Conditions	EV-driver opens the eMSP application.
Events Flow	1- EV-driver selects the register button to enter the registration page. 2- EV-driver fills the required blank fields. 3- EV-driver selects the "Register" button to submit his/her information. 4- eMSP sends an email to EV-driver for account confirmation. 5- EV-driver conforms to his/her email. 6- eMSP allows EV-driver to sign in to the eMSP system.
Exit Conditions	- EV-driver is registered in the eMSP system.
Exceptions	1- EV-driver does not fill all the needed blanks. In this case, eMSP warns him/her to fill all the blanks. 2- EV-driver does not confirm his/her email. 3- The inserted email address/phone number already exists in the application. In this case, eMSP warns him/her to log in instead of registering. 4- Inserted password is too weak. In this case, eMSP warns him/her to choose a stronger password.

Name	Login
Actor	EV-driver
Entry Conditions	1- EV-driver opens the eMSP application. 2- EV-driver has already registered in the eMSP application.
Events Flow	1- Ev-driver fills the email address/phone number and password. 2- EV-driver clicks on the "Login" button. 3- EV-driver enters his/her personal account.
Exit Conditions	- eMSP allows EV-driver to log in.
Exceptions	1 EV-driver enters the wrong email or password. In this case, eMSP warns him/her to enter the correct data.

Name	Search for charging stations
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- EV-driver is on the main view of his/her personal portal on eMSP
Events Flow	1- Ev-driver clicks on the search button to view the list of charging stations 2- The eMSP system shows the lists of charging stations 3- The user can filter the provided stations based on different factors 4- Ev-driver chooses one station 5- USER views the available types of sockets in the station and their prices 6- The user books the charge space
Exit Conditions	- eMSP shows the requested information by Ev-driver
Exceptions	1- losing internet connection 2- eMSP Fails to Connect to CPMS

Name	Book a charge in a charging station
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- EV-driver is on the main view of his/her personal portal on eMSP
Events Flow	1- Ev-driver clicks on the search button to view the list of charging stations 2- The eMSP system shows the lists of charging stations 3- User can filter the provided stations based on different factors 4- Ev-driver chooses one station 5- The user views the available types of sockets in the station and their prices 6- The user chooses one free(the occupied ones are not clickable) socket and click (Next) button 7- eMSP transfers the user to the (Book) page in which, he/she is able to set the time and vehicle identification number(VIN) 8- EV-driver clicks on (Book) button 9- eMSP set the boom appointment for the user
Exit Conditions	- The eMSP reserve a place for the user in the specific CPO
Exceptions	1- eMSP Fails to Connect to CPMS 2- eMSP fails to save the booking process 3- The user loses his/her internet connection while the booking process 4- The user chooses the wrong type of socket

Name	Pay for obtained service
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- The charging process is finished
Events Flow	1- The eMSP sends a notification to the user to inform him/her that the charging process is finished 2- eMSP shows the price which is calculated by CPMS 3- EV-driver clicks on the pay button 4- EV-driver can choose between paying via his/her virtual wallet or another wallet 5- If the user has an offer code, he/she can insert it into offer code button 6- CPMS calculates the new price according to inserted offer code 7- EV-driver pay the bill 8- eMSP completes the payment process 9- eMSP sends the payment receipt to the user
Exit Conditions	- eMSP does the payment process
Exceptions	1- The user loses internet connection after clicking pay button 2- EV-driver closes the application

Name	**Rate a charging station**
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- The payment process is finished
Events Flow	1- The eMSP sends a message to the user to invite him/her to rate the station which was used by them 2- EV-driver clicks on the rating button 3- EV-driver writes a comment about the service he/she has gotten and rates the station 4- EV-driver clicks on the submit button 5- eMSP sends a notification to the user that his/her comment is submitted
Exit Conditions	- The rating process is finished
Exceptions	1- The user closes the application 2- The eMSP is unable to submit the rating due to weak internet connection or losing connection 3- The user leaves the blanks and try to submit, in this case, the eMSP warns him/her to fill all blanks

Name	**Add money in wallet**
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP
Events Flow	1- The user clicks on the (add money) button. 2- The user writes the amount money he/she wants to add to her/his wallet 3- The user chooses the cart that he/she wants to use to get money from 4- The user clicks on the (next) button 5- The eMSP transfers the user to another page to enter the information of the selected cart 6- The user enters the cart's information 7- The user clicks on the (pay) button 8- The eMSP makes an API to CPMS in order to complete the transfer process 9- The eMSP gives the payment receipt to the user
Exit Conditions	- CPMS does the adding money process
Exceptions	1- The user has another bank account that is not available in the list of banks 2- The user does not enter the money amount or does not select the bank, in this case, the eMSP warns him/her to fill the needed blanks

Name	**Login**
Actor	CPO
Entry Conditions	1- CPO manager opens the CPMS application. 2- CPO manager has already registered in the CPMS application.
Events Flow	1- CPO manager fills the email address/phone number and password. 2- CPO manager clicks on the "Login" button. 3- CPO manager enters his/her personal account.
Exit Conditions	- CPMS allows the CPO manager to log in.
Exceptions	1- CPO manager enters the wrong email or password. In this case, CPMS warns him/her to enter the correct data.

Name	**View charging stations' location**
Actor	CPO
Entry Conditions	1- CPO manager opens the CPMS application. 2- CPO manager has already logged in to the CPMS application.
Events Flow	1- The CPO manager clicks on one of the stations 2- CPMS transfers the CPO manager to the charging station details where he/she can monitor the map of the city 3- CPO manager clicks on one station on the map 4- The details of the specific information of that station
Exit Conditions	- The CPMS shows the requested information by the CPO manager
Exceptions	1- The CPMS loses connection while preparing the requested information

Name	View Internal Status of a charging station
Actor	CPO
Entry Conditions	1- CPO manager has already logged in to the CPMS application.
Events Flow	1- CPO manager clicks on the (charging stations status) button 2- On the next page, clicks on one station 3- On the (Charging station status) page, choose one station 4- CPO manager clicks on the internal status 5- In the (internal status) page, the CPO manager clicks on the batteries energies button to view the status of energies available in the batteries 6- In the (internal status) page, he/she is able to see the number of vehicles that are being charged at that moment 7- CPO manager clicks on one of the car's status 8- The CPMS transfer the CPO manager to the page in which he/she can visit the amount of power absorbed and time left to the end of the charge through its CPMS
Exit Conditions	- The CPMS shows the requested information to the CPO manager
Exceptions	1- The CPMS fails to make an API to CPMS in order to get the required information

Name	View External Status of a charging station
Actor	CPO
Entry Conditions	- CPO manager has already logged in the CPMS application
Events Flow	1- CPO manager clicks on the (external status) button 2- On the (external page), he/she can view the number of charging sockets available and their types (slow, fast, rapid) 3- CPO manager clicks on one of the sockets icons 4- CPMS transfers the CPO manager to the specific socket's page 5- On the specific page of the socket, the CPO manager is able to view the cost of that socket and in case of being full, the estimated time to be freed which is calculated by CPMS
Exit Conditions	- The CPMS shows the requested information to the CPO manager
Exceptions	1- The CPMS fails to make an API to CPMS in order to get the required information

Name	**CPMS add offer**
Actor	CPMS
Entry Conditions	- CPO manager has already logged in the CPMS application
Events Flow	1- CPO manager clicks on the (add offer) button 2- CPMS transfers him/her to the next page in which they can add any offer related to their stations 3- CPMS confirms the offer 4- CPMS sends the offer to the eMSP
Exit Conditions	- CPMS confirms the offer and sends it to the eMSP
Exceptions	1- The CPO manager suggests an unrelated offer 2- CPMS Fails to confirm due to losing an internet connection

Name	**Edit personal information**
Actor	EV-driver
Entry Conditions	- The user has already logged in to the eMSP application
Events Flow	1- The user clicks on the (edit personal information) buttons 2- On the next page, the user changes the personal information 3- The user clicks on the (apply) button to save the changes 4- The eMSP saves the changes and alert the user with a notification
Exit Conditions	- The eMSP applies the changes that have been done by the user
Exceptions	1- The user enters wrong information in one block, in this case, the eMSP sends an error to him/her to enter the correct form of needed information 2- The eMSP Fails to confirm the changes due to losing the internet connection

3.2.3. Sequence Diagrams

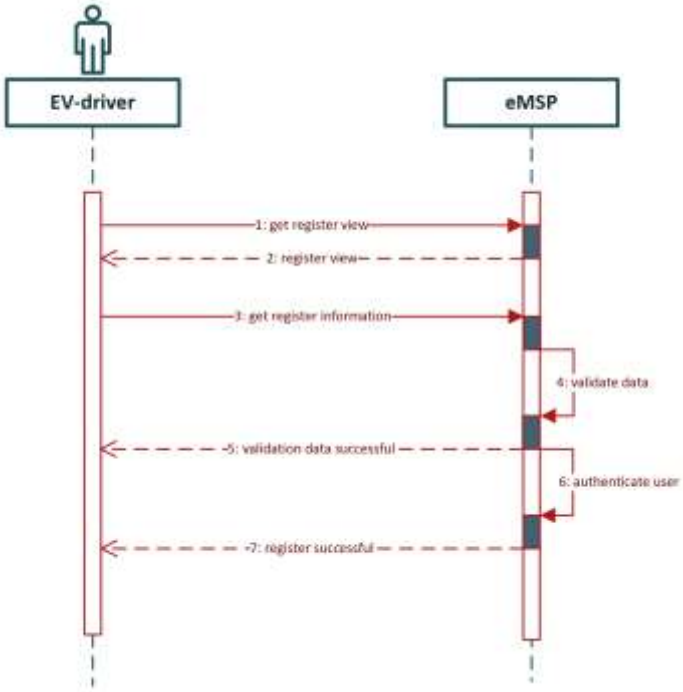

Figure 15-Sequence Diagrams: User registration

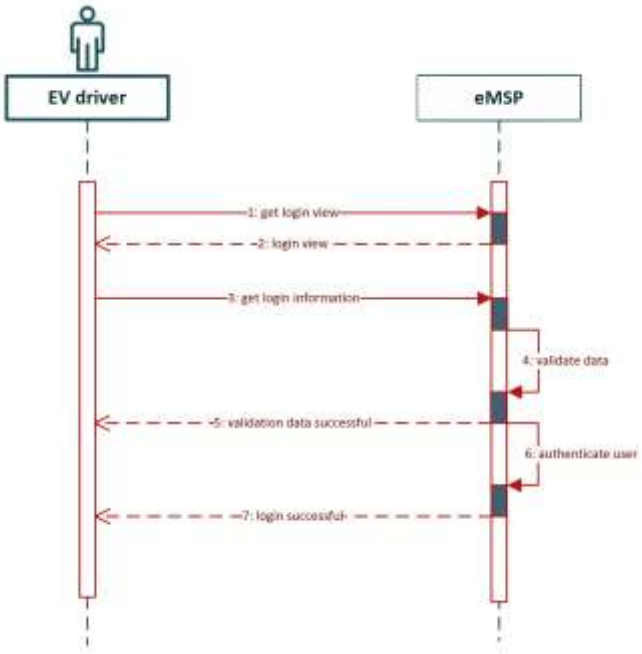

Figure 16-Sequence Diagrams: User login

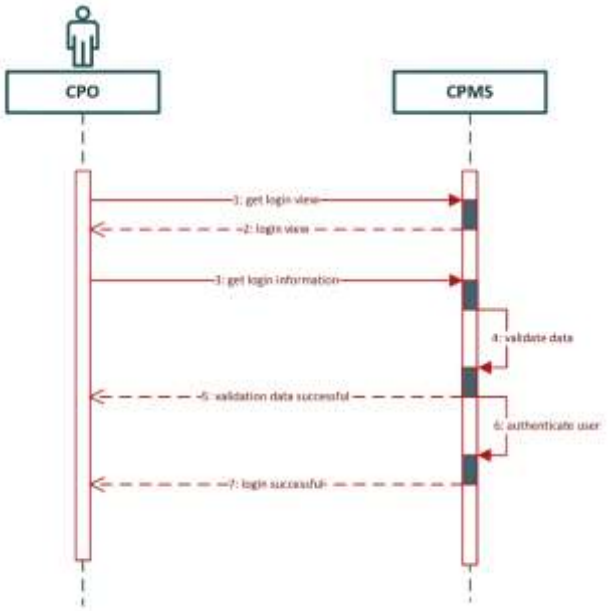

Figure 17-Sequence Diagrams: CPO login

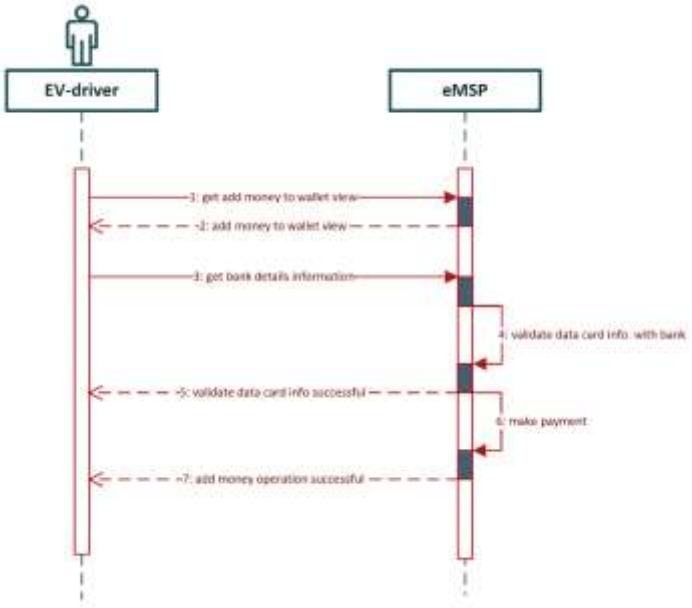

Figure 18-Sequence Diagrams: Add money to wallet

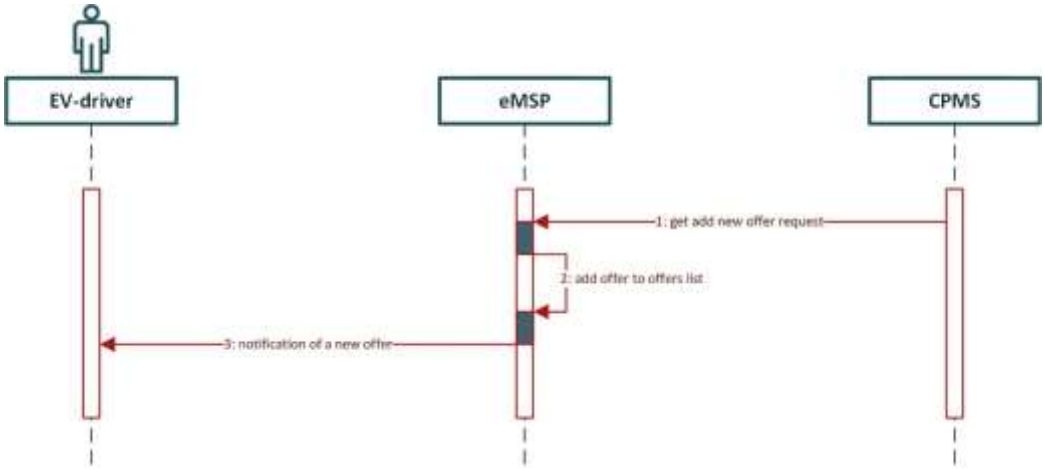

Figure 19-Sequence Diagrams: CPO add offer

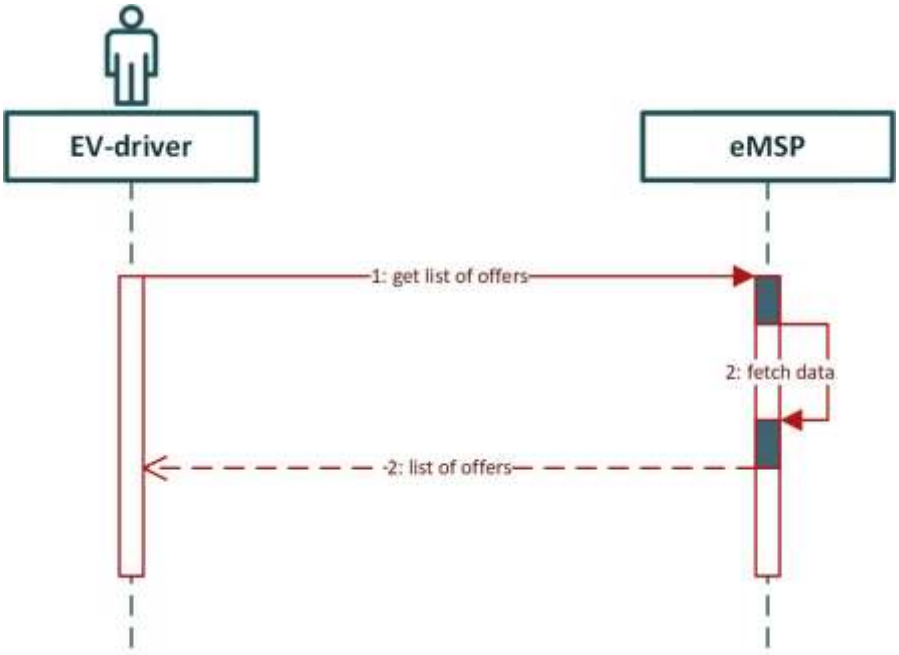

Figure 20-Sequence Diagrams: User view offers

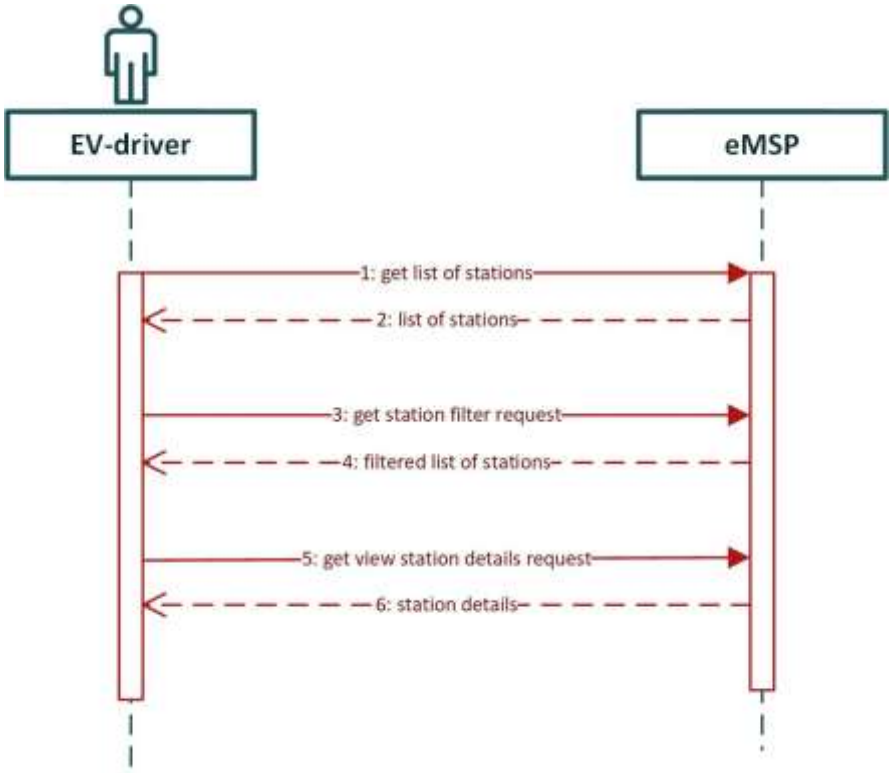

Figure 21-Sequence Diagrams: Search

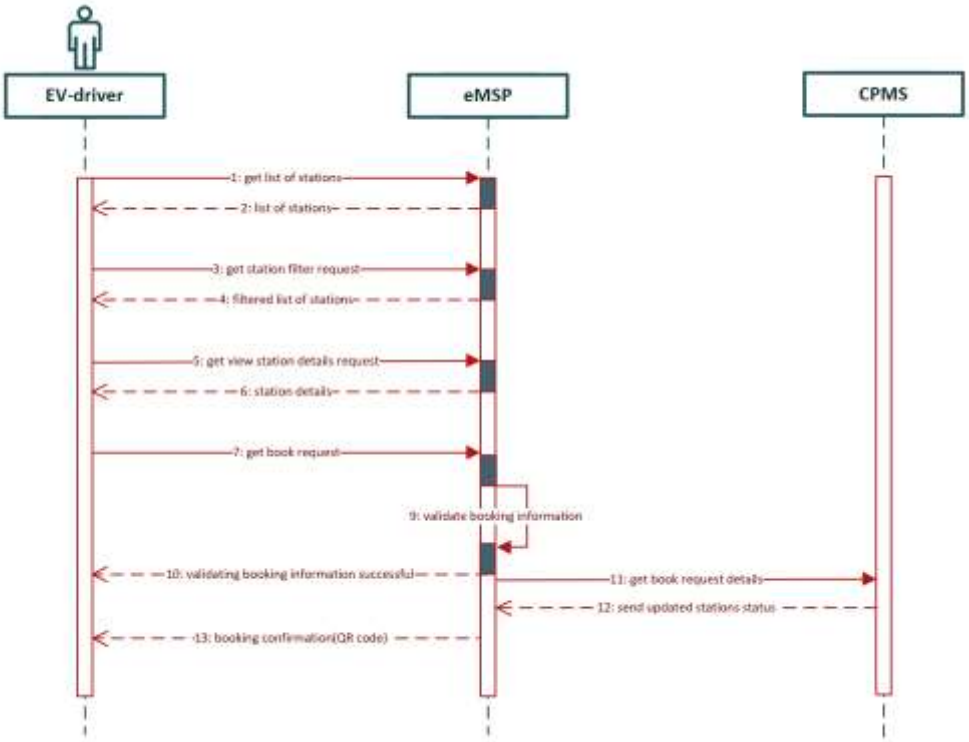

Figure 22-Sequence Diagrams: Book

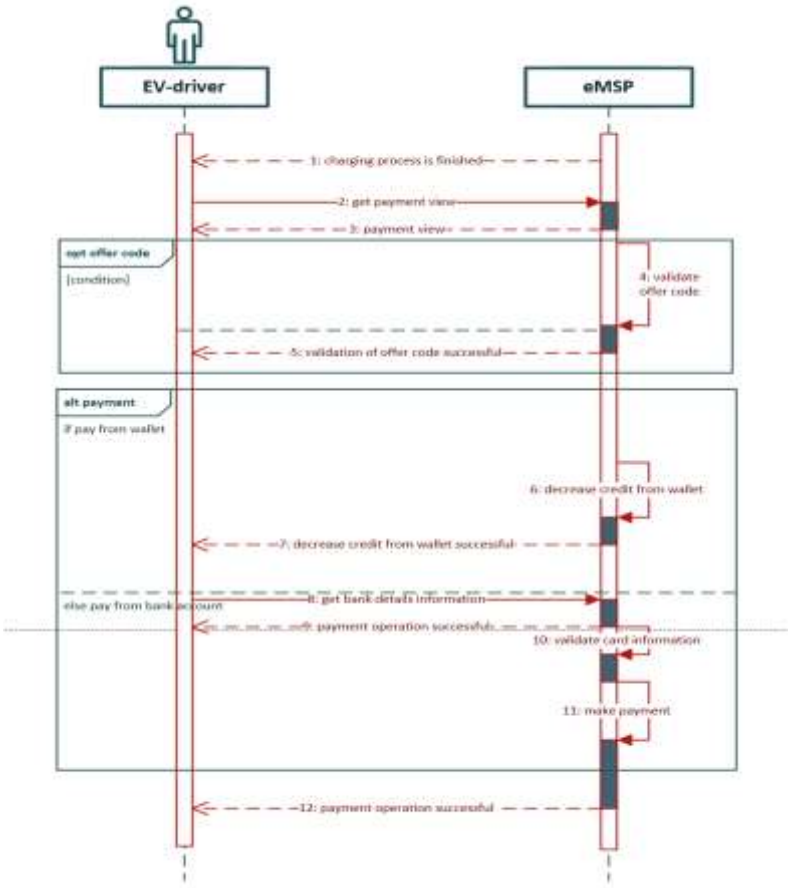

Figure 23-Sequence Diagrams: Pay

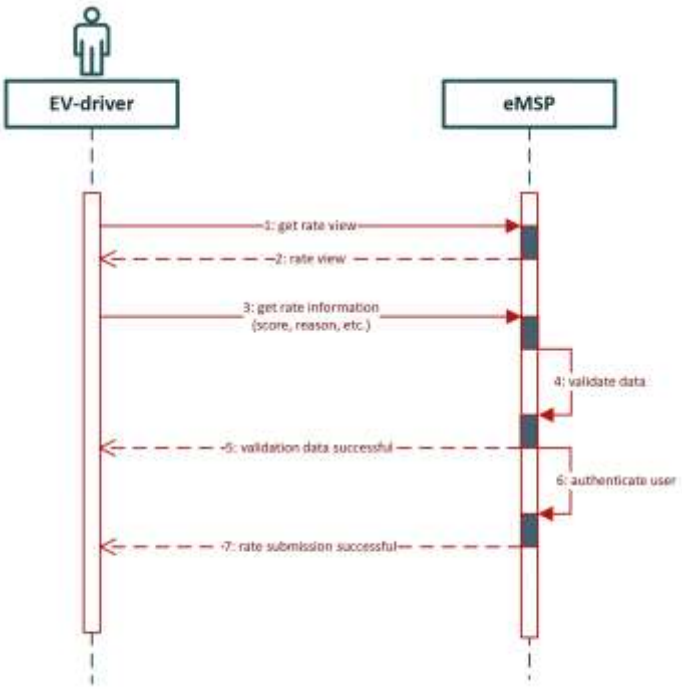

Figure 24-Sequence Diagrams: Rate

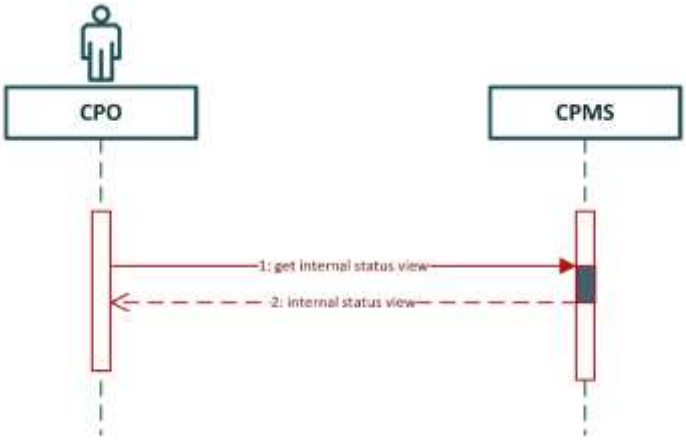

Figure 25-Sequence Diagrams: CPO view internal status

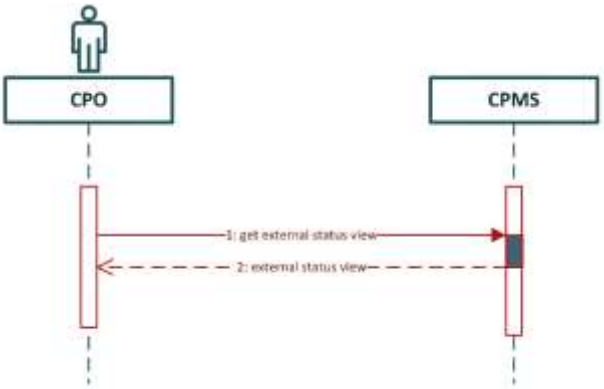

Figure 26-Sequence Diagrams: CPO view external status

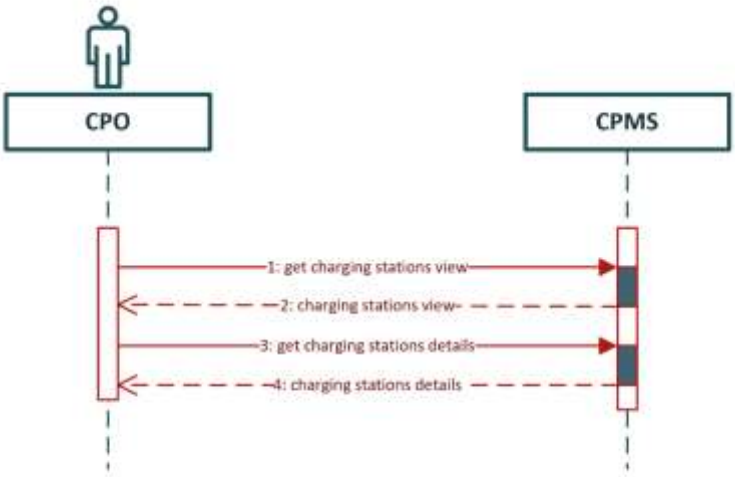

Figure 27-Sequence Diagrams: CPO view charging stations locations

3.2.4. Mapping on Requirements

G1	**eMSP allows EV-drivers to search for charging stations nearby, their cost and any special offer.**
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct
D6	The Current price of energy provided by DSO is correct and updated instantly
D9	The comments and rates which are given by every EV-driver for each station are correct
R9	The eMSP must show the nearest charging stations in their near specific area
R10	The eMSP must show the related stations to the user according to the user's car specifications
R11	The eMSP must allow the user to view other driver's comments on one station and its rate
R12	The eMSP must show the types of each space of the station
R13	The eMSP must show the free spaces of each station to the user
R43	The CPMS allows CPO to view the current price of the energy through DSOs

G2	**eMSP allows EV-drivers to book a charge in a specific charging station 15 minutes before connection.**
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct
D6	The Current price of energy provided by DSO is correct and updated instantly
D8	The user is able to attend the station in 15 min after booking a charging space
R9	The eMSP must show the nearest charging stations in their near specific area
R10	The eMSP must show the related stations to the user according to the user's car specifications
R11	The eMSP must allow the user to view other driver's comments on one station and its rate
R12	The eMSP must show the types of each space of the station
R13	The eMSP must show the free spaces of each station to the user
R15	The eMSP must show the 15 min countdown after the booking process
R16	The eMSP must send a receipt containing a QR code to the user after he/she books a charge space

R17	The eMSP must provide the direction to the specific charge station via the map
R40	The CPMS allows CPO to monitor external status(type(fast, rapid, slow), available sockets, the estimated amount of time until the first socket of that type is freed)

G3	**eMSP allows EV-drivers to sort the available charging point stations based on the selected filter.**
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct
R14	The eMSP must allow users to sort the shown charging stations by different categories and according to the user's car's specification

G4	**eMSP allows EV-drivers to rate the charging station they received the service from.**
D9	The comments and rates which are given by every EV-driver for each station are correct
R33	After finishing payment, the eMSP must allow the user to leave a comment about the station and rate it
R34	The eMSP must allow the user to edit, and remove his/her written comments and ratings
R35	The eMSP must save the user's comment in his/her personal portal
R36	The eMSP must store the history of previous stations used by the user and the bills

G5	**eMSP allows EV-drivers to add a credit card to the wallet inside their account for paying later.**
R6	The eMSP must allow users to add a virtual wallet into his/her personal portal in order to enable them to easily pay via that in the next payments
R7	The eMSP shall allow users to pay either with their virtual wallet or with another card

G6	**eMSP allows EV drivers to pay for the obtained service either using an internal wallet or by credit card.**
D8	The Current price of energy provided by DSO is correct and updated instantly
D7	EV driver pays the bill first and then plugs out his/her car
R25	The eMSP must send a notification to the user that the charging process is finished

R26	The eMSP must allow users to plug out his/her car whenever they wanted
R27	The CPMS must calculate the price of charging and show it to the user via eMSP
R28	The eMSP must allow the user to use his/her tokens which were taken before, for payment
R29	The eMSP must allow the user to use different bank payment gateways to pay the bill
R31	The eMSP must send a notification to the user that the payment process was successful
R32	The eMSP must allow the user to download or print the payment receipt
R41	The CPMS allows CPO to monitor the internal status(for each charging vehicle, amount of power absorbed and time left to the end of the charge)
R43	The CPMS allows CPOs to view the current price of the energy through DSOs

G7	**CPMS allows CPOs to know about the amount of energy used by each vehicle.**
D1	The voltage of charging is compatible with the user's car model.
D2	EV driver inserts the car specifications correctly
D3	The charging stations always cam have batteries and/or are supported by DSOs
D5	Energy supply from Energy Company (DSO) is continuous and without problems
R19	The CPMS must start the charging process after the verification
R23	The eMSP must allow the user to know that his/her car is being charged by the internal batteries of the station or from DSO
R24	The CPMS must end the charging process when the car is fully charged
R39	The CPMS must allow CPO to LOGIN
R41	The CPMS allows CPO to monitor the internal status(for each charging vehicle, amount of power absorbed and time left to the end of the charge)

G8	**CPMS allows CPOs to know the "external" status of charging stations, e.g. location of charging available sockets, type, cost**
R10	The eMSP must show the related stations to the user according to the user's car specifications
R12	The eMSP must show the types of each space of the station
R13	The eMSP must show the free spaces of each station to the user
R24	The CPMS must end the charging process when the car is fully charged
R39	The CPMS must allow CPO to LOGIN
R40	The CPMS allows CPO to monitor external status(type(fast, rapid, slow),available sockets, the estimated amount of time until the first socket of that type is freed)

G9	**CPMS allows CPOs to monitor the charging process to infer when the battery is full**
D1	The voltage of charging is compatible with the user's car model.
D3	The charging stations always cam have batteries and/or are supported by DSOs
D5	Energy supply from Energy Company (DSO) is continuous and without problems
R24	The CPMS must end the charging process when the car is fully charged
R25	The eMSP must send a notification to the user that the charging process is finished

G10	**CPMS allows CPOs to know the "internal" status of a charging station**
D1	The voltage of charging is compatible with the user's car model.
D3	The charging stations always cam have batteries and/or are supported by DSOs
D5	Energy supply from Energy Company(DSO) is continuous and without problems
D10	Each EV-driver uses each station's facilities correctly and after finishing the charging process, the EV-driver put the hose back in its space correctly
R21	The user must be able to see the remaining time of his/her car to be fully charged processed by CPMS through eMSP
R23	The eMSP must allow the user to know that his/her car is being charged by the internal batteries of the station or from DSO
R41	The CPMS allows CPO to monitor the internal status(for each charging vehicle, amount of power absorbed and time left to the end of the charge)
R42	The CPMS allows CPO to monitor the status of the batteries

G11	CPMS allows CPOs to know the current price of energy acquired from the DSOs
D3	The charging stations always cam have batteries and/or are supported by DSOs
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct
D5	Energy supply from Energy Company (DSO) is continuous and without problems
D6	The Current price of energy provided by DSO is correct and updated instantly
R27	The CPMS must calculate the price of charging and show it to the user via eMSP
R43	The CPMS allows CPOs to view the current price of the energy through DSOs

3.3. Performance Requirements

- The eMSP system must be able to serve a great number of users simultaneously.
- The eMSP system must guarantee correct responses.
- The eMSP system must be able to send a response to a query less than 5 seconds since it has been received.
- The eMSP system must be available 99% of the time.
- The CPMS system must be able to serve a great number of users simultaneously.
- The CPMS system must guarantee correct responses.
- The CPMS system must be able to send a response to a query less than 5 seconds since it has been received.
- The CPMS system must be available 99% of the time.

3.4. Design Constraints

3.4.1. Standard Constraints

- The eMSP system requires EV-drivers' permission to retrieve their position.
- Both systems must maintain the data retrieved from the EV-drivers with respect to privacy laws.

3.4.2. Hardware Constraints

- The web browser or the smartphone that the user is using must have the ability to connect to the internet and use GPS services.

3.4.3. Any other Constraints

- The estimations that the CPMS system makes about the remaining time of full charge must be accurate 99% of the time.

3.5. Software System Attributes

3.5.1. Reliability

Both eMSP and CPMS systems must be able to run continuously without any interrupts. The reliability of the systems depends on the services of the systems and should be up 99% of the time. This means the MTTR or downtime should be 3.65 days per year. In order to guarantee this time of downtime, the systems must have an appropriate infrastructure with a full backup system located in a different office that replicates the core services for covering the general failure of the main systems.

3.5.2. Availability

The two systems do not relate to the emergency; thus, we don't need high availability. Moreover, the service is not fully automated because has to rely on the policymakers that follow office hours.

3.5.3. Security

For security issues and to avoid any problems, we need to encrypt the stored data and passwords of each actor of the two systems, EV-drivers and CPOs. Also, if CPMS wants to give some suggestions to EV-drivers based on the information and data it receives through APIs from the navigation system of the car, this information must be encrypted and then stored in the DBMS. Therefore, a layer of protection must be implemented to keep users' personal data safe.

3.5.4. Maintainability

The software must be written in Python and codes must be written with good standards and in OO (object-oriented) structure so that in case of any occurrence of problem or bug, it can be fixed easily and without requiring to change a big portion of the code. Moreover, any bug in one component must have not interrupted with the functionality of other components.

3.5.5. Portability

The software must be designed simply and implemented on different platforms. The software run in different platforms must support Android and iOS operating systems for mobile devices, as well as a Web application for use simply in other platforms.

4. Formal Analysis Using Alloy

4.1. Code

```
// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
// ******************* Signatures ************************
// *****************************************************
// ***************** Things To Know ********************
// 1- Int range in Alloy is from -8 to 7
// *****************************************************
// ********************** Users ************************
// *****************************************************

sig Id {}
sig Email {}
sig Password {}
sig PhoneNumber {}
sig FirstName {}
sig LastName {}
sig Date {}
sig Location {
    latitude: one Int,
    longitude: one Int
}{
    latitude >= -5 and latitude =< 5
    longitude >= -5 and longitude =< 5
}
sig Duration {}
sig BatteryCapacity {}
sig Name {}
sig Address {}
sig VIN {}      // VIN ~ vehicle identification number
sig SocketType {}
sig Text {}

sig EVDriver {
    email: one Email,
    password: one Password,
    firstName: one FirstName,
    lastName: one LastName,
    phoneNumber: one PhoneNumber,

    wallet: one Wallet,
    vehicles: set Vehicle,
    offers: set Offer,
    payments: set Payment   ,
    ratings: set Rating,
    bookings: set Booking
}
```

```
sig Vehicle {
    vin: one VIN,
    Name: one Name
}

sig CPO {
    email: one Email,
    password: one Password,
    firstName: one FirstName,
    lastName: one LastName,
    chargingStations: set ChargingStation
}

sig DSO {
    companyName: one Name,
    currentPrice: one Int,
    Address: one Address,
    chargingStations: set ChargingStation
} {
    currentPrice > 0
}

// *********************************************************
// ***************** Other Entities ********************
// *********************************************************

sig Offer {
    offerName: one Name,
    discountPercent: one Int,
    fromDate: one Date,
    toDate: one Date
} {
    discountPercent > 0
}

sig Wallet {
    createdDate: one Date,
    credit: one Int
} {
    credit > 0
}

sig Booking {
    bookingDate: one Date,
    socket: one Socket
}
```

```
sig Payment {
    paymentDate: one Date,
    amount: one Int,
    offer: one Offer
} {
    amount > 0
}

sig ChargingSession {
    booking: one Booking,
    payment: one Payment,
    chargingDuration: one Duration
}

sig ChargingStation {
    location: one Location,
    sockets: set Socket,
    batteries: set Battery
}

sig Socket {
    powerSupplyAmount: one Int,
    type: one SocketType
} {
    powerSupplyAmount > 0
}

sig Battery {
    capacity: one BatteryCapacity,
    currentPrice: one Int
} {
    currentPrice > 0
}

sig Rating {
    score: one Int,
    reason: one Text,
    chargingStation: one ChargingStation
} { // define score range from 0 to 5
    score >= 0
    score <= 5
}

// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
// ********************** Facts **************************
// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
// Links between User, EVDriver, CPO and their arttributes
```

```
// each email must be linked to only one EVDriver
fact EmailLinkedEVDriver{
    all e:Email |  one evd:EVDriver| e in evd.email
}

// each password must be linked to at least one EVDriver
fact PasswordLinkedEVDriver {
    all p:Password|  some evd:EVDriver | p in evd.password
}

// each first name must be linked to at least one EVDriver
fact FirstNameLinkedEVDriver {
    all fn:FirstName|  some evd:EVDriver | fn in evd.firstName
}

// each last name must be linked to at least one EVDriver
fact LastNameLinkedEVDriver {
    all ln:LastName|  some evd:EVDriver| ln in evd.lastName
}


// each phone number must be linked to only one EVDriver
fact PhoneNumberLinkedEVDriver {
    all pn:PhoneNumber|  one evd:EVDriver | pn in evd.phoneNumber
}

// each wallet must be linked to only one EVDriver
fact WalletLinkedEVDriver {
    all w: Wallet | one evd:EVDriver | w in evd.wallet
}

// each vehicle must have only one EVDriver as owner
fact VehicleLinkedEVDriver {
    all vehicle: Vehicle | one evd:EVDriver | vehicle in evd.vehicles
}

// each offer must be assciated to at least one EVDriver
fact OfferLinkedEVDriver {
    all o: Offer | some evd:EVDriver | o in evd.offers
}

// each must be linked to only one EVDriver
fact RatingLinkedEVDriver {
    all r: Rating | one evd: EVDriver | r in evd.ratings
}
```

```
// each rating must be linked to only one charging station
fact ChargingStationLinkedRating {
    all r: Rating | one cs: ChargingStation | cs in r.chargingStation
}

// each booking must be linked to only one EVDriver
fact BookingLinkedEVDriver {
    all b: Booking | one evd: EVDriver | b in evd.bookings
}

// each payment must be linked to only one EVDriver
fact PaymentLinkedEVDriver {
    all p: Payment | one evd: EVDriver | p in evd.payments
}

// each vehicle must be linked to one VIN
fact VINLinkedVehicle {
    all v: VIN | one veh: Vehicle | v in veh.vin
}

// each email must be linked to only one CPO
fact EmailLinkedCPO{
    all e:Email |  one cpo:CPO | e in cpo.email
}

// each password must be linked to at least one CPO
fact PasswordLinkedCPO {
    all p:Password |  some cpo:CPO | p in cpo.password
}

// each first name must be linked to at least one CPO
fact FirstNameLinkedCPO {
    all fn:FirstName |  some cpo:CPO | fn in cpo.firstName
}

// each last name must be linked to at least one CPO
fact LastNameLinkedCPO {
    all ln:LastName |  some cpo:CPO | ln in cpo.lastName
}

// each charging station must be linked to at least one CPO
fact ChargingStationLinkedCPO {
    all cs:ChargingStation |  some cpo:CPO | cs in cpo.chargingStations
}
```

```
// each charging station must be linked to at least one DSO
fact ChargingStationLinkedDSO {
    all cs:ChargingStation |  some dso:DSO| cs in dso.chargingStations
}

// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.

// Socket & Charging Station & DSO & CPO links
// each location must refer to a diffrent place (latitude & longitude must be
diffrent)
fact {
    all disj loc1, loc2: Location |
    loc1.latitude = loc2.latitude => loc1.longitude != loc2.longitude
}

// each location must be linked to only one charging station
fact LocationLinkedChargingStation {
    all l: Location | one cs: ChargingStation | l in cs.location
}

// each socket must be linked to only one charging station
fact SocketLinkedChargingStation {
    all socket: Socket | one cs: ChargingStation | socket in cs.sockets
}

// each charging station must be linked to at least one socket
fact SocketLinkedChargingStation {
    all cs: ChargingStation| some s: Socket| s in cs.sockets
}

// each battery must be linked to only one socket
fact BatteryLinkedChargingStation {
    all battery: Battery| one cs: ChargingStation | battery in cs.batteries
}

// each charging station must be linked to only one CPO
fact ChargingStationLinkedCPO {
    all cs: ChargingStation | one cpo: CPO | cs in cpo.chargingStations
}

// each DSO must be linked to at least one charging station
fact ChargingStationLinkedDSO {
    all cs: ChargingStation | some dso: DSO | cs in dso.chargingStations
}
```

```
// Booking & Charging Process & Payment Links
// each booking must be linked to only one socket
fact BookingLinkedSocket {
    all b: Booking | one s: Socket | s in b.socket
}


// each booking must be linked to only one Date
fact BookingLinkedDate {
    all b: Booking | one d: Date | d in b.bookingDate
}

// each charging session must be linked to only one booking
// Charging session is created when charging process is started
fact BookingLinkedSession {
    all b: Booking | one session: ChargingSession | b in session.booking
}

// each charging session must be linked to only one payment
fact SessionLinkedPayment {
    all p: Payment | one session: ChargingSession | p in session.payment
}

// eahc payment must be linked to only one offer
fact paymentLinkedOffer {
    all p: Payment | one o: Offer | o in p.offer
}
// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.
// ********************* Runing ************************
// ~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.~.

//run {}
// Run the model
pred show {
    #EVDriver > 2
    #Vehicle > 2
    #CPO > 2
    #DSO > 1
    #ChargingStation > 2
    #Socket > 2
    #Offer = 2
}
run show for 10
```

4.2. Results

4.3. Generated

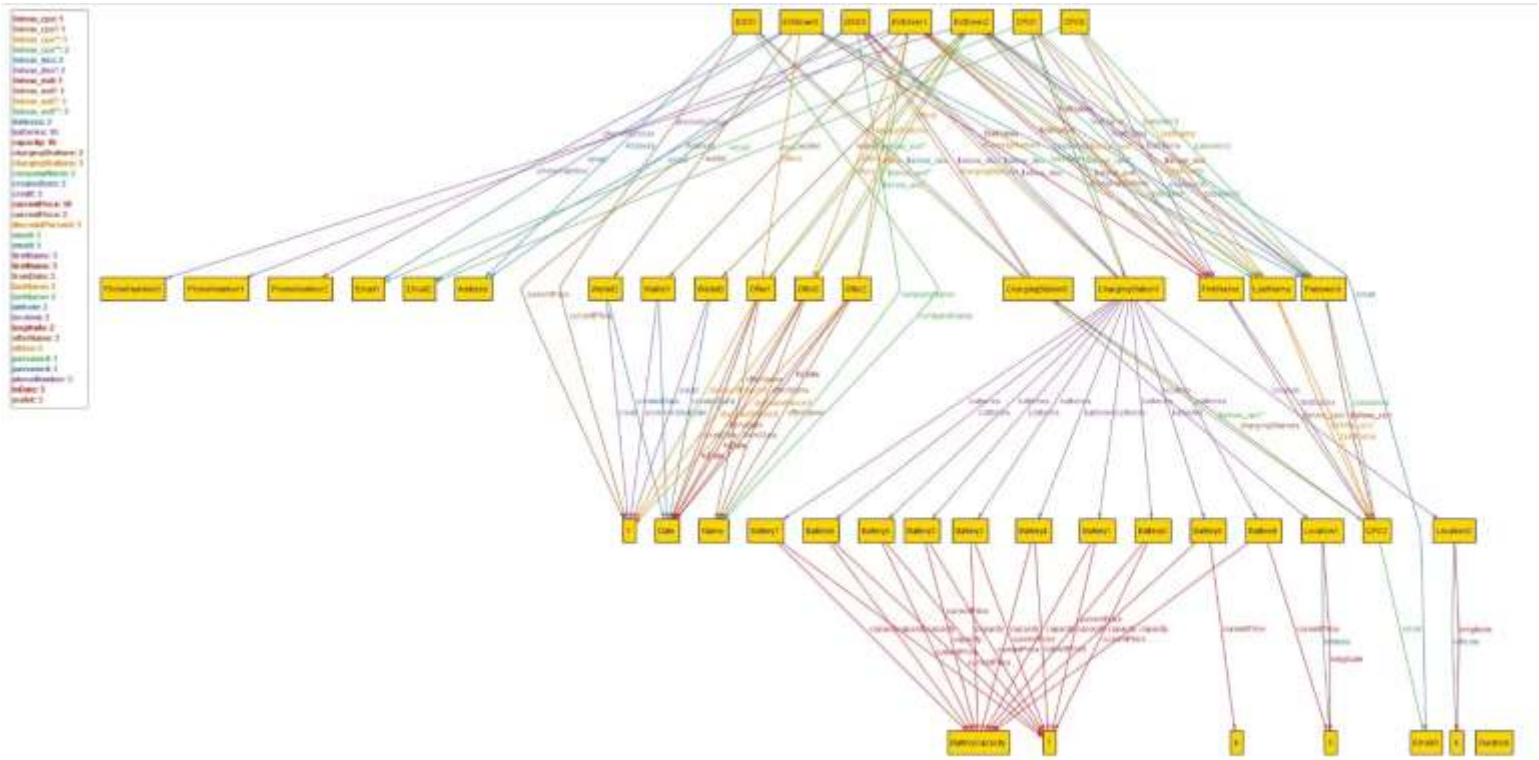

Figure 13-Alloy result_1

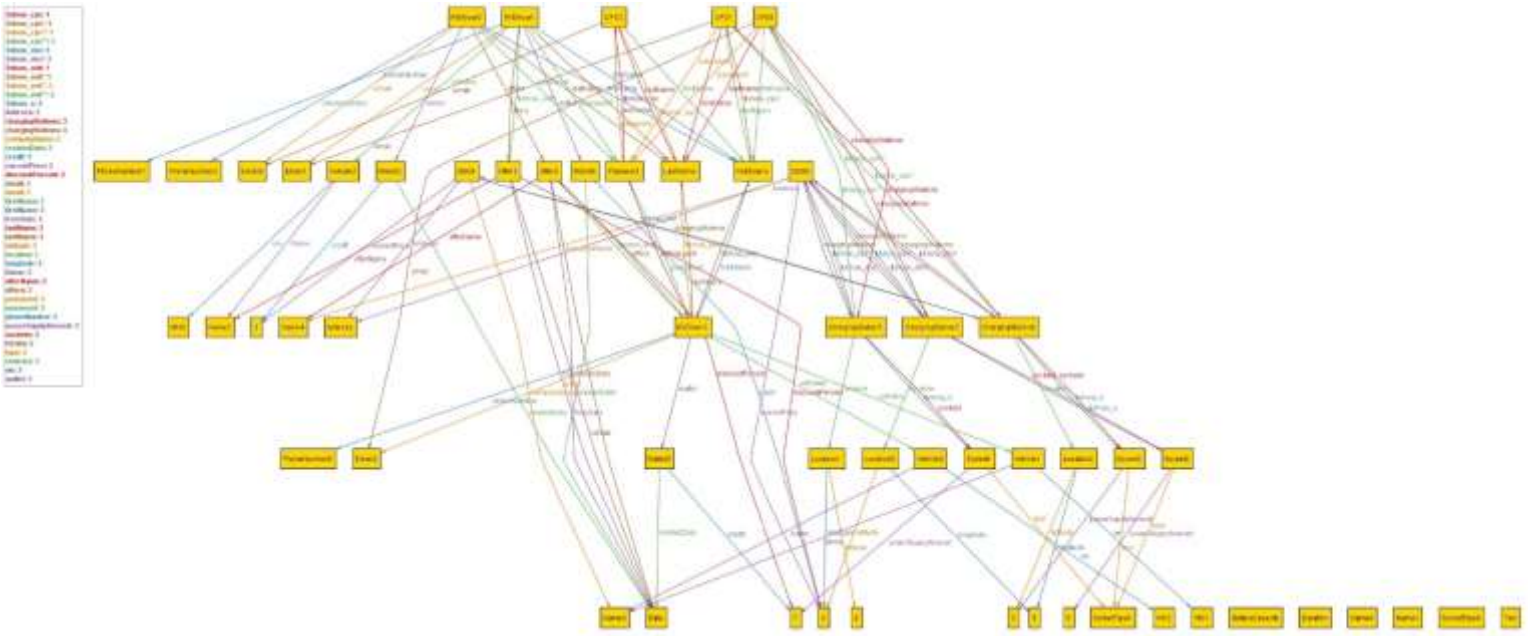

Figure 13-Alloy result_2

5. Effort Spent

- Student 1:

Topics	Hours
Introduction	8 h
Overall description	9 h
Specific Requirement	10 h
Formal analysis	7.5 h
Reasoning	9.5 h
Total	***44 h***

- Student 2:

Topics	Hours
Introduction	8 h
Overall description	8.5 h
Specific Requirement	12 h
Formal analysis	5.5 h
Reasoning	9.5 h
Total	***43.5 h***

6. References

- Specification Document: “Assignment RDD AY 2022-2023_v2.pdf”
- Course slides
- “OCPI-2.2.1.pdf”