POLITECNICO
MILANO 1863

M.Sc. Computer Science and Engineering

Software Engineering 2 project

eMall - e-Mobility for All

Requirements Analysis and Specification Document

Christian Di Diego, Federica Di Filippo, Carmine Faino

23 December 2022

GitHub Repository:

https://github.com/FedericaDiFilippo/DiDiegoDiFilippoFaino

1 Introduction

This document represents the Requirement Analysis and Specifications Document (RASD). The aim of the document is to analyse the functional and non-functional requirements needed to develop and implement an application. It also describes the application domain, the constraints, the actors, and their interactions with the system.

1.1 Purpose

The purpose of the project is to describe and develop eMall – e-Mobility for All, which is a system created with the purpose of limiting carbon emissions due to mobility needs. The application wants to help drivers of electric vehicles to better organize charges, to interfere as little as possible with their daily program. Furthermore, the application allows CPO to manage the charging stations they are assigned to. To guarantee these features, the application must interact with a GPS system and combine data from DSO, to show the most correct and updated information.

1.1.1 Goals

Goal	**Description**
G1	Allow User to get information about nearby charging stations
G2	Allow User to manage a charge
G2.1	Allow User to book the charge
G2.2	Allow User to start the charging process in the booked socket
G2.3	Allow User to pay for the service received
G3	Allow User to receive suggestion about the right moment to charge his vehicle
G4	Allow Admin to get information about energy cost and sources from DSO
G5	Allow Admin to select either to use energy from batteries or directly from DSO
G6	Allow Admin to select from which DSO to acquire energy
G7	Allow Admin to know location and status of charging stations (both internal and external)
G8	Allow Admin to modify status of charging stations (both internal and external)

1.2 Scope

The document will deal with the electric mobility word. The principal agents are the electric vehicles' owners and CPOs, that interact through their employees registered in the application. In fact, User can check the prices and special offers published by each CPO, about all the available charging stations. On the other side, each CPO's employee can modify the prices of energy of the stations he manages and publish new offers on the application.
According to the World and Machine paradigm the machine is identified with the portion of the system to be developed, while the environment is the world that interact with the system. With this classification, the document distinguishes three types of Phenomena:

1.2.1 World Phenomena

Identifier	Description
WP1	A person wants to charge his electric vehicle
WP2	A person buys or sell an electric vehicle
WP3	Prices of energy change
WP4	CPO's employee wants to check the status of the charging stations he manages
WP5	CPO's employee decides about where to acquire energy
WP6	CPO's employee decides whether to use energy stored or from DSO
WP7	CPO's employee decides about costs and special offers

1.2.2 Shared Phenomena

Shared Phenomena - controlled by the world:

Identifier	Description
SP1	A Customer registers an account on the eMSP application.
SP2	A Customer, that wants to enjoy the extra functionalities, share his calendar and position.
SP3	A Customer check the locations of nearby charging stations.
SP4	A Customer books a charge from eMSP application for a specific socket and timeframe.

SP5	A Customer modifies the vehicles' information on his account.
SP6	A Customer reaches the booked socket and starts the charge of the vehicle he has booked for.
SP7	A CPO's employee publishes prices and special offers for a certain charging station.
SP8	A CPO's employee checks energy prices and sources.
SP9	A CPO's employee selects from which energy provider to acquire energy.
SP10	A CPO's employee selects whether to store energy in batteries (if they are present in the charging station) or not.
SP11	A CPO's employee selects whether to use or not energy stored in batteries for vehicles' charging process.
SP12	A CPO's employee checks the internal and external status of a charging station.

Shared Phenomena – controlled by the machine:

Identifier	**Description**
SP13	The System notifies a Customer when he is almost out of charge and needs to stop to a charging station.
SP14	The System assigns an OTP to a Customer who as booked a charge, to make him recognisable at the charging station.
SP15	The System notifies the Customer when the charging process is finished, and he can disconnect the vehicle from the socket.
SP16	The System plans the charging process according to the Customer's schedule.
SP17	The System notifies Customers when there are special offers in the nearby charging stations.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

UserID	the identifier code that a User inserts when booking a charge to be recognised at the charging stations
OTP	one-time passcode

1.4 Revisions history

1.4.1 Reference Documents

- The specification document “Assignment RDD AY 2022-2023_v3”

1.5 Document Structure

The document is divided into six sections:

Section 1 introduces the problem and describes the principal purposes of the eMall application. Furthermore, this section gives a general explanation of the goals and the phenomena related with the application.

Section 2 gives a more detailed description of the Shared Phenomena and of the principal functions that must be implemented in the application. It also shows the most common usage scenarios, focusing on the differences between actor and on their characteristics. Here a distinction between each element interacting with the system is presented through the class diagram.

Section 3 is the principal section of the document. It contains all the specific requirements and explores in a more detailed way (through Use Cases) the scenario presented in Section 2. In the last part of the section there is an explanation about how the Software System Attributes should be implemented in the application. In this section Design Constraints are also mentioned.

Section 4 gives a more formal analysis using Alloy. It creates a system’s model implementing the class diagram shown in Section 2.

Section 5 explains the effort spent by each group member in order to complete the project and shows a list of all the activities done.

Section 6 contains the references present in the document.

2 Overall description

2.1 Product perspective

2.1.1 Scenarios

Here some possible scenarios of usage for the application:

S1. Cesare has still bought his first electric car and he wants to easily manage charges. He creates an account to eMSP application, inserting his name, surname, email and a password. Then he logs in for the first time. Now he can register a vehicle inserting the brand, the model and the car plate (if it has one) of the vehicle. At this point he is registered and a UserID has been associated to him. To pay for charges, he connects his bank account to the application, that will automatically take the money for a charge after it's finished.

S2. Marco is already registered to eMSP application and has shared is calendar and navigation system. While going home from work, suddenly, the car notifies him that he will soon be out of charge, and urgently needs to stop in a charging point. So, he opens the application, checks on the map of charging points the nearest one and books a charge. He chooses one of the available charging sockets and a timeframe. He reaches the station and starts the charge. While waiting for the charge to be completed, Marco drinks a coffee in the bar on the other side of the street. When he receives the notification that the charging process is finished, he comes back to the charging station and disconnect his car from the socket. The cost of the charge is deducted from the bank account connected with the application.

S3. Alessia is planning to come back to Rome, her hometown, to visit her parents. In order not to remain out of charge, she wants to plan the trip. So, she logs in the eMSP application and inserts her path. The application shows her the best charging points where to stop to minimize the loss of time. So, she books the charges in the suggested points.

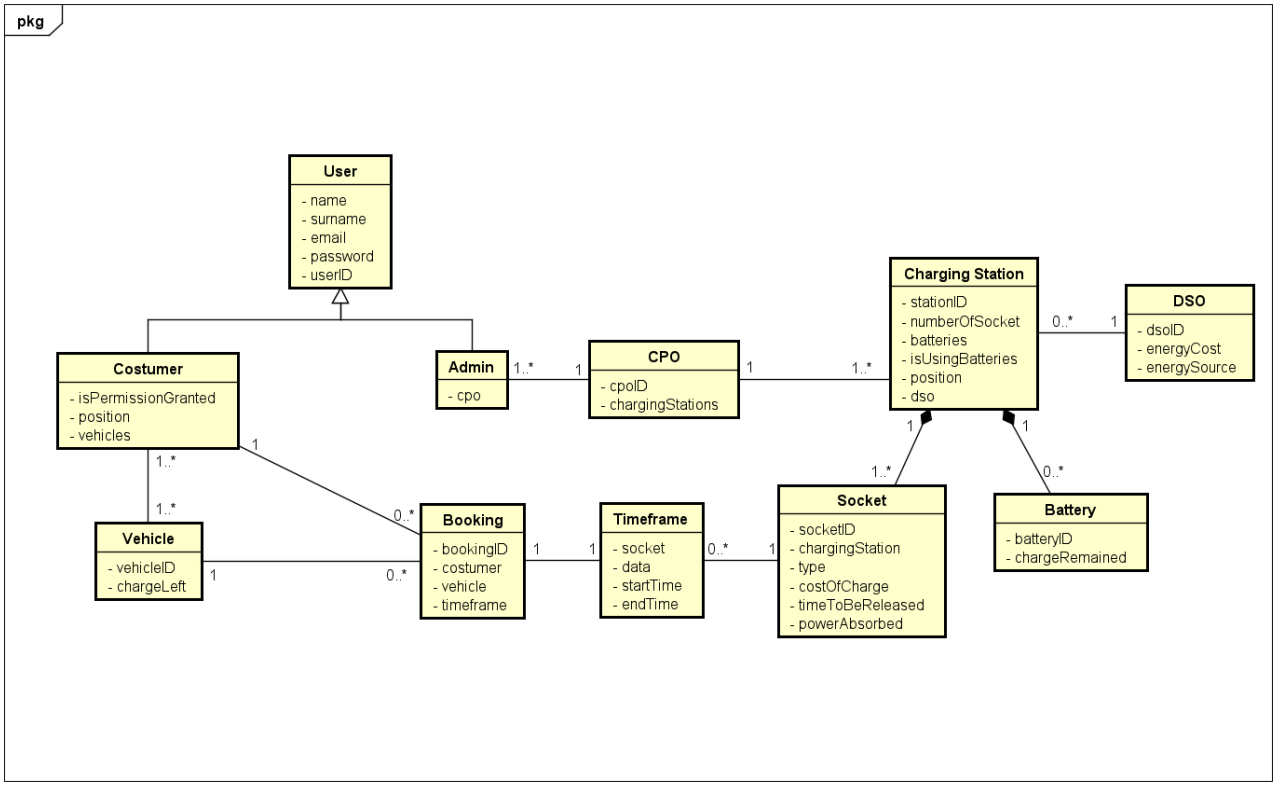
S4. While handling daily Administration of its charging station, a CPO's employee sees from CPMS that the cost of the energy increased a lot. So, he decides to change Distribution System Operator. He compares prices and sources of the available DSOs and choose the most affordable one. During the change from the old DSO to the new one, he makes the charging station related use the energy stored in the present batteries.

S5. Today is Black Friday, so the CPO of the charging station of Milan decides to publish a special offer for all Users. Only for today the charge will be paid 30% off. Fernando receives the notification from eMSP, so he decides to charge all his electric vehicles.

S6. Sara, an Admin of a CPO registered to the application, checking prices of energy, notices that it has increased a lot. So, in order to save money, she decides to use the energy stored in batteries in the charging stations that permit it, until prices get low.

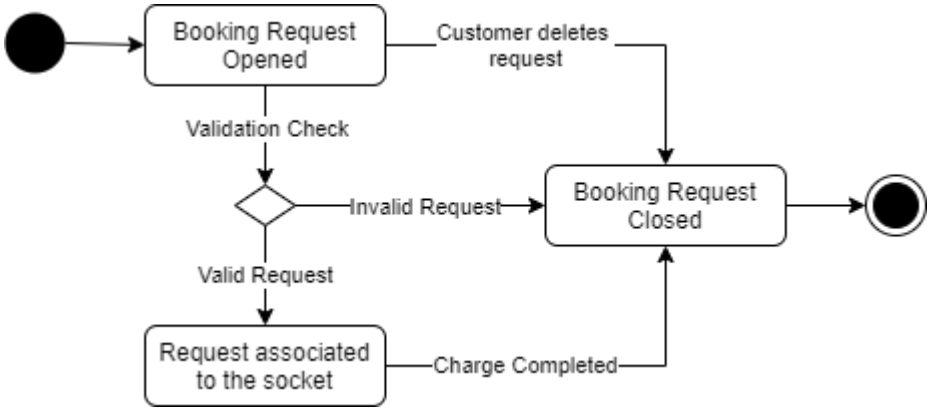
2.1.2 Class Diagram

The class diagram gives a high-level representation of the whole system:

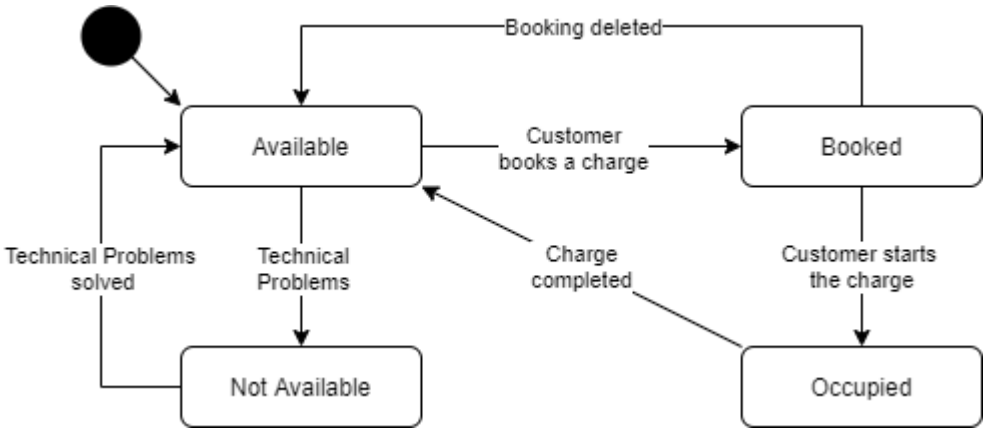


2.1.3 State diagram

The State diagrams describe the behaviour of the system, focusing on the main possible states of each object and on how they evolve when an event occurs. This analysis helps to clarify the most critical aspects of the system, while the simplest ones, such as Registration, are left out for simplicity. In the registration phase User can choose whether to share or not position and calendar.



Booking Request



Socket status

2.2 Product function

This section provides a summary of the main functions that the software will perform, with respect with the goals described in section 1:

PF1. BOOKING FUNCTION:
The app must guarantee the possibility to registered User to book a socket in a charging station. The main page of the application is a map containing all the available charging stations, distinguished by type (slow/fast/rapid) by using different icons. A Customer can insert a timeslot, a date and a location to filter the results shown in the map. When he chooses a charging station, another page with all the available timeslot for the inserted date is shown. To book the charge he just needs to select one or more timeslots and press on the book button. Once a charging station and the timeslots have been selected, User receives an OTP in order to identify himself at the charging station.

PF2. CHARGING FUNCTION
To start a charge, the Socket must scan The Customer's UserID and the OTP generated during booking. So, the Customer opens the app, goes to the "Bookings" section, and select the right one. To make the procedure easier, a new page containing the both the codes needed is shown. When User connects the vehicle to the socket, the application monitors the charging status. Once the charge is finished a notification appears on the Customer's smartphone, so that he can disconnect the vehicle and proceed to payment. The application allows payments just with the app, so the User must have inserted a Valid payment method during the registration. After the charge the amount of money spent is automatically withdraw from that payment method, so the User doesn't have to do anything.

PF3. BATTERY MONITORING FUNCTION
For Users that have granted calendar and navigation system sharing, the application monitors the charge of the vehicle's battery. If it finds out that the battery charge is not sufficient to complete the planned itinerary, it sends a notification to the Customer and suggests a near station where to charge. The Customer can go to the "Notifications" section and open it. When he presses on the right notification, a new page appears, with all the information about the charge left, and the near available charging stations.

PF4. CHANGE OF PROVIDER
Through the application, an Admin of a CPO can be informed about prices and sources of energy, so he can choose from which DSO to acquire energy and select it in the application. The Admin's application contains a special section, which is not available for Customers, called "Sources", that shows a list of all the available DSO. By selecting one of them, all the information about energy provided by the DSO will appear. To choose a new provider, an Admin needs just to press the "Select as Provider" button in the DSO's page. The application will automatically carry out the change.

PF5. SWITCH BETWEEN SUPPLIERS
Through the application, an Admin of a CPO can choose whether to get energy from batteries (if present in charging station) or from DSO. The Admin's application contains another special section, which is called "My Charging Stations", that shows a list of all the charging stations managed by him. If he selects one of them in the list, and a new page will appear with all the information about it. If batteries are present, also their status is shown here. Through this page, he can switch from batteries to DSO and vice versa.
Each change is followed by an authorization notification, that must be confirmed.

2.3 User characteristics

1. Unregistered User: Person that is not registered jet and need to create account to interact with the application. To create an account, he just needs to insert First Name, Last Name, email and a password, but to actually use the application, he must insert the information about at least one electric vehicle and a valid payment method.

2. Customer: Person who has already registered a client account and owns an electric vehicle. He can use all the Services of the application (excluding the ones reserved for Admin).

3.Admin: employee of a CPO that has been pre-registered from the CPO to the application. Through the application, he can enter special functionalities, that allow him to check the status of his charging stations, to modify energy costs, to publish special offers, and to change energy supplier.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain assumptions

DA1.	All Users have the possibility to access to the application, with stable internet connection and the possibility to share position.
DA2.	GPS modules of Customers' smartphones are working properly.
DA3.	Each User has a unique UserID, which is used to recognise him or book charges.
DA4.	There are connected APIs which are used to handle communication between eMSP, CPOs, Users, and DSOs.
DA5.	During the registration phase, Customers insert correct data.
DA6.	Data coming from DSO are assumed to be correct.
DA7.	Timeframes are organized in half hours, so a Customer must book a fixed time slot, which can be made of one or more half of hour in the same booking (the User will pay just for the time used). If a User books a charge and doesn't use it, he will pay the whole time booked.
DA8	User can connect their vehicle to the smartphone

2.4.2 Dependencies

Identifier	Description
D1.	The Application will use the GPS of the Customer's smartphone.
D2.	The Application will use the Internet connectivity of the Customer's smartphone.
D3.	Sockets will use an external service to scan UserID and OTP.
D4.	The Application will User an external service to calculate itinerary and timing is accurate.

3.Specific Requirements

3.1 External interfaces

3.1.1 User Interfaces

The User interface of eMall is an application available both on smartphone and web browser. The app must be compatible with the main operating systems. Clients can use it in both versions, but to get all the features it must be used on smartphone (position access, status of battery and others are not available on web).
Admin can use the smartphone app only to receive notifications and other monitoring functions (such as to check charging station's status) Every action that modifies the system must be done on the web browser of a secure network, in order to ensure the safety of the infrastructure.

3.1.2 Hardware Interfaces

Since all the process require an internet connection, every component must have the possibility to connect to internet. Therefore, every charging station needs to be connected and every User. Moreover, the sockets must be able to recognize the amount of power required by the vehicle's battery.

3.1.3 Software Interfaces

The application must interact with a navigation system and with the bank system to permit payments.
Each Admin must have the access to the safe network (extranet) of the CPO for which he works.

3.1.4 Communication Interfaces

All systems (CPO, DSO, eMSP) must interact through API interfaces to guarantee the communication.

3.2 Functional Requirements:

For CUSTOMER:

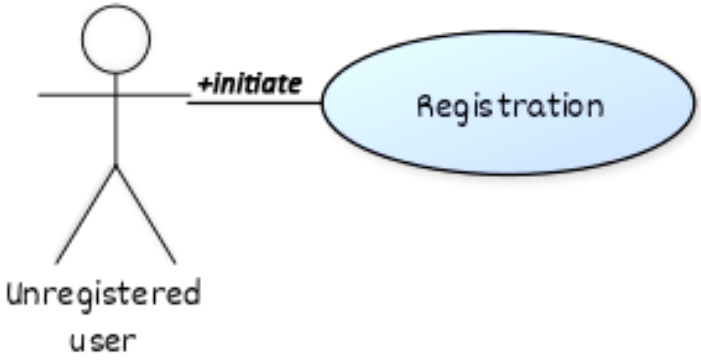
FR1.	The System must allow Customer to register an account by filling a registration form.
FR2.	The System must allow Customer to login to the application by entering his email and password.
FR3.	The System must allow Customer to insert or modify data about his vehicles.
FR4.	The System must allow Customer to modify his personal data (FirstName, LastName, payment method).
FR5.	The System must allow Customer to see the map with all nearby stations.
FR6.	The System must allow Customer to get information about a specific charging station
FR7.	The System must allow Customer to filter the map with specifics about location, date and time.
FR8.	The System must allow Customer to book a charge in the chosen charging station and timeslot.
FR9.	The System must allow Customer to delete a booking.
FR10.	The System must be able withdraw the money requested for the charge.
FR11.	The System must be able to notify the User when the charging level of his vehicle is low.
FR12.	The System must be able to calculate where to stop for charges according to the planned itinerary.
FR13.	The System must be able to monitor charging process and notify User when charging operation is completed.

For ADMIN:

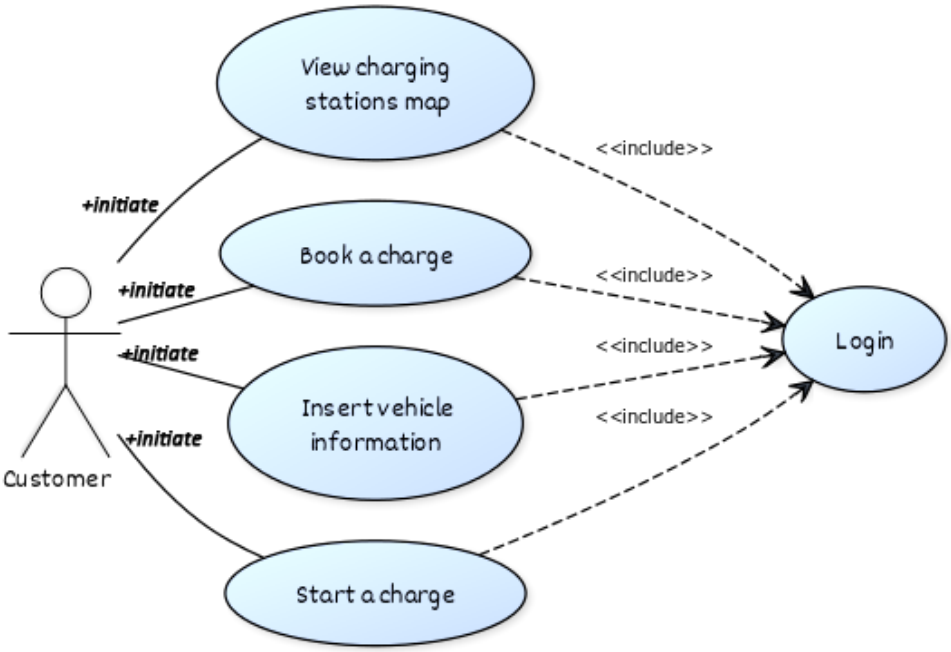
FR14.	The System must allow Admin to login by entering his email and password.
FR15.	The System must allow Admin to check the status of the charging stations he manages:
FR15.1	The System must allow Admin to check the internal status.
FR15.2	The System must allow Admin to check the external status.
FR16.	The System must allow Admin to get information about energy costs and sources.
FR17.	The System must allow Admin to insert special offers in a charging station.
FR18.	The System must allow Admin to modify energy prices in a charging station.
FR19.	The System must allow Admin to select from which DSO to acquire energy.
FR20.	The System must allow Admin to check batteries status and to select if to use energy from batteries or not.
FR21.	The System must allow Admin to insert data about a new charging station assigned to him.
FR22.	The System must allow Admin to modify data about an already existing charging station.

3.2.1 Use Cases Diagrams

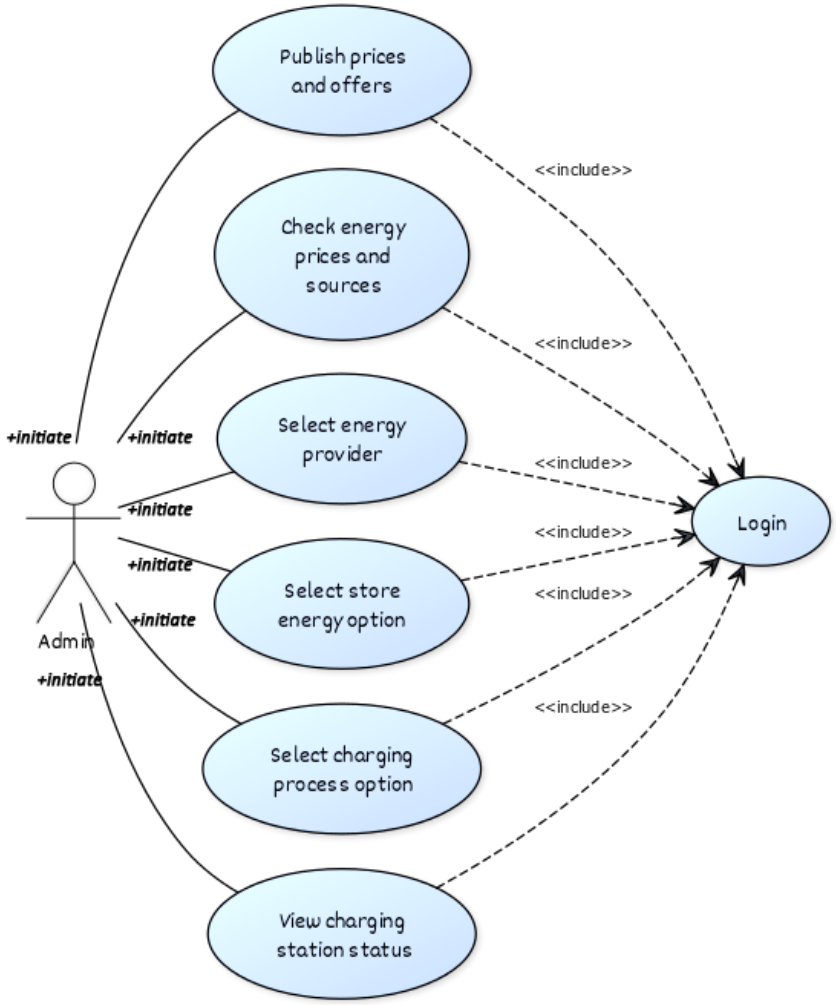
1. Unregistered User



2. Customer User



3. Admin User



3.2.2 Use cases

UC1. User registering an account on eMSP

Actor	Unregistered User
Entry conditions	A User needs to register an account to book charging stations
Event flow	1. The User opens the web site or the smartphone app 2. The User goes in the registration page and enters his data
Exit condition	The registration process is completed
Exceptions	-

UC2. Users share their calendar and position

Actor	User(client)
Entry conditions	The User wants to have access to all the features offered by eMSP
Event flow	1. The User opens the setting page of the application 2. The User selects to give the application permission to access the GPS and calendar
Exit condition	Share process is completed
Exceptions	-

UC3. Customer views location of charging stations

Actor	**User(client)**
Entry conditions	The User wants to see where the charging stations are located
Event flow	1. The User opens eMSP application 2. The User will be shown a map with symbols indicating the position of the charging stations 3. If the User has given the gps permission, he will be shown his position on the map and the closest charging stations
Exit condition	-
Exceptions	-

UC4. Customer books a charge from eMSP application

Actor	User(client)
Entry conditions	The User wants to book a charging station
Event flow	1. The User selects the charging station he would like to book 2. The User will be shown available timeframes of the selected charging station 3. The User selects a timeframe and gives the confirmation
Exit condition	Booking process is completed
Exceptions	-

UC5. Customers inserts vehicle's information

Actor	User
Entry conditions	User wants to insert info of his vehicles
Event flow	1. The User selects the option to insert vehicle info 2. The User inserts info about the vehicle (brand, model, car plate (if it has one))
Exit condition	User has a new vehicle in his profile
Exceptions	-The model of the User's vehicle is not recognized by the app

UC6. Customers start a charge

Actor	User
Entry conditions	User wants to start to charge a vehicle
Event flow	1. The User physically connects the car to a socket of the charging station 2. The User selects the option to start the charging process
Exit condition	The process of charge begins
Exceptions	-During the charging a physical problem occur

UC7. Admin publishes prices and special offers

Actor	User (Admin)
Entry conditions	True
Event flow	1. Admin opens the app and logs in 2. He opens the page of a determined charging station 3. Admin publishes the new price or offer on the page 4. Admin clicks on "save" button to save the new price/offer
Exit condition	-Offers/prices have correctly been published -the new announcement is deleted by CPO
Exceptions	-The eMSP loses connection during the process, the Admin is notified

UC8. Admin checks energy prices and sources

Actor	User(Admin)
Entry conditions	The Admin wants to check info about DSO's prices and sources
Event flow	1. The Admin selects the option to view DSOs info 2. The Admin views the list of providers, their prices and their energy resources.
Exit condition	The Admin exit from the info page.
Exceptions	

UC9. Energy provider selection

Actor	User (Admin)
Entry conditions	The energy cost on the market change and changing option is set to "manual"
Event flow	1. The Admin is notified about the changing. 2. The Admin views the list of providers, their prices and their energy resources. 3. The Admin valuates whether the actual DSO is still the best option.
Exit condition	The Admin decides the DSO
Exceptions	-

UC10. Store energy option

Actor	User (Admin)
Entry conditions	The energy cost on the market change and changing option is set to "manual"
Event flow	1. The Admin is notified about the change. 2. The Admin views the general status of the batteries. 3. The Admin valuates whether store or not energy
Exit condition	The Admin decides the operation.
Exceptions	-

UC11. Charging process option

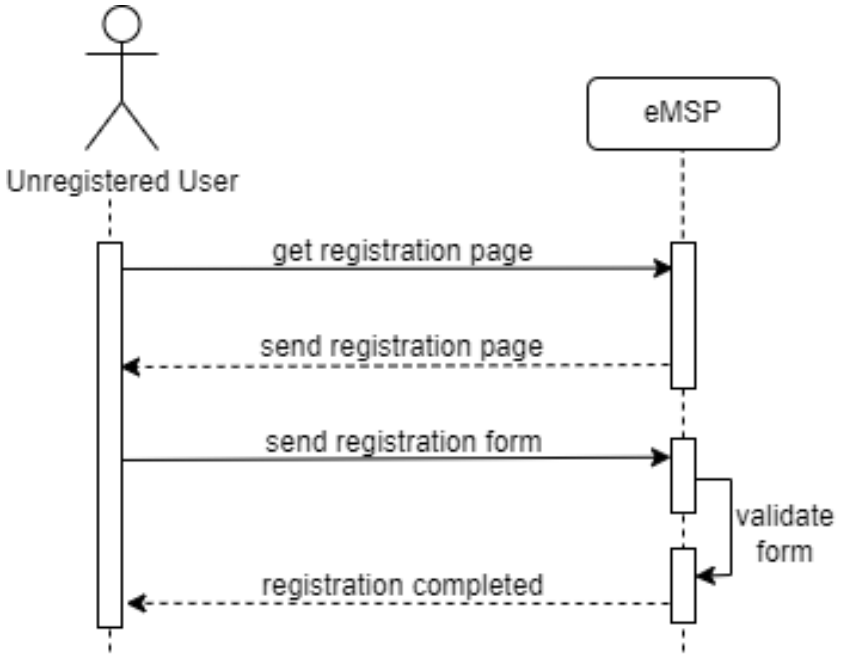
Actor	User (Admin)
Entry conditions	The energy cost on the market change and changing option is set to “manual”
Event flow	1. The Admin is notified about the change. 2. The Admin views the general status of the batteries. 3. The Admin valuates whether to use or not energy from the batteries for the charging processes
Exit condition	The Admin decides the operation.
Exceptions	-

UC12. CPO view status of a charging station

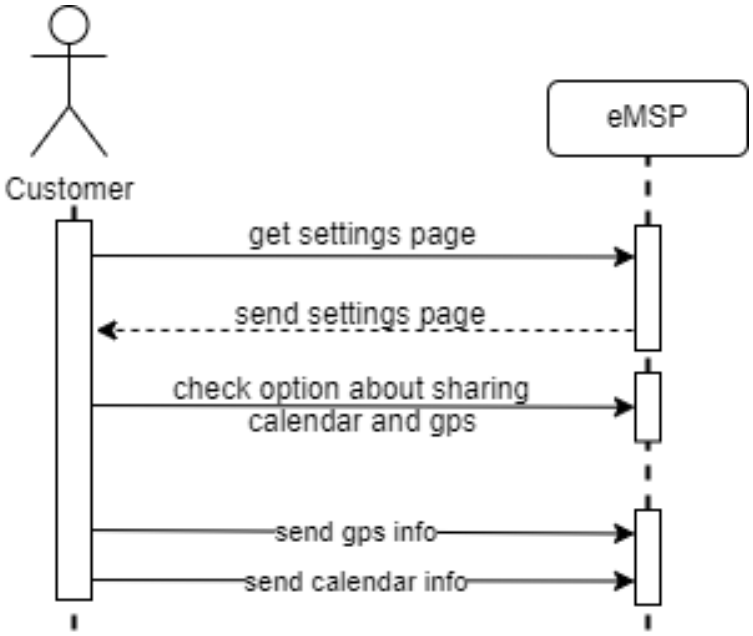
Actor	User (Admin)
Entry conditions	The Admin wants to view the status of a charging station
Event flow	1. The Admin selects the option to view the status of a charging station 2. The Admin view the list of charging stations owned by the CPO 3. The Admin selects the charging station for which wants to discover info
Exit condition	The Admin exit the page.
Exceptions	-

3.2.3 Sequence diagrams

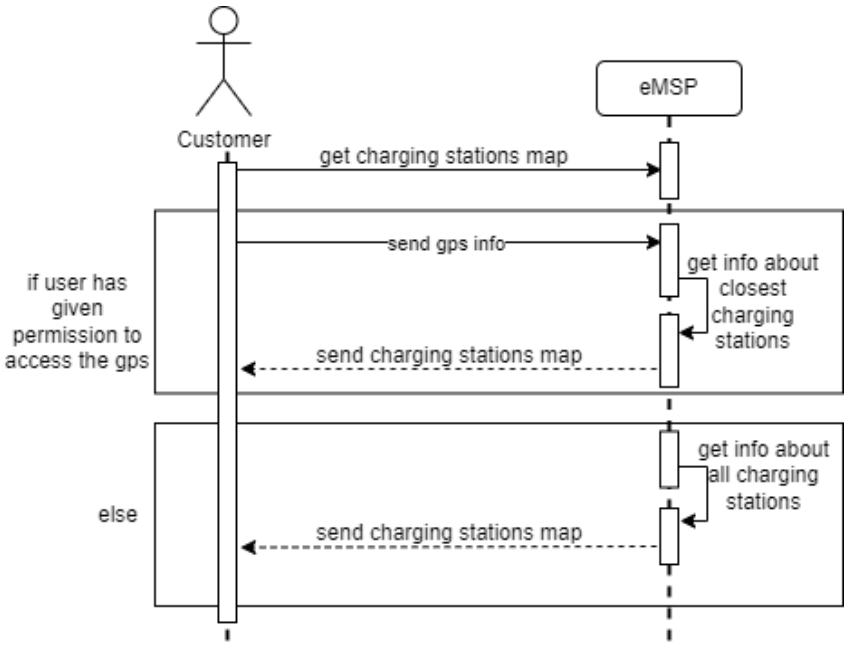
1. Customer registration



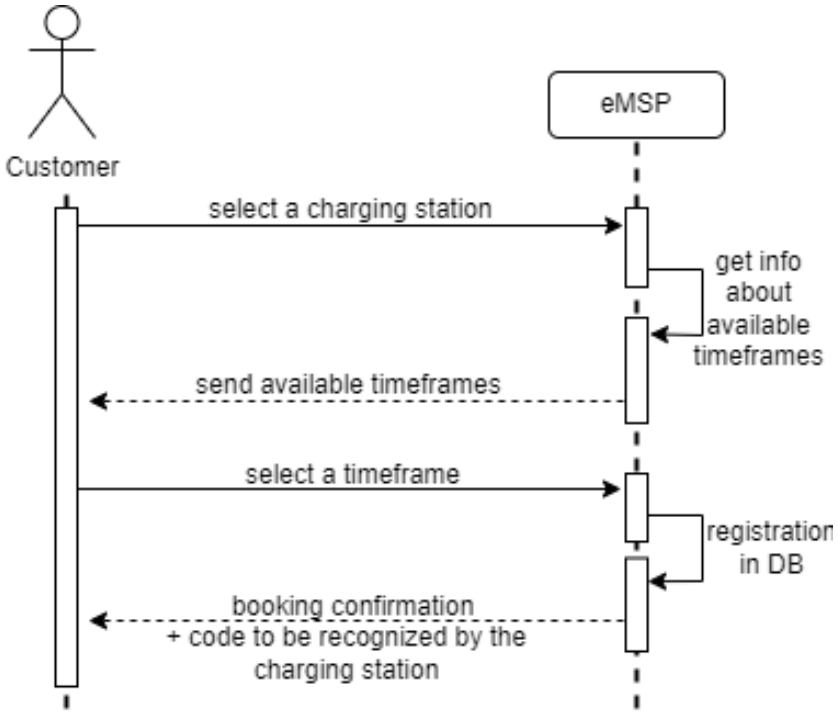
2. A Customer, that wants to enjoy the extra functionalities, share his calendar and position.



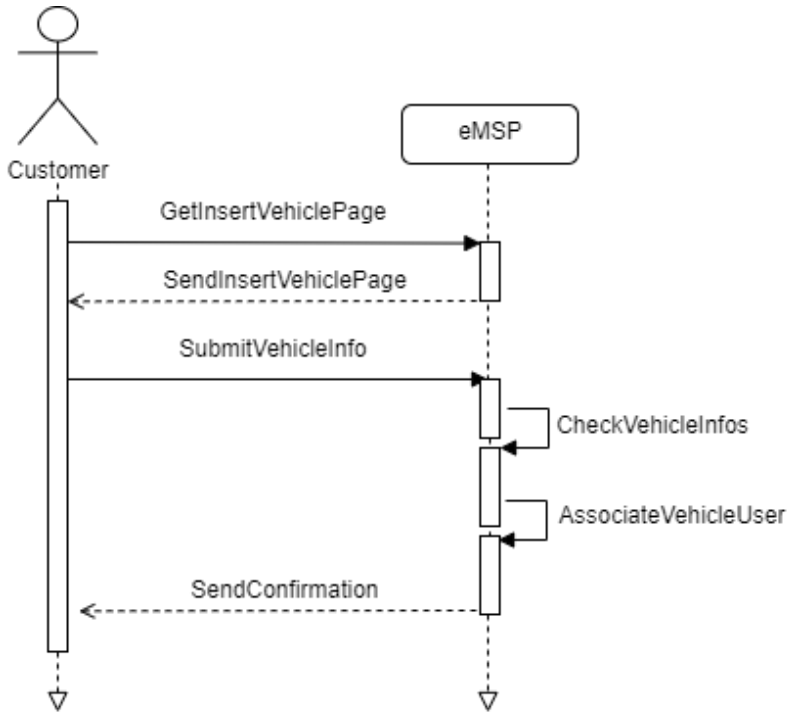
3. A Customer check the locations of nearby charging stations.



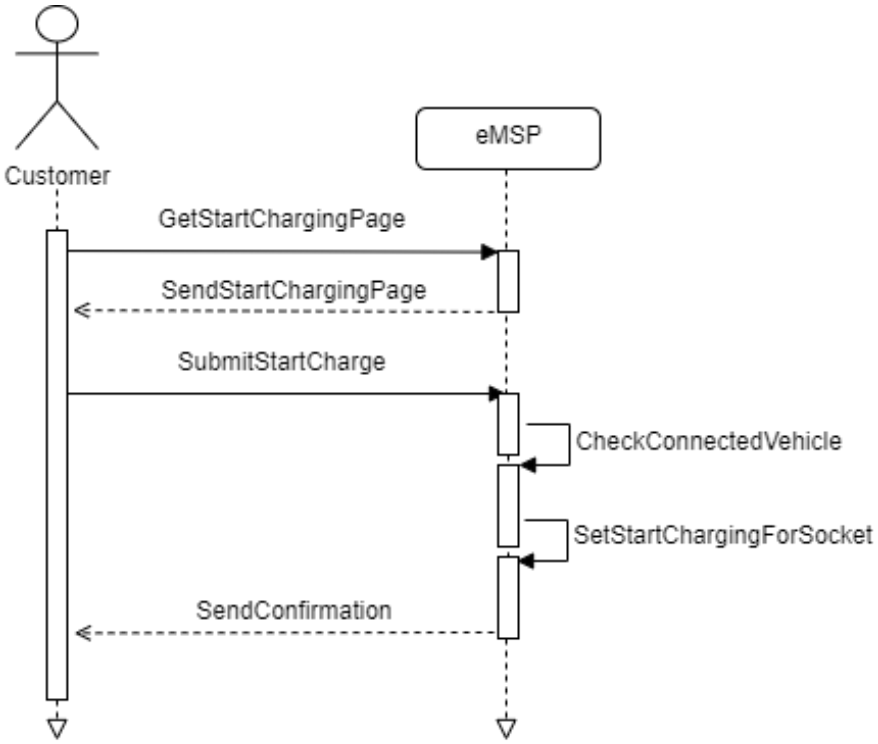
4. A Customer books a charge from eMSP application for a specific socket and timeframe.



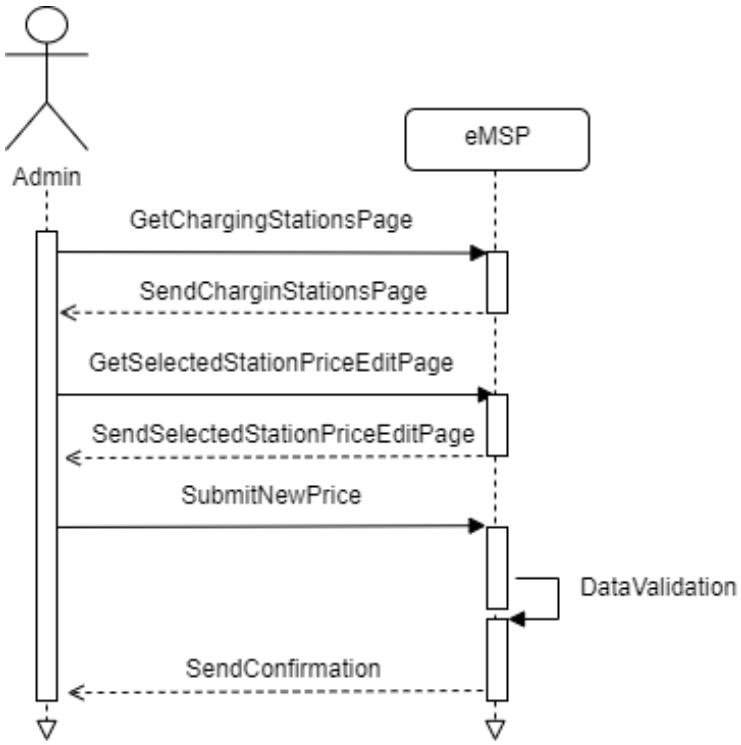
5. A Customer modifies the vehicles' information on his account.



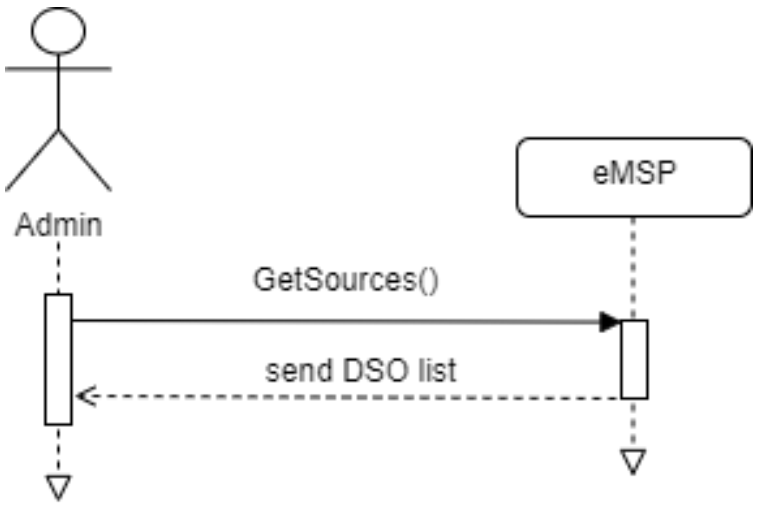
6. A Customer reaches the booked socket and starts the charge of the vehicle he has booked for.



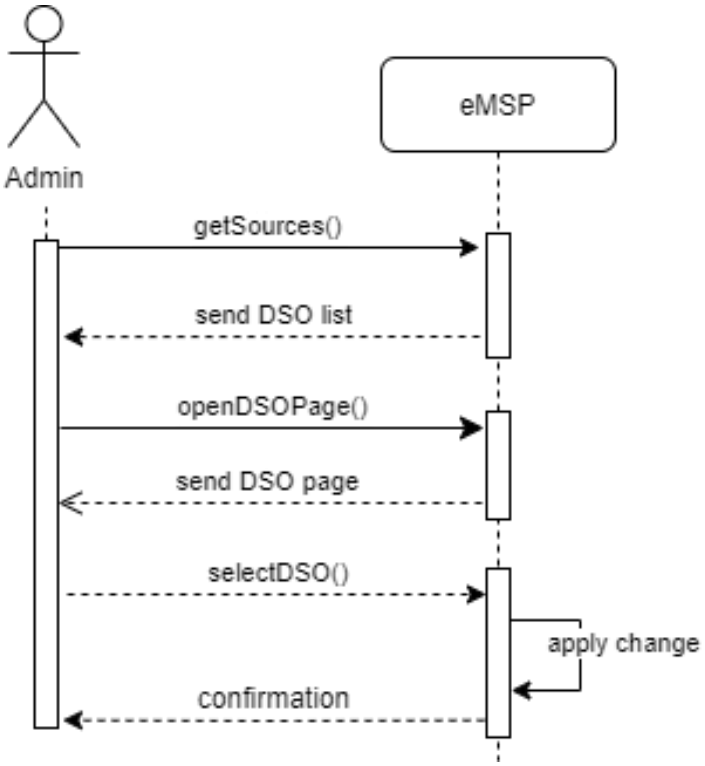
7. A CPO's employee publishes prices and special offers for a certain charging station.



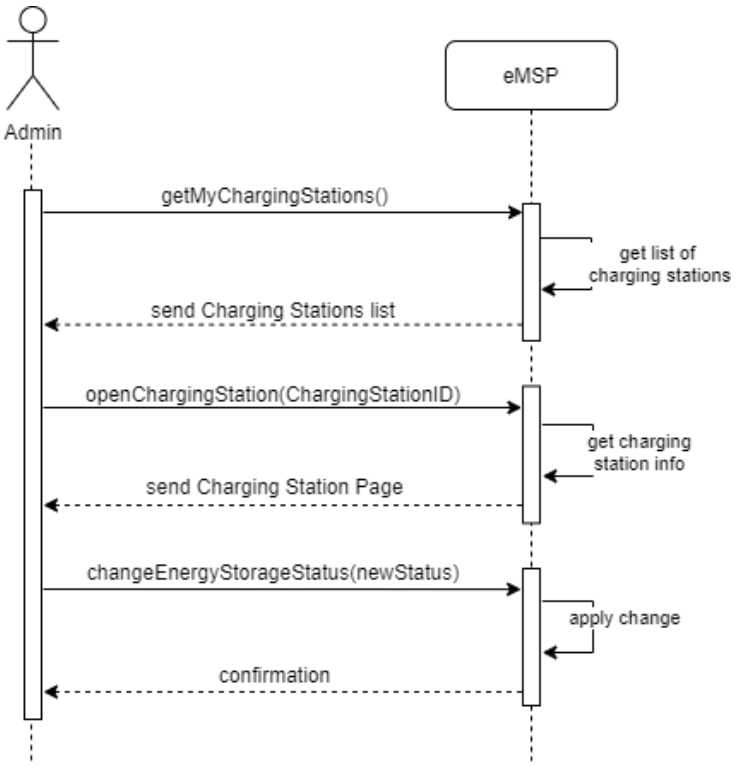
8. A CPO's employee checks energy prices and sources.



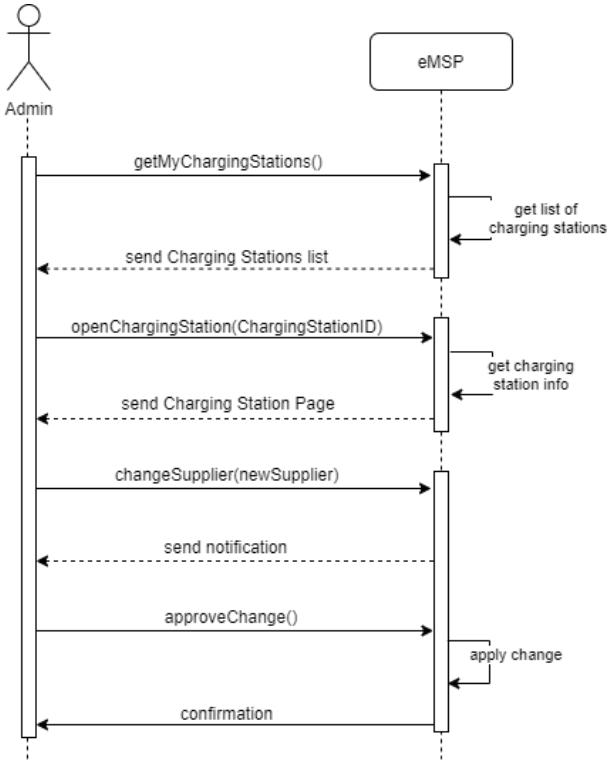
9. A CPO's employee selects from which energy provider to acquire energy.



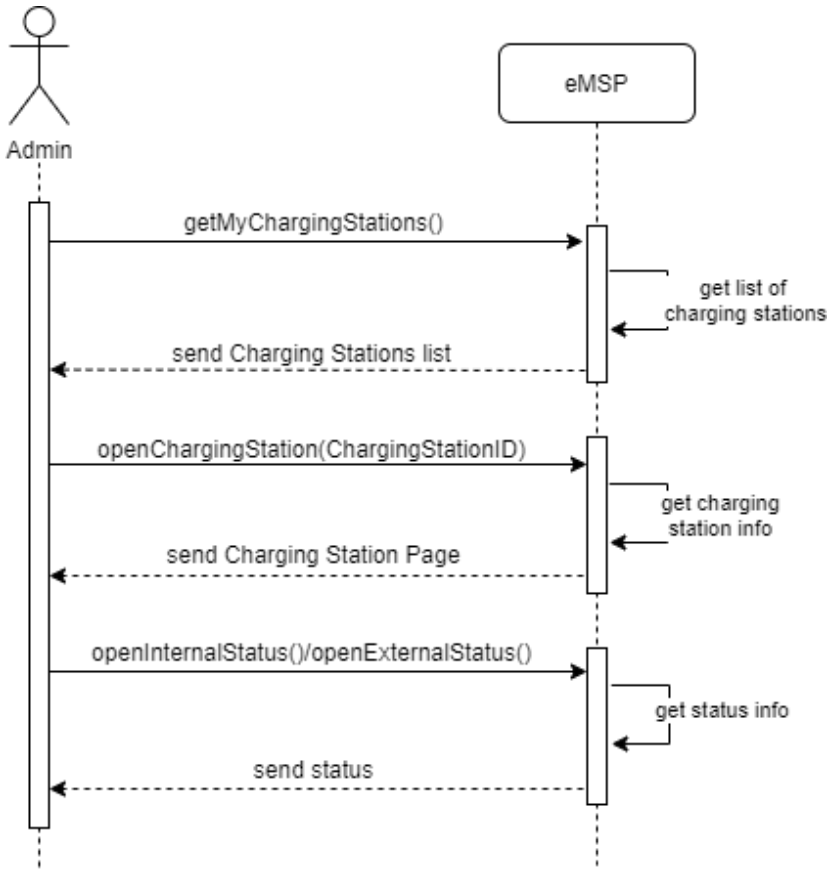
10. A CPO's employee selects whether to store energy in batteries (if they are present in the charging station) or not.



11. A CPO's employee selects whether to use or not energy stored in batteries for vehicles' charging process.



12. A CPO's employee checks the internal and external status of a charging station.



3.2.4 Mapping on requirements

This section gives a summary of the mapping between the requirements and the domain assumptions needed to guarantee the achievement of each goal.

Goal	Domain Assumptions	Functional Requirements
G1	D1,D2,D4	FR1,FR2,FR5,FR6,FR7
G2	D1,D3,D4,D7	FR1,FR2,FR4,FR8,FR9,FR10,FR13
G3	D2,D8	FR1,FR2,FR3,FR11,FR12
G4	D1,D2,D4,D6	FR14,FR16
G5	D1,D3,D4	FR14,FR15,FR20,FR22
G6	D1,D3,D4,D6	FR14,FR15,FR19
G7	D1,D3,D4	FR14,FR15.1,FR15.2
G8	D1,D3,D4	FR17,FR18,FR21,FR22

3.3 Performance requirements

The application server must be able to handle a very high number of requests, without any impact on performances. The application must rapidly update the status of bookings to avoid overlaps.

Regardless of the kind of connection between the application and the vehicle, the sharing of information must be fast.

The application must be as light as possible and must be downloadable on all compatible devices. This is possible, for example, by deleting useless animations.

3.4 Design Constraints

3.4.1 Standard Compliance

Since a lot of personal data (personal data, bank account, car plates...) are shared with the application, the privacy of Users must be granted. All data shall be treated according to GDPR.

3.4.2 Hardware Limitations

The application is available both on smartphone and web browser.

Customer: the application works on both the system, but some functionalities require the possibility to locate the device (e.g. using GPS).

Admin: the application for the Administrators is designed mainly as a web application to be used on a PC, however all the functionalities that do not require a modification of the settings (such as just reading the costs proposed by DSOs) are available also on smartphones.

3.5 Software system attributes

The system will store all the data submitted to it in a standardized form. In this way, it will be easier to catalogue, retrieve, and run queries on the data.

3.5.1 Reliability

The system should be available 24/7 in order to allow Customers to book charges at every time. However, it can be accepted to have periods of 2-4 hours in which the server is down because of maintenance, but Customers must be notified in advance.

3.5.2 Availability

The System should be up for 99,99% of the time. That means that the average time between the occurrence of a fault and service recovery (MTTR or downtime) should be not greater than 3.65 days per year.

3.5.3 Security

To ensure a secure system, eMall uses only encrypted communication protocols. In addition, the data stored in the System must be encrypted and the passwords must be hashed before being stored. Finally, the System should be protected against intrusion from agents that are not authorized to access it.

3.5.4 Maintainability

The System should be organized in modules and be well documented, in order to make maintenance, upgrades and integration easier. The code must be well documented, and a testing routine must be provided and it has to cover at least 80% of the entire codebase, excluding interfaces code.

3.5.5 Portability

The software must run on different web browsers including Google Chrome, Safari and Mozilla Firefox. In addition, it should be available for iOS and Android devices.

4 Formal analysis

4.1 Alloy code

In this section a formal analysis of the system using Alloy is presented.

```
sig Name {}
sig Surname {}
sig Password {}
sig Email {}
sig Position {}
sig EnergySource {}

sig Date {
    day : one Int,
    month : one Int,
    year : one Int,
} {
    day > 0 && day <= 31 && month > 0 && month <= 12 && year > 0
}

abstract sig Bool {}
one sig TRUE extends Bool {}
one sig FALSE extends Bool {}

sig Float{
    leftPart: one Int,
    rightPart: one Int
}

sig Time {
    hour : one Int,
    minute : one Int
} {
    hour >= 0 && hour < 24 && minute >= 0 && minute < 60
}

abstract sig Type {}
one sig Slow extends Type {}
one sig Fast extends Type {}
one sig Rapid extends Type {}
```

```
abstract sig User {
    name : one Name,
    surname : one Surname,
    email : one Email,
    password : one Password,
    UserId : one Int
} {
    UserId >= 0
}

sig Admin extends User {
   cpo : one CPO
}

sig Customer extends User {
    isPermissionGranted : one Bool,
    position : Position,
    vehicles : set Vehicle
}

sig Vehicle {
    vehicleID : one Int,
    chargeRemained : Int
} {
    vehicleID >= 0 && chargeRemained >= 0
}

sig CPO {
    cpoID : one Int,
    chargingStations : set ChargingStation
} {
    cpoID >= 0
}
```

```
sig ChargingStation {
    stationID : one Int,
    numberOfSockets : one Int,
    usingBatteries : one Bool,
    batteries : set Battery,
    position : one Position,
    dso : one DSO
} {
    (#batteries >= 0) && (stationID >= 0) && (numberOfSockets > 0) &&
((#batteries = 0 && usingBatteries = FALSE) || #batteries > 0)
}

sig Socket {
    socketID : one Int,
    chargingStation : one ChargingStation,
    type : one Type,
    costOfCharge : Float,
    timeToBeReleased : Int,
    powerAbsorbed : Float
} {
    socketID >= 0 && timeToBeReleased >= 0 && ((powerAbsorbed.leftPart > 0
|| (powerAbsorbed.leftPart = 0 && powerAbsorbed.rightPart > 0)) iff
timeToBeReleased > 0)
}

sig Battery {
    batteryID : one Int,
    chargeRemained : Int
} {
    batteryID >= 0 && chargeRemained >= 0
}

sig DSO {
    dsoID : one Int,
    energyCost : Float,
    energySource : some EnergySource
} {
    dsoID >= 0 && energyCost.leftPart >= 0 && energyCost.rightPart >= 0
}
```

```
sig Timeframe {
    socket : one Socket,
    data : one Date,
    startTime : one Time,
    endTime : one Time
} {
    startTime.hour < endTime.hour || (startTime.hour = endTime.hour &&
startTime.minute < endTime.minute)
}

sig Booking {
    bookingID : one Int,
    Customer : one Customer,
    vehicle : one Vehicle,
    timeframe : one Timeframe
} {
    bookingID >= 0
}

//Predicates and facts
fact uniqueEmails {
    no disjoint u1, u2 : User | u1.email = u2.email
}



//All charging station in different location
fact uniqueChargingStationPosition {
    no disjoint cs1, cs2 : ChargingStation | cs1.position = cs2.position
}

//All IDs are different
fact uniquePersonId {
    no disjoint u1, u2 : User | u1.UserId = u2.UserId
}

fact uniqueVehicleId {
    no disjoint v1, v2 : Vehicle | v1.vehicleID = v2.vehicleID
}
```

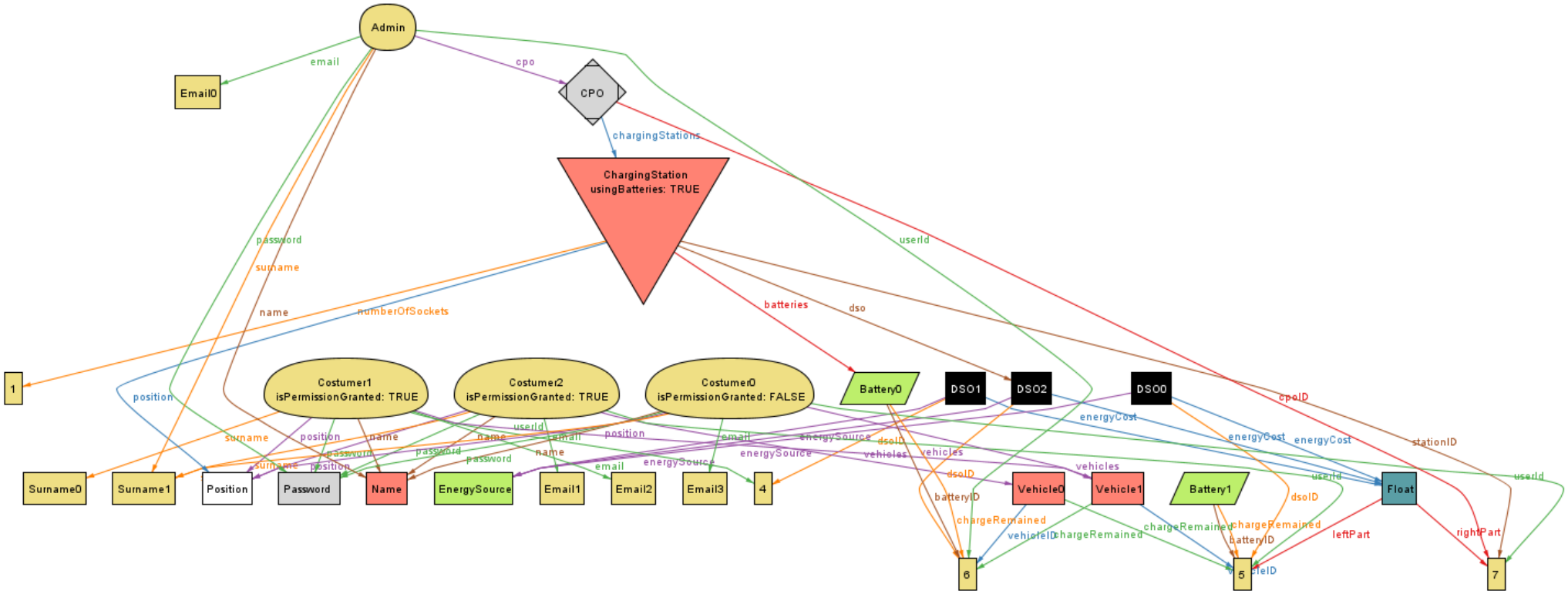
```
fact uniqueCPOId {
    no disjoint c1, c2 : CPO | c1.cpoID = c2.cpoID
}

fact uniqueChargingStationId {
    no disjoint cs1, cs2 : ChargingStation | cs1.stationID = cs2.stationID
}

fact uniqueSocketId {
    no disjoint s1, s2 : Socket | s1.socketID = s2.socketID
}

fact uniqueBatteryId {
    no disjoint b1, b2 : Battery | b1.batteryID = b2.batteryID
}

fact uniqueDSOId {
    no disjoint d1, d2 : DSO | d1.dsoID = d2.dsoID
}

fact uniqueBookingId {
    no disjoint b1, b2 : Booking | b1.bookingID = b2.bookingID
}

//Not different bookings for the same socket and timeframe
fact notDuplicateBookings {
    no disjoint b1, b2 : Booking | b1.timeframe.socket = b2.timeframe.socket &&
    (((b2.timeframe.endTime.hour > b1.timeframe.startTime.hour) ||
(b1.timeframe.startTime.hour=b2.timeframe.startTime.hour &&
b1.timeframe.startTime.minute > b2.timeframe.startTime.minute)) &&
    ((b2.timeframe.endTime.hour < b1.timeframe.startTime.hour) ||
(b1.timeframe.startTime.hour=b2.timeframe.startTime.hour &&
b1.timeframe.startTime.minute < b2.timeframe.startTime.minute))) ||
    (( (b2.timeframe.startTime.hour > b1.timeframe.startTime.hour) ||
(b2.timeframe.startTime.hour = b1.timeframe.startTime.hour &&
b2.timeframe.startTime.minute > b1.timeframe.startTime.minute)) &&
    (( b2.timeframe.startTime.hour < b1.timeframe.endTime.hour) ||
(b2.timeframe.startTime.hour = b1.timeframe.startTime.hour &&
b2.timeframe.startTime.minute < b2.timeframe.startTime.minute )))
}
```

```
//Not duplicate bookings for a vehicle in the same moment
fact notDuplicateBookingsForAVehicle {
    no disjoint b1, b2 : Booking | b1.vehicle = b2.vehicle &&
    (((b2.timeframe.endTime.hour > b1.timeframe.startTime.hour) ||
(b1.timeframe.startTime.hour=b2.timeframe.startTime.hour &&
b1.timeframe.startTime.minute > b2.timeframe.startTime.minute)) &&
    ((b2.timeframe.endTime.hour < b1.timeframe.startTime.hour) ||
(b1.timeframe.startTime.hour=b2.timeframe.startTime.hour &&
b1.timeframe.startTime.minute < b2.timeframe.startTime.minute))) ||
    (( (b2.timeframe.startTime.hour > b1.timeframe.startTime.hour) ||
(b2.timeframe.startTime.hour = b1.timeframe.startTime.hour &&
b2.timeframe.startTime.minute > b1.timeframe.startTime.minute)) &&
    (( b2.timeframe.startTime.hour < b1.timeframe.endTime.hour) ||
(b2.timeframe.startTime.hour = b1.timeframe.startTime.hour &&
b2.timeframe.startTime.minute < b2.timeframe.startTime.minute )))
}

//There is only one CPO which manage a charging station
fact notDifferentManagers {
    no disjoint c1, c2 : CPO | no disjoint cs1, cs2 : ChargingStation | cs1 = cs2 &&
cs1 in c1.chargingStations && cs2 in c2.chargingStations
}

//All batteries are linked to a Charging Station
fact allBatteriesLinked {
    all b : Battery | one cs : ChargingStation | b in cs.batteries
}

//All Charging Stations have different batteries
fact oneBatteryBelongsToOneChargingStation {
    all b : Battery | no disjoint cs1, cs2 : ChargingStation | b in cs1.batteries && b
in cs2.batteries
}

pred world1 {
    #Customer = 3
    #ChargingStation = 1
    #Vehicle < 6
    #Booking < 3
    #Socket < 3
```

```
    #Timeframe < 7
    #DSO > 2
    #CPO < 2
    #Admin = 1
}

run world1 for 4
```

4.2 Generated alloy world

In the next page, a world generated by Alloy Analizer with. In this world the only charging station present has batteries and it is using it. Furthermore, is showed how a vehicle can be shared between multiple Customers.


Admin
Email0
email
cpo
CPO
chargingStations
ChargingStation
usingBatteries: TRUE
password
surname
name
numberOfSockets
userId
batteries
dso
1
Costumer1
isPermissionGranted: TRUE
Costumer2
isPermissionGranted: TRUE
Costumer0
isPermissionGranted: FALSE
Battery0
DSO1
DSO2
DSO0
cpoID
energyCost
stationID
position
Surname0
Surname1
Position
Password
Name
EnergySource
Email1
Email2
Email3
4
userId
energySource
dsoID
vehicles
batteryID
Vehicle0
Vehicle1
Battery1
Float
chargeRemained
vehicleID
leftPart
rightPart
6
5
7

5 Effort spent

Task	Time Spent (h)
Discussion about the Assignment	1
Introduction	4
Overall Description	3
Definition of Scenarios	5
Specific Requirements	7
Use Case and sequence Diagrams	6
Formal Analysis using Alloy	6
Revision	5
Total	37