A.Y. 2022-2023 Software Engineering 2 - RASD Project

Document by Felipe Bagni, Felipe Azank and Sabrina Wolff

December 17, 2022

Table of Contents

1 Introduction

1.1 Purpose

The global climate crisis displays one of the biggest threats to humanity and the planet. The loss of biodiversity, the impact on society and extreme weather conditions are only a few potential consequences if we don't act quickly and reduce the emissions of greenhouse gases in every sector possible. One important sector in this fight is the transportation sector, especially cars that emit 27% of total greenhouse gas emissions in the U.S, making it the largest contributor.[1] To deal with this problem more and more people should be encouraged to buy electric cars instead of traditional ones. But one problem with electric cars is that you can't just drive to a gas station and refill in a few minutes. You need to know where charging stations are and moreover if they are available. Depending on the car's battery the charging process can take between 4-10 hours[2] and therefore needs to be scheduled carefully to fit in the daily schedule of the people using electric cars.

To make electronic mobility more accessible and to reduce the carbon footprint of the transportation sector, eMall tackles this problem by offering a software that allows you to know about the nearest charging station, its prices and if it is free for you to use. It simplifies the charging process of electric cars and makes them therefore more suitable for everyday use.

This document will present the requirements and goals to be achieved to provide a guideline for the further development.

1.1.1 Goals

We start with presenting the goals of the development. For this we will first explain shortly the abbreviations we used. eMSP is the software owners of electric cars can use for all things around the charging process of the car. A CPO is the owner of at least one charging station and they each use one CPMS software to manage their charging stations and energy supply. DSOs are actors that offer energy supply to the CPOs.

Goal	Description
G1	Car owners use eMSP to look for the best (depending on charging station's location, price, special offers, type, availability) charging station nearby
G2	Car owners use eMSP for the charging process of their car (book a charge in a specific charging station for a certain timeframe, start charging, pay for service)
G3	eMSP will proactively suggest the car owner to charge depending on his car's battery, current special charging offers and the car owner's schedule.
G4	CPOs use it's respective CPMS to handle energy needs of the connected charging stations (acquisition of energy from DSO or of connected batteries,

[1] https://www.epa.gov/transportation-air-pollution-and-climate-change/carbon-pollution-transportation
[2] https://monta.com/uk/blog/how-long-does-it-take-to-charge-an-electric-car/

	distribution of energy to the connected stations)
G5	CPMS gives CPO overview and internal/external status information about his charging stations
G6	CPOs use CPMS to handle the pricing (set price or set special offer) for a charging
G7	CPMS performs and monitors a charging process at a charging station for a connected car

1.2 Scope

eMSP is a subsystem of eMall that allows owners of electric cars to easily find and book a charging station. The CPMS is another subsystem of eMall that allows charging point operators to handle their energy acquisition and distribution and based on this the cost of their service. Moreover the CPMS performs the charging process itself.

As a group of three students we have to assume that the eMSP subsystem interacts with CPMSs of multiple CPOs and the CPMS is connected to multiple DSOs.
Therefore the actors we need to consider are the car owner who wants to charge his car and the charging point operator who wants to manage his charging stations.
Moreover the target area of the eMall system is initially thought to be Italy.

1.2.1 World Phenomena

Identifier	Description
WP1	Car owner decides to charge his car
WP2	DSOs change energy prices
WP3	Car owner blocks a charging station while using it
WP4	Car battery is getting empty/ is full
WP5	Charging station’s battery is full/empty

1.2.2 Shared Phenomena

Identifier	Description
SP1	eMSP shows charging stations nearby

SP2	Car owner books a charging station
SP3	CPMS starts a charging process
SP4	Car owner pays for a charging
SP5	CPMS ends a charging process
SP6	CPMS shows current price of energy of DSO
SP7	CPO selects energy acquisition from DSO, battery or a mix of both
SP8	CPO sets the price/ special offer for a charging station
SP9	CPO stores energy in batteries of charging station, if it exists
SP10	CPMS shows internal/external status information about CPO’s charging stations
SP11	eMSP notifies Car owner when charging is finished
SP12	eMSP notifies Car owner when charging is suggested
SP13	CPO activates CPMS’ automatic mode

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Word	**Definition**
Charging station	A station with at least one charging socket where electric cars can recharge their batteries. In some stations additional batteries are connected that can store energy or give the energy to a connected car through the charging station.
Charging Socket	A socket is part of a charging station. It is the cable connecting the charging station and electric car.
Type of Charging sockets	There are different charging stations depending on the velocity of their charging process. The options are slow, fast and rapid.
Battery	A battery is connected to a charging station and can store energy or give the energy to a connected car through the charging station.
Charging Process	The charging process describes the process where the charging station gives energy to the car’s battery until the car’s battery is full.
Charging Station’s External Status Information	External information of a charging stations consists of: Number of charging sockets available, their type such as slow/fast/rapid, their cost, and, if all sockets of a certain type are occupied, the estimated

	amount of time until the first socket of that type is freed
Charging Station's Internal Status Information	Internal information of a charging stations consists of: Amount of energy available in its batteries, if any, number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the charge

1.3.2 Acronyms and Abbreviations

Word	**Description**
RASD	Requirements Analysis and Specification Document
WP X	World Phenomena number X
SP X	Shared Phenomena number X
GX	Goal number X
DX	Domain assumption number X
RX	Requirement number X
eMall	e-Mobility for All
eMSP	e-Mobility Service Providers
CPO	Charging Point Operator
CPMS	Charge Point Management System
DSO	Distribution System Operators

1.4 Reference Documents

The requirements and conclusions mentioned in this document are all derived from the specification document “Assignment RDD A.Y. 2022-2023 Software Engineering 2”.

1.5 Document Structure

This document consists of six sections. The first section was about the purpose of this document. It explained the fundamental definitions and acronyms used in this document and the application domain of the software that is described in the following sections.

In the following section we will start with an overall description of the software. We see the scenarios in which the software is used and the main user characteristics, aligned with the main domain assumptions considered in order to develop the project.

In section three, the specific requirements are elucidated, in the same section, it is possible to see the presence of the main use cases of the system as well as the sequence diagrams of the main activities, this process helps to clarify the functional requirements, also seen in this section. At the end of this part, it is possible to analyse the performance requirements, design constraints and software systems attributes.

The fourth part consists in the Alloy code used to verify the system in a more objective way, part five consists in the effort spent by the members of the project and finally, the sixth part consists in the references section.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios based on shared phenomena

1. **Car Owner wants to use the eMSP system for the first time**
   The car owner Alice has an electric car and wants to register to the eMSP system, so that she is always able to find the best suiting and cheapest charging station close to her.

   She starts the system and sees the Login page. She selects the “Register” button. In the registration form she gives all the relevant data, such as her name, Email, password, her car’s battery data, her bank account information and allows access to her phone’s locationing service and her calendar.

   Now she sends the registration form to the eMSP system, which sends her an acknowledgment Email.

2. **Car Owner uses eMSP to book a charge**
   The car owner Alice drives around in her electric car, when a message appears on her board computer that the car runs low on battery. To find a suitable charging station she stops in a parking lot and starts the eMSP system. She logs into her account and selects the “Find closest charging station” function.

   The eMSP will now locate her position and search for nearby charging stations that have free sockets. Then a list will appear on her phone displaying the location of the closest stations, the type of it (slow, fast or rapid) and the cost of a charge including potential special offers.

   Alice selects an option by pressing on one of the list elements. A booking window will open that shows the form necessary for the booking. She needs to choose a timeframe in which she wants to charge her car and decides on charging it for one hour from 3pm to 4pm. The form shows the offer for this hour is 10€, which Alice accepts. Now she drives to the selected charging station and the eMSP will collect the money from the bank account connected with Alice’s eMSP account.

3. **Car Owner performs a charging process**
   Alice arrives with her electric car at a charging station for which she previously booked a timeslot of one hour to charge her car by using the eMSP system. She parks her car in one of the available free sockets at the beginning of the chosen timeslot.

   Now she attaches the charging station's cable to the car and opens the eMSP system. There she can select her active bookings and press "Start". Then the eMSP system sends a request to the CPMS system, which checks the bookings, validates the request and starts charging according to the amount of power supplied by the socket. Her car is charging now, monitored by the CPMS system, and she goes away for a walk.

   After one hour she returns to her car. The CPMS stops the charging because the battery is full and notifies the eMSP of the end of charging. The eMSP system sends Alice a notification that the charging process is finished. Now Alice can unplug her car and drive away.

4. **The eMSP system suggests proactively a charging to the car owner**
   Alice uses her car every day to drive to work at 8am and to drive home at 5pm. Her eMSP system has access to her calendar and navigation system, for which it knows this schedule. After yesterday's drive back home her car battery is down to 30%, which is also known by the eMSP system.

   Before Alice gets to her car now, the eMSP system sends her a notification suggesting to charge her car close to her working place. At 8:30am there is a special offer of a charging station next to her work and there are still sockets available for her to book. Since she doesn't need her car during working time, she can leave it there and after work it will be charged completely. Alice accepts this offer and books the charging station's socket for 8:30am.

5. **CPO wants to use the CPMS system for the first time**
   The CPO John has some charging stations and wants to register to the CPMS system, so that he can manage his stations and energy supply.

   He starts the system, sees the Login page and selects the "Register" button. In the registration form he gives all the relevant data, such as his name, Email, password and relevant data about his charging stations (amount of sockets, charging velocity, whether a battery is connected, location, technical data regarding the charging process itself).

   Now he sends the registration form to the CPMS system, which sends him an acknowledgment Email.

6. **CPO wants to check the status information about his charging stations**
   The CPO John logs into his account and selects the status information page. There he can decide if he wants to see the internal or external status information.

   He first clicks on the external status information and the CPMS system shows to him a list of all his charging stations. For each one he finds the number of charging sockets available, their type such as slow/fast/rapid, their cost, and, if all sockets of a certain type are occupied, the estimated amount of time until the first socket of that type is freed.

   Then he clicks on the internal status information and the CPMS system shows to him again a list of his charging stations. For each one he finds the amount of energy available in its batteries, if any, number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the charge.

7. **CPO chooses to acquire energy from a DSO and Battery**
   The CPO John logs into his account and clicks on the “Energy Acquisition” page and then on the “Energy Offers” tab. There the CPMS system shows him a list of all the available DSOs and their price of energy. He chooses the first one and because it is the best offer. A page opens where he accepts the offer and pays for the energy. Nevertheless it doesn’t suffice for John’s energy needs but the other DSO offers are too expensive for John. Therefore he clicks on the same page on the “Batteries” tab, which shows him another list of all the charging stations that have one or more connected batteries and their charging status. He sees that all of his three charging stations have each one fully charged battery so he selects to acquire energy from the batteries too. In total the energy is sufficient now for the charging station’s needs.

8. **CPO sets a price for his charging stations**
   The CPO John is already logged in and wants to change the price for his charging stations. He clicks on the “Pricing” button and finds a form where he can set the new price for the charging station. He can also select the “Special offer” button and then select for how long the offer should be offered and the price for it. He selects that a charge costs 10€ per hour and submits the form by clicking on “Accept”. Now the CPMs system updates the information and car owners will see the updated price when searching for John’s charging stations.

9. **CPO activates automatic mode for CPMS**
   The CPO John is already logged in and wants to activate the automatic mode of the CPMS. The CPMS will then decide by itself the energy acquisition, when to save energy in batteries or use the batteries for charging and the ideal price or time for special offers. For this he toggles the “Activate Automatic Mode” on the main page and the mode is activated as long as John doesn’t deactivate it.

10. **CPO stores energy in batteries**
    The CPO John is already logged in and saw a really good offer from a DSO. He uses the energy for his charging stations but also wants to store this cheap energy in the charging station’s battery. So after accepting the offer he clicks on the “Batteries” tab under “Energy Acquisition” and clicks on the button “Save surplus energy in

batteries”. Now the CPMS system will start charging the charging station’s battery if there is more energy available than used in the moment.

2.1.2 Domain model

Class Diagram

Here in the class diagram of the domain, we see all the entities participating in our goal domain. Car owner and CPO are both Users because they use the two subsystems eMSP and CPMS.

Each CPO uses one unique CPMS system that communicates with multiple DSOs to gather information regarding energy prices. He owns charging stations that have certain status information and each has a different amount of charging sockets. A charging socket can be slow, fast or rapid charging. Next to the charging sockets, a charging station can have batteries.

A car owner owns a car that also has a battery. He uses the eMSP system that communicates with multiple CPMS systems to find the best suitable charging stations for the car owner.

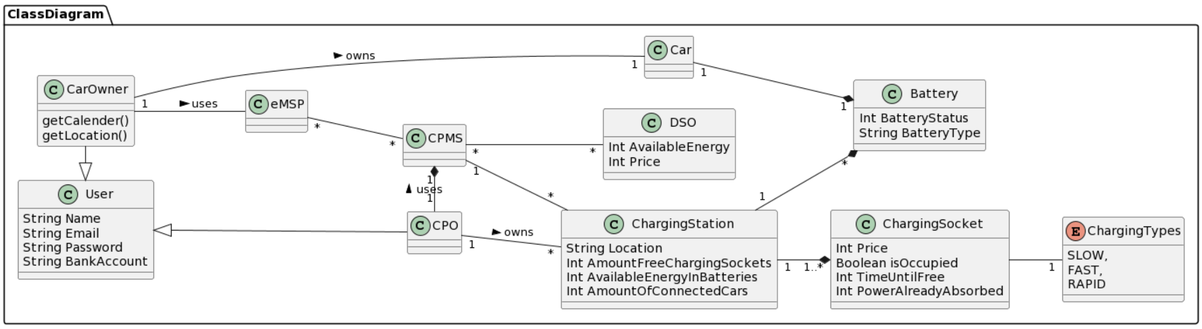

Figure 1: Domain Class Diagram

State Diagram

Furthermore, it is possible to design the state machine of the states comprehended in the charging station’s processes.

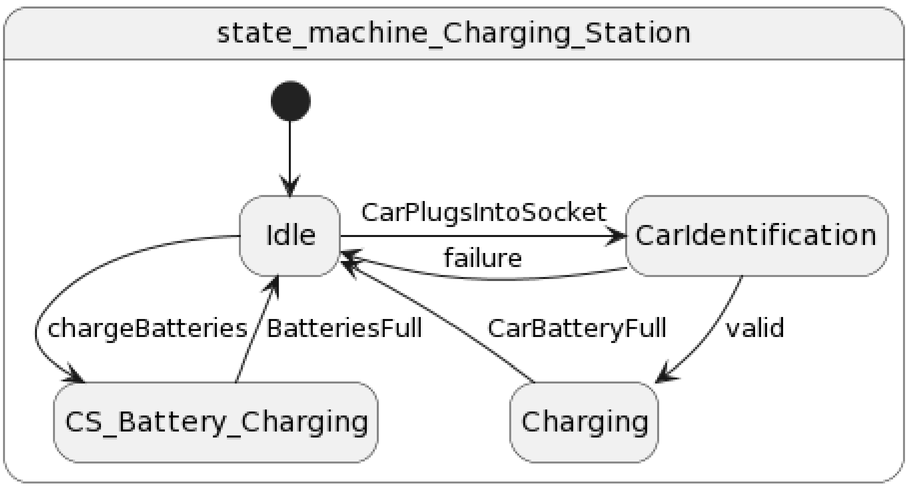

Figure 2: State Diagram

2.2 Product functions

In the following we will focus on the main functionalities of the two subsystems eMSP and CPMS.

2.2.1 Booking procedure of charging an electric car

One of the most important tasks of the eMSP is to manage the charging process. It keeps an eye on the battery of the car owner and if the battery is low and there is a possible charging station close to the owner's location or fitting into the owner's schedule with a good price offer, the system will recommend it to the car owner. This works proactively so that the owner doesn't need to take care of the charging himself.
But if he still wants to search for stations himself, he can also use the system to find nearby charging stations and decide on location, type and offer, which one he wants to book. The booking and payment procedure is also part of the system's functionality so that the owner can do the complete booking procedure at one place.

2.2.2 Charging procedure of electric car

Once the car owner booked a charging station and plugged it into the car, he can also start charging with the eMSP system. The system will communicate with the CPMS system of the charging station to activate the charging and will also inform the owner as soon as his car is finished with charging.

2.2.3 Checking status information about charging stations

The management of the charging stations is the main task of the CPMS system. To do that the CPO needs to keep informed about the status information of his charging stations. Therefore the CPMS provides him with all the internal and external status information of the charging stations.

2.2.4 Acquiring energy from a DSO or connected battery

The other important part of the management process is to decide from where to acquire energy. The CPMS offers multiple solutions for this. First the CPO can decide to acquire energy from a DSO. The CPMS will provide him with the current prices and the CPO can choose from which DSO to buy his energy. If the energy prices are good enough, the CPO can also decide to charge the batteries of his charging stations, in case there are any. Later, when the prices are worse or the offer of the DSO ends, he can decide to use the energy saved in the batteries instead of acquiring energy from another DSO. And lastly, he can also decide to use a mixture of these two solutions.

2.2.5 Adjusting prices for charging stations or select special offers

After managing the acquisition of the energy, the last part of the managing process is to set a price per hour for the charging stations of the CPO. The CPO can either simply update the price or select a special offer, meaning he gives a specific discount for a chosen time frame.

2.2.6 CPMS Automatic mode

If the CPO does not want to decide himself, he can select the Automatic mode of the CPMS. The CPMS will now handle all decisions by itself, meaning it will handle the energy acquisition, when to save energy in batteries or use the batteries for charging and the ideal price or time for special offers.

2.3 User Characteristics

In this section we present the two actors using the software.

1. **Car Owner**
   A registered Car Owner uses the system to book charges for his electric car. He can find charging stations and select the one with the best offer. He will also get suggestions by the eMSP to charge his car, when the situation is suitable, for example there is a special offer close to the owner's location or the charge would fit well into the owner's schedule.
   We expect that the car owner wants a simple system that is easy and fast to use. There should be no long procedure between locating the closest charging station and actually booking one. We expect the owner to be in the middle of his daily schedule where he wants to use as little time as possible to find the best offer for his charge. So the system should support the decision as much as possible, so the owner will get the best charging option in the least amount of time.

2. **Charging Point Operator**
   A registered Charging Point Operator uses the system to manage his charging stations which means checking their status information and acquiring energy from the best possible sources.
   We expect that the CPO wants a well-organised system, where he can find a high amount of different information in clearly divided sectors. It should be easy to use but still provide the necessary complexity and detail of information, which is needed in order to decide on energy sources.

2.4 Domain Assumptions

Identifier	Description
D1	The whole schedule of the user is found on the calendar on the car owner's phone
D2	The location service of the car owner's phone is accurate and constantly working
D3	There is an API for the communication between all entities of eMSP and

	CPMS and between CPMS and DSO.
D4	The charging stations and connected batteries are fully functional
D5	Car owners will free up the charging socket after their charging process is finished
D6	Car owners insert correct data about their car’s technical information
D7	Car owners and CPO insert correct bank account information, from which the payment can be successfully paid
D8	CPOs insert correct technical data and further information about their charging stations
D9	There exists an API for the eMSP and CPMS system to forward the payment request to. It will handle the communication with the bank and complete the payment procedure.
D10	There exists an algorithm the CPMS will use when the automatic mode is activated
D11	The socket will be freed up exactly when the charging time is up
D12	DSOs always deliver the energy in the stipulated time
D13	It's always possible to predict when the charging will be completed
D14	The DSO is not considered as an agent for the CPMS system or the eMSP system. It is just an energy provider from which the CPMS can retrieve information via an API.
D15	We assume each Car Owner has exactly one car registered on his account

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface of eMSP is a mobile app that will allow car owners to use the eMSP system on their smartphones. The app has to be user-friendly, in other words, it has to be fast and easy to use (no technical background should be needed to use the app). Moreover, the response time should be really low, with very straightforward processes.

Whereas, the CPMS is a website that will be used by CPOs. The idea of a website is that it is a better way to visualise and organise the information in different and well-divided sectors, mainly if it is used in a computer, but it is still user-friendly on mobile devices. Due to its uses for deciding on energy sources, it should be simple to use while still offering the essential complexity and detail of information.

3.1.2 Hardware Interfaces

Since both subsystems revolve around inputting and visualising data, both are fully operable with internet connection, the only hardware interface needed to access eMSP is a working smartphone with a GPS. Similarly, to access CPMS it is needed a working computer or smartphone with a web browser.

Moreover, the system relies on the use of sensors that are able to measure useful data for both eMSP and CPMS, such as measuring the battery level of a car or batteries of charging stations or to check if a charging socket is available or occupied. However, these factors combined with the transportation of energy are managed externally and the data is retrieved through the usage of APIs.

3.1.3 Software Interfaces

The main software interface is a domain assumption concerning the algorithm for CPMS automatic mode. And also the smartphone using eMPS can download apps and location services and calendar are present.

3.1.4 Communication Interfaces

The system's functionality depends on interaction with other services for a number of its features. There are different interfaces of communication, all of them are through APIs. Below is presented a list of the necessary functions offered by each interface.

- eMSP with Smartphone
  - eMSP must get data from calendar (car owner schedules) and location services.
- eMSP with Battery:
  - eMSP needs to get information concerning the car’s battery.
- eMSP with CPMS
  - eMSP needs information concerning charging stations, such as charging station’s location, price, special offers, type of sockets. Also, eMSP needs to send the “start charging” signal for CPMS.
- eMSP or CPMS with Bank API
  - In order to pay and/or charge for services.
- CPMS with Charging Stations
  - External and Internal information of a charging station.
- CPMS with Battery
  - Information concerning connected batteries.
- CPMS with DSO
  - Information about energy prices and availability.
- Charging Station with Battery
  - Information concerning both car’s batteries and the ones of the charging station itself.

3.2 Functional Requirements

3.2.1 Use Case Diagrams

1. Registration/ Login Procedure of Car Owner and CPO

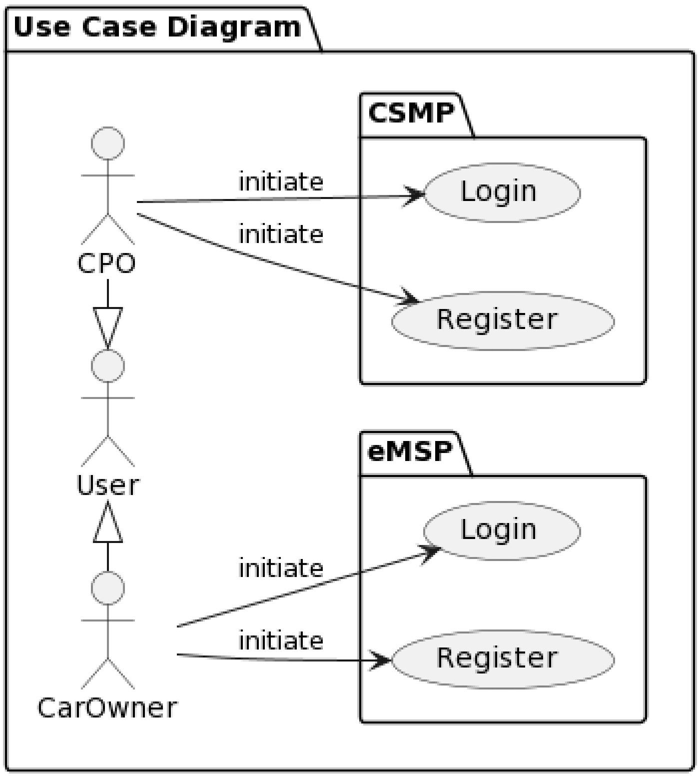

Figure 3: Use case diagram for a registration/ login procedure

2. eMSP - Book a charging process

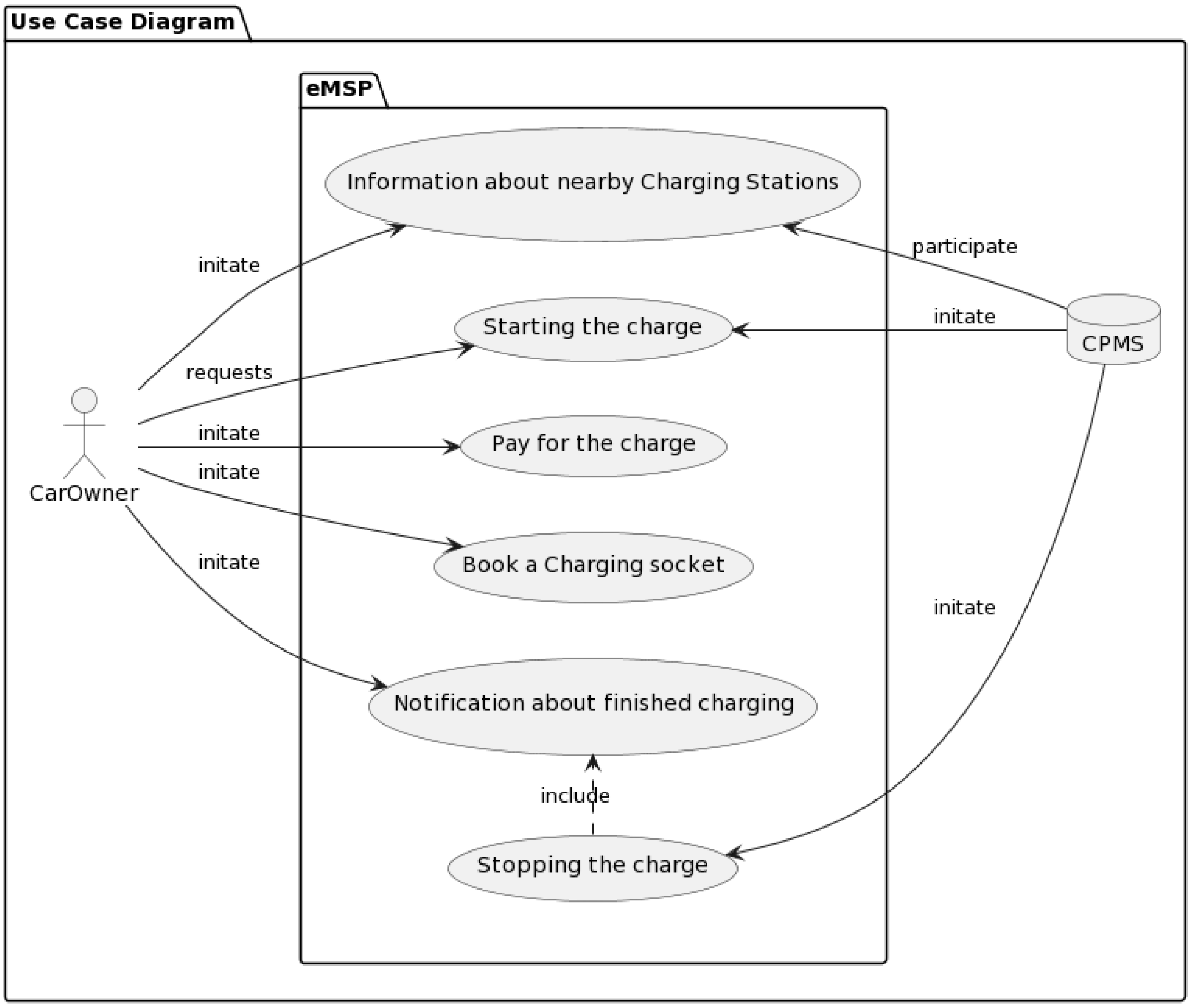

Figure 4: Use case diagram for booking a charge

3. eMSP - Suggest charging station

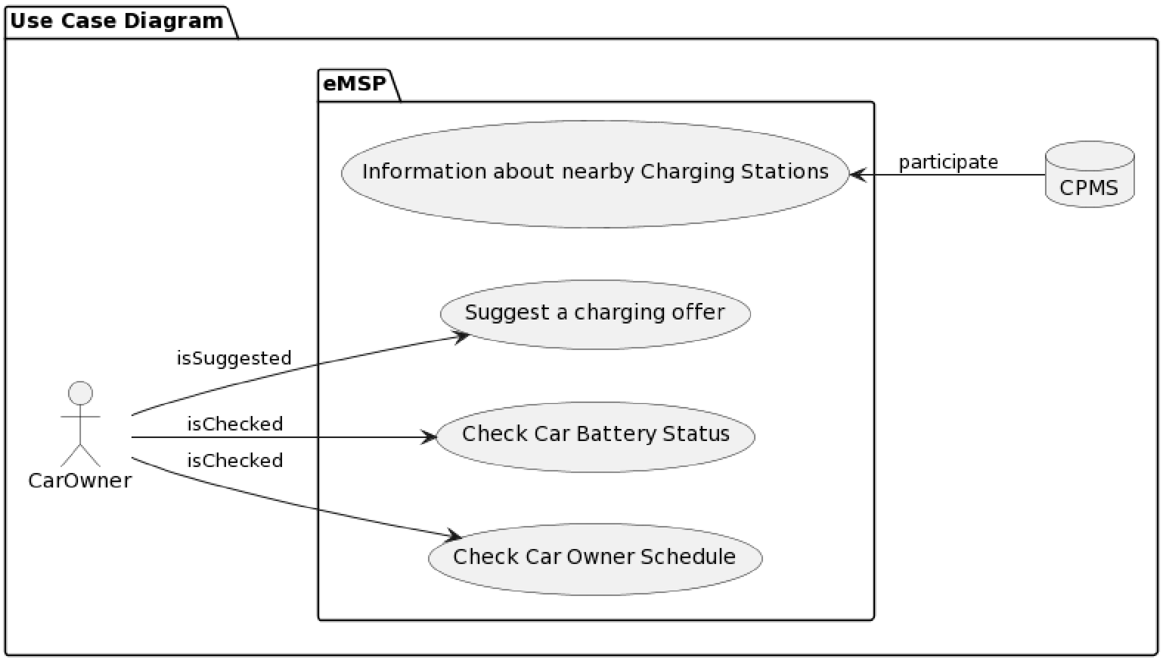

Figure 5: Use case diagram for a suggestion by eMSP

4. CPMS - Manage Charging Stations

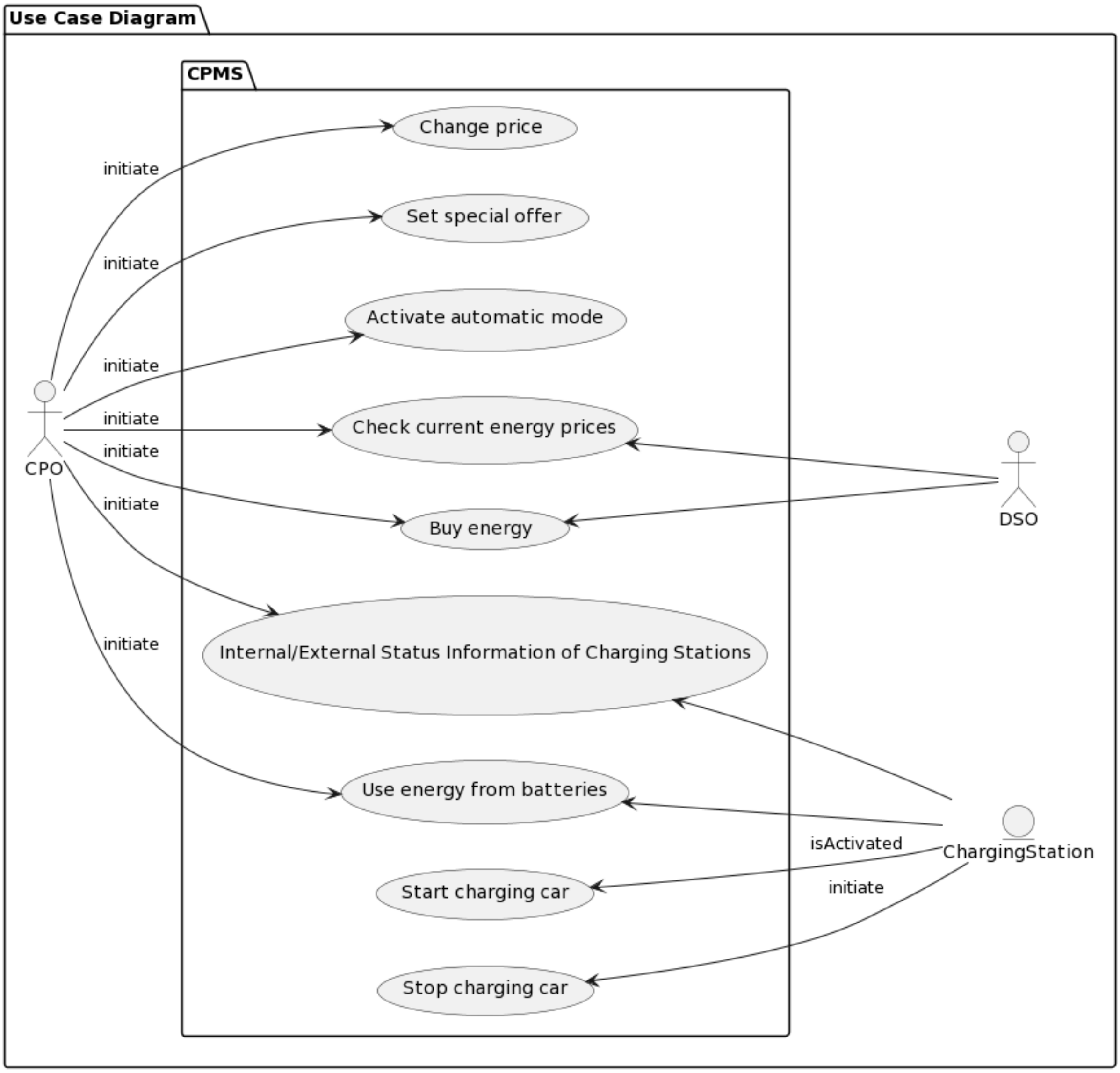

Figure 6: Use case diagram for the management of Charging Stations by the CPMS

3.2.2 Use Cases

1. Car Owner registers on eMSP system

Actor	Car Owner
Entry conditions	The car owner does not have an account and is on the initial view of the eMSP system
Event flow	1. The car owner presses the “Register button” 2. The registration form is loaded 3. The car owner enters name, Email, password, his car’s battery data, his bank account information and allows access to his phone’s locationing service and calendar.

	4. He presses “Submit” 5. The eMSP system processes the information and displays a success message
Exit condition	An account for the car owner is created
Exceptions	1. The car owner does not enter all mandatory data 2. Phone does not have locationing service 3. Phone does not have calendar 4. Incorrect data on car’s battery 5. Incorrect bank account information → **In all cases eMSP system will notify the car owner**

2. Car Owner logs into the eMSP system

Actor	Car Owner
Entry conditions	Car Owner is registered, not logged in and on the eMSP system main page.
Event flow	1. Car Owner presses “Login” button 2. “Login” page is shown 3. Car Owner inserts his Email address and password 4. Car owner presses “Submit” 5. eMSP processes the information and displays a success message
Exit condition	Car Owner is logged into eMSP system
Exceptions	1. Car Owner inserts wrong password 2. Car Owner inserts wrong Email → **In all cases eMSP system will notify the car owner**

3. CPO registers on CPMS system

Actor	CPO
Entry conditions	The CPO does not have an account and is on the initial view of the CPMS system
Event flow	1. The CPO presses the “Register button” 2. The registration form is loaded 3. The CPO enters name, Email, password, his car’s battery data, his bank account information and relevant data about his charging stations (amount of sockets, charging velocity, whether a battery is connected, location, technical data regarding the charging process itself) 4. He presses “Submit” 5. The CPMS system processes the information and displays a success message

Exit condition	An account for the CPO is created
Exceptions	1. The CPO does not enter all mandatory data 2. Incorrect data on charging stations 3. Incorrect bank account information → **In all cases CPMS system will notify the CPO**

4. CPO logs into the CPMS system

Actor	CPO
Entry conditions	CPO is registered, not logged in and on the CPMS system main page.
Event flow	6. CPO presses “Login” button 7. “Login” page is shown 8. CPO inserts his EMail address and password 9. CPO presses “Submit” 10. CPMS processes the information and displays a success message
Exit condition	CPO is logged into CPMS system
Exceptions	3. CPO inserts wrong password 4. CPO inserts wrong EMail → **In all cases CPMS system will notify the CPO**

5. Car Owner books a charge

Actor	Car Owner
Entry conditions	Car Owner is logged into the eMSP and on the initial view
Event flow	1. The car owner clicks on “Book Charging Station” 2. A page is shown with all available charging stations nearby. The exact location, offer and type is shown. 3. The car owner selects one option. 4. A form appears to ask the car owner how long he wants to charge 5. He selects a timeframe and submits 6. A new page is shown presenting the booking details (which station for how long and the price for it). 7. The car owner clicks on “Book” 8. The eMSP system forwards the payment and bank account information to an API that performs the payment procedure 9. The information is processed and the eMSP system displays a success message
Exit condition	The car owner has a booking for a charging station
Exceptions	1. There are no free Charging Stations nearby

	2. The selected time is not possible (e.g. negative amount of hours) 3. The payment cannot be processed due to bank account problems → **In all cases eMSP system will notify the car owner and cancels the current order**

6. Charging procedure of car owner using eMSP

Actor	Car Owner
Entry conditions	The Car Owner is logged into the system and on the main page. He parked his car next to a charging station and has a booking for this charging station
Event flow	1. The car owner plugs the charging socket into his car 2. He selects “Active Bookings” 3. A page is shown presenting the active bookings 4. He selects the booking and presses the “Start” button 5. The charging socket reads out the identification of the car 6. The information is processed in the CPMS system and the charging starts. 7. The CPMS system monitors the charging. As soon as the battery of the car is full, it stops charging and notifies the eMSP system. 8. The eMSP system notifies the car owner that the car is ready to unplug 9. The car owner unplugs the car and drives away 10. The current booking is no longer visible in “Active Bookings”
Exit condition	The car owner has a fully charged car
Exceptions	1. The identification of the car is not valid when the car owner drove to the wrong charging station 2. The battery is broken and will not charge 3. The car owner does not unplug the car → **In all cases eMSP system will notify the car owner**

7. eMSP system suggests charging offer to Car Owner

Actor	eMSP system
Entry conditions	The Car Owner has an eMSP account and allowed access to his phone’s calendar and locationing system.
Event flow	1. The car’s battery is under 30% 2. The eMSP system checked the calendar selects a suitable time slot for charging the car 3. Close to the car owner’s location it checks for fitting charging stations and selects one

	4. The eMSP system notifies the Car owner presenting the possible time slot, location of charging station and price.
Exit condition	The car owner is presented with an offer for a charge that he can accept
Exceptions	1. The car owner did not insert all his schedule data into the calendar and does not have free time during the time slot → **In this case the car owner can simply reject the offer** 2. The locationing of the phone does not work to figure out the nearby charging stations 3. The calendar of the phone does not work to figure out free spaces for charging → **In there cases the eMSP will notify the car owner**

8. CPO checks status information about charging stations

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the “Charging Stations” tab 2. The CPMS system shows a list with all charging stations presenting the internal and external status information
Exit condition	The CPO sees the information he wanted
Exceptions	1. There are no charging stations available → **In all cases CPMS system will notify the CPO**

9. CPO buys energy from DSO

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the “Energy Acquisition” tab. 2. He sees a list of possible DSOs and another tab with the name “Batteries” 3. He chooses one of the DSOs 4. A form opens where he can select the amount of energy he wants and how much it costs 5. He submits the form by clicking “Order” 6. The CPMS system forwards the payment and bank account information to an API that performs the payment procedure 7. The CPMS system displays a success message.
Exit condition	The CPO has bought energy from the DSO

Exceptions	1. The payment cannot be processed due to bank account problems → **In this case CPMS system will notify the CPO**

10. CPO chooses to acquire energy from Charging Station's batteries

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the "Energy Acquisition" tab. 2. He sees a list of possible DSOs and another tab with the name "Batteries" 3. He clicks on "Batteries" 4. The CPMS system displays the possible batteries and their charging status 5. He can select the batteries by clicking "Use Energy" and afterwards click "Accept" 6. The CPMS system acquires energy from the chosen batteries now
Exit condition	The CPO chose the Charging Station's batteries as energy source
Exceptions	1. There are no charged batteries → **In this case CPMS system will notify the CPO**

11. CPO sets "Special offer" for his Charging Stations

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the "Pricing" tab 2. The CPMS shows a form where he can insert a price or select special offer 3. The CPO selects special offer 4. Now he can select the timeframe and price for this timeframe 5. He submits by pressing "Accept"
Exit condition	The CPMS sets a special offer for the chosen timeframe and returns afterwards to the price before
Exceptions	1. The time frame is not possible → **In this case CPMS system will notify the CPO**

12. CPO sets Price for his Charging Stations

Actor	CPO

Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the “Pricing” tab 2. The CPMS shows a form where he can insert a price or select special offer 3. The CPO inserts a new price 4. He submits by pressing “Accept”
Exit condition	The CPMS sets a new price for all charging stations
Exceptions	1. The price is not possible (e.g. negative) → **In this case CPMS system will notify the CPO**

13. CPO activates automatic mode

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the toggle on the main page to activate the automatic mode
Exit condition	The CPMS manages all decisions by itself
Exceptions	

14. CPO stores energy in Charging Station’s batteries

Actor	CPO
Entry conditions	CPO is logged into the CPMS system and on the main page
Event flow	1. The CPO clicks on the “Energy Acquisition” tab. 2. He sees a list of possible DSOs and another tab with the name “Batteries” 3. He clicks on “Batteries” 4. The CPMS system displays the possible batteries and their charging status 5. He selects the batteries he wants to charge by clicking on “Save Energy” and submits via the button “Accept” 6. The CPMS system saves energy in the chosen batteries now
Exit condition	The Charging Station’s batteries are charging
Exceptions	1. There are no attached batteries at the owned charging stations 2. There is not enough energy to charge the batteries on top of charging connected cars → **In all cases CPMS system will notify the CPO**

3.2.3 Requirements

eMSP system

Requirement	Description
R1	The eMSP system shall allow an unregistered car owner to register an account by entering name, Email, password, his car's battery data, his bank account information and allows access to his phone's locationing service and calendar
R2	The eMSP system shall allow a registered car owner to search for charging stations nearby
R3	The eMSP system allows the registered car owner to login with his registered account by entering his Email address and password
R4	The eMSP system allows the registered car owner to book a charge
R5	The eMSP system allows the registered car owner to start a booked charge
R6	The eMSP system suggests the registered car owner a suitable offer based on his location and calendar information
R7	The eMSP system must be able to notify the car owner of incorrect data during the registration
R8	The eMSP system must be able to notify the car owner of wrong login data during the login
R9	The eMSP system must be able to notify the car owner when his selected timeframe in the booking is not possible
R10	The eMSP system must be able to notify the car owner if there is no charging station nearby when he searches for them
R11	The eMSP system must be able to notify the car owner if the payment procedure during the booking failed
R12	The eMSP system must be able to notify the car owner if he plugged his car into a wrong charging station
R13	The eMSP system must be able to notify the car owner if his car does not charge
R14	The eMSP system must be able to notify the car owner if he forgets to unplug his car
R15	The eMSP system must be able to access the locationing system of the car owner
R16	The eMSP system must be able to access the calendar information of the car owner
R17	The eMSP system must be able to communicate with an API handling the

	communication with the bank accounts to handle the payment procedure
R18	The eMSP system must be able to communicate with all the CPMS systems of the charging stations nearby
R19	The eMSP system shall allow the car owner to view his active bookings
R20	The eMSP system shall allow the car owner to start a charge
R21	The eMSP system shall notify the car owner when his car is ready to unplug
R22	The eMSP system checks for charging suggestions when the car battery is below 30%
R23	The eMSP system must be able to check the car owner’s car battery

CPMS system

Requirement	Description
R24	The CPMS system must be able to communicate with an API showing the DSO information
R25	The CPMS system must be able to communicate with an API handling the communication with the bank accounts to handle the payment procedure
R26	The CPMS system must be able to notify the registered CPO of incorrect data during the registration
R27	The CPMS system must be able to notify the registered CPO of wrong login data during the login
R28	The CPMS system must be able to notify the registered CPO if there are no charging stations attached to his account
R29	The CPMS system must be able to notify the registered CPO if the payment procedure during the energy acquisition failed
R30	The CPMS system must be able to notify the registered CPO if he tries to charge batteries but there aren't any
R31	The CPMS system must be able to notify the registered CPO if he tries to charge batteries but there isn't enough energy
R32	The CPMS system allows the registered CPO to check status information about his charging stations
R33	The CPMS system allows the registered CPO to buy energy from DSO
R34	The CPMS system allows the registered CPO to save energy in batteries at his charging stations

R35	The CPMS system allows the registered CPO to use the energy from the batteries connected to his charging stations to charge connected cars
R36	The CPMS system allows the registered CPO to login with his registered account by entering his Email address and password
R37	The CPMS system shall allow an unregistered CPO to register an account by inserting name, Email, password, his car's battery data, his bank account information and relevant data about his charging stations (amount of sockets, charging velocity, whether a battery is connected, location, technical data regarding the charging process itself)
R38	The CPMS system shall start a charge by giving the signal to the charging station to start charging the connected car
R39	The CPMS system shall inform the eMSP system when the car battery is fully charged
R40	The CPMS system shall allow the registered CPO to see a list of the available charging station batteries
R41	The CPMS system shall allow the registered CPO to see a list of the available DSOs
R42	The CPMS system can check if the connected car is the one for which the charge was booked
R43	The CPMS system shall allow the registered CPO to set a special offer
R44	The CPMS system shall allow the registered CPO to set the price for his charging stations
R45	The CPMS system shall allow the registered CPO to activate the CPMS' automatic mode

3.2.4 Mapping on Requirements

This is the mapping of each goal, the domain assumptions we have in this case and the requirements that fulfil the goals. For each goal we have several requirements explaining in detail the demands we have on the system.

Goal	**Domain assumption**	**Requirement**
G1	D3, D6	R1, R3, R2, R10, R18
G2	D3, D4, D5, D7, D9	R1, R3, R4, R5, R9, R11, R12, R13, R14, R17, R18, R19, R20, R21
G3	D1, D2, D3	R1, R3, R6, R15, R16, R22, R23
G4	D4, D8, D9, D10	R24, R25, R29, R30, R31, R33, R34, R35, R36, R37, R41, R45

G5	D8	R28, R32, R36, R37, R40, R45
G6	D4	R36, R37, R43, R44, R45
G7	D4	R36, R37, R38, R39, R42, R45

3.2.5 Sequence/ Activity Diagrams

In this section we want to show the corresponding sequence diagrams for the use cases we presented before. For simplicity we will show only once the login procedure for the car owner or CPO and will afterwards assume they are already logged in.
Moreover we will only consider the successful interactions since the exceptions are already mentioned in the Use Cases.

1. Car owner registration

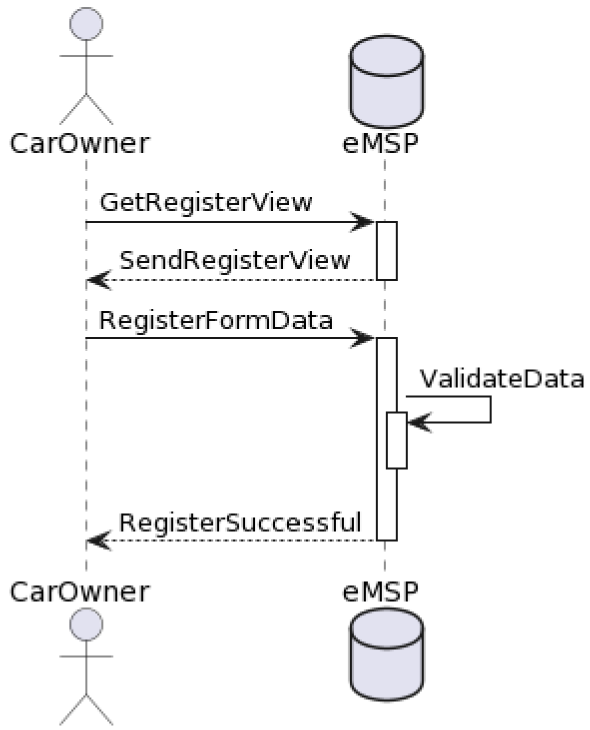

Figure 7: A car owner registers to eMSP

2. CPO registration

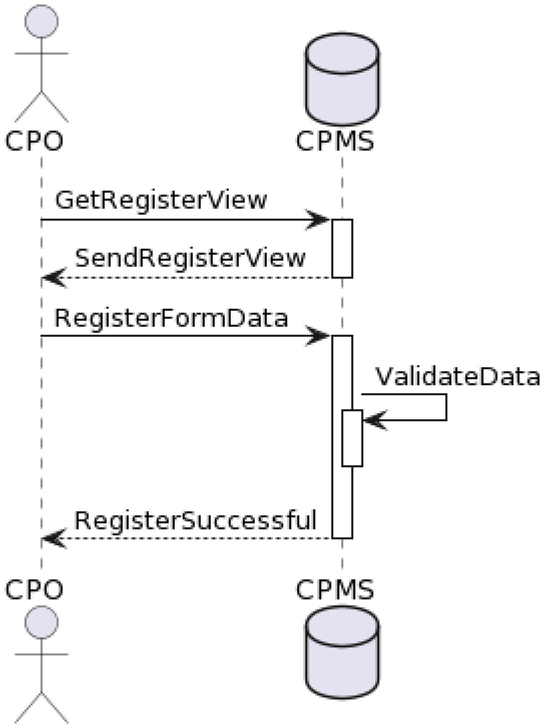

Figure 8: CPO registers to the CPMS

3. Car owner login

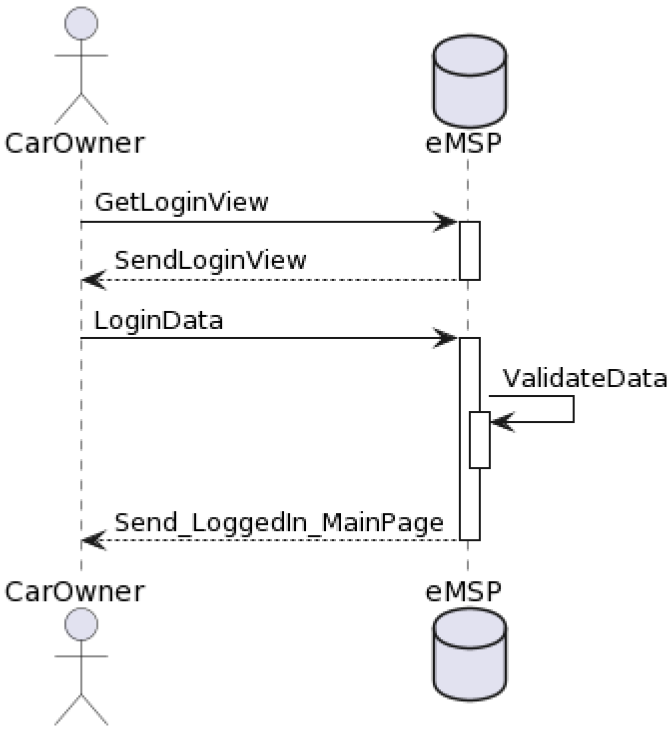

Figure 9: Car Owner logs into the eMSP

4. CPO login

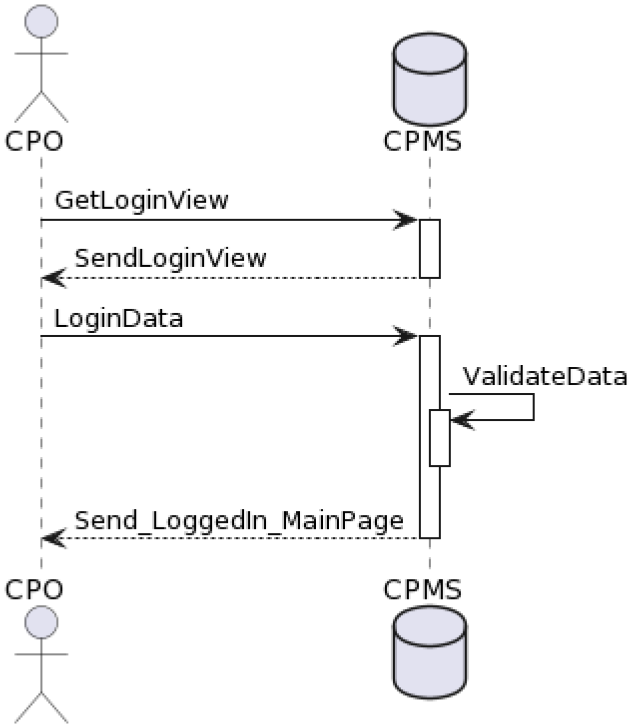

Figure 10: CPO logs into the CPMS

5. Car Owner books a charge

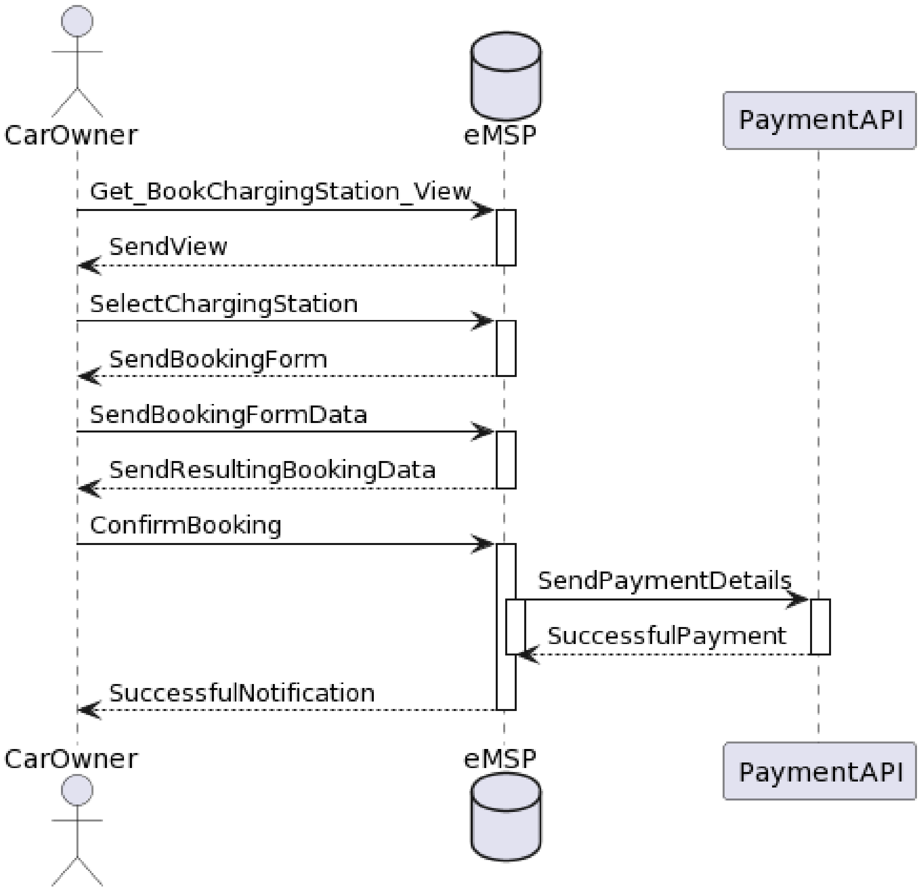

Figure 11: Car Owner books a charge on eMSP

6. Charging procedure of car owner using eMSP

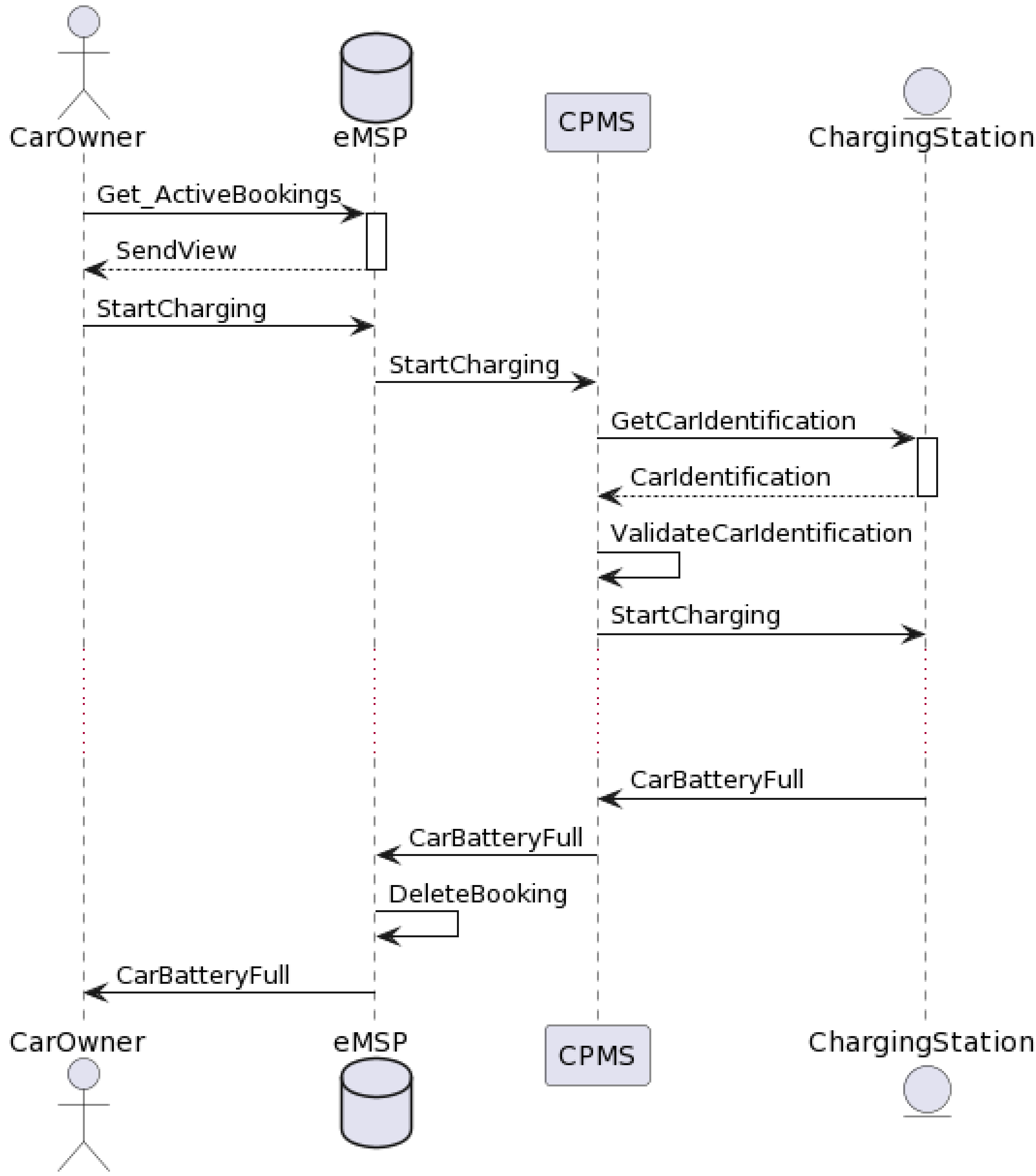

Figure 12: Car Owner charges his car

7. eMSP system suggests charging offer to Car Owner

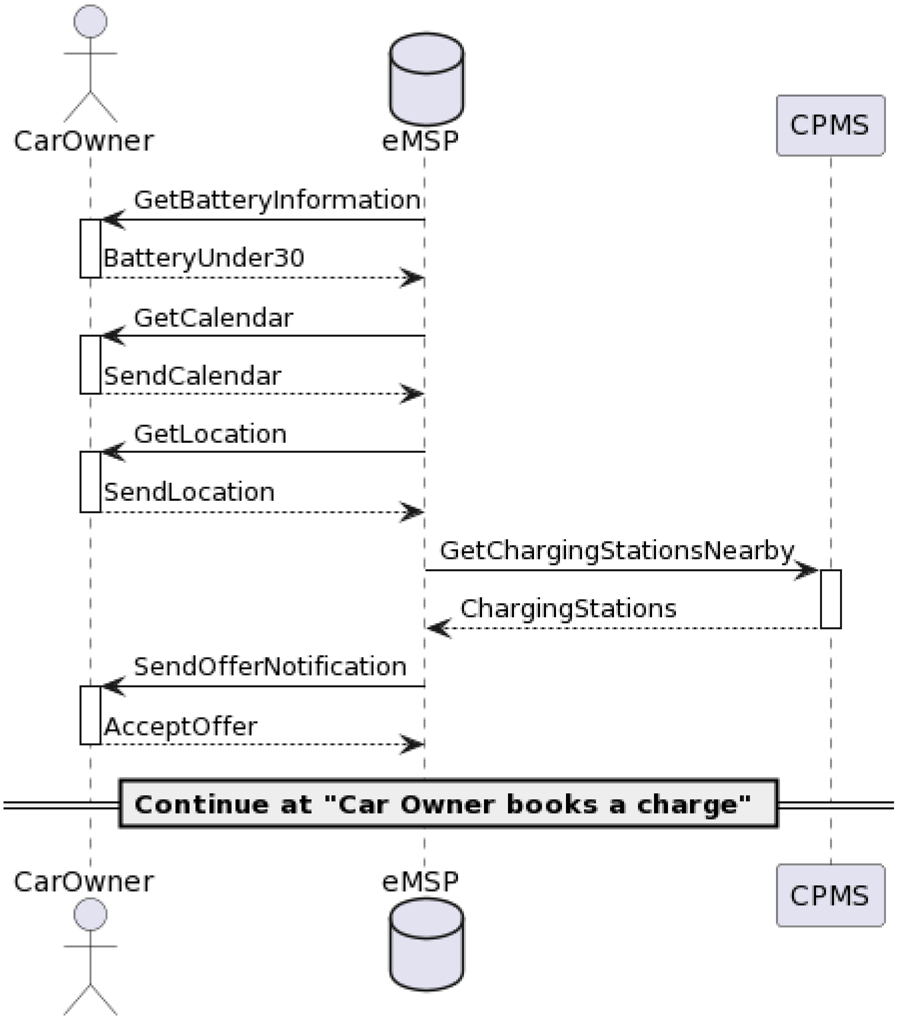

Figure 13: eMSP suggests the car owner to charge the car

8. CPO checks status information about charging stations

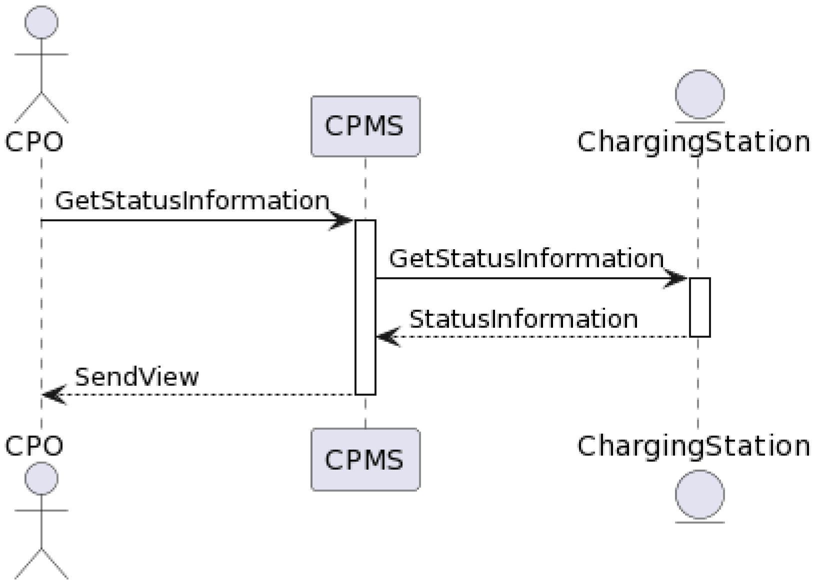

Figure 14: CPO checks the status information of his charging stations

9. CPO buys energy from DSO

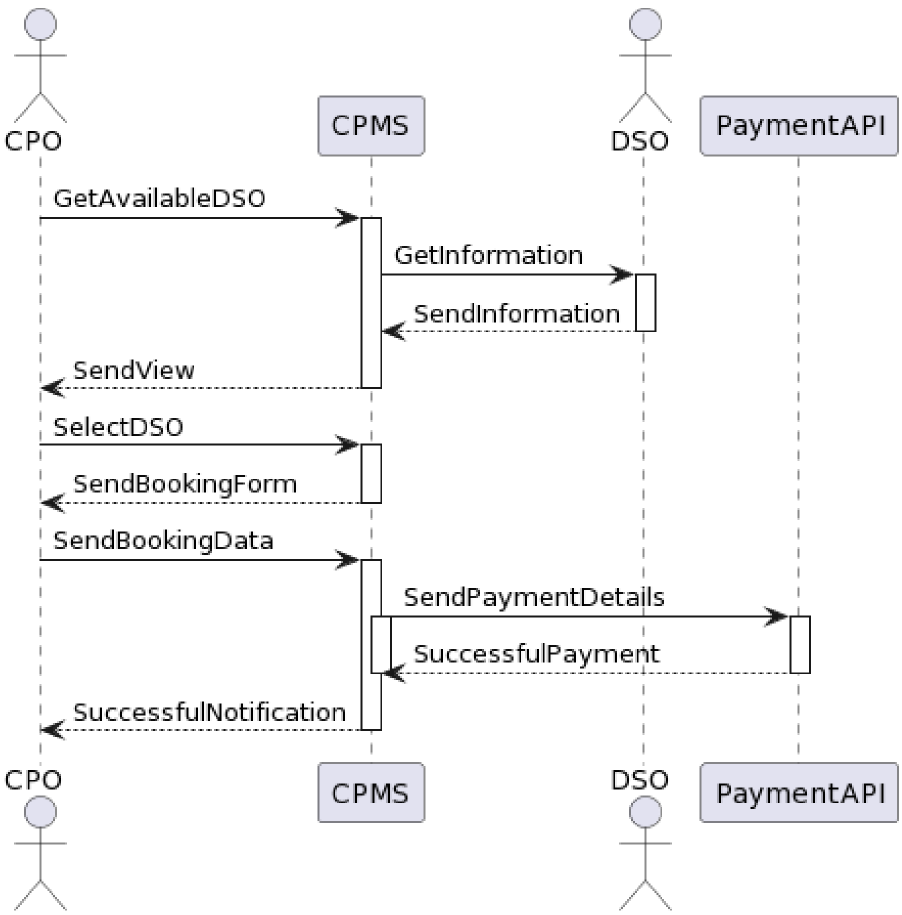

Figure 15: CPO buys energy from the DSO

10. CPO chooses to acquire energy from Charging Station's batteries

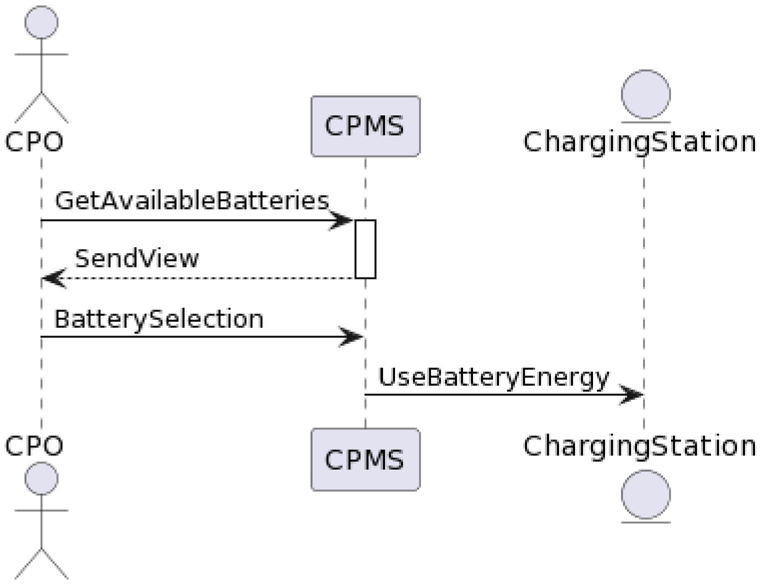

Figure 16: CPO uses energy from charging station's batteries

11. CPO sets "Special offer" for his Charging Stations

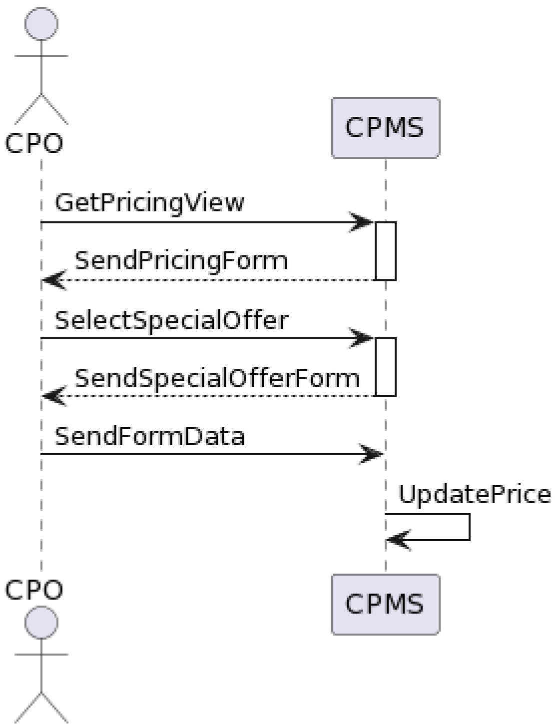

Figure 17: CPO sets special offer

12. CPO sets Price for his Charging Stations

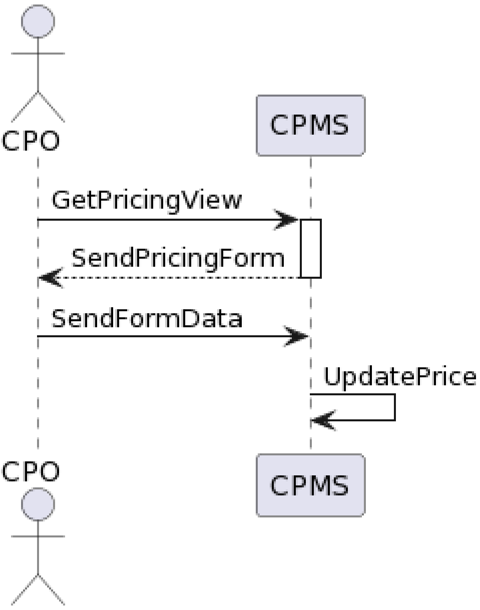

Figure 18: CPO sets price for the charging stations

13. CPO activates automatic mode

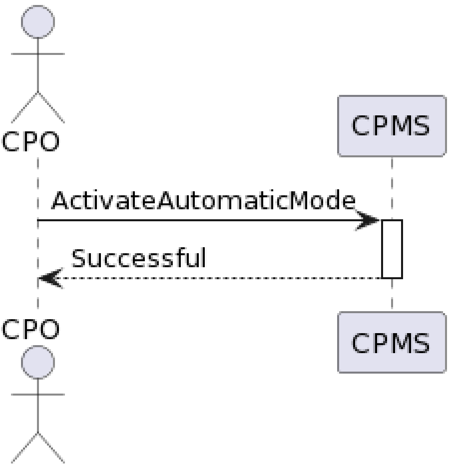

Figure 19: CPO activates the automatic mode of the CPMS

14. CPO stores energy in Charging Station's batteries

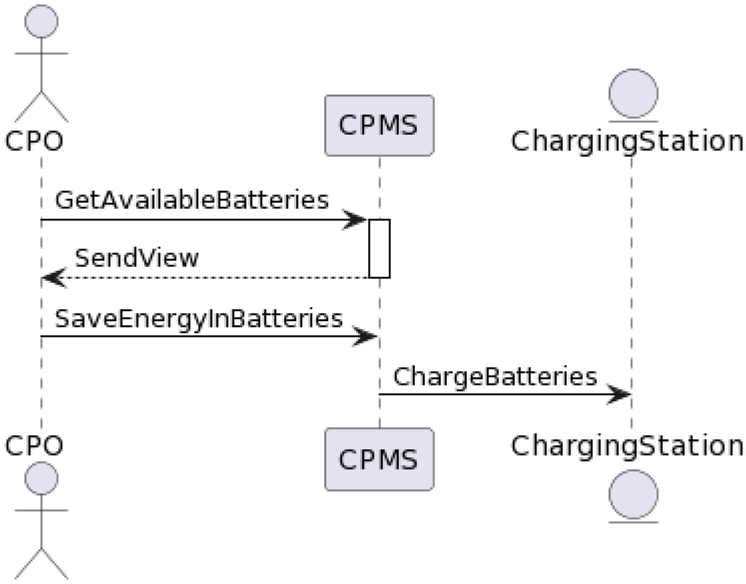

Figure 20: CPO stores energy in the charging station's batteries

3.3 Performance Requirements

The eMSP should be as lightweight as possible since it will be used on mobile phones as an app while people may be on streets, it should not be assumed that all users have access to high speed internet connection, car owners may find slower internet connection in some zones of the city. Moreover, eMSP loading and actions should be quickly and effectively, considering its reliability on calendar and location services of the smartphone. Another performance requirement is the ability to handle a high number of concurrent users of at least 70,000, which is approximately the number of electric cars in Italy.

Performance wise, the CPMS also should be able to handle an elevated number of concurrent users. It is estimated that there will be less than the number of concurrent users for eMSP, considering that there are around 15,000 charging stations for cars in Italy, we can consider at least this number of concurrent users for CPMS. CPMS should be lightweight as well, since relevant information may vary in real-time for CPOs decisions, for instance energy prices by DSOs. Thus, CPMS should have a low latency and high throughput, so data is visualised effectively as it may change in real time and these changes are paramount for CPOs.

3.4 Design Constraints

3.4.1 Standards Compliance

All personal data from users need to be processed according to the GDPR, with the respective privacy policy being explicitly agreed on by the user during registration. This includes the handling and protection of the car owner's methods of payment data (e.g credit card number), location and schedule.

The application for the users (eMSP) should be deployed and available in the main native app platforms, such as Google Play and App Store. While the platform for the management system (CPMS) should function fully on all widely used web browsers (e.g Chrome, Safari, Edge, Firefox).

Finally, all external APIs used to gather data from one system to another need to strictly follow their guidelines.

3.4.2 Hardware Limitations

In order to make it possible for the system to find the best charging stations options available, the eMSP application needs the current location of the user. Given this, the hardware to be used needs: connection to the internet and GPS connection possible to stream its location. For the CPMS platform, only an internet connection is needed.

3.4.3 Other Constraint

In order for the process of charging to be accomplished, the mobile using the eMSP application needs to allow and execute at least one payment method.

3.5 Software System Attributes

3.5.1 Reliability

The system should be highly reliable and needs to be running most of the time, once that the process of charging an electric vehicle may happen at any time during the 24 hours of a day. Regular maintenance needs to be done with a high regularity given that a lot of new CPOs and stations are coming to the market.

3.5.2 Availability

Both the eMSP and the CPMS are of fundamental importance for every electric-vehicle driver looking for a charging station. The failure of the system may occur in extremely negative occurrences (such as a driver running out of battery and needing to call towing services).

Given the risk of leaving the system down, the ideal number should be at least “three zero” availability (99.9%), which results in approximately 8.76 hours throughout a year that was considered a fair amount based on similar applications.

Furthermore, as stated in the “Assignment RDD” document, it is necessary to try elaborating a system connection with outside APIs aiming to prevent a whole system failure if the API or external system fails.

3.5.3 Security

It is extremely necessary to deploy a protection of the private data inside the system, once that much of the present information is high in value to the users and CPOs, such as the payment methods and information used, the type of cars and batteries, availability of stations and strategic data to the operators. Which leads to the obligation of password encryption and API endpoint authorization.

3.5.4 Maintainability

Considering that the electrical vehicle industry is rapidly changing and growing, it is extremely interesting to develop both the eMSP and CPMS with the most current software patterns followed by a clear documentation, once that, in the close future, a lot of new features may become interesting to be developed.

3.5.5 Portability

As stated in the “Standard Compliance” section, there must be a compatibility of the eMSP with the current mobile possibilities in the same manner that there must be a compatibility of the CPMS platform with the most used web browsers.

4 Formal Analysis using Alloy

4.1 Alloy code

For the formal analysis of the system, the Alloy code was developed in order to impose some of the constraints described in this document and verify if there are feasible systems produced.

The code can be seen below where the model of the system is created. Every class declared has attributes and predicates that connect them with each order in order to explore the relation between parts of the system, as well as testing possible errors in the formulation of the domain model.

```
// Signatures

sig Name, Password, Email, BankAccount {}
sig Location {}
sig BatteryType, BatteryStatus {}

sig SystemEMSP {
   connection: one SystemCPMS
}
sig SystemCPMS {
   connection: one SystemEMSP
```

```
}

abstract sig ChargingTypes {}
one sig SLOW extends ChargingTypes {}
one sig FAST extends ChargingTypes {}
one sig RAPID extends ChargingTypes {}

abstract sig Bool {}
one sig True, False extends Bool {}

abstract sig User {
	name: one Name,
	email: one Email,
	password: one Password,
	bankAccount: one BankAccount
}

sig CPO extends User {
	station: some ChargingStation,
	account: one CPMS
}

sig ChargingStation {
	location: one Location,
	freeSockets: one Int,
	energyBatteries: one Int,
	connectedCars: one Int,
	sockets: some ChargingSocket,
	owner: one CPO,
	batteries: set Battery,
	system: one SystemCPMS
} {
	freeSockets >= 0
	energyBatteries >= 0
	connectedCars >= 0
}

sig ChargingSocket {
	station: one ChargingStation,
	price: one Int,
	isOccupied: one Bool,
	timeUntilFree: one Int,
	powerAlreadyAbsorbed: one Int,
	type: one ChargingTypes
} {
	price >= 0
	timeUntilFree >= 0
	powerAlreadyAbsorbed >= 0
}

sig DSO {
```

```
            energy: one Int,
            price: one Int
} {
            price >= 0
            energy >= 0
}

sig CPMS {
            dso: some DSO,
            system: one SystemCPMS
}

sig EMSP {
            system: one SystemEMSP
}

sig Battery {
            status: one BatteryStatus,
            type: one BatteryType
}

sig Car {
            battery: one Battery
}

sig CarOwner extends User {
            car: one Car,
            account: one EMSP
}

// PREDICATES AND FACTS

// unique emails for every user
fact uniqueEmails {
    no disjoint u1 , u2 : User | u1.email = u2. email
}

// all emails connected to a user
fact allEmailConnected {
    all e: Email | one u: User | e = u.email
}

// unique back account for every user
fact uniqueBankAccounts {
    no disjoint u1 , u2 : User | u1.bankAccount = u2.bankAccount
}

// all bank accounts connected to a user
fact allBankAccountsConnected {
    all b: BankAccount | one u: User | b = u.bankAccount
}
```

```
// unique name for every user
fact uniqueNames {
   no disjoint u1 , u2 : User | u1.name = u2.name
}

// all names connected to a user
fact allNamesConnected {
   all n: Name | one u: User | n = u.name
}

// all passwords connected to a user
fact allPasswordsConnected {
   all p: Password | some u: User | p = u.password //all all, because there can be
equal passwords (verify)
}

//---

// unique CPMS account for every CPO
fact uniqueCpmsAccount {
   no disjoint c1 , c2 : CPO | c1.account = c2.account
}

// all CPMS are connected to one CPO
fact allCPMSConnected {
   all c: CPMS | one u: CPO | c = u.account
}

// ---

// unique location for every charging station
fact uniqueChargingStationLocation {
   no disjoint c1, c2 : ChargingStation | c1.location = c2.location
}

// all Locations connected to a Charging Station
fact allLocationsConnected {
   all l: Location | one c: ChargingStation | l = c.location
}

// ---

// unique set of charging stations for every CPO
fact uniqueChargingStationSetForCPO {
   no disjoint c1 , c2 : CPO | c1.station = c2.station
}

// all CPOs connected to charging stations
fact allCPOConnectedAllChargingStation{
   all c: CPO, s: ChargingStation | s in c.station iff s.owner = c
```

```
}

// ---

// all sockets connected one charging station
fact allSocketsConnectedOneChargingStation{
   all so: ChargingSocket, st: ChargingStation | so in st.sockets iff so.station = st
}

// unique set of charging sockets for every charging station
fact uniqueChargingSockets {
   no disjoint s1 , s2 : ChargingStation | s1.sockets = s2.sockets
}

// ---

// all BatteryStatus connected to a Battery
fact allBatteryStatusConnected {
   all s: BatteryStatus | one b: Battery | s = b.status
}

// all BatteryType connected to a Battery
fact allBatteryTypeConnected {
   all t: BatteryType | one b: Battery | t = b.type
}

// --

// unique car battery for every car
fact uniqueCarBattery {
   no disjoint c1 , c2 : Car | c1.battery = c2.battery
}

// unique car battery for every car and charging station
fact uniqueBatteryCarStation {
   no disjoint c: Car, s: ChargingStation | c.battery in s.batteries
}

// --

// unique car for every carowner
fact uniqueCarCarOwner{
   no disjoint u1 , u2 : CarOwner | u1.car = u2.car
}

// all Car are connected to one CarOwner
fact allCarCarOwnerConnected {
   all c: Car | one u: CarOwner | c = u.car
}
```

```
// unique EMSP account for every CarOwner
fact  {
   no disjoint c1 , c2 : CarOwner | c1.account = c2.account
}

// all EMSP are connected to one CarOwner
fact allEMSPConnected {
   all e: EMSP | one u: CarOwner | e = u.account
}

// --

// all EMSP interfaces connected to system EMSP
fact allSystemEMSPConnected {
   all e: EMSP | one s: SystemEMSP | s = e.system
}

// all CPMS interfaces connected to system CPMS
fact allSystemCPMSConnected {
   all c: CPMS | one s: SystemCPMS | s = c.system
}

// all ChargingStations connected to system EMSP
fact allSystemCPMSChargingStationConnected {
   all c: ChargingStation | one s: SystemCPMS | s = c.system
}

// system EMSP connected with system CPMS
fact allSystemConnected {
   all s1: SystemCPMS, s2: SystemEMSP | s1 = s2.connection iff s1.connection = s2
}

// --

// Dynamic Modeling

// add DSO in a CPMS
pred addDSO [cp,cpnew: CPMS, ds: DSO]{
   cpnew.dso = cp.dso + ds
}

//add Socket in Station
pred addSocket [cs,csnew: ChargingStation, s: ChargingSocket]{
   csnew.sockets = cs.sockets + s
}

//add Battery in Station
pred addBattery [cs,csnew: ChargingStation, b: Battery]{
   csnew.batteries = cs.batteries + b
}
```

```
//add Station in CPO
pred addStation [cp,cpnew: CPO, cs: ChargingStation]{
   cpnew.station = cp.station + cs
}

// --

// Examples of possible assertions

assert NoCarWithSameBattery {
   no disjoint c1, c2: Car | c1.battery = c2.battery
}

assert NoSameBankAccountForUsers {
   no disjoint u1 , u2 : User | u1.bankAccount = u2.bankAccount
}

//check NoCarWithSameBattery for 8
//check NoSameBankAccountForUsers for 8

// --

pred world1 {
   #CPO=1
   #ChargingStation=2
   #Battery=3
   #Car=1
   #ChargingSocket=3
}

pred world2 {
   #CPO=3
   #Car=1
   #EMSP=1
   #Battery=3
   #SystemEMSP=1
   #SystemCPMS=1
}

run world2 for 6
```

4.1.1 First Model

In the first model (represented in the code as “world1”) the main objective is to test-out the main relations and constraints that compose the system, they are:

- Each CPO needs at least one Charging Station
- Each Charging Stations needs at least one Charging Socket

- Each Car Owner implies the existence of an electric Car and, given that, the presence of a car Battery
- All CPOs and Car Owners are unique
- All Charging Sockets need to be connected to a Charging Station (being able to have more than one Socket per Station)

4.1.2 Second Model

The second model (represented in the code as “world2”) focuses on testing the system connections and data sharing inside the system, in order to make it possible to have the data allocated in the right spots to run the optimization algorithms and automated decisions in price. Given that, the main objectives were:

- Assure that every Car-Battery data is connected to the EMSP and the EMSP is connected to the System ESMP
- Assure that every CPO is connected to an CPMS and these to the System CPMS

4.1.3 Dynamic Model

Finally, a dynamic model was created in order to test-out possible events and actions that happen in the system

4.1.4 Resulting Worlds

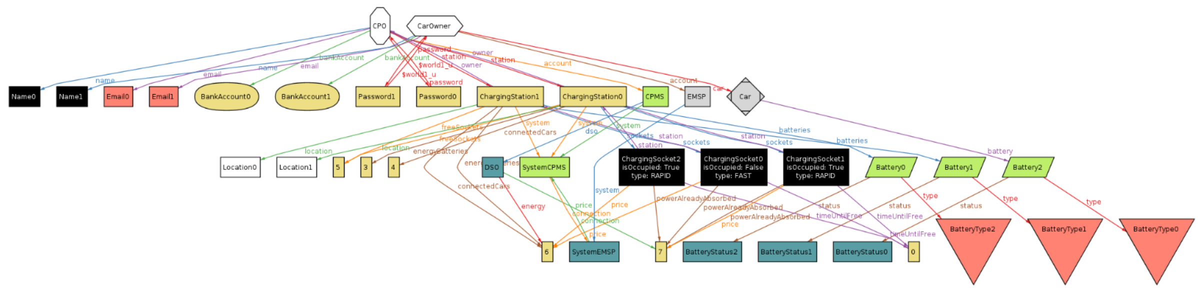

Figure 21: model one result

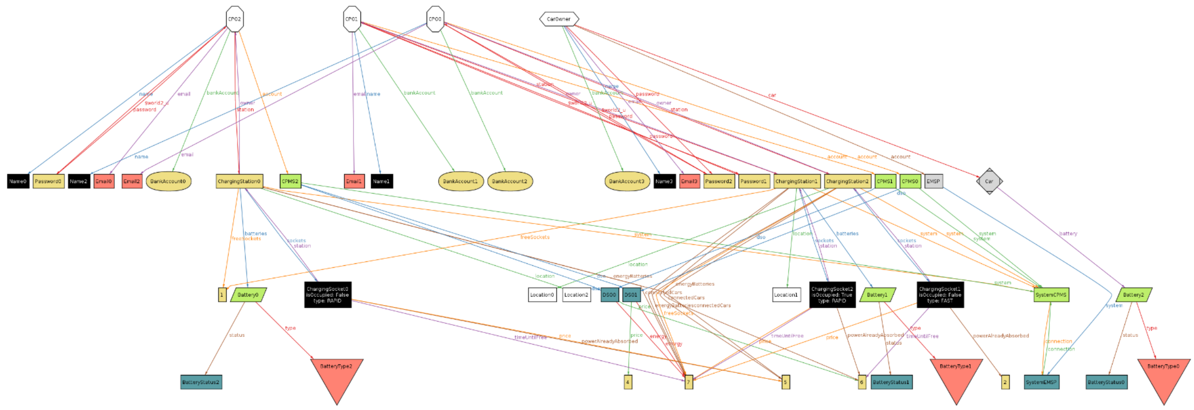

Figure 22: second model result

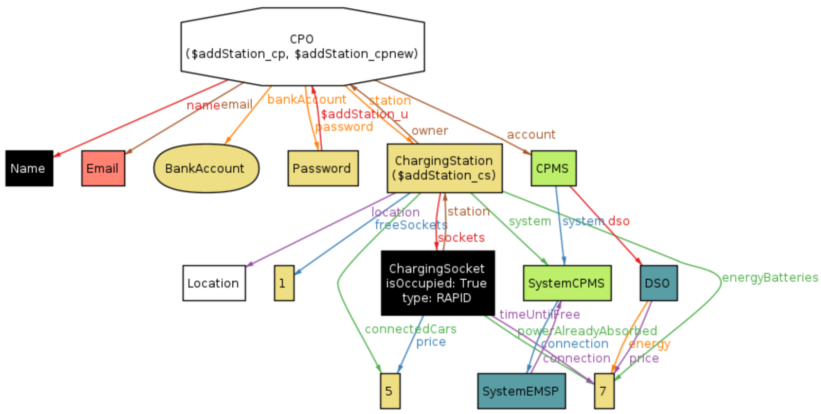

Figure 23: World obtained by running dynamic model (specifically: addStation)

5 Effort spent

1. Azank

Task	Time spent
Introduction	5 h
Overall description	4 h
Specific requirements	8 h
Formal analysis	15 h

Reasoning	10h
Total	44 h

2. Bagni

Task	Time spent
Introduction	3h
Overall description	4h
Specific requirements	11h
Formal analysis	18h
Reasoning	7h
Total	43 h

3. Wolff

Task	Time spent
Introduction	5 h
Overall description	15 h
Specific requirements	12 h
Formal analysis	4 h
Reasoning	7 h
Total	43 h