POLITECNICO
MILANO 1863

RASD

Requirement Analysis and Specification Document
a.a. 2022/2023

Version: 1.0
Date: 23/12/22

Eutizi Claudio
Person ID: 10812073
Student ID: 995635
email: claudio.eutizi@mail.polimi.it

Perego Gabriele
Person ID: 10488414
Student ID: 987104
email: gabriele2.perego@mail.polimi.it

Contents

1 Introduction

This *Requirements Analysis and Specification Document (RASD)* aims to provide an overview of the eMall project. The following document will help the reader to understand the purpose of the project i.e. in which environment the application operates and which services offers to its users. In particular way it will illustrate goals and how these may be reached, guaranteeing the meeting of certain functional and nonfunctional requirements.

1.1 Purpose

Billions of tons of CO2 are released into the atmosphere every year as a result of coal, oil, and gas production. Human activity is producing greenhouse gas emissions at a record high, with no signs of slowing down. While science tells us that climate change is irrefutable, it also tells us that it is not too late to stem the tide. This will require fundamental transformations in all aspects of society; one of the most debated ones undoubtedly regards the mobility and the increasingly widespread usage of electric vehicles. When it comes to climate change and air quality, electric vehicles are clearly preferable to petrol or diesel ones and the benefits will further increase going forward, as world will adopt more renewable energy sources in the future.

The **e-Mobility for All (eMall)** is a software application designed to support the charging process of electric vehicles. The eMall purpose can be summarized in two main aims:

- Allow drivers to charge easily, quickly and effectively their electric vehicle through out a dedicated user-friendly software interface called **eMobility Service Provider (eMSP)**. This system wants to be an intermediary between the end users and the Charging Point Operators (CPOs). The electric vehicle drivers will find in eMall all the information they need in order to carefully plan, monitor and manage the charging process of their electric vehicle, in such a way that it introduces minimal interference and constraints on their daily schedule.

- eMall wants to be also focused on supporting the Charging Point Operators (CPOs) that own and manage the charging stations. In particular, support is given via a subsystem called **Charging Point Management System (CPMS)**. Every CPO has its dedicated CPMS that administers its IT infrastructure and provide a simple access point to the CPOs operators that want to monitor the charging stations status, make decisions and apply changes. The CPMS handles also the acquisition of energy from the so-called Distribution System Operators (DSOs) and can automatically handle decisions in substitution to the manual intervention of the CPO human operator. In order to avoid the Operators to do too much repetitive operations, some of these can be also automatically performed by an autonomous system installed in the CPMS infrastructure. This autonomous system's implementation and functionalities are out of the scope of this RASD; it is, in fact, considered as an already existing and working subsystem that operates autonomously, but whose role is subordinate to the Operator's intervention, that has always an higher priority.

Finally, eMall guarantees an effective and reliable interaction between eMSP and CPMSs, providing full support to users on both sides.

1.1.1 Goals

In this section, we will go to extract the main goals of eMall. The goals indicated below are divided with respect to the eMall system section they belong to.

- [$G.1$] - Allow to identify Users, Drivers and Operators.
- [$G.2$] - Allow Users to visualize external information of charging stations (e.g. charge price, socket status, socket charging type etc.).
- [$G.3$] - Allow Users to visualize the position of the charging stations in a selected area.
- [$G.4$] - Allow Drivers to reserve a socket in a selected charging station for a certain time frame.
- [$G.5$] - Allow Drivers to control (start and stop) and monitor the charging process.
- [$G.6$] - Allow Drivers to pay for the charging service.
- [$G.7$] - Allow Operators to visualize information about the internal status of charging stations (e.g. station battery percentage, number of charging vehicles, amount of absorbed power etc.) and to intervene to implement changes in the charging stations' settings.
- [$G.8$] - Allow Operators to visualize information about DSOs and to choose the one from which to acquire energy.
- [$G.9$] - Allow Operators to set the price of a charge and set special offers in a selected charging station.

1.2 Scope

In this section we want to give a brief analysis of the machine, world and shared phenomena.
The Machine is the application software that we want to develop i.e. eMall.
The World is the external environment, namely the part of the real world that is affected by our system.
These two actors communicate and influence each other. World phenomena are events that take place in the real world and taken by themselves do not have a direct impact on the System.

Shared phenomena can be of two kinds: controlled by the Machine and observed by the World, or vice versa controlled by the World and observed by the Machine. In the tables shown below the phenomena are clearly separate in eMSP and CPMS phenomena.

1.2.1 World and Machine Table : eMSP

Phenomenon	**Controlled by**	**Shared**
A Driver wants to charge an electric vehicle	W	No
A Driver does not arrive in time to the reserved socket	W	No
The machine shows data about the status of a selected charging station	M	Yes
The machine reserves a socket of a charging station for a certain time frame	M	Yes
The machine sends a QR-Code ticket to be scanned by the driver at the charging point	M	Yes
The machine shows data about the charging process of the electric vehicle (e.g. battery percentage, price, kWatts, spent & remaining time)	M	Yes
The machine shows if the payment has been successfully completed or eventually any payment issue occurred	M	Yes
The machine notifies that the charging process is started or finished	M	Yes
A User opens the eMSP application	W	Yes
A User signs up to the eMSP application	W	Yes
A Driver logs in the eMSP application	W	Yes
A User selects a charging station for view data about it	W	Yes
A Driver books a socket in a selected charging station	W	Yes
A Driver scans the QR-Code to confirm he is arrived at the socket	W	Yes
A Driver connects/disconnects the electric vehicle to/from the reserved socket	W	Yes
A Driver starts or stops charging the electric vehicle	W	Yes
A Driver inserts data of payment	W	Yes

Table 1: World and Machine table: eMSP

1.2.2 World and Machine Table: CPMS

Phenomenon	Controlled by	Shared
A DSO changes the price of the energy sold	W	No
The batteries in a certain charging point are empty	W	No
There is a fault in the electrical line	W	No
The machine shows data about the "external" status of a selected charging station	M	Yes
The machine shows data about the "internal" status of a selected charging station	M	Yes
The machine shows data about the DSOs	M	Yes
A Operator sets price and offers of sockets of a charging point	W	Yes
A Operator visualizes the list of DSOs	W	Yes
A Operator selects the DSO from which to buy energy	W	Yes
A Operator sets the supply settings of a charging station i.e. energy from batteries, DSO or mixture	W	Yes

Table 2: World and Machine table: CPMS

1.3 Definitions and abbreviations

1.3.1 Acronyms

- **RASD**: Requirement Analysis and Specification Document.
- **eMSP**: e-Mobility Service Provider.
- **CPO**: Charging Point Operator.
- **CPMS**: Charging Point Management System.
- **DSO**: Distribution System Operator.
- **UML**: Unified Model Language.
- **API**: Application Programming Interface.
- **QR-Code**: Quick Response-Code.
- **EV**: Electric Vehicle.
- **OCPI**: Open Charge Point Interface.
- **CS**: Charging Station.
- **CP**: Charging Point.
- **FC**: Fiscal Code.

1.3.2 Definitions

- **User**: person or entity who wants to use the application and is not registered to it.
- **Driver**: User who owns an electric vehicle, can use the charging stations for charging purposes and is registered to the application.
- **Charging Point Operator (CPO)**: entity or organization that owns and manages all the EV charging infrastructure assets.
- **e-Mobility Service Providers(eMSP)**: eMall subsystem used by Drivers for their charging purposes.
- **Charge Point Management System (CPMS)**: eMall subsystem that manages the CPO IT Infrastructure. It is also an interface that allows Operators to handle technical and economic aspects of the charging stations they manage.
- **Operator**: human operator who works into the staff of a CPO and have access to the CPMS functionalities.
- **Distribution System Operator (DSO)**: entity responsible for the operation and management of electricity distribution networks, from which the CPOs can buy energy.
- **Charging Station**: station in which Drivers can charge their electric vehicles.
- **Charging Point**: structure of a charging station with sockets where the Drivers can connect their electric vehicle.
- **Charging Socket**: electric socket to be connected to the EVs. It can be of different kinds depending on the charging velocity they provide.
- **Fiscal code**: a 16 characters code used in Italy to uniquely identify a person.

1.3.3 Abbreviations

- $[G.n]$ = n-th goal.
- $[R.n]$ = n-th functional requirements.
- $[D.n]$ = n-th domain assumption.
- $[UC.n]$ = n-th use case.

1.4 Revision history

Version	Date	Details
1.0	23/12/22	RASD first deadline draft

Table 3: revision history

1.5 Reference Documents

Title	Authors	Links
R&DD Assignment AY 2022-2023	M. Camilli, E. Di Nitto, M. Rossi	
The World and the Machine	Michael Jackson	Online PDF
Alloy Official Documentation	MIT Software Design Group	Alloy documentation

Table 4: table of documents and references

- ISO/IEC/IEEE 29148:2018(E)
- StarUML for diagrams
- Moqups App for mockups

1.6 Document Structure

The rest of the document is organized as follow:

Introduction (section 1): this section is the introductory part of the document and specifies the purpose and the scope of the whole application, underlining the goals that have been set. Here we also give an introduction about the world in which the system will be collocated, highlighting the boundaries and the interactions between the machine and the world.

Overall Description (section 2): This section gives an overall description of the eMall functionalities offered by its two subsystems eMSP and CPMS, underlining the main goals that have been set. More in details, this section provides scenarios and class diagrams in order to give a general view of the application structure and some statecharts to explain the evolution of some crucial domain parts.

Specific Requirements (section 3): contains an in-depth description and explanation of the system that we want to develop. hardware and software interfaces the user can exploit in order to use the eMall application are described. Then the functional and nonfunctional requirements, associated with use cases analysis and domain assumptions, define the interaction between the users and the System.

Formal analysis using Alloy (section 4): This section exploits Alloy in order to generate a formal model of some critical parts of the domain. Some images of significant instances of this model are provided.

Effort Spent (section 5): This section includes information about the number of hours each group member has worked for this document.

2 Overall description

2.1 Product perspective

To better understand the peculiarities of the application to be designed, it is important to detail the domain for which it is intended. In this chapter a detailed analysis of the scenarios and a visual representation of the domain model will help to achieve the scope.

2.1.1 Scenarios

Scenario 1: Charge Leklerk needs to charge his EV which has a low percentage of battery.
Mr.Leklerk decides to try out a new application eMall, that his friend Carlos Jr. suggest to him. Mr.Leklerk downloads the eMall application on his smartphone, opens the app and, from the homepage, selects the Login icon. He has not an account already, so he decides to create one. He is required to insert: name, surname, phone number, email address, fiscal code and a valid password; he proceeds by inserting his personal data and confirms the agreement of eMall terms and conditions. After successfully completed the registration phase, Mr.Leklerk looks for the nearest CSs to his position on the map, selects one of them, visualizes which sockets are available and book one.

Booking is successful, Mr.Leklerk receives the QR-code ticket and visualizes on the screen that he has 15 minutes to reach the reserved socket, to connect the socket and his EV and to scan the QR-code (mandatory in this order) in order not to loose the reservation. After 10 minutes Mr.Leklerk arrives at the socket, connects the socket cable to his EV and successfully scans the QR-code. At this point, Mr.Leklerk visualizes from the screen that he must enter payment data, within 5 minutes, to proceed with the charging of the vehicle, otherwise he will loose the reservation.

In 1 minute of time, without any special problems, Mr.Leklerk enters the data of his prepaid card, saves these data directly and proceeds with the confirmation.

Once the payment data is successfully entered, the system previously freezes €100 from Mr. Leklerk payment method and then the socket starts charging the vehicle. Mr.Leklerk visualizes on his screen all the progress related to the charging process in real time. In particular, he visualizes: the battery status of the EV, the kWatts used, the money for kWatts used, the initial time of the charging process and and the estimated final time to reach 100 % vehicle charge.

Mr.Leklerk then decides to go for a coffee in a nearby bar waiting for his electric vehicle to completely charge. After about 2 hours, Mr.Leklerk receives a notification from the application: the maximum charging capacity has been reached and the charging process is finished.

Once the charge has been successfully completed, the system withdraws the correspondent total charge price from the previously freezed €100. Mr.Leklerk then goes out the bar, reaches the vehicle, safely detaches the cable and goes away, on a new adventure.

Scenario 2: It will be not a good day for Max Van Der Stappen.
After a busy work day, Max leaves the office and after entering the car he realizes that it

is at 15% battery. Without further ado, Max opens the eMall app, logs in as Driver and looks for a CS with available sockets.

Max successfully books a free socket, but the traffic of the city does not allow him to reach it within 15 minutes, so Max loses the reservation. Then, he tries to book the same socket again but notices that its prices are changed and have increased. Then he decides to book another socket of a nearby CS.

In less than 15 minutes, Max arrives in time to the socket booked, connects the vehicle to the socket and successfully scans the new QR-code received.

In a short time, he enters the payment data correctly, selecting the data of a card already saved, and starts the charging process. Max decides to go for a ride in the sports store next to the CS while the vehicle charges. However, after 1 hour he realizes that evening he has the anniversary dinner with his girlfriend and he is late for it. Then, he leaves the store and through the application stops the charging process in advance, visualizing that the battery of the vehicle has reached 80 %.

After that, he disconnects the vehicle to the socket and goes home.

Scenario 3: Riccardo Bensoni's working day at "The Rooster CPO" headquarter will be very demanding.

The IT system that allows the automatic management of the charging stations is down, so some delicate operations need to be handled manually. Firstly, Mr. Bensoni logs in the eMall application using his desktop PC from his office entering his personal business email address, Identification Number and Password and clicks "login".

The first task the company assigned to him is to change another DSO from which to acquire energy due to an unjustified increase of the current DSO price.

Mr. Bensoni, who is already logged in, goes in the "myDSO" section and selects "List of DSO". Ordering the list by price, he selects the "Energy4All" that is not the cheapest one (still cheaper than the current one), but has a high percentage of energy produced with renewable sources. This choice has been made because of internal CPO policy that oblige to choose DSO that provide renewable energy percentage above the 70After some seconds, the DSO confirms the deal and the modification is successfully registered.

The second thing he has been tasked to do is to modify the price of the "FAST" sockets of the charging station in the Lambrate neighborhood, adding a 10% discount in order to encourage the Drivers to use those sockets.

Mr. Bensoni enters the neighborhood in the map view and, for each CS, he adds the offer.

The third task is to modify the supply settings of the Via Golgi 42 charging station. This is because the batteries are about to empty and the current supply settings are set with 70% of energy from batteries and 30% from the DSO line; so it is necessary to set the supply settings to 100% from DSO.

Mr.Bensoni then clicks on "my Charging Stations", inserts the charging station address in the search bar and selects it.

In the CS menu, he selects "Supply Settings", where he finds an overview of the charging station energy providing settings. Then he selects "Line Only" and submits.

2.1.2 Class Diagrams

Below it is shown a high level class diagram of the whole application. There are a few details for which may be necessary a more in-depth analysis. The **User** class is a superclass that indicates a generic user that wants to use the application. They are allowed to do some operations i.e. registration as a Driver, the visualization of the nearest charging stations (based on Coordinates) and their the external status. The classes *Operator* and *Driver* extend the class User and represent the other two main actors of the system.

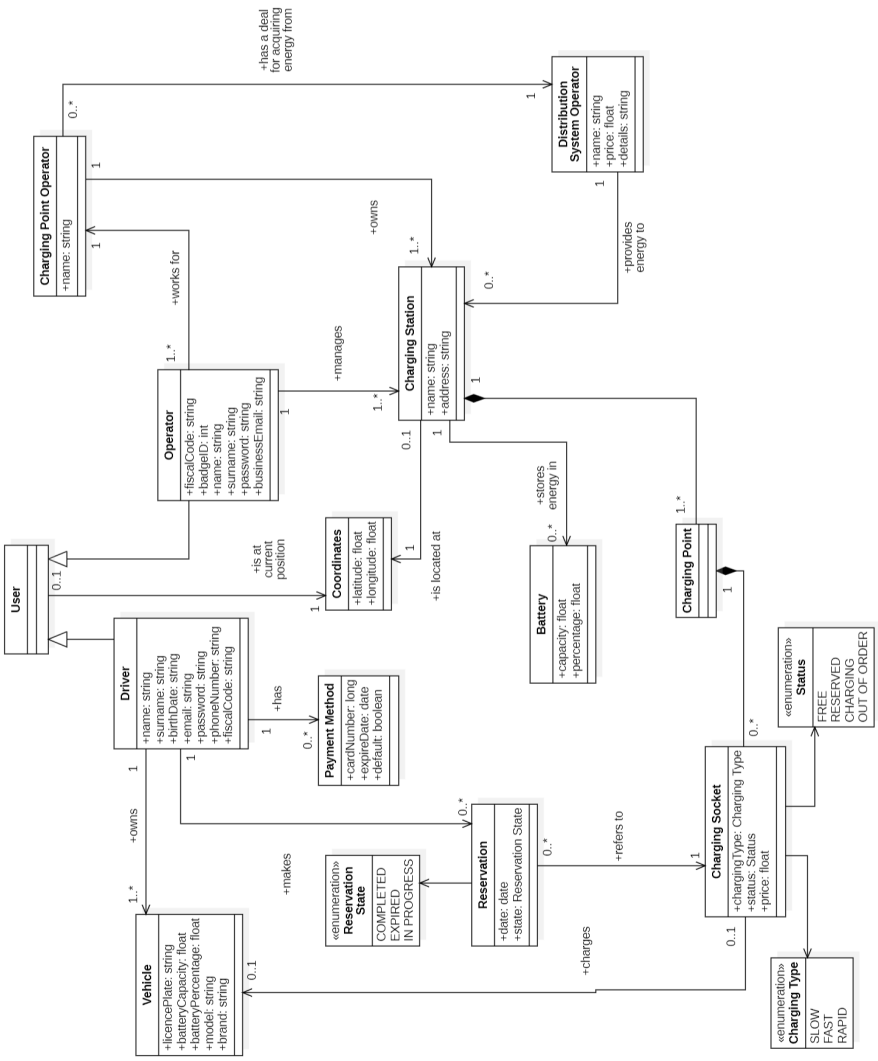

Figure 1: Main eMall Class Diagram

The next diagrams show in detail the distinction of the two application's subsystems, eMSP and CPMS, and the interaction between them.

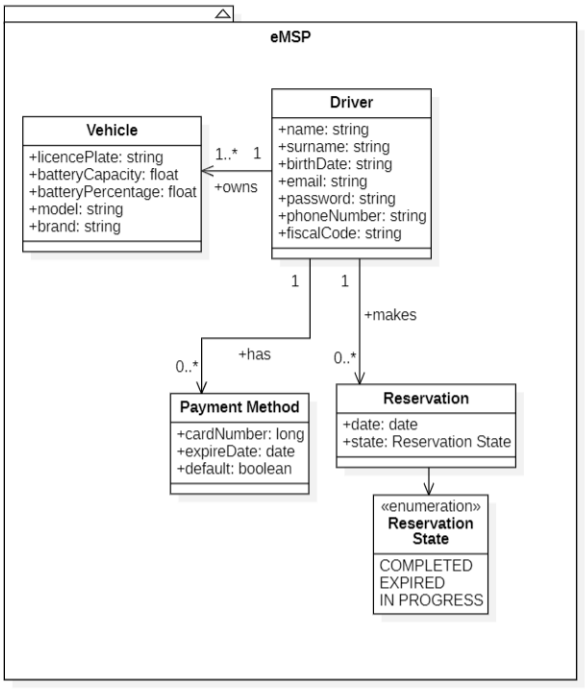

Figure 2: eMSP module Class Diagram

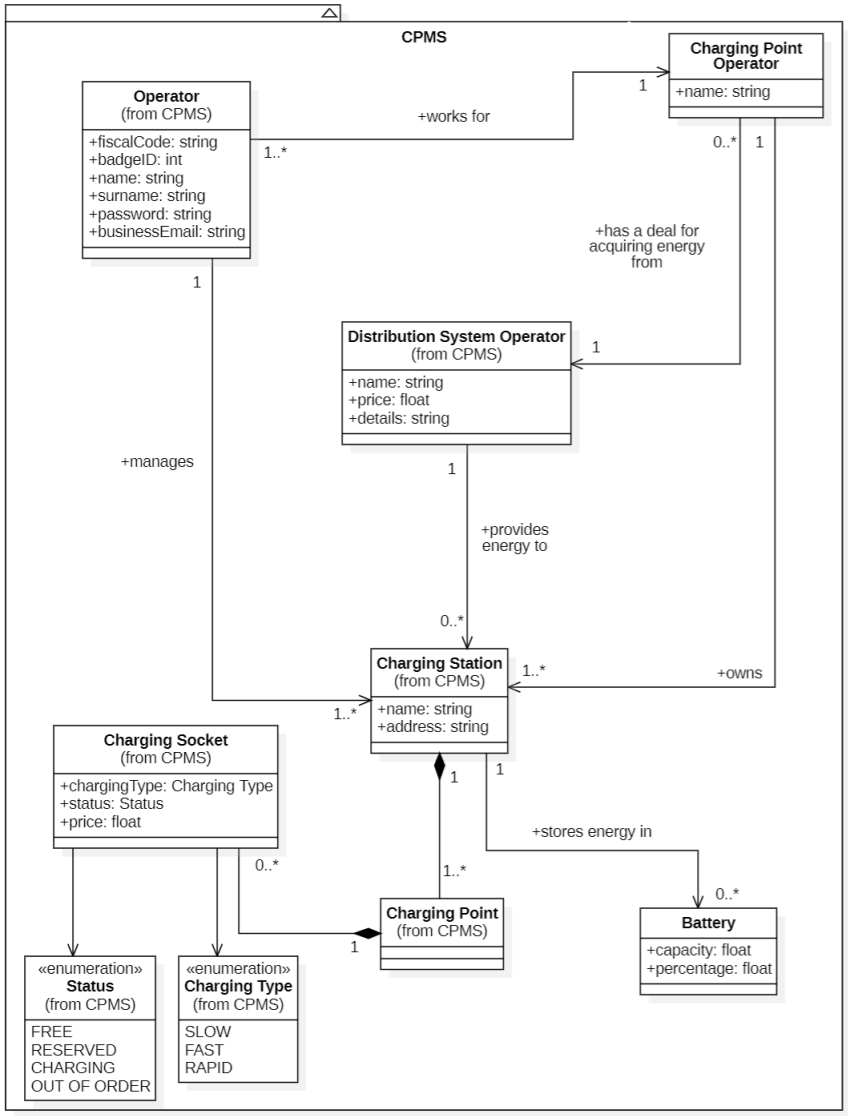

Figure 3: CPMS Class Diagram

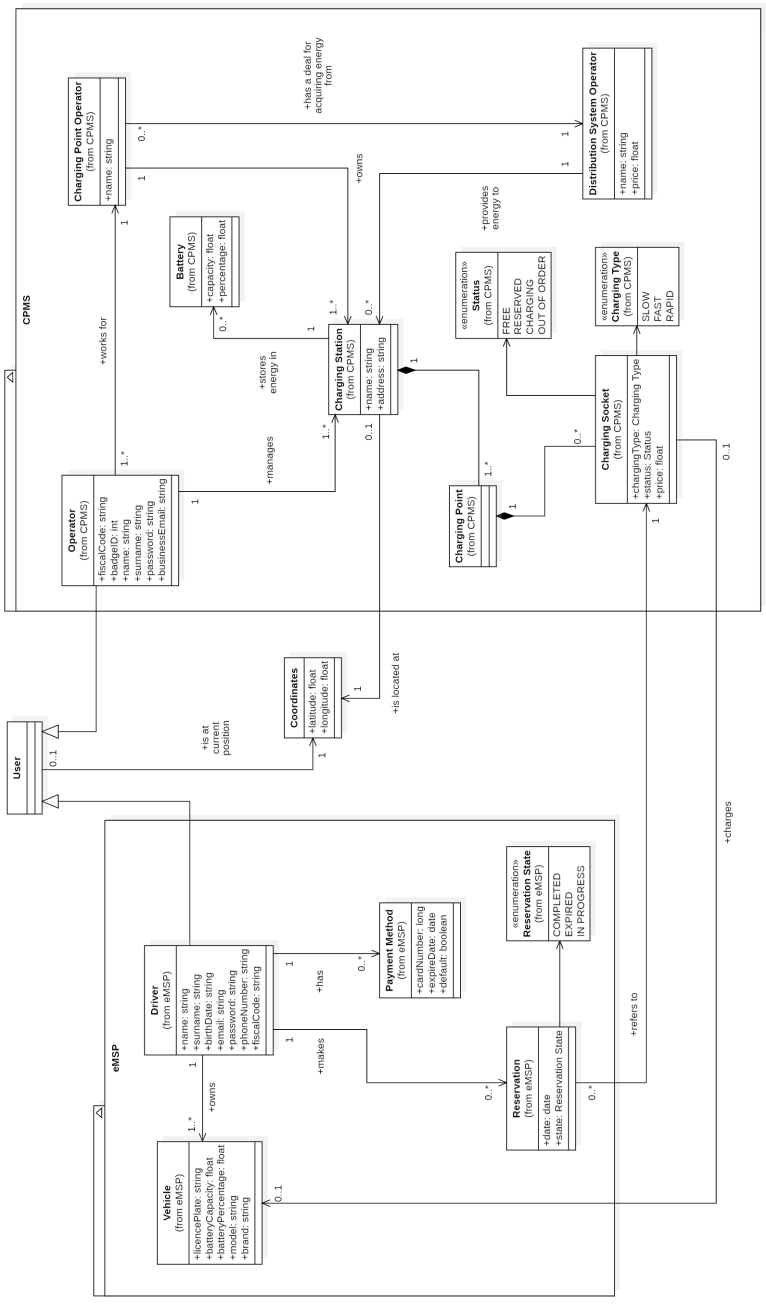

Figure 4: Module-oriented class diagram

In this last module-oriented class diagram can be noticed how the eMSP subsystem can interact with multiple CPMS related to different CPOs. Different Drivers, using the

eMSP functionalities, are able to localize charging stations managed by different CPMSs and to reserve their sockets for the charge of their owned EVs.

2.1.3 Statecharts

The statechart diagrams shown below are used to describe the main sequences of events that the system handles in its common scenarios.

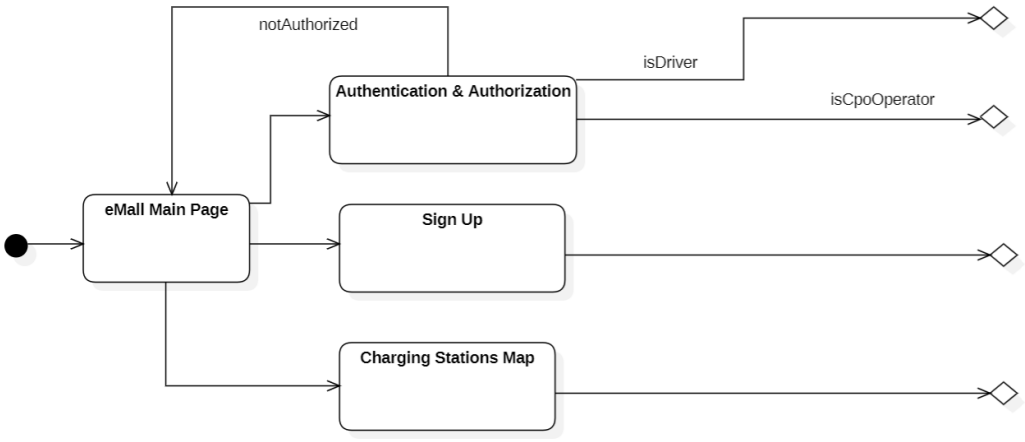

Figure 5: Main Statechart

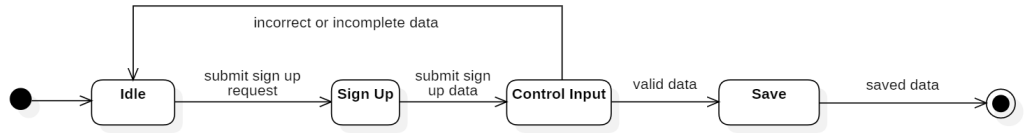

Figure 6: New Driver Registration Statechart

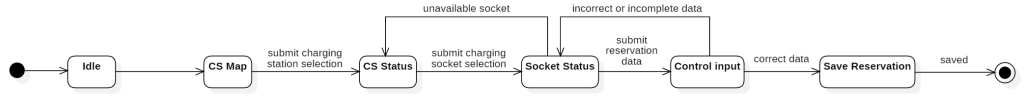

Figure 7: Socket Reservation Statechart

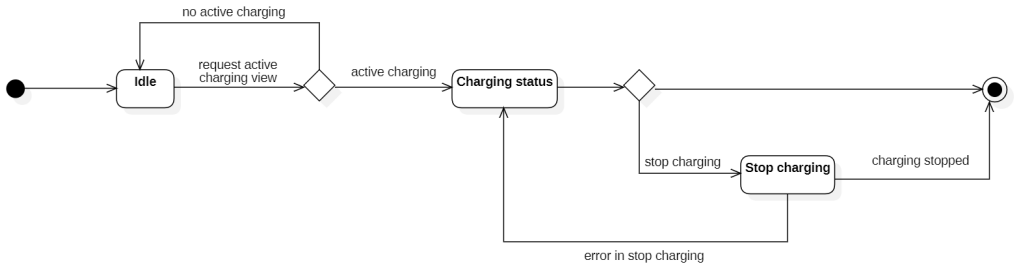

Figure 8: Active Charging Status View Statechart

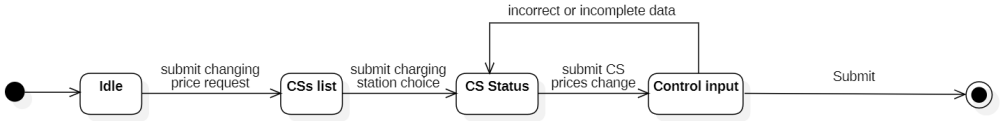

Figure 9: Changing Price And Offers Of A Charging Station Statechart

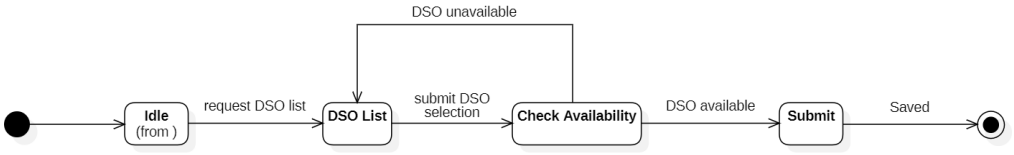

Figure 10: Choosing DSO Statechart

2.2 Product functions

In this section we will present, in a descriptive way, the main functionalities of the eMall application. Starting from the involved scenarios, the most important requirements will be extracted with the aim of conducting a more precise and formal analysis of them in the next chapter.

2.2.1 Generic functionalities

There are some functionalities of eMall that can be performed by all the Users.

- **Registration and Login**
  In order to use the eMSP booking and charging functionalities, Users must sign up in the application providing name, surname, email address, FC and a valid password. In addition, Users have to agree on Terms and Conditions of Use, that includes for them an agreement on privacy and data collection.
  Drivers, who are already registered, may want to sign in the application. In this case they have to insert email address and the password they have used for the sign up operation.
  Operators that work for the CPO and need to use the CPMS functionalities are required to provide: business email address (provided by their CPO), a valid password and their identification number to log in. A dedicated section of the application will appear and they will be able to exploit the CPMS-dedicated functionalities. Due to security reasons, new CPO Operators will be provided with the login credentials by the CPO organization itself, so they do not have to sign up arbitrarily. They will be recognized directly by the CPMS.

- **Visualization of the charging stations on the map**
  Each kind of User, only if they authorize the access, can be traced by GPS. In this way they can visualize in a map the presence of close CSs to their position or in a selected area (in this second case, the GPS authorization is not strictly required).

- **Visualization of "external" status of a Charging Point**
  Each kind of User can select a certain CS on the map and visualize its "external" data i.e. location, number of CPs, available and occupied charging sockets for each CP, the estimated time when the occupied sockets will be freed, type of each socket (slow/fast/rapid) and cost of each socket.

2.2.2 eMSP functionalities

We will now explain the eMSP functionalities that Drivers can use.

- **Book a socket**
  The Driver can reserve in advance a socket of a CP. After booking, Driver will receive a QR-code ticket that must be scanned by the charging point QR-code reader to confirm the Driver's presence. The Driver has therefore 15 minutes, as per OCPI protocol, to arrive at the booked socket with is EV and scan the QR-code ticket at the appropriate CP. For those 15 minutes the socket is reserved. If the Driver does not show up within this time frame, the socket will be freed for other reservations.

- **Payment and start of the charging process**
  In order to launch the charging process, the Driver must connect the EV to the socket and provide one valid method of payment in the application. After correctly scanning the QR-Code ticket, the reserved socket is unlocked. Now, the Driver have must connect the EV to the CP, with the socket's cable, and must insert a valid payment method among those already added or to add another one. Meanwhile, a 5 minutes timer starts. If the Driver does not have a valid payment method yet, he must insert the data about a valid one before the timer expires in order not to lose the reservation. The system will then reserve for the charge an amount of money $x \leq$ €100 and $x \leq$ maximum credit available in the account related to the selected payment method. If the transaction is successful, the charging process will automatically start. In case of payment issues i.e invalid data inserted or no money available in the payment method, the Driver can retry entering again the payment data or choosing another payment method within the five minutes; period of time after which, in case of transaction failure, the reservation expires. The reservation expires also if the Driver does not connect the EV to the CP with the socket.

- **Saving of credit cards**
  The application shall also allow the Driver to save the payment data he prefers, in order to be faster and more effective during the payment phase. In this way, during the payment phase, the Driver will only have to select a payment method already saved to start the charging process in a few seconds.

- **Visualization of data during the charging process**
  During the charging process, the Driver can see real-time data about it, such as: kWatts used for charge the EV, price per kWatts used, time spent since the start of the charging process, remaining time to reach 100 % charge.

- **Stop of the charging process**
  The Driver has the possibility to stop in advance the charging process via a button on the application. Otherwise, the charging process stops automatically when the EV reaches 100 % battery.

2.2.3 CPMS functionalities

We will now explain the CPMS functionalities that Operators can use.

- **Visualize the internal status of a charging station**
  The Operators can visualize the "internal" status of a selected charging station, such as: the amount of energy available in its batteries (if batteries are presents), the number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the charge.
- **Data visualization and selection of DSO**
  The Operators can visualize: the list of available DSOs, from where they can select a new DSOs from which buy energy; the list of DSOs from which they already buy energy and deselect them.
- **Setting of price and offers**
  The Operators can modify the prices of a selected Charging Station and they can set special offers according to the market needs.
- **Decide the supply settings**
  If batteries are present in a CS, the Operators can decide how to get energy for charging the EV, such as: get energy from station batteries, get energy from DSOs, or a mix of these two modalities.

2.3 User characteristics

- **User**: person who: can download the eMall application, is not registered to it and can only benefit from the Generic functionalities.
- **Driver**: User who owns an electric vehicle, can use the charging stations for charging purposes and is registered to the application. Drivers can benefit from Generic functionalities and also from the so-called eMSP functionalities.
- **Operator**: person who works for a CPO and can make some business choices through dedicated functionalities. Operators can benefit from Generic functionalities and also from the so-called CPMS functionalities.

2.4 Assumption, dependencies and constraints

2.4.1 Domain assumptions

Domain assumptions are descriptive assertions assumed to hold in the world.

- $[D.1]$ = Drivers and Operators have different access to the functionalities of eMall i.e. eMSP and CPMS.
- $[D.2]$ = The data (FC, email, etc...) provided during registration is truthful and belongs to the person who creates the account.
- $[D.3]$ = User's device have to be connected to the Internet to guarantee the proper functioning of the application.
- $[D.4]$ = The Drivers behave civilly with respect to CPOs' infrastructures they use.
- $[D.5]$ = When a Driver arrives at the reserved socket, it is found free.
- $[D.6]$ = The Driver remains in the reserved seat only for the time needed for the charge.
- $[D.7]$ = For safety reasons, the cable that connects the EV to the socket remains locked for the entire process of charge.
- $[D.8]$ = Each Operator has his own list of CSs to manage.
- $[D.9]$ = Operators' manual decisions do not conflict with automatic CPMS decisions.
- $[D.10]$ = The interaction between the various providers (eMSPs, CPMSs, and DSOs) occurs through uniform APIs.
- $[D.11]$ = There exist uniform API for accurate GPS localization of Users and CSs.
- $[D.12]$ = There exist uniform API that manages the payments of the charges.

2.4.2 Constraints

- Each Driver must create only one account.
- Driver can not have more than one active reservation at a time.
- A Socket can not be reserved by two different Drivers for the same timeframe.
- When a QR-Code ticket is generated, the system doesn't allow to generate another ticket with the same QR-Code.
- Each CS must have one DSO which supplies electricity to it.
- A CS is managed by no more that one Operator.
- The Operator decisions about the CSs management have the priority with respect to the automatic CPMS system decisions.
- If a Socket is out of order, the Driver can not reserve it.

3 Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

In this section, we are going to show the user interfaces of eMall application. The eMall application is available for mobile and tablet, so as to be suitable everywhere without excessive limits. In addition, a PC version of eMall is only available for Operators to better manage CPMS functionalities from their workplace. Graphical interfaces shall have a simple structure and shall be easy to use by all type of Users. The interfaces between the system and Users are presented using mockups. Since some functionalities are common to all the Users, others are just for one of the two categories: Drivers and Operators.
For simplicity, we will show only mockups for mobile application.

- **eMall Logo**

Figure 11: eMall Logo

The eMall Logo (fig. 11) is composed by: the name of the application "eMall", the slogan "e-mobility for all" and finally a green and light-blue car with simple shapes with a lightning bolt inside.
This simple image fully captures the purpose of the service: offer a simple, fast and effective service for the charging and managing of electric vehicles.

- **General homepage: visualization of CSs on the map**

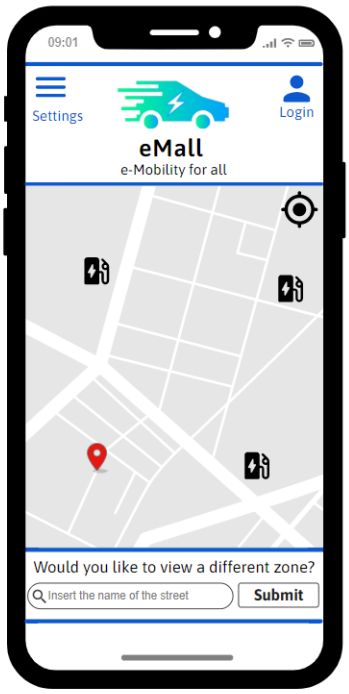

Figure 12: Visualization of CSs on a map

The first screen that a User sees as soon as he enters eMall is the map where to search for CSs (fig. 12).
It is composed by: a white background; the eMall logo on the top-center; the login icon on the top-right; the settings menu on the top-left; the map at the center, which shows the actual position of the User and the CSs closer to him; the search bar at the bottom-center with the button "Submit" beside. The search bar gives the possibility to the User to select a different zone and look for specific CSs in selected area.

- **Visualization of the external status of the CSs selected**

Figure 13: a. Free Socket, b. Not Free Socket

The User can select a CS on the previous map for visualize its "external" status in that moment.

From top to the bottom the User can see: a grey background; a exit icon at the top-left used for return to the map; the icon of a CS with its address position, serial number and the date; the bar of the number of the CPs present in the CS from which the User can select one; the bar of number of sockets present in the selected CP, from which the User can select a particular socket; the data about the selected socket, such as the code, the type, if it is free or not, the current price, the booked-time if it was not free and the special offers if presents.
In particular way: fig. 13.a shows the interface of a free socket; fig. 13.b shows the interface of a not free socket with the time to wait until it gets freed.

- **Registration and Login**

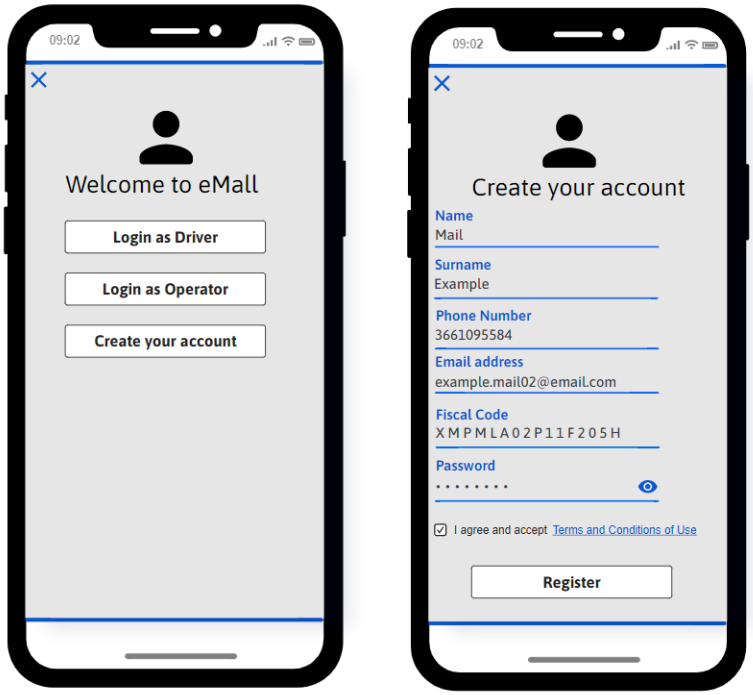

Figure 14: eMall Registration: a) Selection interface, b) Create an account

By clicking on the login icon, the User will see (fig. 14.a) three buttons to enter the application as Driver or Operator, and finally a button to register.
The buttons have the following functionalities:

- **Create your account** (fig. 14.b): for the sign up in the eMall application, a User has to provide: name, surname, phone number, email address, FC and a valid password. the User has also to accept the "Terms and Condition of Use".
- **Login as Driver** (fig. 15.a): the Driver, User already registered, has to insert email address, password and press "Login" button to log in the application. A Driver has also the possibility to recover the password if he forgot it.
- **Login as Operator** (fig. 15.b): a Operator has to insert: business email address, his identification code, password and press "Login" button to log in the application and use the CPMS functionalities.
  A operator has also the possibility to recover the password if he forgot it.

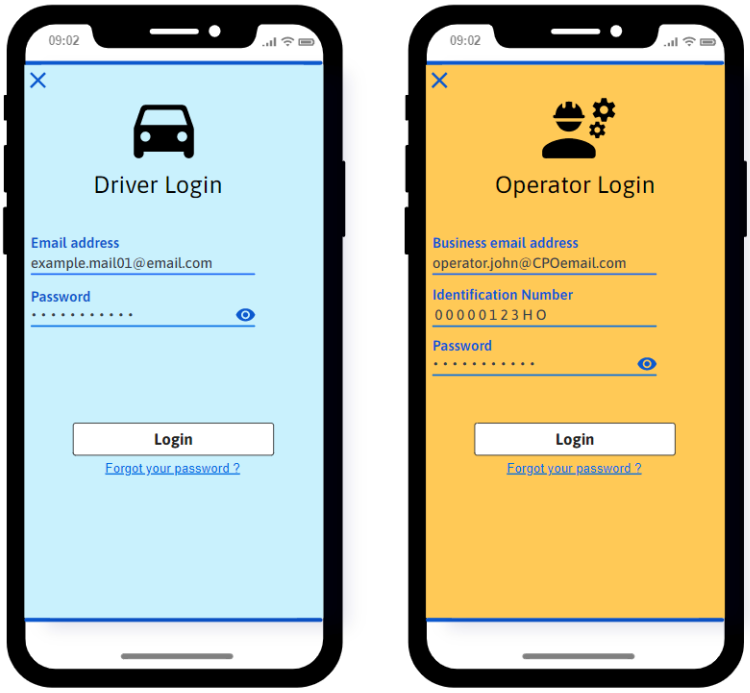

Figure 15: eMall Login: a) Driver, b) Operator

- **Driver homepage**

  When a Driver logs in the application, he can see an interface (fig. 16.a) similar to the main interface (**??**) but with: a light blue background; logout icon at the top-right; the CSs on the map colored by blue and a options menu eMSP at the top-left. By clicking on the eMSP menu, the Driver can visualize some functionalities (fig. 16.b).
  We describe now, in a qualitative way, these options:

  - **Profile**: the Driver can visualize details about his profile, such as: name, surname, email address etc.
  - **Active Charging**: the Driver can visualize, if are in progress, data about the charging process.
  - **Reservations**: the Driver can visualize the booking done before, or in progress, and their details.
  - **Payment Options**: selecting this, the Driver can visualize the payments done previously and can save a new payment method.
  - **Settings**: the Driver can visualize and modify some settings of the application, such as: language, change password etc.

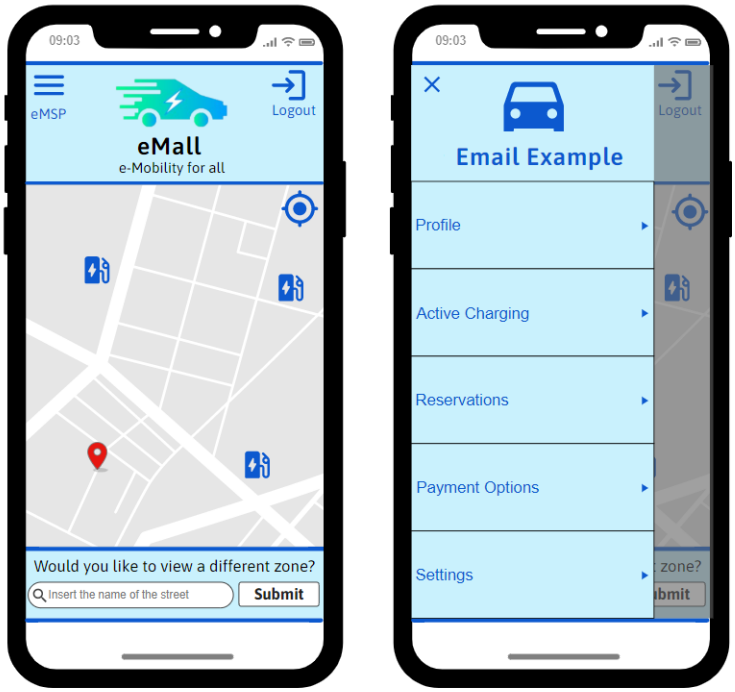

Figure 16: a) Driver principal page, b) Driver options menu

- **Operator homepage**

  When a Operator logs in the application, he can see an interface (fig. 17.a) similar to the main interface (fig. 12) but with: an orange background; the logout icon at the top-right; the CSs on the map colored by blue and a options menu CPMS at the top-left. By clicking on the CPMS menu, the Operator can visualize some functionalities (fig. 17.b).
  We describe now, in a qualitative way, these options:

  - **Profile**: the Operator can visualize details about his profile, such as: name, surname, business email address etc.
  - **myChargingStations**: The Operator can enter the main menu that allows him to have access to CPMS functionalities (i.e. view the internal status of the CPs, set price and offers etc...).
  - **CPO**: the Operator can visualize data about the CPO for which works.
  - **DSO**: the Operator can visualize the DSO from which the CPO buy electricity and change that DSO with another one via a list of DSOs.
  - **Settings**: the Operator can visualize and modify some settings of the application, such as: language, change password etc.

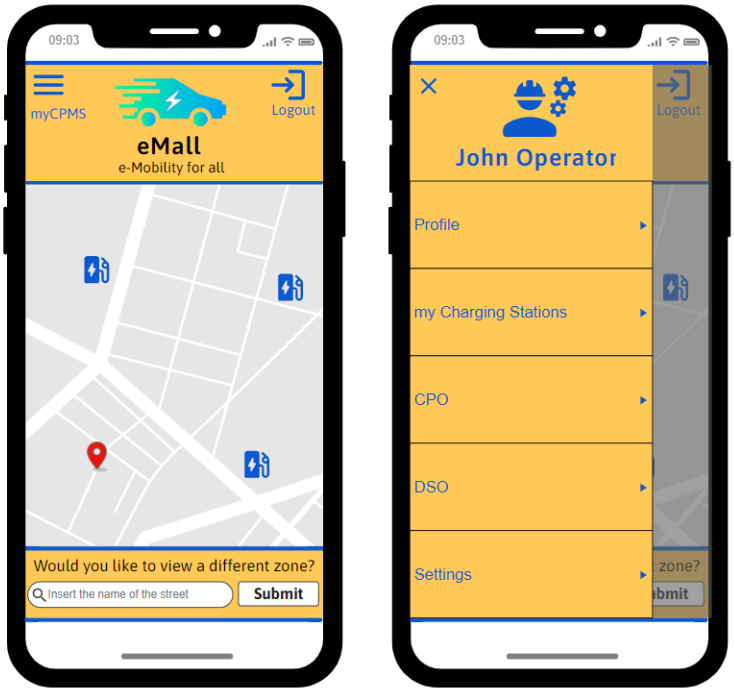

Figure 17: a) Operator principal page, b) Operator options menu

- **Booking a socket**

We qualitatively describe here the user interface for a socket reservation in the eMall application:

- selecting a CS, the Driver can view its "external" status (fig. 18.a) and, if the socket is free, book the socket through the button "Book the socket".
- Then, the Driver has 15 minutes to scan the QR-Code to the Charging Point. The application shows to the Driver (fig. 18.b) from the top to the bottom: data about the CS, a timer, the data about the reservation and a button showing the QR-code ticket to be scanned. (fig. 18.d).
- If the Driver does not scan the QR-Code before the timer expires, a notification will appear.(fig. 18.c). This means that the time is expired and the Driver has lost the reservation.

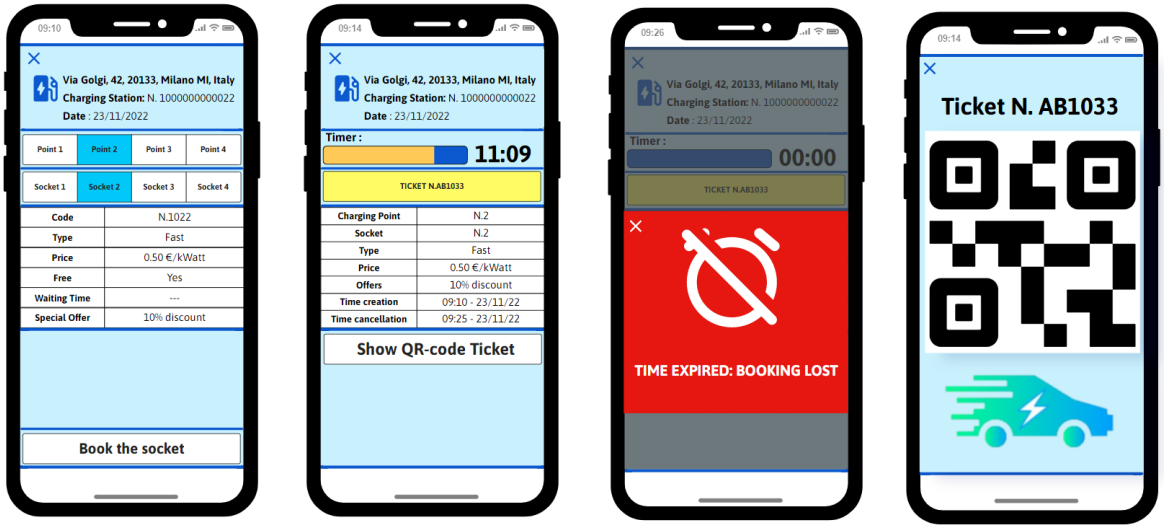

Figure 18: a) External status, b) Booking Status, c) Booking Lost, d) QR-code ticket

Figure 19: a) Past reservations with a ticket, b) Past reservations with no ticket

The Driver can visualize all the reservation selecting "eMSP" and after "Reservations". The Driver will visualize all the "ended reservations" of the bookings made in the past. These bookings can be: green if successfully made and red if not happened, with all the details about.

Below, the Driver can visualize:

- **Open Ticket** (fig. 19.a): in this case clicking on it the Driver will visualize the state of the booking (fig. 19.b).
- **No reserved socket**: so no ticket available (fig. 19.b). In this case, the Driver has the possibility to book one socket clicking on the link "Book a socket".

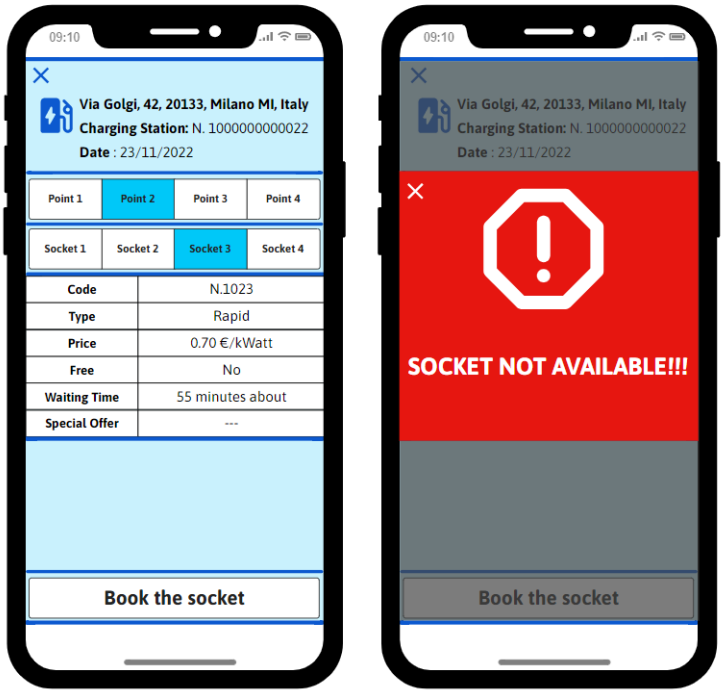

Figure 20: a) External status, b) Booking no available

Finally, the Driver will visualize the message "Socket not available" (fig. 20.b) in cases where:

- The Driver tries to book an already reserved or occupied socket (fig. 20.a).
- Driver tries to book an out of order socket.

- **Payment data and start of the charging process**

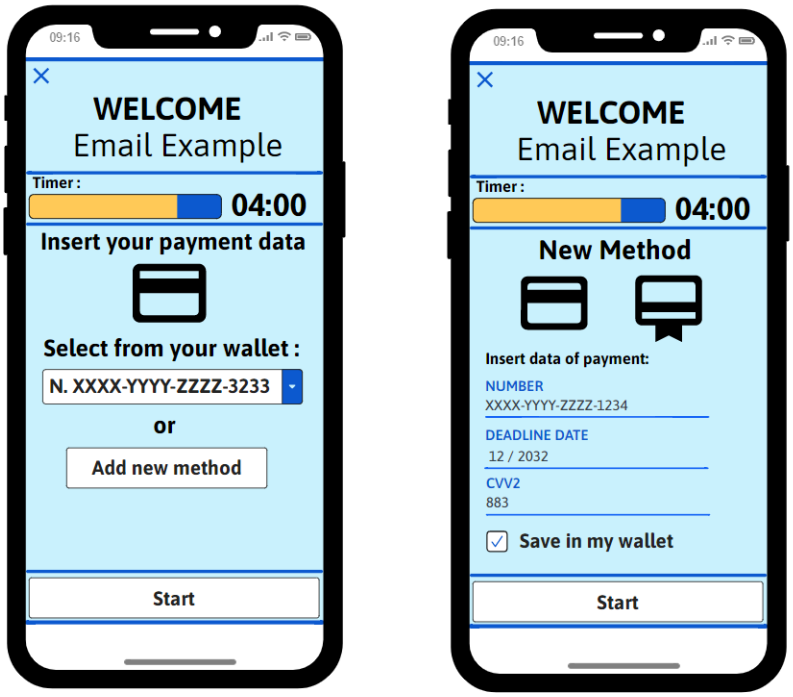

Figure 21: a) Payment Interface, b) Change method interface

After the correct QR-code scan, the Driver have to insert payment data, in less than 5 minutes, for start the charging process. In particular,the Driver can visualize:

- **Payment interface** (fig. 21.a): the Driver can: choose a credit card previously saved by the wallet, or change payment method clicking on the button "Add new method".
- **New Method** (fig. 21.b): in this page, the Driver can insert payment data of another credit card o similar, and can save them in the wallet through the appropriate box "Save in my wallet".

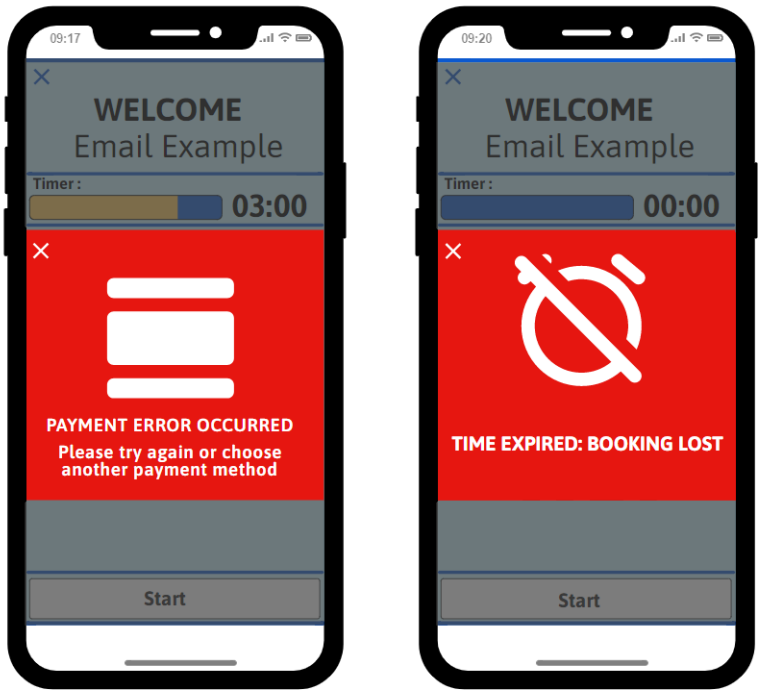

Figure 22: a) Incorrect payment data, b) Payment time expired

After entering the payment data, the Driver have only to click on the "Start" button to start the charging process.
The Driver can visualize also some different notifications of error during the payment process. For example:

- **Incorrect payment data inserted** (fig. 22.a): if the data of payment are not correct, a notification will appear to point it out.
- **Payment time expired** (fig. 22.b): if the Driver does not insert payment data in time, a notification will appear to point out that the reservation is lost.

- **Visualization of charging process**

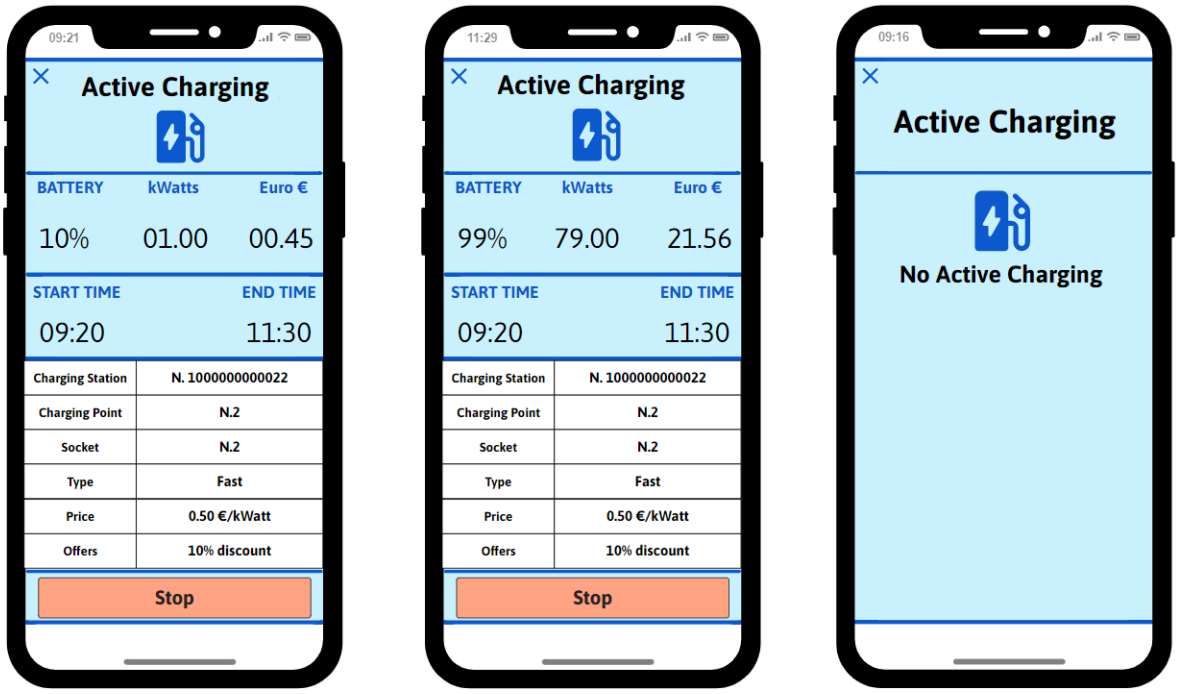

Figure 23: a) Beginning of the charging, b) End of the charging, c) No Active Charging

If the payment data is correct, the charging process will automatically start, and the Driver will be able to visualize in real time the progress of the charging process. In particular, clicking on "eMSP" menu and after on "Active Charging", the Driver will visualize the charging status of the process if it is in place, otherwise the Driver will visualize (fig. 23.c) that there are **No Active Charging** in place. Talking about the charging process in place, we have:

- **Start of the charging process** (fig. 23.a): the figure shows the initial status of the charging process. It shows: the current battery percentage of the EV, the kWatts inserted, the money per kWatt, the "start time" of the process and the supposed "end time" of the process and the data about the reservation did. The "end time" indicates the time in which the battery will be at 100%.
- **End of charging process**(fig. 23.b): the figure the final stages of the charging process.

The Driver is informed about the end of the charging process by a notification (??.c).

The stop of a charge occurs if: he stops the process in advance clicking on the "Stop" Button, or the EV charge gets 100% of battery. Closing the notification (clicking on the "X" icon in top left) the Driver can visualize all the details of the happened charging process:

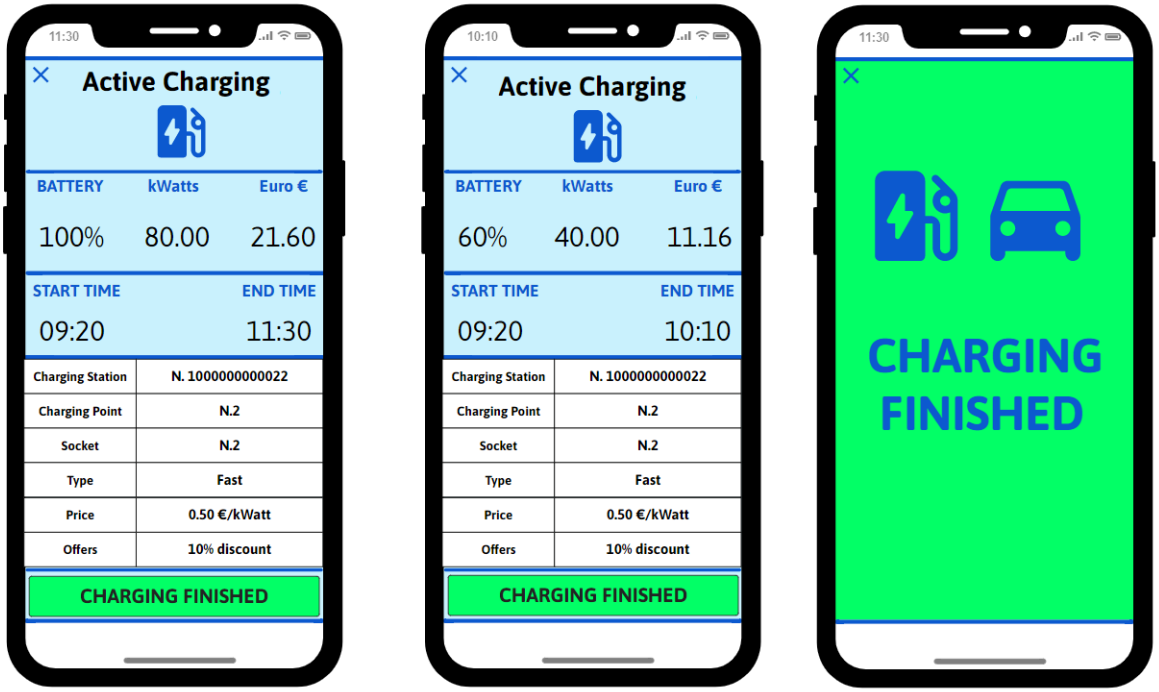

Figure 24: a) 100% of battery reached, b) Process stopped in advance, c) End of charge notification

- **Full charge of battery** (fig. 24.a): the figure shows the end of the process when the battery arrives at the 100% of charge.
- **Process stopped in advance** (fig. 24.b): the figure shows the end of the process when the Driver stops in advance the charging process.

It is possible to come back to the map interface by clicking on the top left icon.

- **CPMS functionalities: Externals Status of CS by Operator**

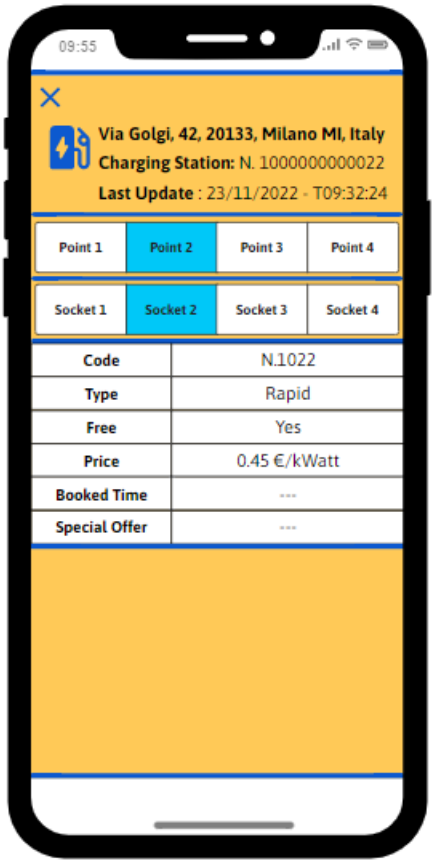

Figure 25: Charging Stations External Status by Operator

By clicking on a certain CS on the map, the Operator can view his foreign status (fig. 25).

From top to bottom the User can see: an orange background; a exit icon at the top-left, used for return to the map; the icon of a CS with its address position, serial number and the last update done with date and hour; the bar of the number of the CPs present in the CS from which the User can select a particular CP; the bar of number of sockets present in the CP selects before, from which the User can select a particular socket; the data about the socket selected before, such as the code, the type, if it is free or not, the current price, the booked-time if it was not free and the special offers if presents.

- **CPMS functionalities: Station Status Menu**

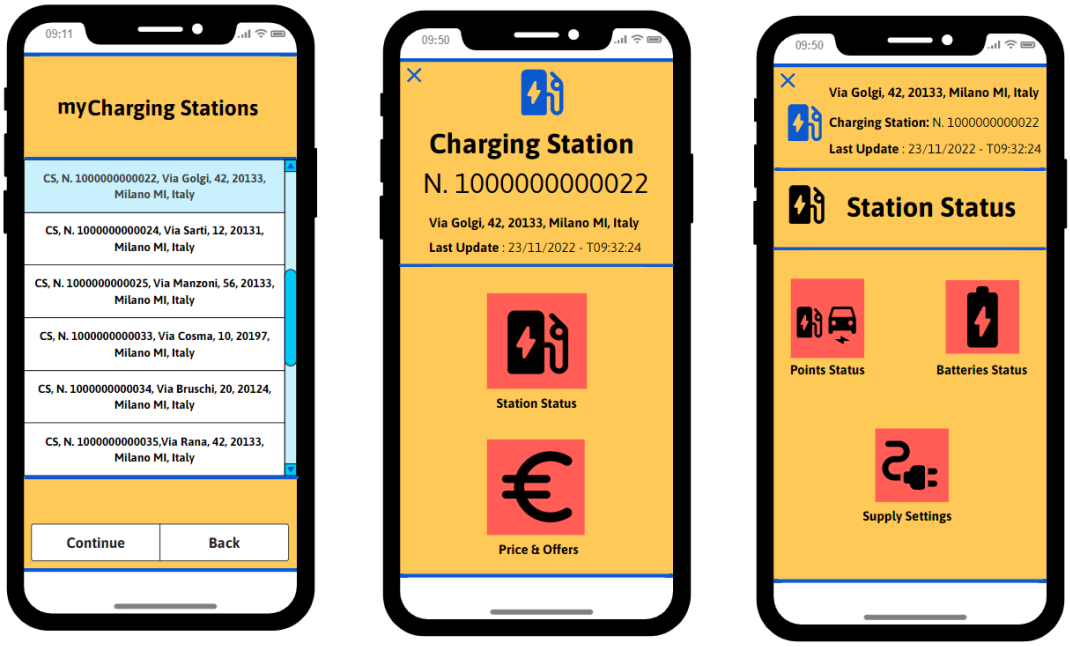

Figure 26: a) Selection of the CS, b) CPMS functionalities menu, c) Station Status menu

By clicking on myCharging Status, the Operator can see the list of CSs (fig. 26.a) that manages and select one, and click on "Continue" button, for visualize its internal data.

After choosing the CS, the Operator will see a screen (fig. 26.b) with the main data of the CS (serial number, address and the last update done of its settings) and two icons below.

The icons represent: Station Status, where the Operator can view and modify some internal parameters of the CS, and Price & Offers, where the Operator can view and modify prices and offers of the charging station.
After selecting the "Station Status" icon, the Operator will see a screen (fig. 26.c) with three icons: Point Status, where he can visualize the information concerning each CP of the CS; Batteries Status, where the Operator can visualize data about the batteries present in the CS; Supply Settings, where the Operator can view and change the type of supply of the CS.

The Operator can come back to the previous screen clicking on the "X" icon at the top left of the screen.
The "Last Update" indicates the last change made by either the Operator or the CPMS.

- **CPMS functionalities: Point Status and Batteries Status**

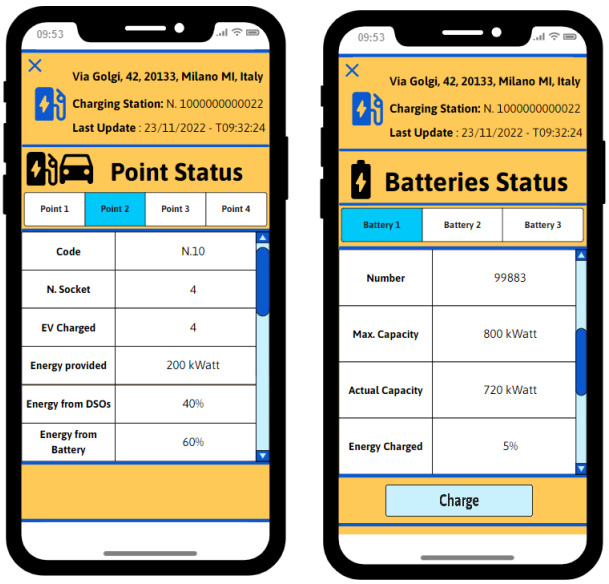

Figure 27: a) Point Status, b) Batteries Status

By clicking on the "Point Status" icon, the Operator can view (fig. 27.a) the features of each CP of the selected CS. For example: the code of the CP, the number of sockets it has, the number of EV connected at that time, how much energy it absorbs and other data.

Through a scroll bar, the Operator can see all the features related to CP selected.

By clicking on the "Batteries Status" icon, the Operator can visualize (fig. 27.b) the characteristics of the batteries (if present) of the selected CS.

In particular, the Operator can visualize for each battery, for example: the serial number, the maximum capacity, the actual capacity, the energy used for charge EVs, in percentage, in that moment and other data. Through a scroll bar, the Operator can see all the features related to the battery selected.

- **CPMS functionalities: Price & Offers and Supply Settings**

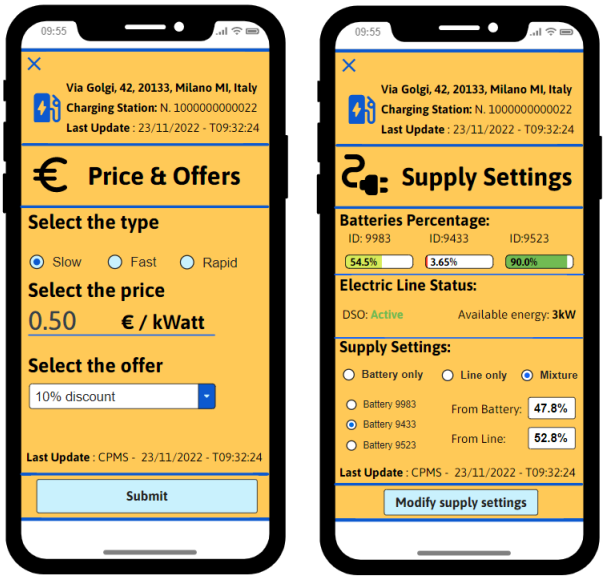

Figure 28: a) Price & Offers, b) Supply Settings

By clicking on the "Price & Offers" icon, the Operator can view (fig. 28.a) a screen related to the price and offers functionalities and can set them. In particular, the Operator: can select which type of socket change (and this change will be applied to all sockets of that type of the CS), and can set the new price for that type of socket. Optionally, the Operator can select a offer for the CS (selecting one from a special menu). For confirm the price and offer chosen, the Operator have to click on the "Submit" button.

By clicking on the "Supply Settings" icon, the Operator can view (fig. 28.b) a screen related to the energy provider functionalities and can set them. The Operator visualizes: the percentage of each battery and the status of the electric line between the DSO and the CS. Below, the Operator can change manually the supply settings. In particular, the Operator can choose three kind of supplies: battery only, DSO (line) only, and a mixture of them. In the mixture setting , the Operator can manually decide from which battery take electric energy and how much electric energy take , in percentage, from the batteries and from the DSO. Clicking on the "Modify supply settings" button, changes will be applied.

In both "Price&Offers" and "Supply Settings", the Operator can see the features of the last update done like: the entity that did the changes (CPMS or Operator), the date and the hour.

- **CPMS functionalities: List of DSOs and myDSO**

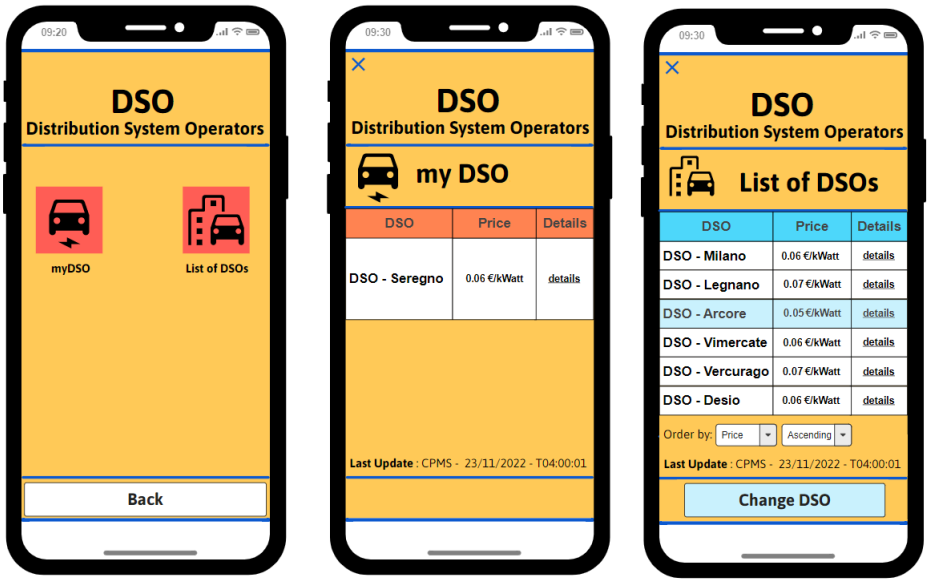

Figure 29: a)DSO menu, b) myDSO, c) List of DSOs

The Operator can manually decide from which DSOs to buy electric energy to charge the EVs connected to CSs.

By clicking on "myCPMS" and then "DSO" the Operator can visualize the DSO main menu (fig. 29.a) in which he decide the operation to do. By clicking on the icon "myDSO" the Operator can view the actual DSO that provide electricity (fig. 29.b) to the CSs of the CPO. In particular, the Operator can visualize the name of the DSO, the price at which it sells electricity and view details about it.

By clicking on the "List of DSOs" icon, the Operator can view the list of DSOs available, their price at which they sell electricity, view details related to them and order the list by a parameter e.g. price(fig. 29.c). In order to change the DSO, the Operator have to select the new DSO and click on the "Change DSO" button. The change will take place automatically and the Operator can view the just chosen DSO by going to "myDSO".

In both "myDSO" and "List of DSOs",the Operator can see the features of the last update done: the entity that did the changes (CPMS or Operator), the date and the hour.

3.1.2 Hardware interfaces

The main hardware interfaces for the system are the Charging Points of the CSs. These devices are supposed to be able to: properly scan the QR-code of the User's device, send electricity to the connected vehicle and ensure proper communication between the data of the EVs in charge and the CPMS.

3.1.3 Software interfaces

The system takes advantage of some external services which are necessary to ensure its correct and complete functioning. Since the User can visualize the CS in a specific area and select them, the application requires the usage of a map of the city in which it is being used. One possible option is to use Google Maps as it provides accurate real-time information for mapping. Also, the interaction between the various providers (eMall, eMSPs, CPOs, and DSOs) occurs through uniform dedicated APIs.

3.1.4 Communication interfaces

The form of communication used is the HTTPS Protocol: to safely communicate through the internet with the Web Server and the DBMS.

3.2 Functional requirements

3.2.1 Use Case Analysis

In this section we want to give a more in-depth view of the eMall application's use cases. Some of the main operations have been already described in a graphical way in section 3.1.1 with the aid of the mockups.

RegisterDriver

Actor	**Driver**
Entry Condition	The Driver has no eMall account, is on the initial view of the application and wants to sign up.
Flow of events	1. (a) The Driver opens the application and selects "Login" and then "Create Your Account". (b) The System responds and presents a sign up form to the Driver. (c) The Driver enters name, surname, email address, FC, phone number and password in the form. (d) The Driver accepts the Terms and Condition of Use and selects "Register". (e) The eMall system processes the acquired data and displays a success message.
Exit Conditions	An account has been successfully created and the System has stored a new Driver account.
Exceptions	(a) The Driver does not fill all the mandatory fields. (b) The entered email address does not exist. (c) The entered data is not valid. (d) The email or the FC of the Driver are connected to an already existing account. (e) The System always notifies the Driver the exception.

Table 5: UC.1

2. **viewChargingStationExternalStatus**

Actor	**Registered or Unregistered Driver**
Entry Conditions	The User is in the initial view of the application.
Flow of events	(a) The User selects a CS on the map view and selects "See Details". (b) The System provides a view with data about the external status of the CS i.e. price for each charging type, the special offers (if any), number of sockets and their type, number of free sockets, if all the sockets of a type are occupied and the amount of time until a socket of that type is freed.
Exit Condition	The view has been correctly presented to the User.
Exceptions	
Special Requirements	

Table 6: UC.2

3. **LoginDriver**

Actor	**Registered Driver**
Entry Conditions	The Driver is in the initial view of the application and wants to log in
Flow of events	(a) The Driver opens selects "Login" in the initial page of eMall. (b) The Driver selects "Login as Driver". (c) The System responds and presents a login form to the Driver. (d) The Driver fills the form with the needed credentials: email address or phone number and password. (e) The Driver selects "Login". (f) The System processes the acquired data and displays the Driver's main page.
Exit Condition	The login has successfully carried out.
Exceptions	(a) The Driver does not insert all the mandatory data. (b) The entered email or phone number and/or the entered password are incorrect. (c) eMall always notify the Driver the exception, if occurred.
Special Requirements	

Table 7: UC.3

4. **ReserveSocket**

Actor	**Registered Driver**
Entry Conditions	The Driver is in the initial view of the application and wants to reserve a socket
Flow of events	(a) The Driver chooses a charging station on the map view. (b) The System responds with a view of the external status of the selected charging station. (c) The Driver selects a socket of a certain CP and selects on "reserve". (d) The System registers the data about the reservation.
Exit Condition	(a) The reservation of the socket has been successfully carried out. (b) The System generates a QR-Code to be scanned by the Driver at the charging point's appropriate QR-Code Reader.
Exceptions	(a) The Driver tries to reserve an already reserved, occupied or unavailable socket. (b) eMall always notify the Driver the exception.
Special Requirements	(a) The generated QR-Code is available no later than 30 seconds after a successful reservation. (b) The generated QR-Code expires in 15 minutes. (c) At the moment of the reservation, the price of the energy for that reserved socket is locked and can not be modified until the Driver completes the charge or the reservation expires.

Table 8: UC.4

5. **StartCharging**

Actor	**Registered Driver**
Entry Conditions	The Driver is at the charging station and wants to start a charge.
Flow of events	(a) The Driver approaches a QR-Code to the Charging Point QR-Code Reader. (b) The System scans the QR-Code and provides in the application a view with the Driver's payment methods. (c) The Driver selects or adds a payment method and selects "Start". (d) The System reserves a fixed maximum amount of money from the Driver's chosen payment method. (e) The System unlocks the reserved charging socket. (f) The Driver plugs the socket in the electric vehicle.
Exit Condition	The charging process of the Driver's electric vehicle successfully starts.
Exceptions	(a) The selected payment method is invalid, expired or has no money in it. (b) The Driver tries to scan the QR-Code more than 15 minutes after the reservation. (c) eMall always notify the Driver the exception.
Special Requirements	(a) The unlocking of the reserved socket occurs no longer than 10 seconds after the successful money transaction. (b) The Driver has a timer of 5 minutes from the QR-Code scan to choose or add the payment method and to start the charging.

Table 9: UC.5

6. **InsertPaymentMethod**

Actor	**Registered Driver**
Entry Conditions	The Driver is logged in eMall and on the initial view.
Flow of events	Case 1: Adding a method out of the charging process. (a) The Driver selects on "My eMall" and then on "Payment". (b) The System responds with the view of the already added payment methods, if any. (c) The Driver selects "add a new payment method". (d) The System responds sending a form to fill with the data of the new payment method. (e) The Driver inserts the new payment method data and selects "Submit". (f) The System processes the acquire data and stores the new payment method information. Case 2: Adding a method after QR-Code scan at the charging point. (a) The Driver selects on "insert method". (b) The System responds sending a form to fill with the data of the new payment method. (c) The Driver selects "add a new payment method". (d) The Driver inserts the new payment method data, choose whether to save the new method in the wallet or not and selects "Submit". (e) The System processes the acquire data and stores the new payment method information.
Exit Condition	The new payment method has been successfully added.
Exceptions	(a) The Driver does not fill the form with all the mandatory data. (b) The Driver fills the form with non valid data i.e. expired payment method, incorrect number and/or CVV2. (c) The System can not establish a connection with the bank related to the payment method. (d) eMall always notify the Driver the exception.
Special Requirements	

Table 10: UC.6

7. **viewActiveChargingStatus**

Actor	**Registered Driver**
Entry Conditions	The Driver is logged in eMall and in the initial view of the application.
Flow of events	(a) The Driver selects "Active Charging". (b) The System provides a view with data about the active charging of the Driver's EV i.e. total price, EV battery percentage, start & end time kWatts provided and an overview of the booked charging socket.
Exit Condition	The active charging view has been correctly presented to the Driver.
Exceptions	(a) The Driver has no active chargings. (b) eMall always notify the Driver the exception.
Special Requirements	

Table 11: UC.7

8. **StopCharging**

Actor	**Registered Driver**
Entry Conditions	The Driver is logged in eMall, on the initial view, and has an active charging.
Flow of events	(a) The Driver selects on "My eMall" and then on "Active Chargings". (b) The System responds with the view of the status of the active charging. (c) The Driver selects on "Stop". (d) The System withdraws the amount of money corresponding to the provided quantity of energy. (e) The System responds with a message of successful interruption of the charging process. (f) The System shows to the Driver the report of the occurred charging process. (g) The Driver unplugs the socket from the EV and puts it in place. (h) The System updates the status of the socket, setting it as "Free".
Exit Condition	The payment for the charging has been successfully performed.
Exceptions	(a) An error with the payment process occurres. (b) eMall always notify the Driver the exception.
Special Requirements	

Table 12: UC.8

9. **OperatorLogin**

Actor	**Registered Operator**
Entry Conditions	The Operator is in the initial view of the application and wants to log in.
Flow of events	(a) The Operator selects "Login" in the initial view of eMall. (b) The Operator selects "Login as Operator". (c) The System responds and presents a form to the Operator. (d) The Operator fills the form with the needed credentials: email address, identification number and password. (e) The Operator selects "Login". (f) The System processes the acquired data and displays the Operator's main page.
Exit Condition	The login operation has been successfully carried out.
Exceptions	(a) The Operator does not insert all the mandatory data. (b) The entered data is incorrect. (c) eMall always notify the Driver the exception, if occurred.
Special Requirements	

Table 13: UC.9

10. **ViewChargingStationPointsStatus**

Actor	**Registered Operator**
Entry Conditions	The Operator is logged in and in the initial view of the application.
Flow of events	(a) The Operator selects "myCPMS" in the initial view. (b) The Operator selects "my Charging Stations". (c) The System responds with a list of the CSs the Operator is tasked to manage. (d) The Operator choose a charging station and then selects "Continue". (e) The Operator selects "Station Status" and then "Points Status". (f) The System provides a view of the internal status of each charging point of the charging station i.e. number of vehicles being charged, amount of power absorbed by each vehicle and estimated time left to the end of the charge.
Exit Condition	The System shows correctly the internal status information of the charging station.
Exceptions	
Special Requirements	

Table 14: UC.10

11. **ViewChargingStationBatteriesStatus**

Actor	**Registered Operator**
Entry Conditions	The Operator is logged in and in the initial view of the application.
Flow of events	(a) The Operator selects "myCPMS" in the initial view. (b) The Operator selects "my Charging Stations". (c) The System responds with a list of the CSs the Operator is tasked to manage. (d) The Operator choose a charging station and then selects "Continue". (e) The Operator selects "Station Status" and then "Batteries Status", that is available only if the selected CS is provided of any battery. (f) The System provides a view of the internal status of the charging station i.e. battery ID, maximum capacity, actual capacity and energy charged.
Exit Condition	The System shows correctly the battery status information of the charging station.
Exceptions	
Special Requirements	

Table 15: UC.11

12. **ModifyPriceAndOffersOfAChargingStation**

Actor	**Registered Operator**
Entry Conditions	The Operator is logged in and in the initial view of the application.
Flow of events	(a) The Operator selects "myCPMS" in the initial view. (b) The Operator selects "my Charging Stations". (c) The System responds with a list of the CSs the Operator is tasked to manage. (d) The Operator choose a charging station and then selects "Continue". (e) The Operator selects "Station Status" and then "Price and Offers". (f) The System provides a view where to set the price of each type of socket and to set any special offer. (g) The Operator sets the prices and, eventually, a special offer.
Exit Condition	The System processes the data and sets prices and, eventually, special offers for the selected charging station.
Exceptions	(a) The Operator inserts incompatible price or/and a special offer (e.g. negative price or negative percentage discount). (b) eMall always notify the Driver the exception, if occurred.
Special Requirements	

Table 16: UC.12

13. **chooseDSO**

Actor	**Registered Operator**
Entry Conditions	The Operator is logged in and in the initial view of the application.
Flow of events	(a) The Operator selects "myCPMS" in the initial view. (b) The Operator selects "myDSO". (c) The System shows data (i.e. name, energy price) about the DSO that is now providing energy. (d) The Operator selects "Choose another DSO". (e) The System shows a list of the available DSOs with their prices and details. (f) The Operator selects a DSO and then presses "Choose this DSO". (g) The System processes the acquired data and establishes a connection with the chosen DSO.
Exit Condition	The DSO choice has been successfully completed.
Exceptions	(a) The System does not receive any answer from the DSO, so it can not establish the connection. (b) eMall always notify the Driver the exception, if occurred.
Special Requirements	

Table 17: UC.13

14. **viewAndSetSupplySettings**

Actor	**Registered Operator**
Entry Conditions	The Operator is logged in and in the initial view of the application.
Flow of events	(a) The Operator selects "myCPMS" in the initial view. (b) The Operator selects "my Charging Stations". (c) The System responds with a list of the CSs the Operator is tasked to manage. (d) The Operator choose a charging station and then selects "Continue". (e) The Operator selects "Supply Settings". (f) The System shows: • data about the state of the charging station's batteries, if any, and about the energy line from which the DSO provides electricity i.e. battery charge percentage, DSO state (active/unavailable) and, if active, the energy it is providing. • percentages of energy provided by the DSO and by the CS battery. (g) The Operator selects "100% battery" (if any battery is available) or "100% DSO" or "Mixture", indicating, in this case, the percentages. (h) The Operator selects "Submit". (i) The System processes the acquired data and sets new supply settings for the selected CS.
Exit Condition	The supply settings modification has been successfully completed.
Exceptions	(a) The Operator presses "Submit" without selecting any supply method. (b) The Operator has chosen "Mixture" but with a sum of percentages different from 100%. (c) eMall always notify the Driver the exception, if occurred.
Special Requirements	

Table 18: UC.14

3.2.2 Use Case Diagrams

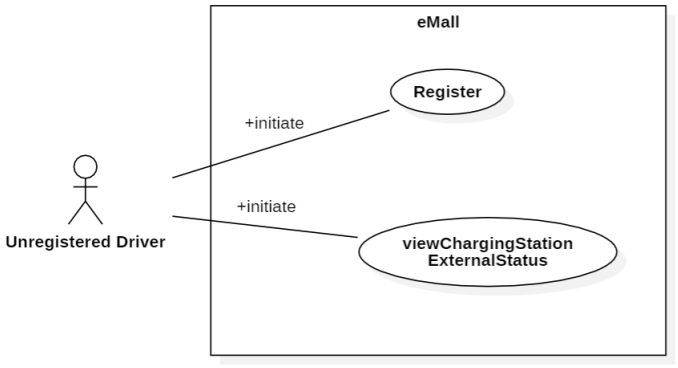

Figure 30: Unregistered Driver Use Case Diagram

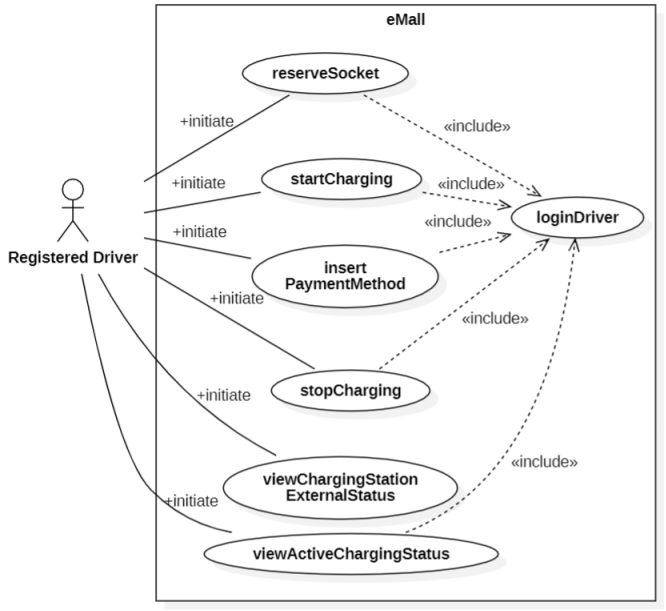

Figure 31: Registered Driver Use Case Diagram

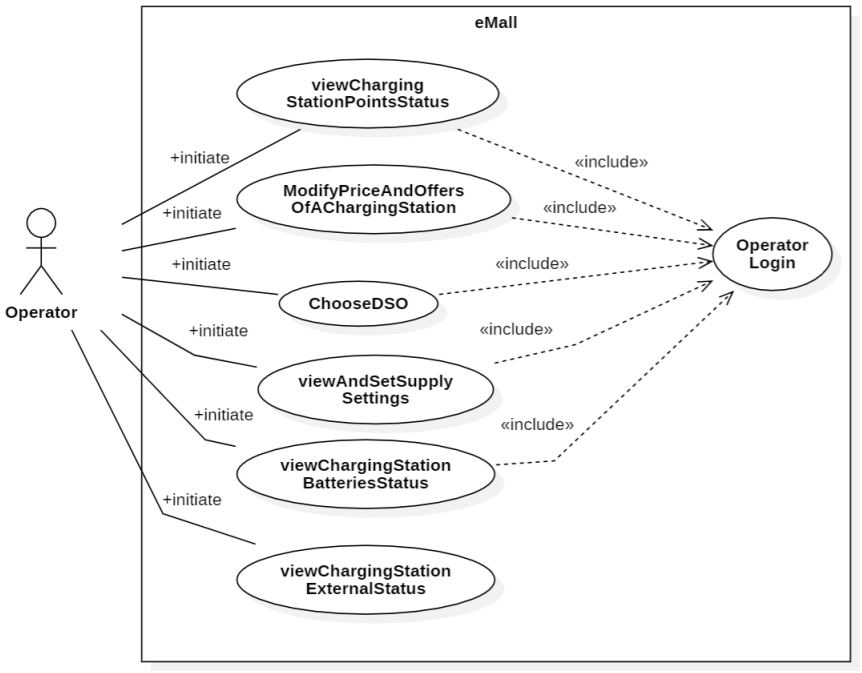

Figure 32: Operator Use Case Diagram

3.2.3 Sequence Diagrams

In this section will be shown the sequence diagrams regarding the previously described use cases. Here the interactions between eMSP and CPMSs will be highlighted, also comprising the interactions with external actors that are involved in the system's operations.

- **Driver Registration**

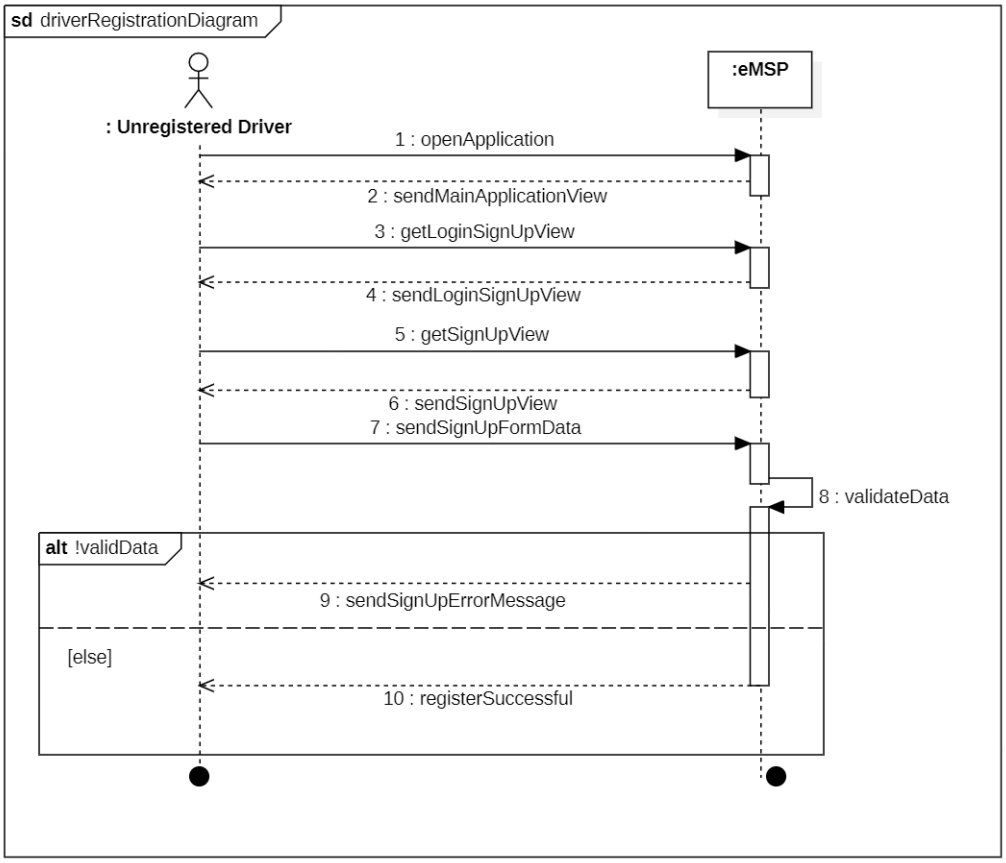

Figure 33: Driver Registration Sequence Diagram

- **View Charging Station External Status**

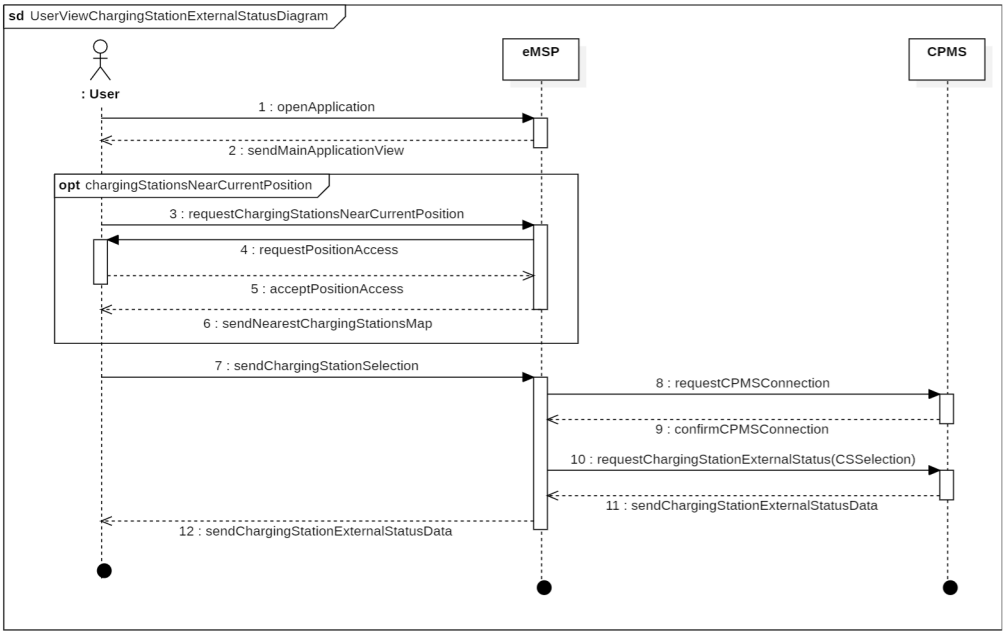

Figure 34: Sequence Diagram of viewing the external status of a charging station. either Operators or Registered/Unregistered Drivers have access to the data about the external status of a selected charging station.

- **Book Socket**

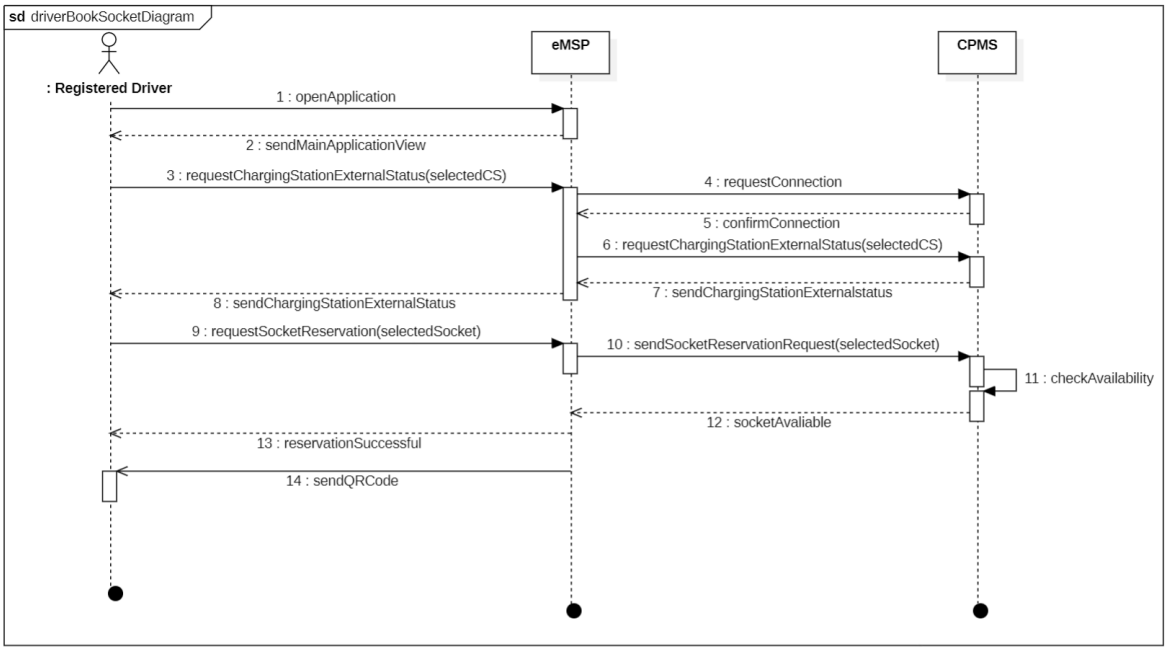

Figure 35: Socket Reservation Sequence Diagram

- **Start Charging**

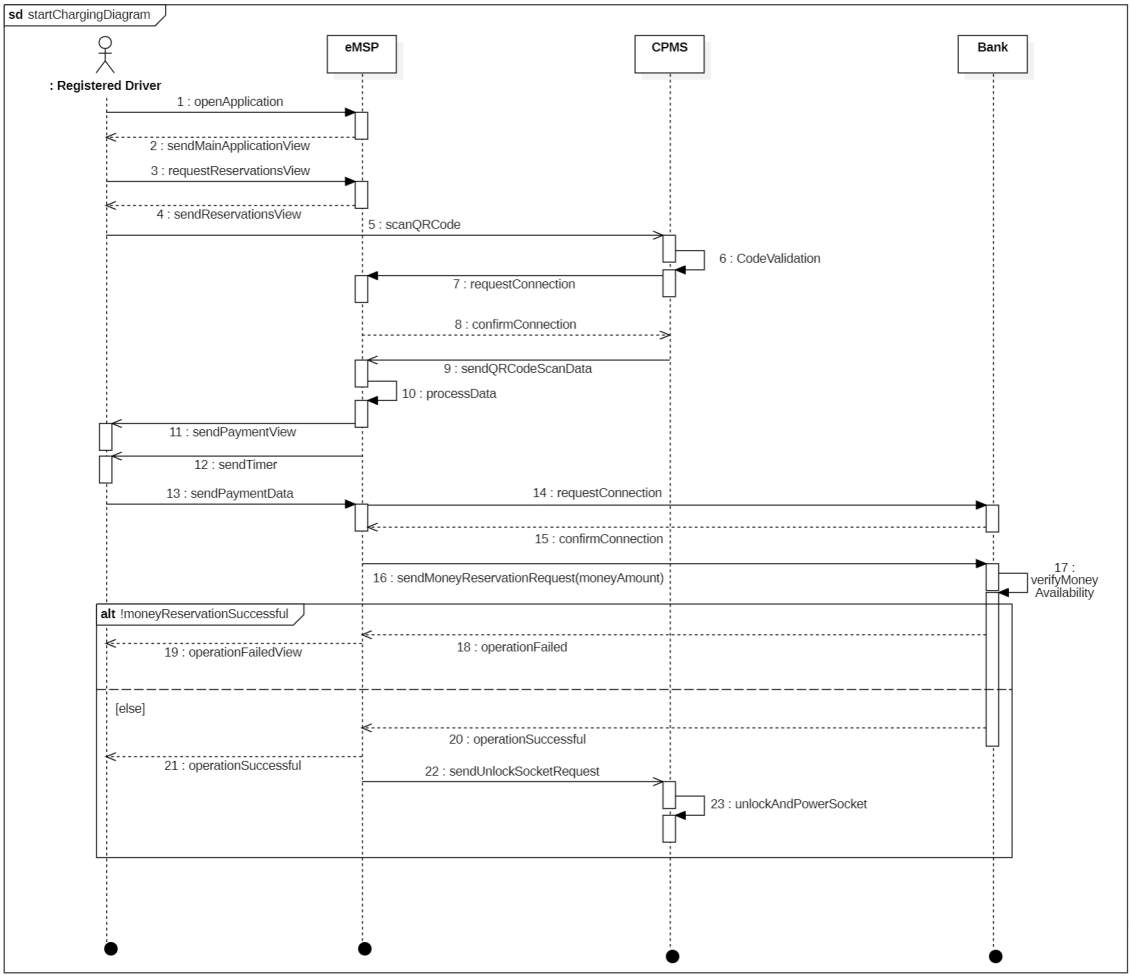

Figure 36: Start Charging Sequence Diagram

- **Insert New Payment Method**

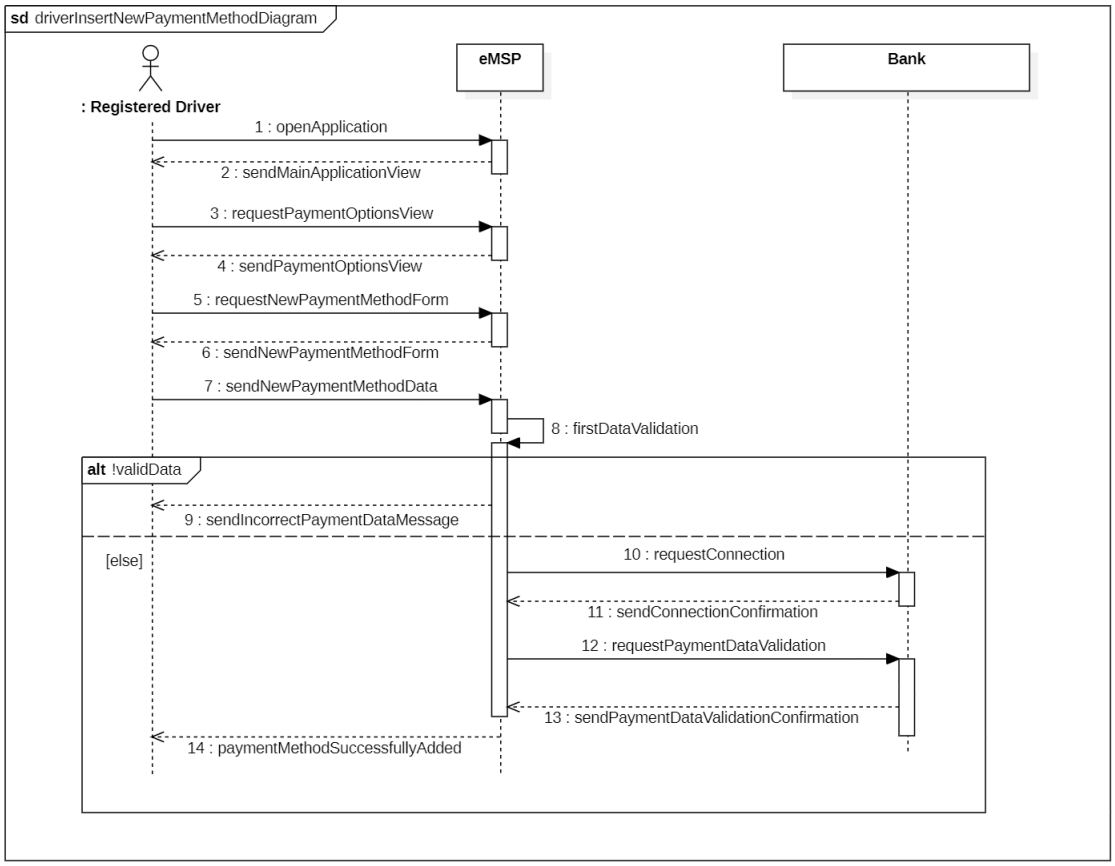

Figure 37: Insert New Payment Method Sequence Diagram

- **View Active Charging Status**

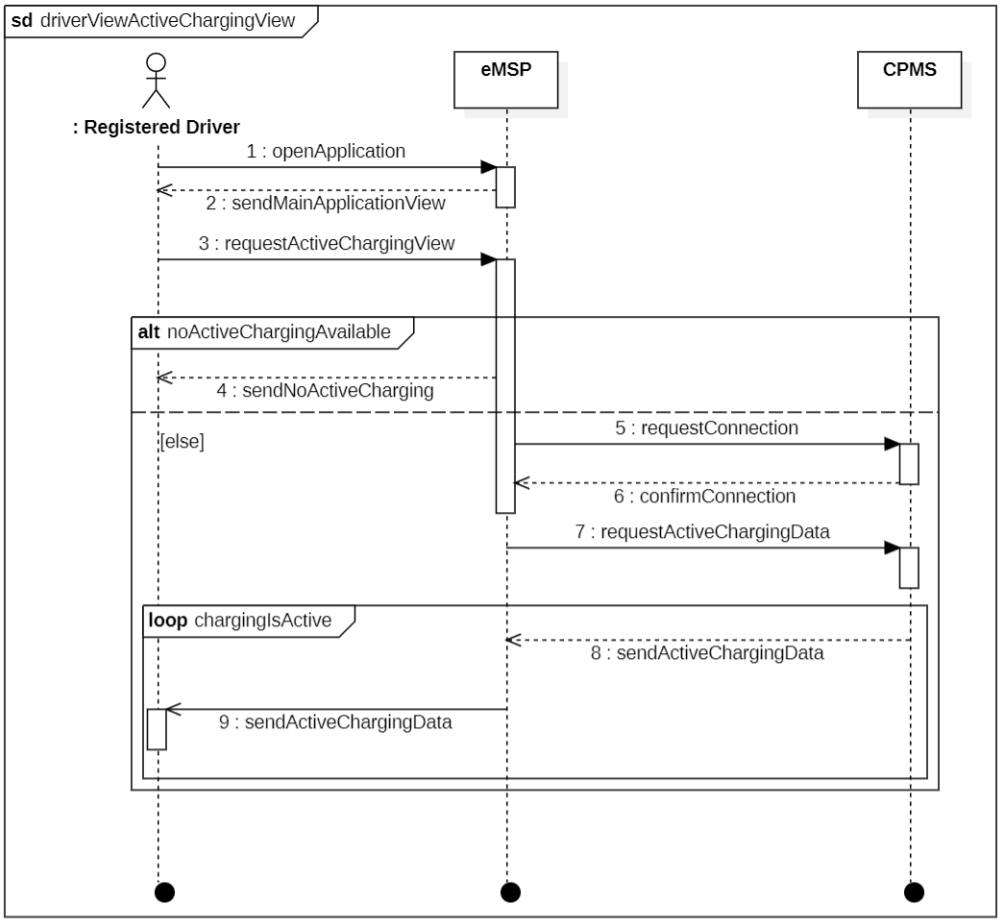

Figure 38: View Active Charging Status Sequence Diagram

- **Stop Charging**

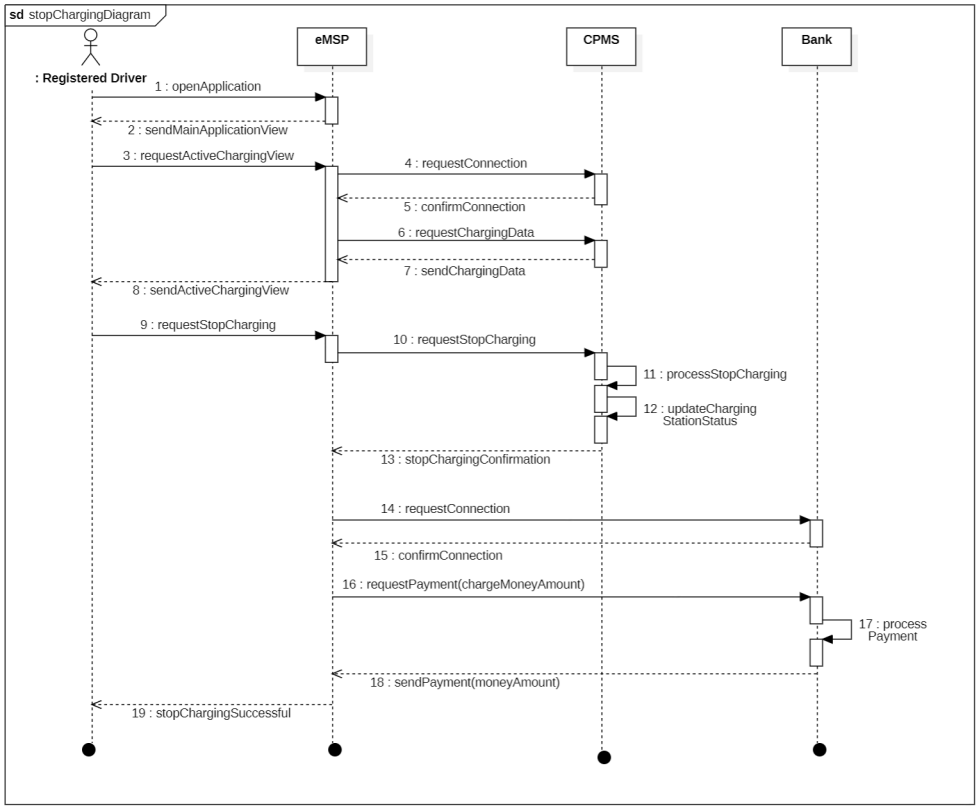

Figure 39: Stop Charging Sequence Diagram

- **Operator Login**

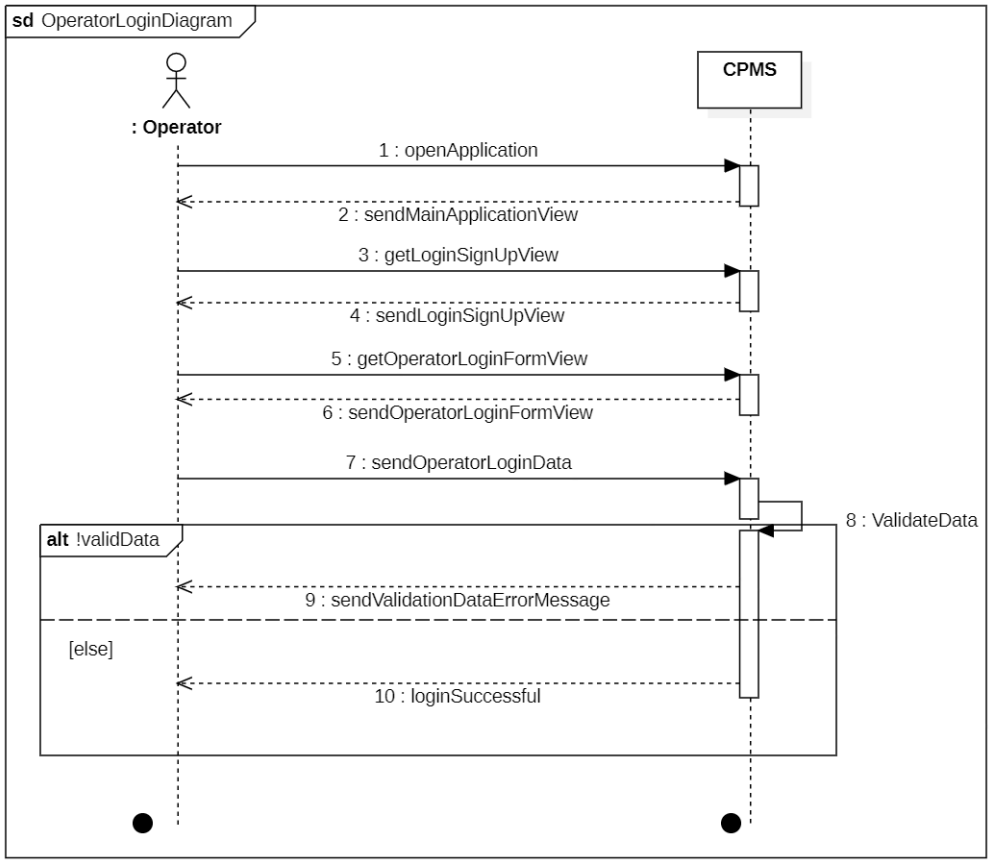

Figure 40: Operator Login Sequence Diagram

- **View Charging Points Status**

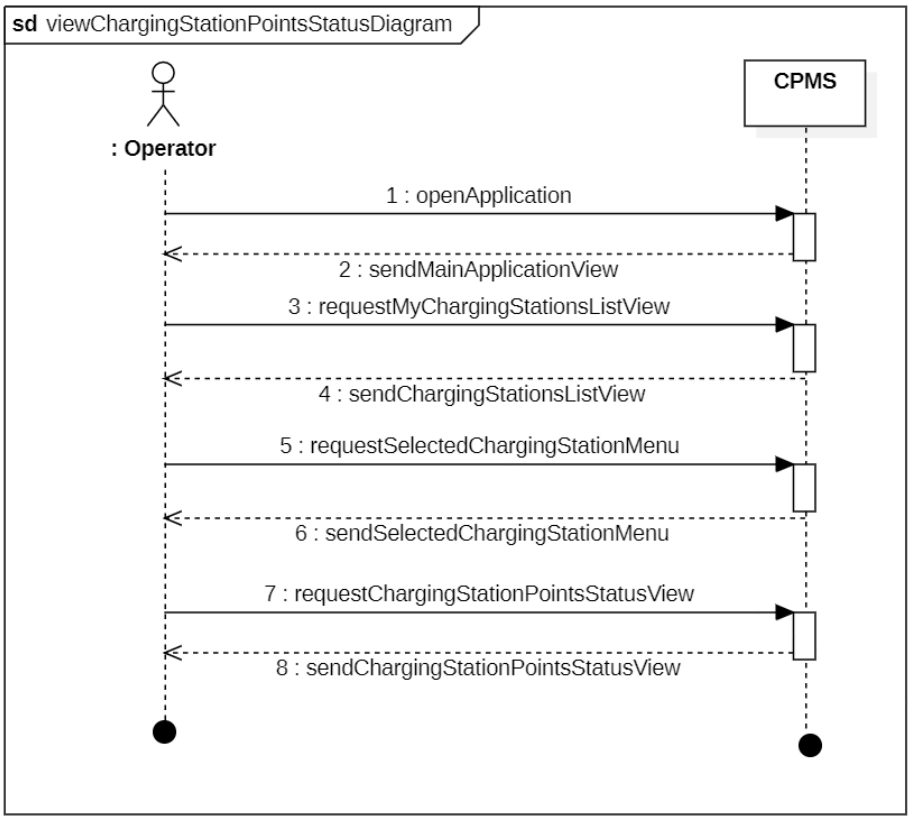

Figure 41: View Charging Station Points Status Sequence Diagram

- **View Charging Station Batteries Status**

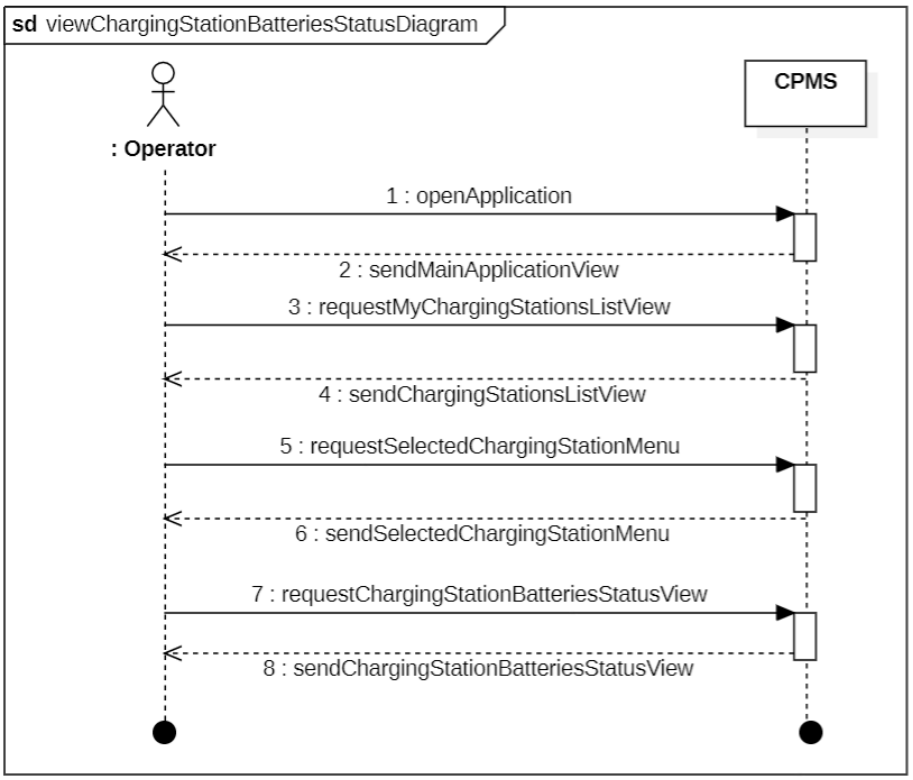

Figure 42: View Charging Station Batteries Status Sequence Diagram

- **Modify Charging Station Price And Offers**

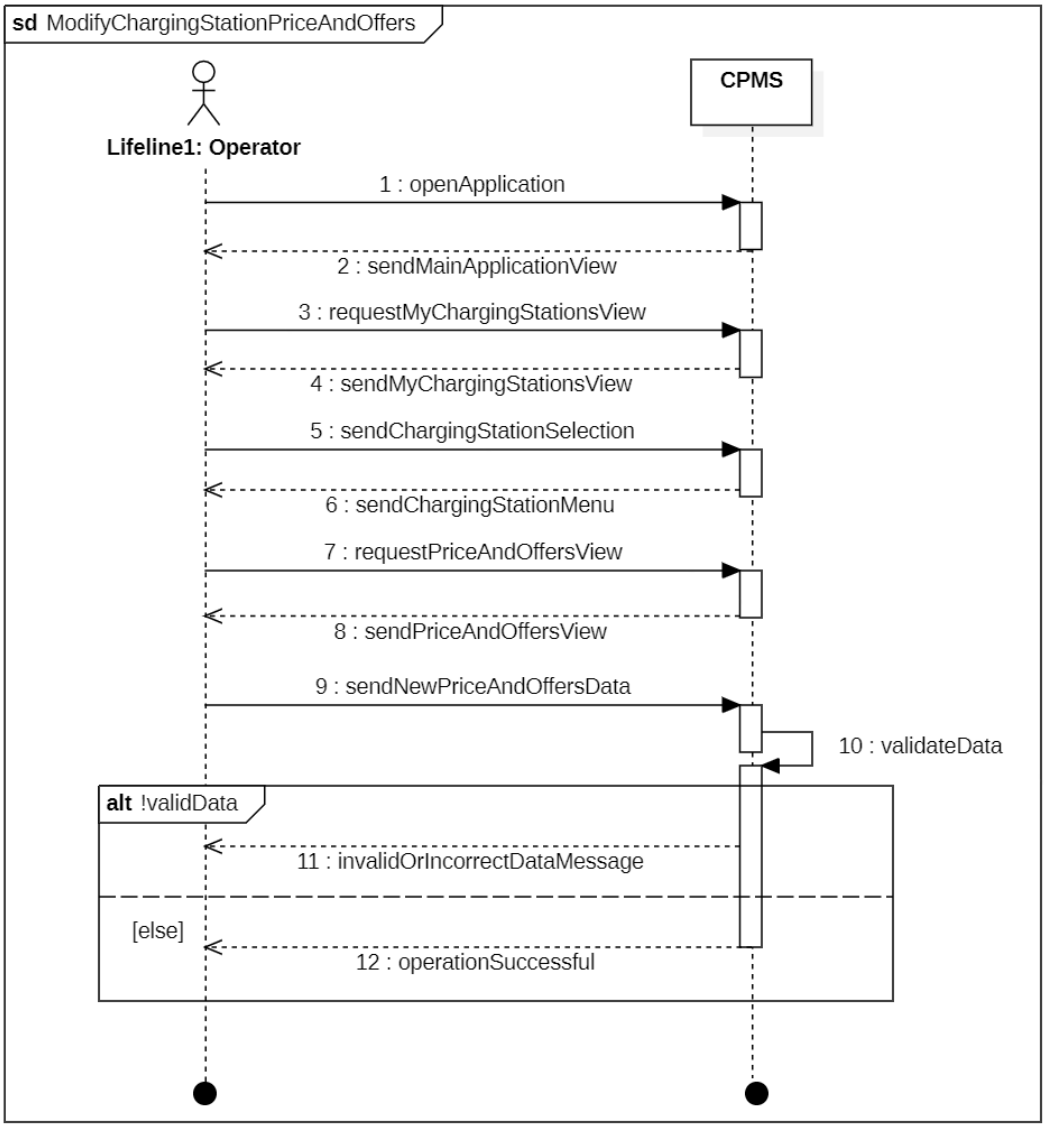

Figure 43: Modify Price And Offers Sequence Diagram

- **Choose DSO**

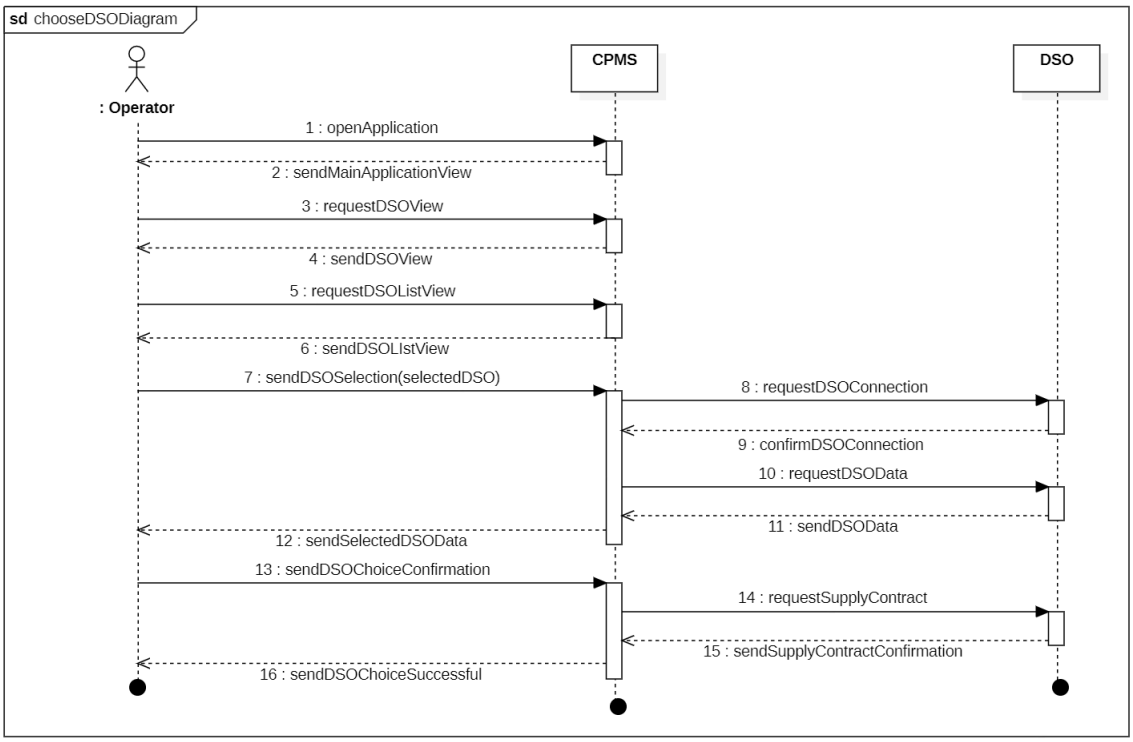

Figure 44: Choose DSO Sequence Diagram

- **View And Set Supply Settings**

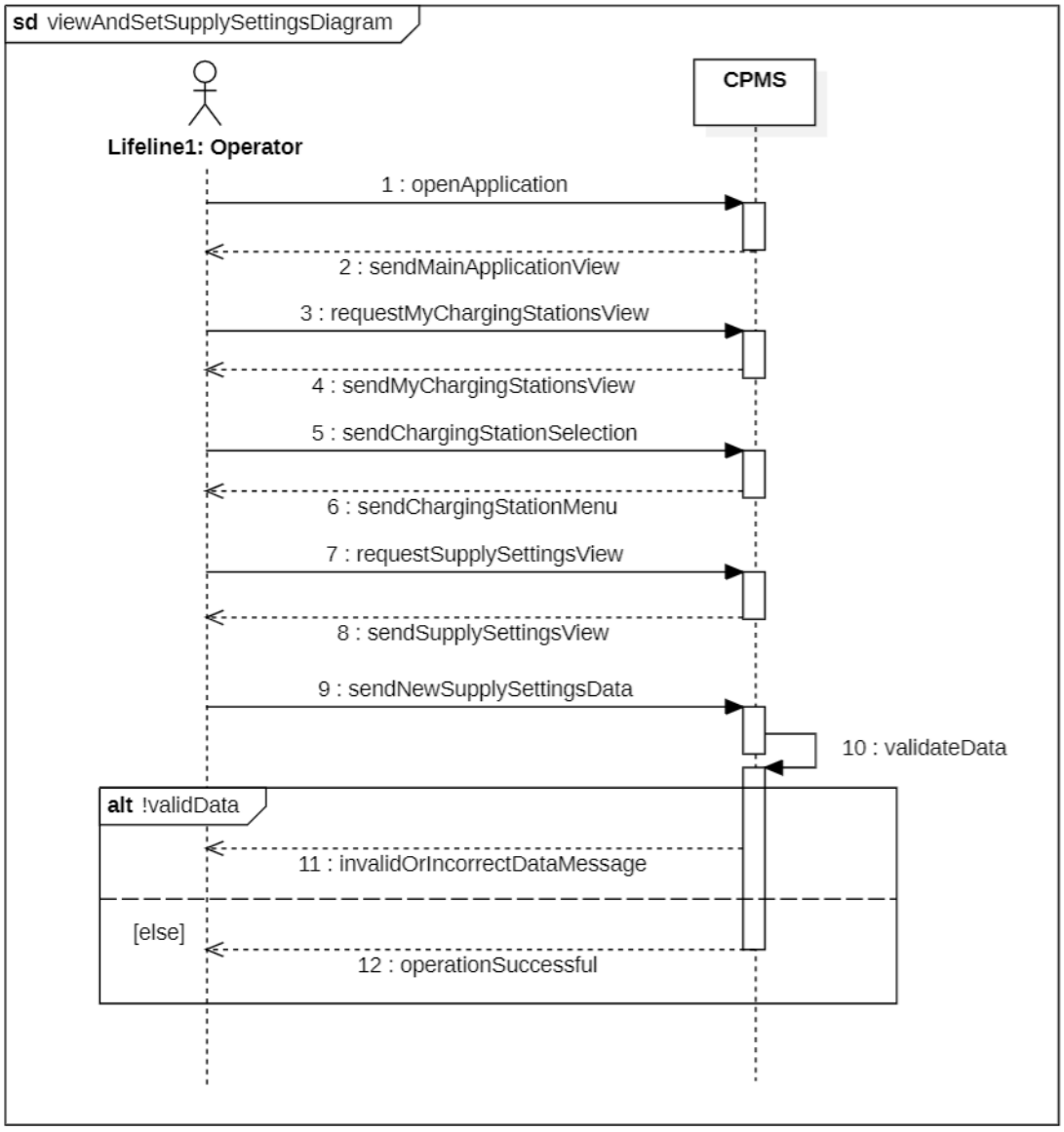

Figure 45: View And Set Supply Settings Sequence Diagram

3.2.4 Requirements and Goals Matrix

Requirements

- $[R.1]$ = The system shall allow a unregistered User to register an account as a Driver.
- $[R.2]$ = The system must allow registered Drivers to login.
- $[R.3]$ = The system must allow registered Operators to login.
- $[R.4]$ = The system shall allow a User, Driver and Operator, to see the map of CSs.
- $[R.5]$ = The system shall allow a User, Driver and Operator, to select a CS on the map.
- $[R.6]$ = The system shall allow a Driver to reserve a socket of a selected CS.
- $[R.7]$ = The system shall allow a Driver to receive a unique QR-code ticket for a successful reservation.
- $[R.8]$ = The system shall give a timer of 15 minutes to the Driver to scan the QR-code at the Charging Point (otherwise, the booking is rejected).
- $[R.9]$ = The system shall be able to correctly associate the socket with the Driver who reserved it.
- $[R.10]$ = The system shall allow a Driver to insert payment data.
- $[R.11]$ = The system shall give a timer of 5 minutes to the Driver in order to connect the EV to the socket and insert valid payment data (otherwise, the booking is rejected).
- $[R.12]$ = The system shall allow a Driver to visualize correct and real-time data about the charging process.
- $[R.13]$ = The system shall allow a Driver to stop the charging process.
- $[R.14]$ = The system shall allow the Driver to book a socket if and only if the Driver has no active reservations.
- $[R.15]$ = The system shall allow the Driver to book a socket if and only if the socket is available.
- $[R.16]$ = The system shall allow a Operator to select one of the CSs managed.
- $[R.17]$ = The system shall allow the Operator to visualize data about the "internal status" of the managed CSs.
- $[R.18]$ = The system shall allow the Operator to visualize the status of the batteries of a charging station, if any.
- $[R.19]$ = The system shall allow the Operator to charge the batteries of a charging station, if any.

- [$R.20$] = The system shall allow the Operator to select the price and the offers of a certain type of socket.
- [$R.21$] = The system shall allow the Operator to select the supply settings of the CSs.
- [$R.22$] = The system shall allow the Operator to visualize the list of available DSOs, with their prices and details.
- [$R.23$] = The system shall allow the Operator to visualize the current DSO from which the CPO is acquiring electric energy.
- [$R.24$] = The system shall allow the Operator to change the current DSO with another one from the list of available DSOs.
- [$R.25$] = The system shall allow a correct and coherent communication between the operations of the CPMS automatic system and the manual interventions handled by the Operators.
- [$R.26$] = The system shall be able to notify Drivers or Operators of incorrect actions.
- [$R.27$] = The software of the CPs must be consistent with the system.

Goals Matrix

GOALS	**DOMAIN ASSUMPTIONS**	**REQUIREMENTS**
G.1	D.1, D.2, D.3, D.10	R.1, R.2, R.3, R.25
G.2	D.3, D.10, D.11	R.4, R.5, R.25
G.3	D.3, D.10, D.11	R.4, R.5, R.25
G.4	D.3, D.10	R.2, R.4, R.5, R.6, R.7, R.8, R.14, R.15, R.26
G.5	D.3, D.4, D.5, D.6, D.7, D.10	R.2, R.9, R.10, R.11, R.12, R.13, R.26, R.27
G.6	D.3, D.10, D.12	R.2, R.10, R.11, R.26
G.7	D.3, D.8, D.9, D.10	R.3, R.16, R.17, R.18, R.19, R.25, R.26
G.8	D.3, D.8, D.9, D.10	R.3, R.17, R.22, R.23, R.24, R.25, R.26
G.9	D.3, D.8, D.9, D.10	R.3, R.16, R.17, R.20, R.25, R.26

Table 19: Goal Matrix table

3.3 Performance requirements

In this section we will specify some of the static and dynamic numerical requirements of the system and of the interaction between the users and the application. eMall system have to manage a lot of booking requests from different Drivers at the same time, so there will be a continuous stream of information that must be properly managed. Taking into account this large number of booking requests, the system needs to be able to store historical data. At the beginning we should guarantee that our system will be able to process requests for up to 100'000 active individual Users. In order to guarantee scalability, if the number of registered Users increases, modifies will be applied in order to meet the future needs. We prefer a step-by-step approach in order not to unnecessarily allocate resources.

3.4 Design constraints

3.4.1 Standards compliance

The standard longitude and latitude measures are used for the position. Regarding sensitive data the users provide to the application, the entire application is subject to the General Data Protection Regulation (GDPR), EU regulation on the processing of personal data and privacy, in order to legally protect our Users. Therefore the system ensures that only the name and location of any CS will be exposed. The same compliance will be adopted in order to ensure the registered customers data protection.

3.4.2 Hardware compliance

eMall application should be designed in order to be employed on smartphones, tablets and PCs. In particular, eMall application for Personal Computers is only available for Operators in order to have access to CPMS functionalities from their office. In order to ensure the system usage to different Users, the application must work independently of the hardware where it runs. Each device must meet the screen requirements for correct QR code scanning of charging socket's scanners. In addition, it must use the GPS positioning with the only purpose of localizing the User position and retrieving the position of any CS. Devices, CSs and monitoring stations must have a working internet connection (Wi-Fi, 5G/4G/3G) in order to submit any request to the system. In addition, very old devices can not run the application (but this point is better explained in Portability paragraph).

3.4.3 Other constraints

An important aspect to manage is the conflict that can arise between the reservation status of the sockets and the changes of some parameters by the Operators or by the CPMS.
When a socket is reserved, its prices and any offers are locked.
If a price or offer change is made, that change will be saved in a "next state" of the sockets. The change will be not applied immediately to the reserved sockets, but in a second moment, when the sockets are released.
When a socket is released (for the end of a charge or for a time expired) its status will be

updated.

Every CPO must verify that the sockets of each CS work properly, regarding both the correct scanning of tickets and the internet connection. The monitoring of the correct functioning of CSs must be constant and carried out by authorized and competent workers.

3.4.4 Software system attributes

3.4.5 Reliability

The reliability of eMall should be high enough, in order to give continuity to the system and operate properly for a long period of time. These needs are related to the fact that a new booking could be received at any time, or the CPMS could make automatic choices that could influence the future decisions of Drivers.

3.4.6 Availability

The system needs real time responses in order to complete booking and charging processes and to show correct data about the sockets of the CSs. With this constraint, the system availability must be ensured at least for 99.9 % or three-nines, that corresponds to an approximated system downtime of 8.76 hours per year. Is preferred to perform the maintenance during the night.

3.4.7 Security

Security is very important for our application, in particular because it exchanges and stores many sensitive data. All information regarding Drivers and Operators is sensitive. The system must prevent any attack that could steal data (especially data about credit cards stored in the application) or make the service unavailable. The data exchanged will also be encrypted to prevent any interception. The system will never expose sensitive data to external actors without consent.

3.4.8 Maintainability

The IEEE Standard Glossary of Software Engineering Terminology defines maintainability as: "The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment." The system development must ensure modification in order to fix any problem. It also must be suitable for future functionality extensions and supplements. This purpose is reached with a well commented, clean and readable implementation, using coherent design patterns. The system will also respond to major technological innovations.

3.4.9 Portability

In order to allow a large number of users (Drivers and Operators) to use our services, eMall mobile application should be developed for the most popular Operative Systems:

Windows, OSX, Android and iOS. Since the computational burden of our application is enough high, in developing-phase we will not consider old devices.

4 Formal analysis using Alloy

In this section a formal description of the domain of the system and its properties is given using Alloy.
We will focus on some critical aspects that are essential for the correct functioning of the application. Attention has been given to the following aspects:

- How Users are managed by the system:
  - Two different Drivers can not have the same Fiscal Code.
  - Two different Drivers can not have the same email address.
  - Two different Operators can not have the same Identification Number (OperatorID).
  - Two different Operators can not have the same business email address.
  - All inserted Fiscal Codes correspond to a registered Driver.
  - All inserted OperatorIDs correspond to a registered Operator.
  - An inserted email address corresponds to a registered Driver.
  - An inserted business email address correspond to a registered Operator.
  - A password is always associated to a Driver or Operator.
  - A position (indicated by coordinates) is always associated to a User.
  - Two or more Drivers (or Operators) can have the same coordinates.
  - Two or more coordinates can not be associated to the same Driver (or Operator).
- How reservations are managed by the system:
  - Two or more Drivers can not have the same reservation.
  - One Driver can do one or more reservations.
  - The same time can be associated to one or more reservations.
  - Sockets can have one or more reservations.
  - Each Reservation has: only one Driver, only one socket and only one time.
  - It is not possible to have two or more identical reservations.

- How the connection between the Charging Stations, CPO, Operators and DSO are managed by the system:
  - Each CS is managed by one and only one Operator.
  - Each Operator manages one or more CSs.
  - Each CS is associated to one or more CP.
  - Each CP is associated to one and only one CS.
  - Each socket is associated to one and only one CP.
  - Each CP is associated to one or more Sockets (we suppose that each CP as at least two sockets).
  - Each CS can have zero or more batteries (we suppose that each CS can have at most three batteries).
  - Each battery is associated to one and only one CS.
  - Operators and CSs are associated to one and only one CPO.
  - Each CPO can have one or more Operators and one or more CSs.
  - CPO has one and only one DSO.
  - Each DSO can be associated to one or more CPOs.

4.0.1 Alloy Users Model

```
open util / integer

// * * * * * * * * * * Signatures * * * * * * * * * *

sig CF{}
sig OperatorID{}
sig Email{}
sig BusinessEmail{}
sig Password{}

sig Coordinates{
        coordinates: some User,
}

abstract sig User{
        coordinates : one Coordinates,
}

sig Driver extends User{
        cf: one CF,
        email : one Email,
        password : one Password,
}

sig Operator extends User{
        operatorID : one OperatorID,
        businessEmail : one BusinessEmail,
        password : one Password,
}

// * * * * * * * * * * Facts * * * * * * * * * *

//There are no CFs duplicates (each Driver can register just once)
fact CFNODuplicates{
all d1,d2: Driver, thiscf: CF |
thiscf in d1.cf and thiscf in d2.cf implies d1 = d2
}

//There are no email duplicates(each Driver can register just once)
fact EmailNODuplicates{
all d1,d2: Driver, thisemail:  Email |
thisemail in d1.email and thisemail in d2.email implies d1 = d2
}

//There are no OpertorID duplicates
fact OperatorIDNoDuplicate{
all o1,o2: Operator, thisid : OperatorID |
thisid in o1.operatorID and thisid in o2.operatorID implies o1 = o2
}

//There are no business email addresses duplicates
fact BusinessAddressNODuplicates{
all o1,o2: Operator, thisbe:  BusinessEmail |
thisbe in o1.businessEmail and thisbe in o2.businessEmail implies o1 = o2
}

//All OperatorIDs have to be associated to an Operator
fact IDOperatorConnection{
all id : OperatorID | some o:Operator | id in o.operatorID
}

//All business email address have to be associated to an Operator
fact BusinessEmailOperatorConnection{
all bemail : BusinessEmail | some o:Operator | bemail in o.businessEmail
}

//All CFs have to be associated to a Operator
fact CFDriverConnection{
```

```
all c: CF | some d: Driver | c in d.cf
}

//All Email Address have to be associated to an Operator
fact EmailDriverConnection{
all e : Email | some d:Driver | e in d.email
}

//All Password have to be associated to a Driver
fact PasswordDriverConnection{
all p:Password | some d:Driver | p in d.password
}


//All Password have to be associated to a Operator
fact PasswordOperatorConnection{
all p:Password | some o:Operator | p in o.password
}


//Coordinates have one or more Users, and User has only one coordinates
fact MoreUsersCoordinates{
all c : User , u: Coordinates | c in u.coordinates ⟺ u in c. coordinates
}


//* * * * * * * * * Predicates* * * * * * * * *

pred show{
#Operator ≥2
#Driver ≥2
}

run show for 5
```

The following model (fig. 46) shows an example of how Users are managed.

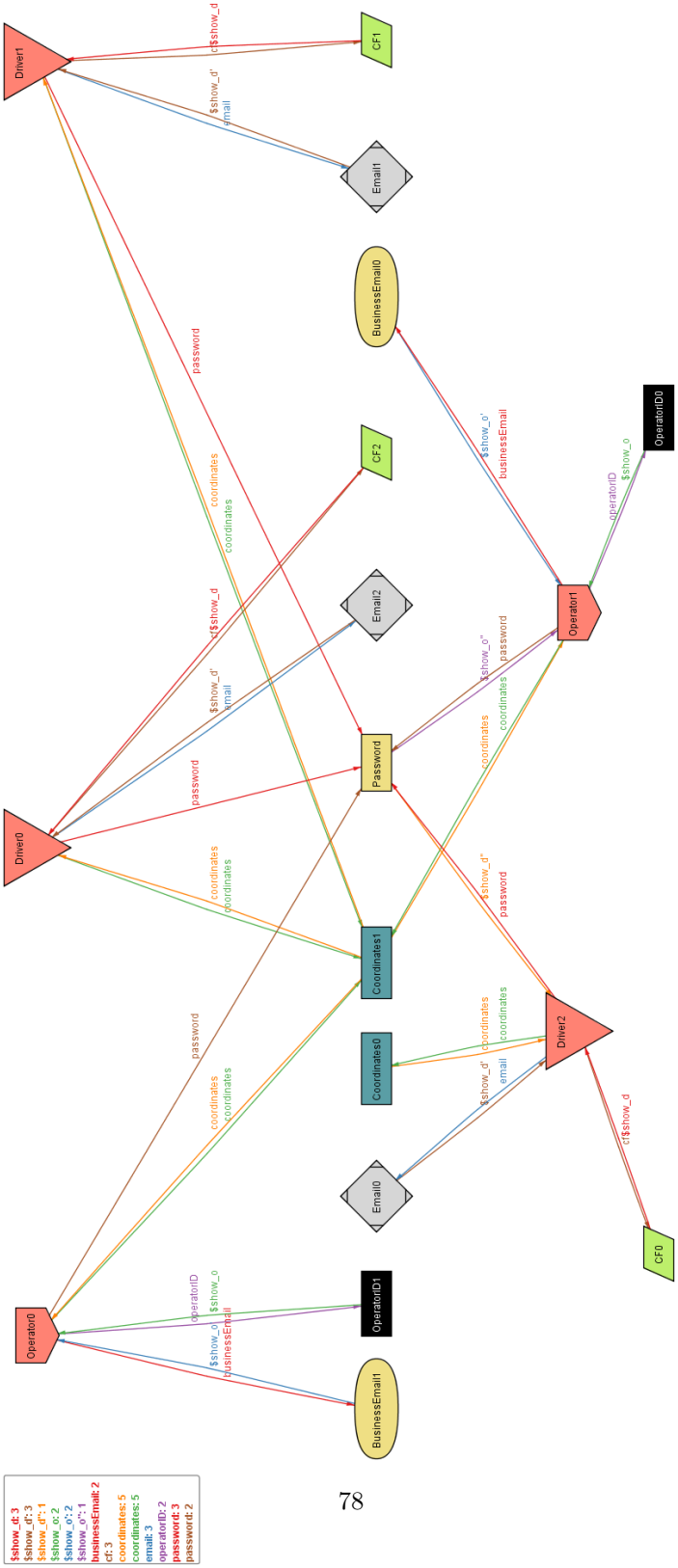

Figure 46: User Alloy Model

4.0.2 Alloy Reservation Model

```
open util / integer

// * * * * * * * * * * Signatures * * * * * * * * * *

sig Time{
        reservation: some Reservation,
}

sig Driver{
        reservation : some Reservation,
}

sig Socket{
        reservation : some Reservation,
}

sig Reservation{
        rd: some Driver,        //rd = reservation for drivers
        rs: some Socket,        //rs = reservation for sockets
        time: one Time
}

// * * * * * * * * * * Facts * * * * * * * * * *

//There are not many Drivers that have the same reservation
fact reservationsNODuplicatesDriver{
all d1,d2: Driver, r: Reservation |
r in d1.reservation and r in d2.reservation implies d1 = d2
}

//There are not many Sockets that have the same reservation
fact reservationsNODuplicatesSockets{
all s1,s2: Socket, r: Reservation |
r in s1.reservation and r in s2.reservation implies s1 = s2
}

//There are not many Sockets that have the same reservation
fact moreReservationsInDifferentTimes{
all d: Driver, r1,r2: Reservation |
(r1 in d.reservation and d in r1.rd) and
(r2 in d.reservation and d in r2.rd) and
r1 ≠ r2 implies r1.time ≠ r2.time
}

//There are not many Sockets that have the same reservation
fact moreReservationsInDifferentTimes{
all s: Socket, r1,r2: Reservation |
(r1 in s.reservation and s in r1.rs) and
(r2 in s.reservation and s in r2.rs) and
r1 ≠ r2 implies r1.time ≠ r2.time
}


//Connection between Time and Reservation: Time must have one Reservation
fact connectionTimeReservation{
all t : Time , r: Reservation |  r in t.reservation ⟺ t in r.time
}

//Connection between Drive and Reservation
fact connectionDriverReservation{
all d : Driver , r: Reservation | d in r.rd ⟺ r in d.reservation
}

//Connection between Socket and Reservation
fact connectionSocketReservation{
all s : Socket , r: Reservation | s in r.rs ⟺ r in s.reservation
}
```

```
//* * * * * * * * * Predicates* * * * * * * * * *

pred show{
#Socket ≥3
#Driver ≥2
#Reservation≥2
}
run show for 10
```

The following model (fig. 47) shows an example of how Reservations are managed.

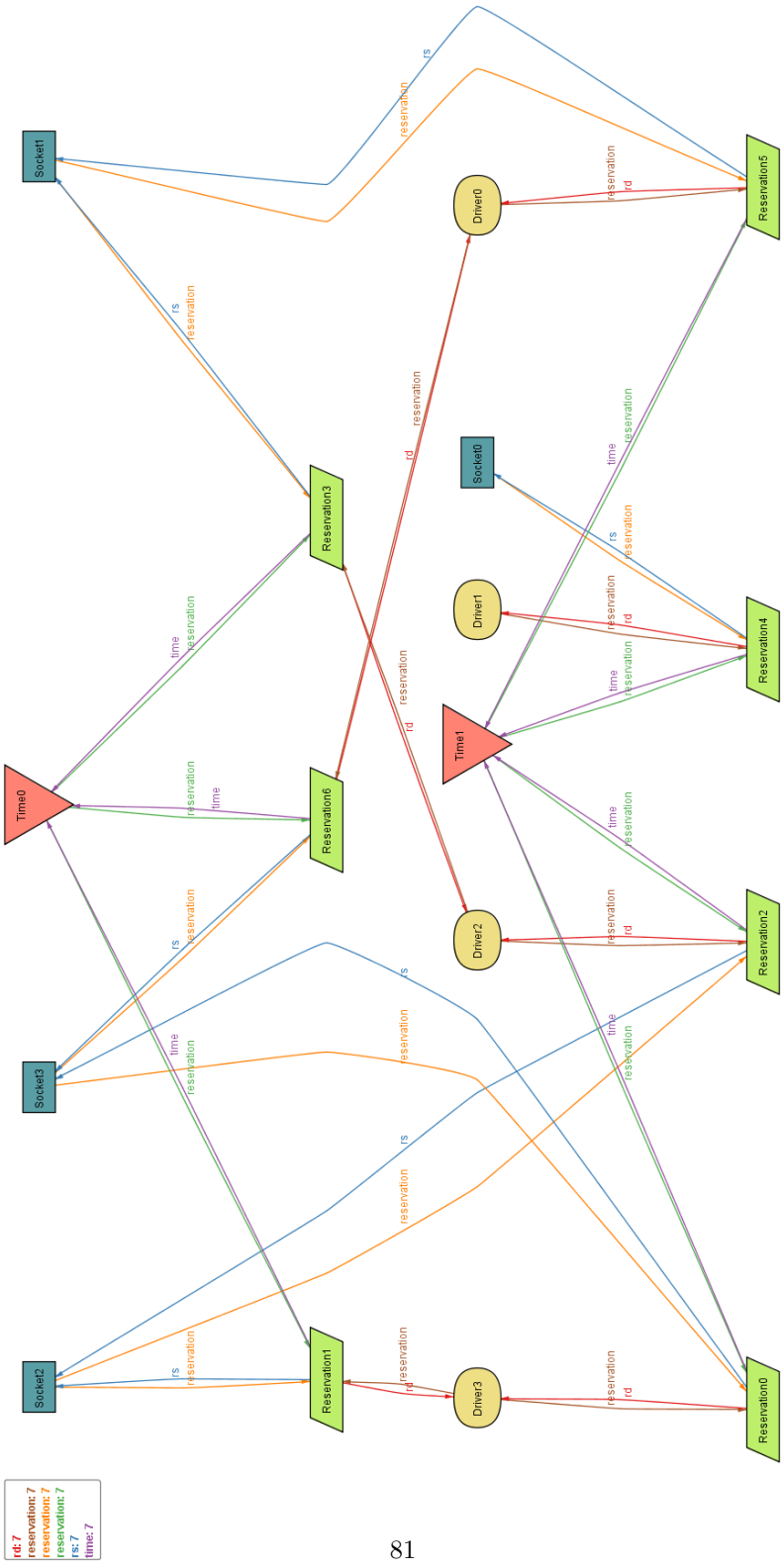

Figure 47: Reservation Alloy Model

```
/* * * * * * * * * * Assertion* * * * * * * * * * */

assert noTwoIdenticalReservations{
all r1,r2:Reservation, d:Driver, s:Socket, t:Time |
d in r1.rd and s in r1.rs and t in r1.time and d in r2.rd and s in r2.rs and t in r2.time implies r1 = r2
}
check noTwoIdenticalReservations for 10

Executing "Check noTwoIdenticalReservations for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  16902 vars. 690 primary vars. 23591 clauses. 188ms.
  No counterexample found. Assertion may be valid. 16ms.

/* * * * * * * * * * Assertion* * * * * * * * * * */

assert noTwoDriversSameReservation{
all  d1,d2:Driver,r:Reservation, s:Socket, t:Time |
s in r.rs and t in r.time and d1 in r.rd and d2 in r.rd implies d1=d2
}
check noTwoDriversSameReservation for 10

Executing "Check noTwoDriversSameReservation for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  16548 vars. 690 primary vars. 22917 clauses. 174ms.
  No counterexample found. Assertion may be valid. 16ms.
```

Figure 48: Assertions for Reservations

This model has been tested with some assertions for confirm his correctness. In particular way, we have identified two essential assertions to be verified, show in (fig. 48).

First assertion: it is not possible to have two or more different reservations with the same Driver and the same socket at the same time.
The Alloy code has confirmed this assertion without giving any counterexamples.

Second assertion: two or more Drivers can not book the the same reservation (same socket at the same time).
Also for this, the Alloy code has confirmed this assertion without giving any counterexamples.

4.0.3 Alloy Charging Stations Model

```
open util / integer

// * * * * * * * * * * Signatures * * * * * * * * * *

sig Socket{}

sig DSO{
        cpo: some CPO,
}

sig Battery{
        cs: one CS,
}

sig CPO{
        operator: some Operator,
        mydso: one DSO,
}

sig Point{
        socket: some Socket,
        cs : one CS,
}

sig Operator{
        cpo: one CPO,
        cs: some CS,
}

sig CS{
        operator : one Operator,
        cpo: one CPO,
        dso: one DSO,
        cp: some Point,
        battery: set Battery,
}



// * * * * * * * * * * Facts * * * * * * * * * *

//Every CS has one DSO from the same CPO
fact CStoDSOsameCPO{
all s: CS, c: CPO |
c in s.cpo implies s.dso = c.mydso
}

//Every Operator has only one CPO
fact oneOperatortoCPO{
all c1,c2: CPO, o: Operator |
o in c1.operator and o in c2.operator implies c1=c2
}

//Every DSO has only one CPO
fact oneOperatortoCPO{
all c1,c2: CPO, d: DSO |
d in c1.mydso and d in c2.mydso implies c1=c2
}

//Every CP has one and only one CS
fact operatorNoDuplicatesCS{
all c1,c2 : CS , p: Point |
p in c1.cp and p in c2.cp implies c1=c2
}

//Every Socket has one and only one CP
fact socketCP{
all p1,p2 : Point , s: Socket |
```

```
s in p1.socket and s in p2.socket implies p1=p2
}

//Every CP has one and only one CS
fact operatorNoDuplicatesCS{
all c1,c2 : CS , p: Point |
p in c1.cp and p in c2.cp implies c1=c2
}

//Every Battery has one and only one CS
fact batteriesInCS{
all c1,c2 : CS, b: Battery |
b in c1.battery and b in c2.battery implies c1 = c2
}


//Connection between Operator and CPO
fact connectionOperatortoCPO{
all o: Operator, c: CPO |
o in c.operator ⟺ c in o.cpo
}

//Connection between Operator and CS
fact connectionOperatortoCS{
all o: Operator, c: CS |
o in c.operator ⟺ c in o.cs
}

//Connection between CS and CPO of the same Operator
fact connectionCStoCPO{
all s: CS, o: Operator |
s in o.cs and o in s.operator implies s.cpo = o.cpo
}

//Connection between Operator and CS
fact connectionPointToCS{
all c : CS, p : Point | c in p.cs ⟺ p in c.cp
}

//Connection between Operator and CS
fact connectionBatteriesCS{
all b: Battery | some c: CS | b in c.battery
}

//Connection between batteries and CS
fact batteryAndCS{
all b: Battery, c: CS | b in c.battery ⟺ c in b.cs
}


//There are CPs with at least one Socket
fact moreSocketInCP{
all p: Point | #p.socket ≥2
}

//There are CS with more batteries
fact morebatteriesInCS{
all c: CS | #c.battery ≤3
}

//* * * * * * * * * Predicates* * * * * * * * *

pred show{
#CPO≥3
#Operator ≥3
#CS ≥4
}
run show for 30
```

The following model (fig. 49) shows an example of how Charging Stations are managed.

This last model, not being completely clear because of the few instances provided, has been tested through the following assertions to verify its correctness.
The assertion (fig. 50) wants to show that a CPO can have more than one Operator.
The assertion (fig. 51) wants to show that an Operator can manage more than one CS.
The assertion (fig. 52) wants to show that a CPO can have one and only one DSO from which take electric energy.

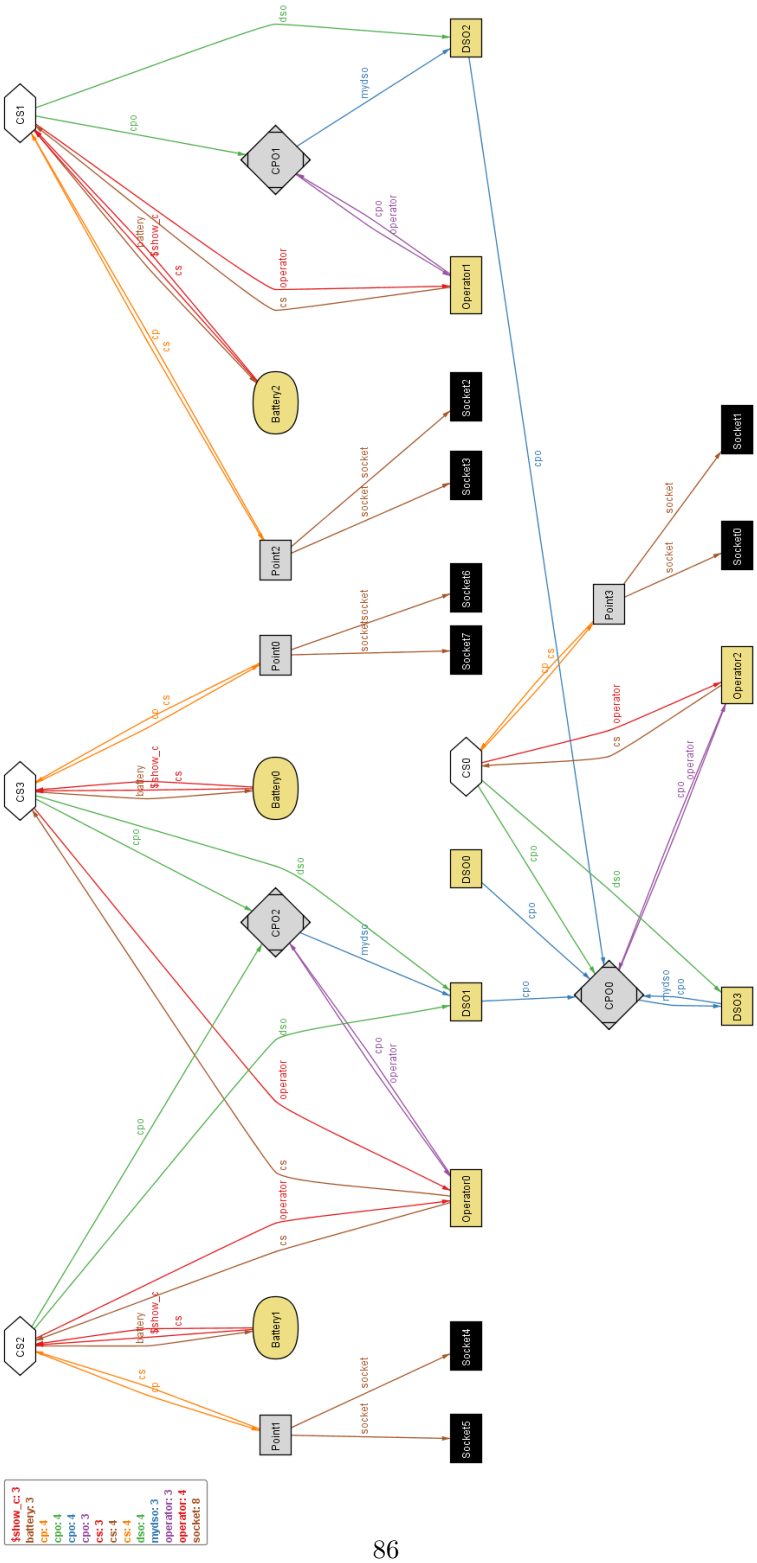

Figure 49: Charging Station Alloy Model

```
//* * * * * * * * * * Assertions* * * * * * * * * *

//Check if a Operator has more CSs
assert moreCSsInSameOperator{
all o:Operator, s1,s2: CS |
s1 in o.cs and s2 in o.cs implies s1=s2
}
check moreCSsInSameOperator for 10
```

```
Executing "Check moreCSsInSameOperator for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  26190 vars. 1500 primary vars. 43020 clauses. 74ms.
  Counterexample found. Assertion is invalid. 72ms.
```

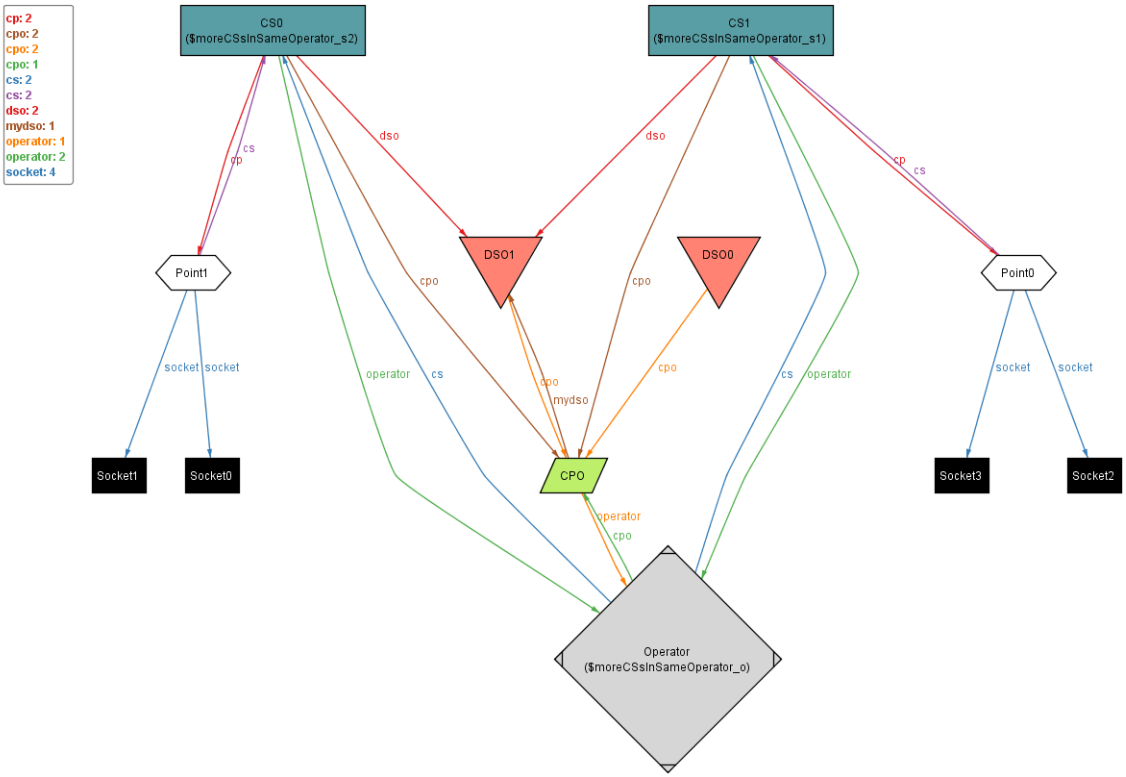

Figure 50: Assertion: CPOs with many Operators

```
//* * * * * * * * * Assertions* * * * * * * * * *

//Check if a CPO has more Operators
assert moreOperatorsInSameCPO{
all c:CPO, o1,o2:Operator |
o1 in c.operator and o2 in c.operator implies o1 = o2
}
check moreOperatorsInSameCPO for 10
```

```
Executing "Check moreOperatorsInSameCPO for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  26190 vars. 1500 primary vars. 43020 clauses. 199ms.
  Counterexample found. Assertion is invalid. 114ms.
```

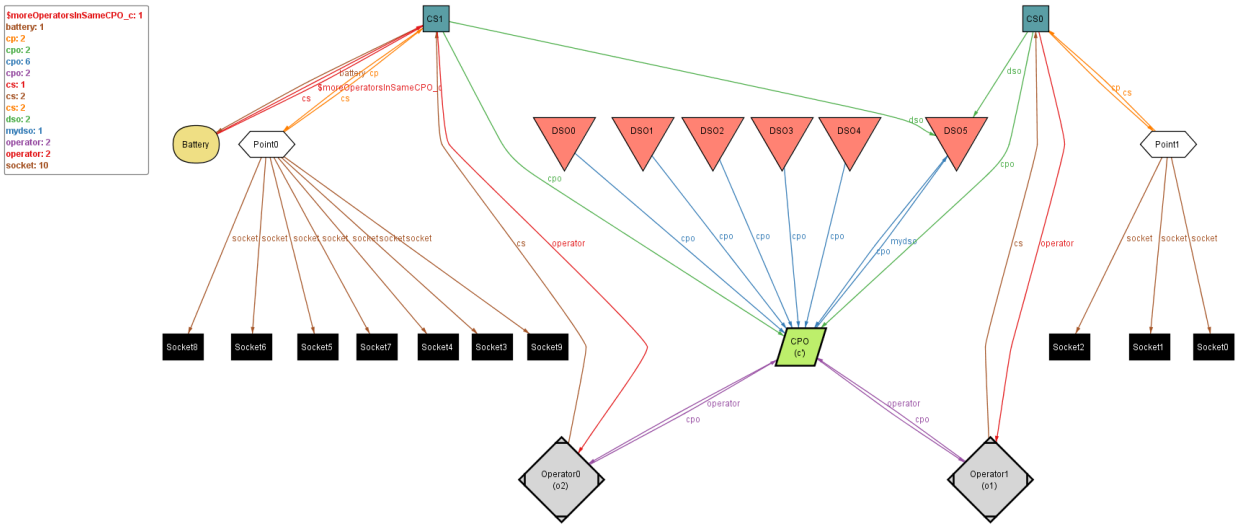

Figure 51: Assertion: Operators with many CSs

```
//* * * * * * * * * Assertions* * * * * * * * * *

//Check if a CPO has more DSO
assert moreDSOtoCPO{
all c1,c2 : CPO, d:DSO |
d in c1.mydso and d in c2.mydso implies c1=c2
}
check moreDSOtoCPO for 10
```

```
Executing "Check moreDSOtoCPO for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  26300 vars. 1500 primary vars. 43230 clauses. 65ms.
  No counterexample found. Assertion may be valid. 8ms.
```

Figure 52: Assertion: CPO with only one DSO

5 Efforts

Individual Work		
	Eutizi Claudio	***Perego Gabriele***
Tasks	**Hours**	**Hours**
Introduction (chapter 1)	4	8
Overall description (chapter 2)	12	9
Specific requirements (chapter 3)	12	9
Formal analysis using Alloy (chapter 4)	4	6
Final Revision	4	4
Total	**36**	**36**

Table 20: Time spent by each team member