RASD assignment 1

Filippo Cinfrignini, Davide Esposito

November 2022

Contents

1 Introduction

1.1 Purpose of the Document

The purpose of this project is to deliver the Requirement Analysis and Specification Document (RASD) for the e-Mobility for All (eMall) system-to-be. The RASD describes to the fullest extent the mentioned system in terms of both functional and nonfunctional requirements, and furthermore specifies the constraints that it needs to respect as well as its boundaries. On one hand this document is addressed to the developers who will implement the requirements. On the other hand, it is intended as a contractual basis for the customers. Therefore it does not contain any specific terminology if not properly defined.

1.2 Goals

Identifier	**Description**
G1	Allow drivers to know information about charging stations nearby
G2	Allow drivers to book a charge for a vehicle in a specific charging station
G3	Allow drivers to manage and consult their bookings in progress
G4	Allow drivers to control charging process
G5	Allow charging station operators to monitor the status of a station
G6	Allow charging station operators to manage a station's energy sources

1.3 Scope

Electric mobility (e-Mobility) is a way to limit the carbon footprint caused by our urban and suburban mobility needs. When using an electric vehicle, knowing where to charge the vehicle and carefully planning the charging process so that it introduces minimal interference and constraints on our daily schedule is of paramount importance.

With this in mind, we aim to develop a new system to aid the e-Mobility market, eMall. On the one hand, it will allow owners of electric vehicles to book charging stations and monitor charging processes. On the other hand, the future system will allow those responsible for charging stations to check the status of the station and manage the station's energy sources.

This document will further expand on the goals and requirements put on the system-to-be with the purpose of guiding its development.

1.3.1 World Phenomena

Identifier	Description
WP1	Drivers leave the station while their vehicle is charging
WP2	Drivers need to know where the nearest charging station is located
WP3	Drivers decides that he wants to charge his vehicle
WP4	Drivers want to save money on the cost of charging their vehicles
WP5	A driver waits for a busy station to become free
WP6	The batteries of a station are discharged
WP7	The cost of energy supplied by a DSO has increased
WP8	Drivers decide to stop the charging process before the vehicle is fully charged

1.3.2 Shared Phenomena

Controlled by the world:

Identifier	Description
SP1	A driver books a currently available charging socket
SP2	A driver cancels a booking shortly after confirmation
SP3	A charging station operator controls the occupied charging sockets
SP4	A driver enters the verifying code to unlock the socket and start with the charge
SP5	An operator checks the amount of energy available in the station batteries

Controlled by the machine:

Identifier	Description
SP6	The system notifies the driver when the charging process is finished
SP7	The system shows the costs and offers for a charging station selected by the user
SP8	The system provides the driver with a verification code after confirming the booking
SP9	The system notifies the user when he does not show up on time at the station

1.4 Definition, Acronyms, Abbreviations

1.4.1 Definitions

- Users of the system are electric vehicle owners using the platform.
- Electric vehicles charge at charging stations.
- The charging stations are owned and managed by Charging Point Operators.

1.4.2 Acronyms

Acronyms	Description
eMall	e-Mobility for All application
eMSP	e-Mobility Service Provider
RASD	Requirements Analysis and Specification Document
CPO	Charging Point Operator
CPMS	Charge Point Management System
DSO	Distribution System Operator
UI	User Interface

1.4.3 Abbreviations

Abbreviations	Description
WPx	World phenomena
SPx	Shared phenomena
GX	Goal number X
DX	Domain Assumption number X
RX	Requirement number X

1.5 Revision History

- V1.0 - December 23rd 2022 : first version

1.6 Reference Documents

[1] assignment Assignment RDD AY 2022-2023_v3.pdf.

[2] assignment Lectures slides of the Software Engineering II course on https://beep.metid.polimi.it/

1.7 Document Structure

The **first chapter** contains the goals that the system should fulfill and the context in which the system is going to work. It contains also some useful information, such as definitions, acronyms, and abbreviations, that can help the reader understand the document.

In the **second chapter** is present the overall description of the system giving further details regarding the system. In this section can be found some real-world scenarios involving the system and the elements to model the environment. Are also described all the kinds of users that interact with the system. This chapter describes also the functions the system offers to the users.

The **third chapter** provides a description of the system's requirements, both functional and nonfunctional as well as their mapping with the relative goals. It also contains a preview of the mobile application's UI along with some details about the hardware and software interfaces. This chapter includes also the use cases, followed by the sequence diagram, directly derived from the scenarios of chapter two.

In the end, the **fourth chapter** presents the Alloy code used to test the soundness of the system to be. The code is complemented with the relative graphical metamodel.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

1. **Owner is looking for charging stations nearby**

   Max recently moved into his new home, and having an electric car needs a station near his home. Before moving into his new home, Max had already downloaded the eMall application, thanks to which he was aware that there are charging stations in his area. So, starting the application already downloaded on his device, the map is shown with the different charging stations available in its area marked with an indicator. With great amazement Max realizes that he has a large number of stations close to his home. Max decides to look for the information on the stations provided by the application.

2. **Vehicle owner checks if there are special offers in a charging station**

   Maria has bought a new electric car, she wants to save up money. When the battery of her car is below 30% of maximum capacity Maria checks on the eMall application if there are any convenient offers available. She opens the application installed on her phone. The application shows a map with all the stations, each marked with an indicator. Maria taps the marker of the station nearest to her home and the app opens a pop-up inside which all the main information about the station is provided, including price details. Discounted prices are highlighted and next to each of these is shown the discount percentage. Maria found a really great discount on a charging station not so far from her.

3. **Vehicle owner book a charge for his vehicle**

   Mark needs to charge his car as the next day he has to leave for a holiday with his family. After opening the application, Mark selects the usual station where he charges his car, selects the type of charging and proceeds by choosing the time slot. Fortunately, Mark found an available charging socket. Application requires credit/debit card credentials before proceeding with booking confirmation. Mark provides the information of his credit card and proceeds to book the charge. After the confirmation, the system saves the booking details and the verification code to be shown at the station. Mark alerts his wife and goes to the station to charge the vehicle.

4. **User charges his vehicle at the station and pays for the service obtained**

   John has arrived at the station in time and looks for the socket he has booked. Once he has found it, he enters the code provided at the moment of the confirmation. The code entered by john is correct, so the charging point informs him that it is possible to insert the socket in the vehicle. John connects the socket to his car and leaves the station to grab a coffee. John receives a notification that his car has finished the charging process. Hence, he comes back, disconnects the socket from the car and leaves. The system calculates the cost of the service based on the energy supplied to the vehicle and automatically takes the payment from the card inserted during the booking phase.

5. **User try to charge at the station without booking**

   Anna sees a charging station and stops by believing that she could charge her electric car. Anna parks in front of the charging socket, then she tries to connect her car but discover that the socket is locked. She then notices the instruction printed on the side of the charging point. She proceeds to read the instructions and download the app on her phone to book a reservation.

6. **User cancels a reservation**

   Anna had decided to charge her vehicle at the station near her home. In fact, after making the reservation, she had quickly prepared to leave home. Unfortunately, she was called in for an important work commitment that she can't lose. So Anna takes her smartphone, access her eMall account and from the 'My reservations' app view cancels the reservation.

7. **Operator that is responsible for a station wants to check its status**

   Nick work as an operator at one of the biggest CPO in the country, Nick wants to see the status of a particular station. Nick is already logged in as an operator. Nick searches the station he wants to check in the list of the stations owned by the CPO where he works. Once he has found the desired station he clicks on the station and a dashboard appear with all the information relevant to him. Now nick can see at a glance everything he needs.

8. **Operator selects the DSO from which acquire energy**

   Frank, an important employee of a CPO, must choose a new DSO for a station, comparing the different prices offered in the market. Then, by accessing the eMall application via his smartphone, Frank can access the list of DSOs available for his CPO stations. Comparing the different prices, Frank finds a DSO that significantly lowered the price of energy compared to the other DSOs. Then, accessing the management panel of the station concerned, Frank selects the DSO as an energy source and selects the one he had previously found.

9. **Operator monitors an ongoing charging process**

   The station in Via Roma is busy and John, an operator on duty, wants to check when the charging of the number 4 column becomes free. So John takes his smartphone and connects to the station management panel. By accessing the sockets control panel, John can see the status of all sockets in the station. Then select the affected socket and find that it is free in less than 10 minutes!

2.1.2 Class Diagram

In Figure 1 is represented the domain class diagram related to eMall. It contains all the elements of the domain in which the system operates and the interaction between such elements. The most relevant parts of the diagram are described, in order to ease the overall understanding of the domain.

- Basically the domain consists of two main users: the drivers (*Driver*), who require and obtain a charge and the operators (*Operator*), who manage and control the different charging stations. The driver accesses the system using his credentials (email and password) and keeps track of other personal information (e.g. first name and second name). The operator accesses the system through an operator code (*operatorCode*) and a password (*password*). The *Operator* works for a CPO.

- Drivers, through the system, book charges at certain stations. Each booking is identified by a code (*codeIdentifier*), which is generated by the system to successively verify that booking, in order to avoid anyone to use another user's booking. When a driver confirms the booking, he must arrive at the station within a certain time (*expirationTime*), beyond which the reservation will be cancelled.

- When a driver makes a reservation, he chooses from the different types of charge available in the station. Each charging station (*Station*) has one or more plugs (*ChargingSocket*), each of which has a type based on the power of the socket (*type* e.g fast, rapid, normal ...). As soon as a booking is confirmed, the selected socket becomes unavailable (*available*) and can be activated, i.e. start the charging process (*inUse*) only if the driver enters the correct code (*expectedCode*). The driver knows the location (*longitude* and *latitude*) and the name (*name*) of the station (*Station*).

- Operators work for a Charging Point Operator (*CPO*) and are responsible for monitoring and managing the activities of stations owned by the CPO. Each station (*Station*) is supplied by one or more batteries (*Battery*) or by the energy from a Distribution System Operator (*DSO*). The station owner *CPO* decides if and from which *DSO* acquire the energy (*in agreement with*).

- A station can work using the energy from different sources. The energy provided can be the available energy inside the batteries (*percentage*) or the energy acquired from a *DSO*.

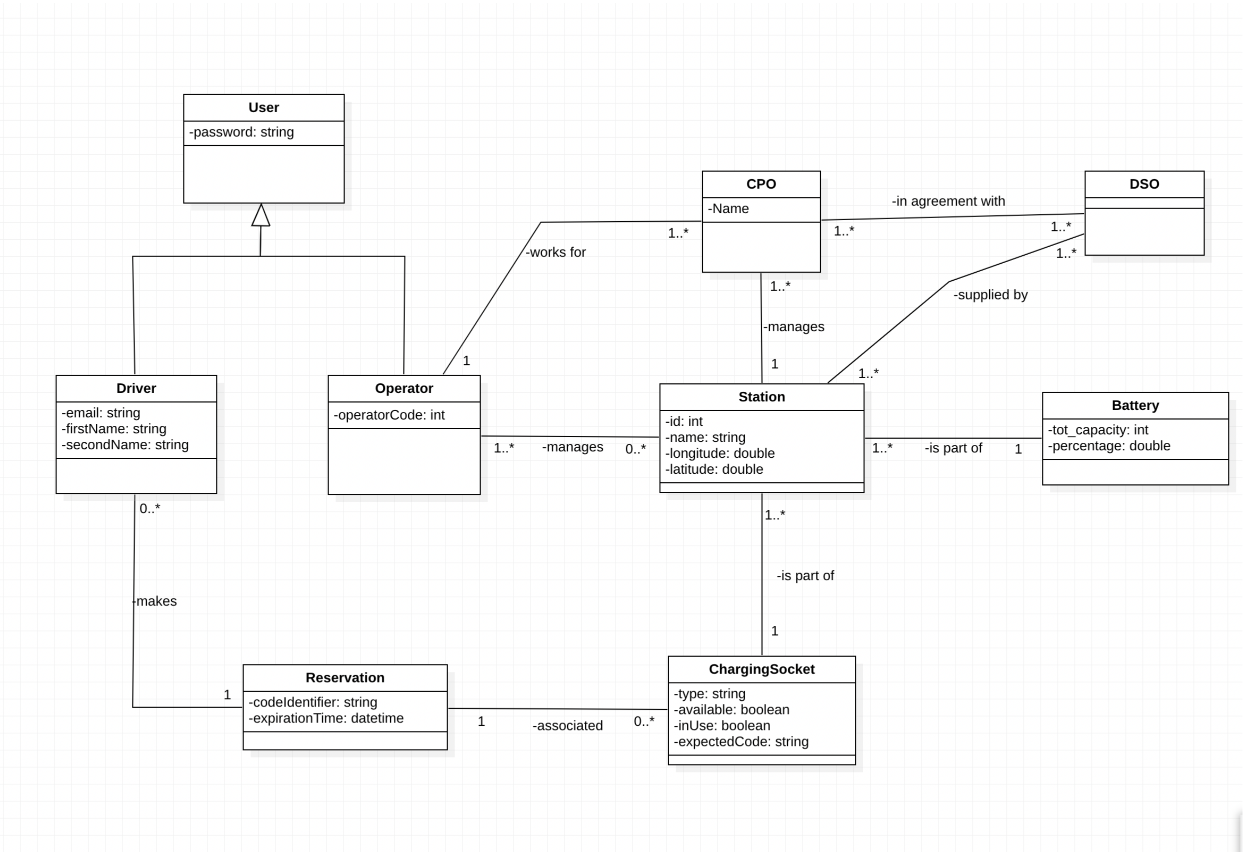


Figure 1: Domain class diagram

2.1.3 Statecharts

The state charts discussed in this section are the ones of the charging socket and of the reservation.

The reservation

From the moment a user confirms a reservation, he will have a certain time limit (*TRV* - Time Required to Validate) within which he must show up at the station. If the time expires or the booking is cancelled, then the reservation will go in the *missed* state. Otherwise, from the moment the user unlocks the charging socket by entering the verification code at the station, the reservation will be in the state *in progress* until the charging process does not end. Important to note that the reservation goes into the state *in progress* when the socket is unlocked by the user.

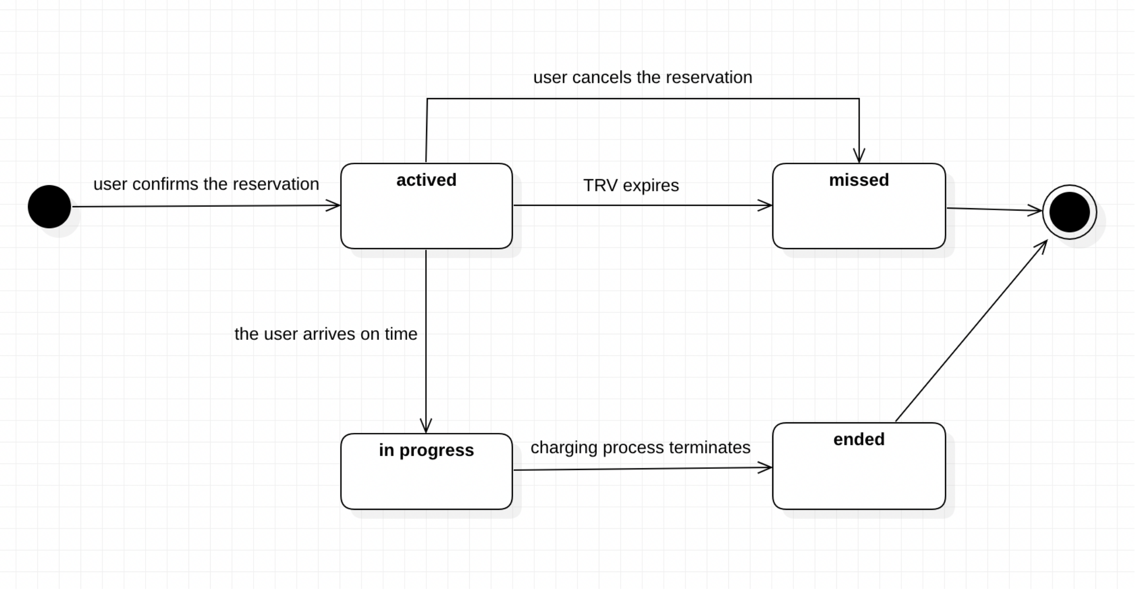

user cancels the reservation
user confirms the reservation
actived
TRV expires
missed
the user arrives on time
in progress
charging process terminates
ended

Figure 2: Reservation state chart

The charging socket

Until no socket reservation has been made, it is in the *available* state. When the socket is booked, wait for the user to arrive at the station and verify his reservation using the code provided by the system. If the time needed to verify *(TRV)* expires or the reservation has been cancelled, the socket will be available again and ready to accept another booking. Instead, if the user arrives at the station and verifies his booking on time, then the socket goes into the *unlocked* state and waits for the user to connect the socket to the vehicle. The charging process starts *(start of supply)* after the vehicle has been successfully connected and the user starts it manually through the application. The process ends *(end of supply)* when the vehicle is fully charged or when the user manually stops it via the application. The *save amount* state calculates the cost of the service obtained by the user based on the current provided.

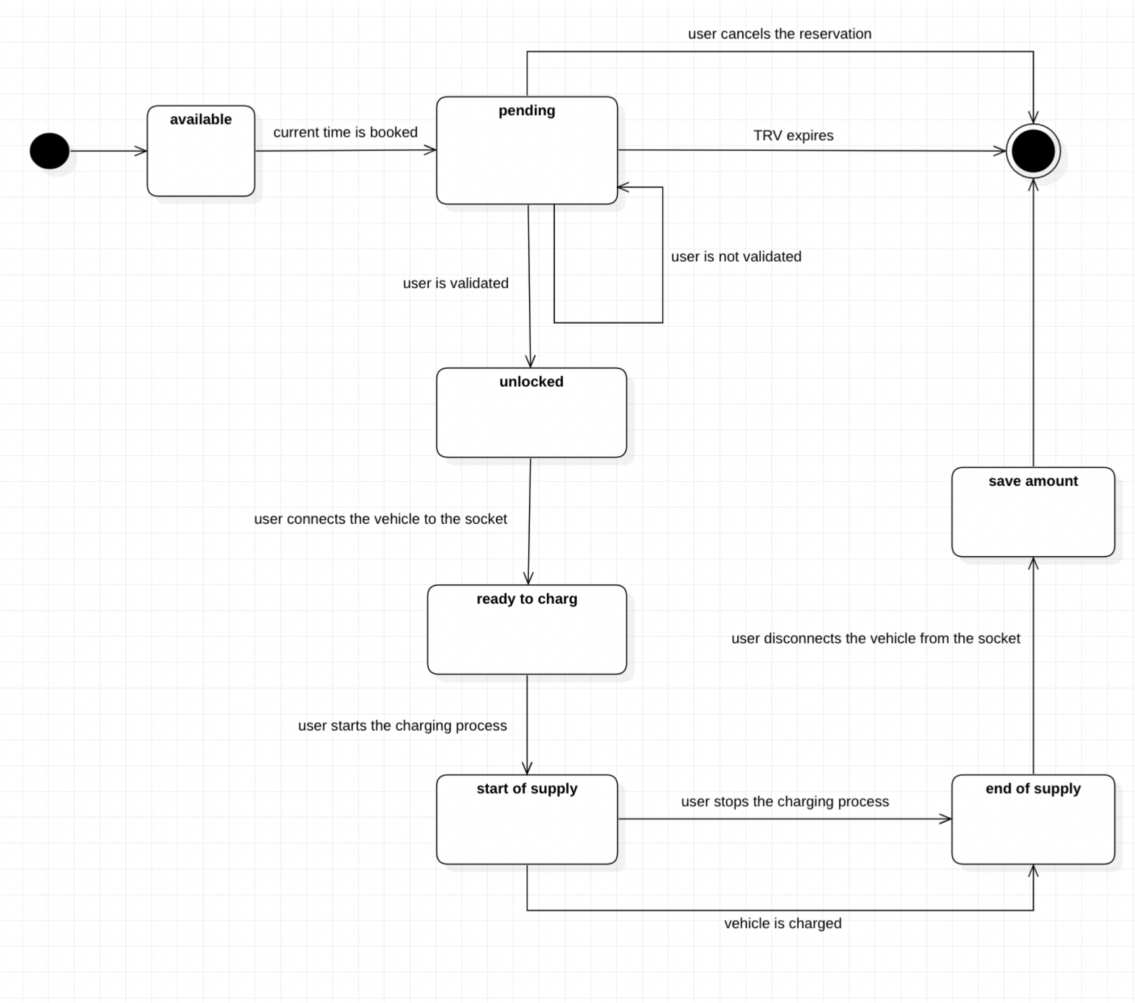

user cancels the reservation
available
current time is booked
pending
TRV expires
user is not validated
user is validated
unlocked
save amount
user connects the vehicle to the socket
ready to charg
user disconnects the vehicle from the socket
user starts the charging process
start of supply
user stops the charging process
end of supply
vehicle is charged

Figure 3: Charging socket state chart

2.2 Product Functions

Electric mobility (e-Mobility) is a way to limit the carbon footprint caused by our urban and sub-urban mobility needs. eMall, the system to be, tackles this opportunities by providing a set of functionalities with the intent of helping the users. The main functions of eMall are better described in the paragraphs beneath.

2.2.1 Provide information of nearby charging stations

One of the main functions of the eMall application is to provide information about charging stations. A Driver can find all the stations near him, and display relevant information such as: the sockets he has, the type of each outlet, the prices and discounts available for each type of charge. Through the mobile application, the user can view the location of the stations via an interactive map, where each charging station is initiated by a specific marker.

2.2.2 Management of bookings

The main function of the eMall system is to offer Drivers the possibility to book a charge. In this way it is facilitated the mobility of electric vehicles. A driver can book a socket to charge his vehicle by going directly to the station and checking his reservation. The system does not manage future reservations and these can only be made when a socket is currently available. From the moment a user makes a booking, he has a certain amount of time to show up at the station and start the charging process Drivers can also cancel their reservation if they think they won't arrive to the station on time.

2.2.3 Management and control of charging processes

Once at the station, the eMall system offers this function to the Driver. Every Driver can start charging his vehicle after having made a reservation in advance trough the eMall application. The system offer a technological solution in order to verify that a reservation has been made. The Driver can consult the app where are stored all the information needed to access this function. The driver that has made a reservation can start and can also terminate the charging process at a charging socket after having connected his electric vehicle. The vehicle will be disconnected automatically if the battery is full in order to prevent any damage to the vehicle batteries.

2.2.4 Management and control of stations

In addition to offering a service aimed at booking and managing vehicle charges, the system allows charging station operators to have access to the management and control functions of the same stations. In particular, the system allows to monitor the stations from the point of view of energy and the use of charging sockets. It is possible to monitor the status of a specific charging socket, its availability and also, if it is in use, how much energy it is supplying for that charge and the estimated time before the process ends. From the energy point of view, however, it is possible to monitor the status of the batteries of a station and the different energy suppliers available (DSOs). DSO are the entities responsible for distributing and managing energy from the generation sources to the final consumers (CPO and then stations). Energy prices vary from DSO to DSO and each CPO selects from which of these to acquire energy based on prices and contractual links between these entities. For this reason, it is important for the manages and monitors stations, to be able to compare the different prices on the market and choose from which operator to acquire energy. Operators will have access to a number of management features, such as the ability to select the energy source (battery or DSO) and decide whether to store or not energy in the batteries of the station

2.3 User Characteristics

The kinds of users that are interacting with the application and their needs. There are two kinds of users that interact with the system: drivers and operators. Below there is a brief description of the user and for each kind of user their needs.

2.3.1 Driver

The Driver is the person who is interested in charging a vehicle, it could be his vehicle or a vehicle that he is using or his vehicle but driven by someone else. The driver needs to consult the position of charging stations and wants to book a socket for charging his vehicle. Obviously the application must account for the model of vehicle the Driver has inserted in the application in order to show only the relevant charging stations and show the correct price for the service. The term Driver generalize who is using these services. This means that such user does not necessarily have to drive the vehicle inserted in the application, a Driver could book a charge for himself or someone that will go to the charging station to charge the vehicle for him.

2.3.2 Operator

The Operator is the person who works for a CPO, a company that owns charging stations. The operator is responsible for the management of the stations and has direct control over the charging stations. With this term we consider the users who have functional and administrative responsibilities on the station. The operator needs to know the status of the company's stations in real time. The operator also needs to decide whether to store or not energy in the batteries of a station, whether to use the energy available in the batteries for charging the vehicles at the station or acquire it from DSOs.

2.4 Assumptions, Dependencies and Implementation Constraints

2.4.1 Domain Assumptions

In the following are specified the assumptions made for the domain. Such assumptions are properties and/or conditions that the system takes for granted, mostly because they are out of the control of the system itself, and hence need to be verified to assure the correct behaviour of eMall.

Identifier	**Description**
D1	There exists APIs to make transactions so that drivers can make payments
D2	The transaction APIs automatically manage different currencies
D3	Each cellphone has services that can provide the position
D4	Each driver books a type of charging compatible with their vehicle
D5	Each driver always connects the socket to the vehicle after validating the reservation
D6	Each charging station has at least one battery
D7	Every station identifies charging sockets with a unique number printed on the side
D8	Each charging socket is provided of a terminal to acquire the verification code
D9	Each charging socket has only one type of charge
D10	A DSO available for a station is also available for all stations owned by the same CPO
D11	DSO and CPO make agreements before a DSO is available as a energy source for stations
D12	There is always at least one DSO available to provide energy
D13	The station receives energy from at most one DSO at the time

2.4.2 Implementation Constraints

[C1] The eMall mobile application has to ask customers to send them notifications

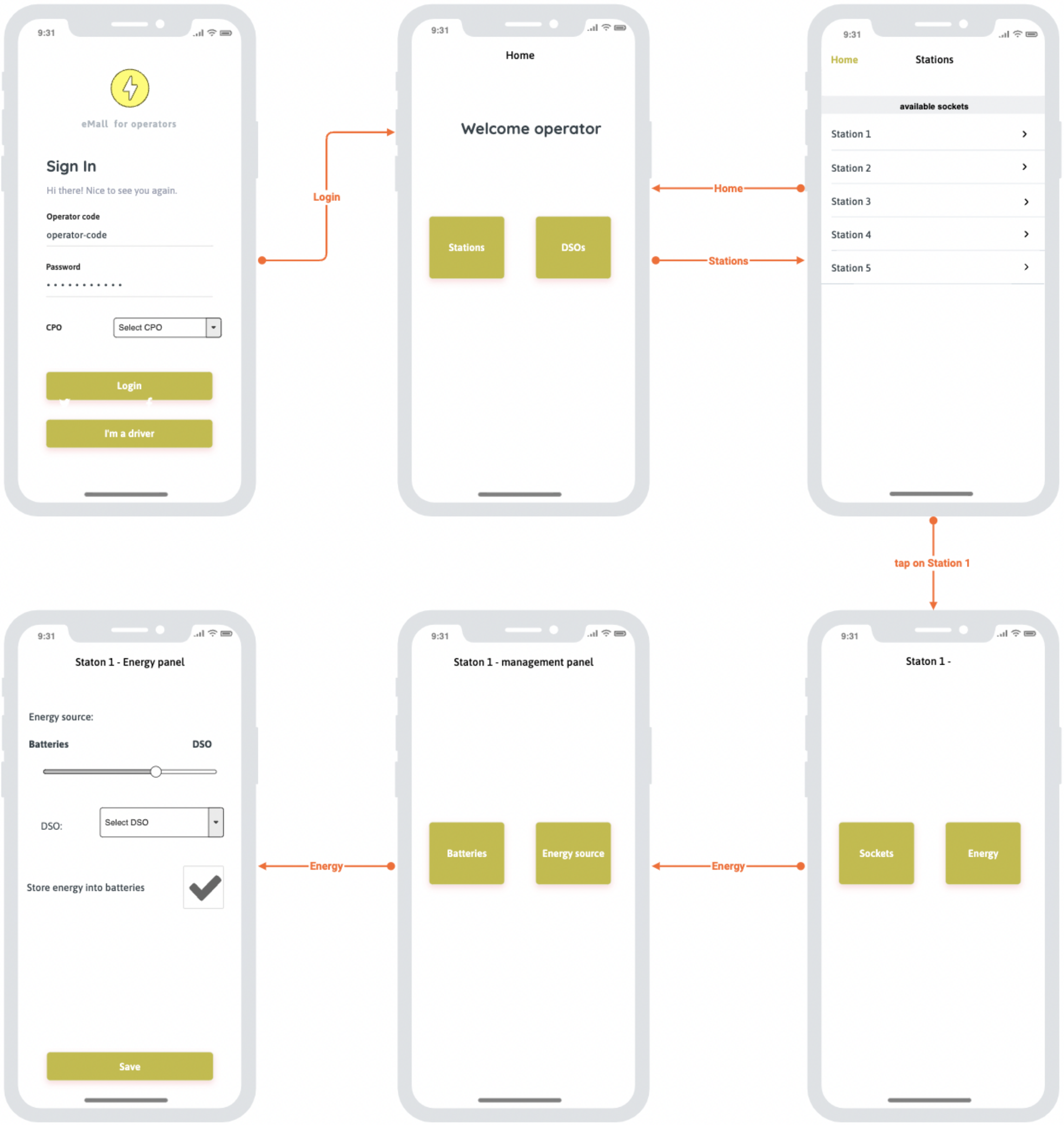
[C2] The eMall mobile application has to ask customers to access their position

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

In this section of the document is presented the mobile application UI: the views both the driver and the operatore are going to see when using the mobile application.
In particular, it shows the succession of opinions to make a booking of a charging socket by a driver user and the subsequent consultation of the details. Instead, for the operator, it shows the succession of views to access to the management panel of the energy source of a station.



Hi there! Nice to see you again.

Operator code: operator-code

Welcome operator

Figure 4: Operator UI mobile app

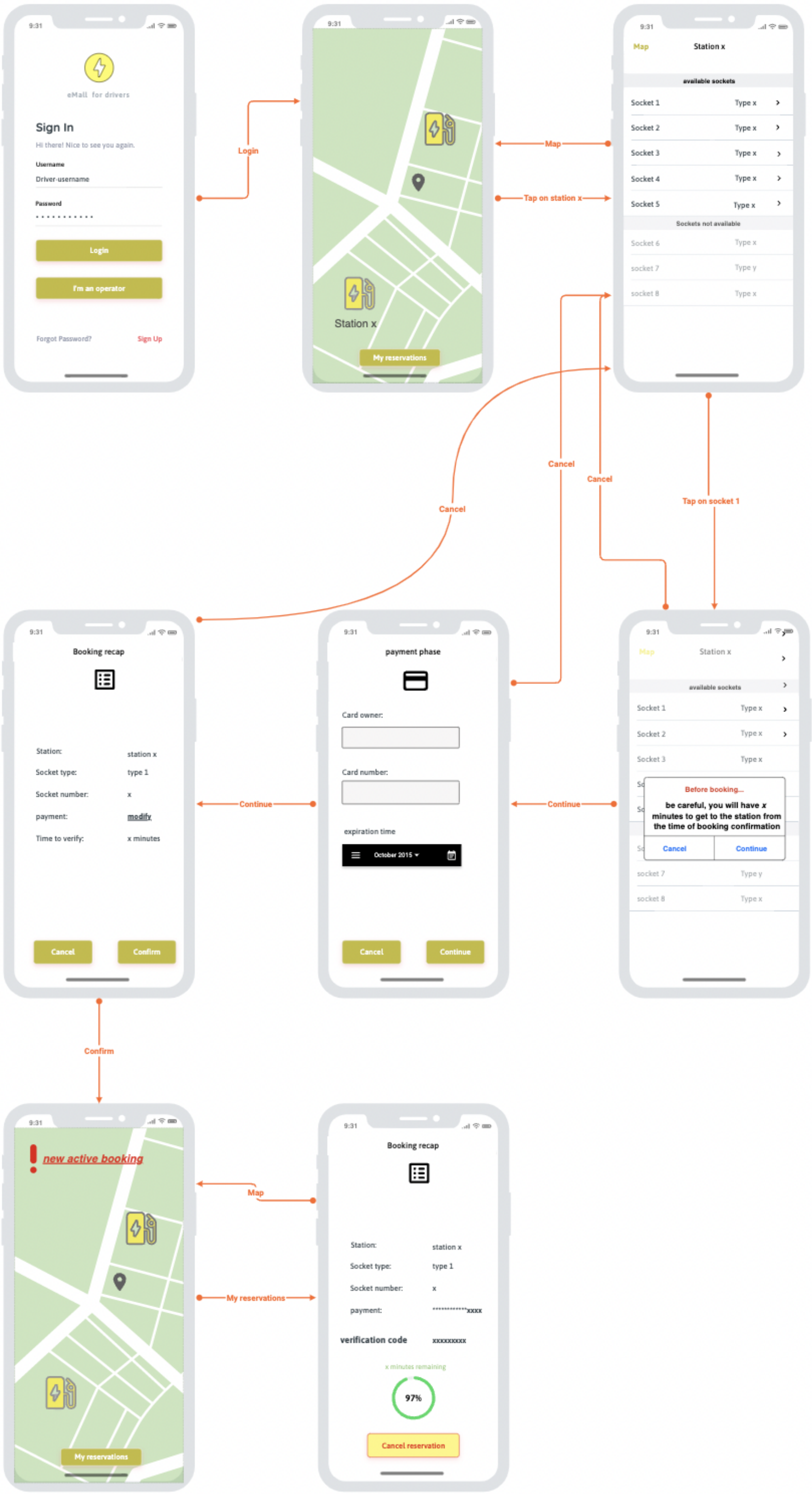


Figure 5: Driver UI mobile app

3.1.2 Hardware Interfaces

Maps
The system communicates with external APIs to provide services related to the position and to offer drivers a map of their surroundings that they can navigate.

Payment manager
The system communicates with external APIs to provide services related to different payment methods and allow the system to verify the veracity of a payment method and allow subsequent transitions so that the user can pay for the service obtained.

3.1.3 Communication Interfaces

The system exploits the internet connection for the communication to and from all devices.

3.2 Functional Requirements

In the following are specified all the requirements that the system has to fulfill. In order to work properly, the system should:

[R1] Allow drivers to log in

[R2] Allow drivers to select a charging station from a map

[R3] Allow drivers to check a charging socket availability

[R4] Allow drivers to check the socket types provided by a selected charging station

[R5] Allow drivers to consult the charging costs of a specific charging station

[R6] Allow drivers to make one booking at a time

[R7] Allow drivers to view details of a booking that has not yet expired

[R8] Allow drivers to cancels a reservation that has not yet expired

[R9] Allow drivers to insert the verification code at the charging socket booked

[R10] Allow drivers to start the charging process for a vehicle

[R11] Allow drivers to stop an ongoing charging process

[R12] Allow drivers to pay at the end of the charging process

[R13] Allow operators to log in

[R14] Allow operators to view the list of stations owned by the CPO they work for

[R15] Allow operators to consult the number of charging sockets occupied

[R16] Allow operators to monitor the estimated time until a busy socket becomes free.

[R17] Allow operators to see the list of the available DSOs.

[R18] Allow operators to consult the current price of energy of a DSO

[R19] Allow operators to select the energy source for a station (batteries, DSO or a mix thereof)

[R20] Allow operators to decide whether to store or not energy in stations' batteries

[R21] Notify drivers when their charging process is ended

[R22] Notify drivers when the time requested to unlock the socket is expired

[R23] Notify drivers the total cost of the service obtained

3.2.1 Use Cases diagrams

In the following are provided the use case diagrams deduced from the scenarios reported in paragraph 2.1.1. They help identify the actors interacting with the system and their role within each use case.

Driver Use Case Diagram

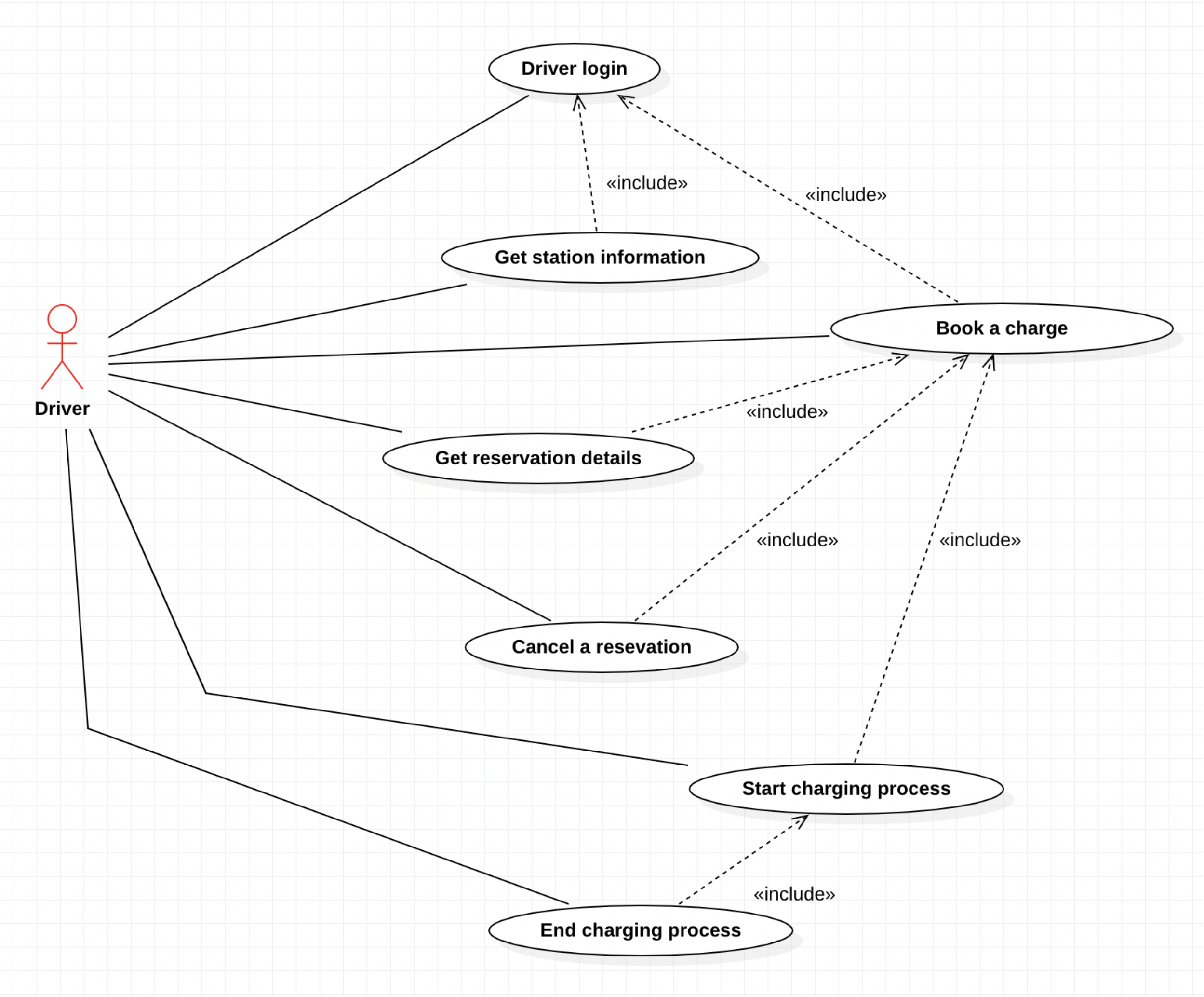


Figure 6: driver use case diagram

Operator Use Case Diagram

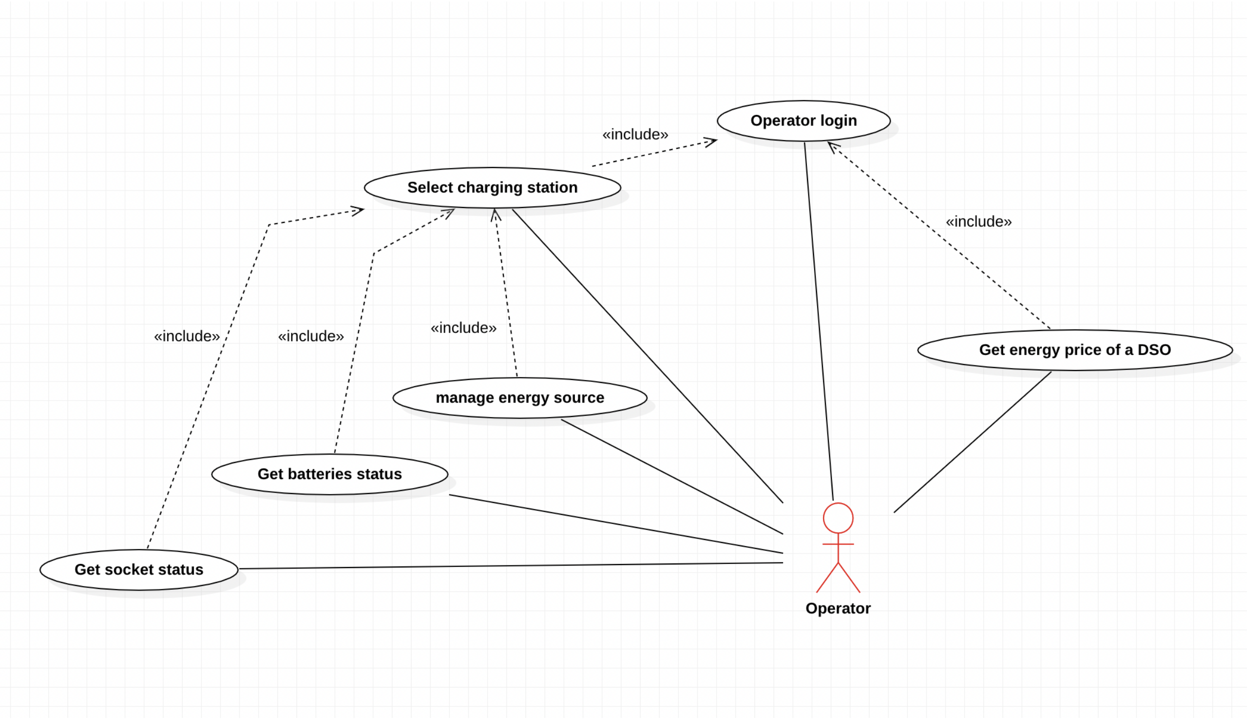


Figure 7: operator use case diagram

3.2.2 Use cases

[UC - 1] User-driver login to eMall

Name	Driver login
Actors	Driver
Entry condition	The driver *d* has installed and started the eMall application on a device
Event flow	(a) *d* presses the "I'm a driver" button (b) The system shows the login view (c) *d* inserts his username and password in the form (d) *d* clicks on the "Login" button (e) The system checks the credentials (f) The application shows the map view
Exit condition	*d* is properly accessed to his eMall account
Exception	(e-1) The data inserted are not valid. The system notifies *d*. The event flow returns to (b) step

[UC - 2] Driver consults the information of a station

Name	Get station information
Actors	Driver
Entry condition	The driver d is logged in the application, the application has access to d position
Event flow	(a) The system shows a map which contains the available charging stations identified by a marker (b) d selects the interested station (c) The system shows the list of all charging sockets of the station with its charging type. Specifically, each item of the list: • is clickable by d if the socket is currently available for a booking • is not clickable by d if the socket is not currently available for a booking (d) d selects an available socket (e) The system shows a view with the charging socket details, its cost and special offers available
Exit condition	d has seen the information of a charging socket of a specific station
Exception	(e-1) The selected station is currently unavailable. The system notifies d and returns to the main view with the updated map
Alternative	d can decide to consult only the charging type available in a specific station, without select no item of the list

[UC - 3] Driver books a charge

Name	Book a charge
Actors	Driver
Entry condition	A charging station has been selected by the driver *d*. There is no active reservation in progress (not yet expired).
Event flow	(a) *d* selects an available charging socket from the list (b) The system shows the time before the booking expires once it is confirmed (c) *d* presses the button "Continue" (d) *d* inserts the details of the payment method and presses the ’Continue’ button (e) The system checks that the inserted data are correct (f) The system shows a recap page containing the details of the reservation: • Name of the station • Number of the charging socket and its type • Payment method details • Time before the booking expires once it is confirmed (g) *d* confirms his choices, pressing the ’Confirm’ button (h) The system register the reservation (i) The system save the details of the reservation and the verification code in ’My reservations’ section of the application.
Exit condition	*d* has booked successfully
Exception	(e-1) The card details inserted are not valid. The system asks *d* to insert the correct data. (e-2) The selected socket has already been booked by another user. The system notifies *d* and returns to the entry condition (e-3) The reservation is not registered properly. The system notifies *d* and returns to the entry condition

[UC - 4] Driver consult the details of a booking

Name	Get reservation details
Actors	Driver
Entry condition	The booking that the driver *d* intends to consult must have been confirmed and must not have expired
Event flow	(a) *d* goes in 'My reservations' section of the application pressing the 'My reservations' button. (b) The system shows a view with all the information regarding the reservation: • Name of the station • Number of the charging socket and its type • Payment method details • Time left before the booking expires • Verification code
Exit condition	*d* has consulted the details of a booking

[UC - 5] Driver cancels a reservation

Name	Cancel a reservation
Actors	Driver
Entry condition	The booking that the driver *d* intends to cancel must have been confirmed and must not have expired
Event flow	(a) *d* goes in 'My reservations' section of the application pressing the 'My reservations' button. (b) The system shows the details view of the active reservation (c) *d* presses on the 'Cancels reservation' button (d) The system asks *d* to confirm the operation (e) *d* presses the 'Cancel' button (f) The system cancels the reservation
Exit condition	The reservation has been successfully cancelled by *d*

[UC - 6] Driver starts a charging process in a station

Name	Start charging process
Actors	Driver
Entry condition	The driver *d* has booked a charging process at a station. The time limit to unlock the booked socket has not yet expired (the reservation is still active). *d* is provided of the verifying code and he can access on the account with which he has booked, through the application, to start the charging process
Event flow	(a) *d* arrives at the station and he approaches the socket booked. (b) *d* inserts the verification code on the input terminal of the charging socket (c) The system notifies that the inserted code is correct (d) The system asks *d* to connect the charging socket to the vehicle (e) *d* connects the socket to the vehicle (f) The system notifies that the charging process can begin. (g) *d* presses the 'Start charge' button via the app (h) The system notifies *d* that the process has started successfully
Exit condition	The charging process starts properly providing energy to the vehicle
Exception	(e-1) The verifying code inserted is not correct. The system notifies *d* (e-2) Time expires before code entry. The system notifies *d* that the reservation is expired

[UC - 7] Driver ends an ongoing charging process in a station

Name	Stop charging process
Actors	Driver
Entry condition	The charging process booked by the driver *d* has already started
Event flow	(a) *d* presses the 'Stop charge' button via the app (b) The charging process is interrupted. the system notifies d and asks to disconnect the socket from the vehicle (c) *d* disconnects the socket from the vehicle
Exit condition	The charging process ends properly. The system calculates the energy supplied e notifies *d* of the cost of the service obtained
Exception	(e-1) The charging process is completed before the user manually interrupts it. The event flow starts from (b) step

[UC - 8] User-operator login to eMall

Name	Operator login
Actors	Operator
Entry condition	The Operator *o* has installed and started the Mall application on a device
Event flow	(a) *o* presses the "I'm an operator" button (b) The system shows the login view (c) *o* inserts his operator code and his password in the fields of the form and selects his CPO from the list of CPOs available. (d) *o* presses the "Login" button (e) The system checks the credentials inserted by *o* (f) The application shows the managing view
Exit condition	*o* has access to the services offered by the application
Exception	(e-1) The data inserted are not valid. The system notifies *o*. The event flow returns to (b) step.

[UC - 9] Operator consults the energy price of a DSO

Name	Get energy price from a DSO
Actors	Operator
Entry condition	The Operator *o* is logged in the application
Event flow	(a) *o* presses the "DSO" button (b) The system shows the list of available DSOs and the current energy price
Exit condition	*o* has seen the current energy price of a DSO

[UC - 10] Operator accesses the management panel of a station

Name	Select charging station
Actors	Operator
Entry condition	The Operator *o* is logged in the application
Event flow	(a) *o* presses the "Stations" button (b) The system shows the list of all charging stations owned by the operator's CPO. (c) *o* selects a charging station
Exit condition	*o* has access to the management panel of a station

[UC - 11] Operator checks a socket status

Name	Get socket status
Actors	Operator
Entry condition	The Operator *o* is logged in the application. *o* has access to the management panel of a station
Event flow	(a) *o* presses the 'Sockets' button (b) The system shows the list of all charging sockets of the station with its charging type and its currently availability. Specifically, each item of the list: • is clickable by *o* if the socket is currently in use • is not clickable by *o* if the socket is not currently in use (c) *o* selects a charging socket currently in use (d) The system shows a view of the ongoing charging process details. Specifically, are showed: • the estimated time for the end of the charging process • the amount of power absorbed by the vehicle
Exit condition	*o* has seen the status of a charging socket of a specific charging station

[UC - 12] Operator checks the batteries status of a station

Name	Get batteries status
Actors	Operator
Entry condition	The Operator *o* is logged in the application. *o* has access to the management panel of a station
Event flow	(a) *o* presses the 'Energy' button (b) *o* presses the 'Batteries' button (c) The system shows the list of all the batteries in the station. Specifically, for each item of the list are showed: • the max capacity of the battery • the currently percentage of energy in the battery
Exit condition	*o* has seen the status of the batteries of a specific charging station

[UC - 13] Operator changes the energy source settings of a station

Name	Manage energy source
Actors	Operator
Entry condition	The Operator *o* is logged in the application. *o* has access to the management panel of a station
Event flow	(a) *o* presses the 'Energy' button (b) *o* presses the 'Energy source' button (c) The system shows a view with the current energy source of the station. Specifically, are showed: • a 'Store energy' button which allows to decide whether to store or not energy into the station's batteries • a drop down menu to select the DSO from which acquire energy • a range slider to decide from which source available (batteries or the selected DSO) acquire energy to charge vehicles at the station. Depending on the position of the pointer you can also set a mix between the two sources. (d) *o* drags the gamma cursor pointer to the rightmost point, selecting only the DSO as the power source. item Next, o select a new available DSO from the drop down menu (e) *o* presses the 'Save' Button (f) The system notifies *o* that the operation has been successful
Exit condition	*o* has changed the energy source settings of a station properly
Exception	(e-1) After confirmation of the operation, the system is not able to apply the selected settings. The system notifies *o* and return to the entry condition (e-2) *o* decides to acquire energy from the batteries even if they are discharged. After the confirmation of the operation, the system is not able to apply the selected settings. The system notifies *o* and return to the entry condition

3.2.3 Sequence diagrams

[UC - 1] User-driver login to eMall

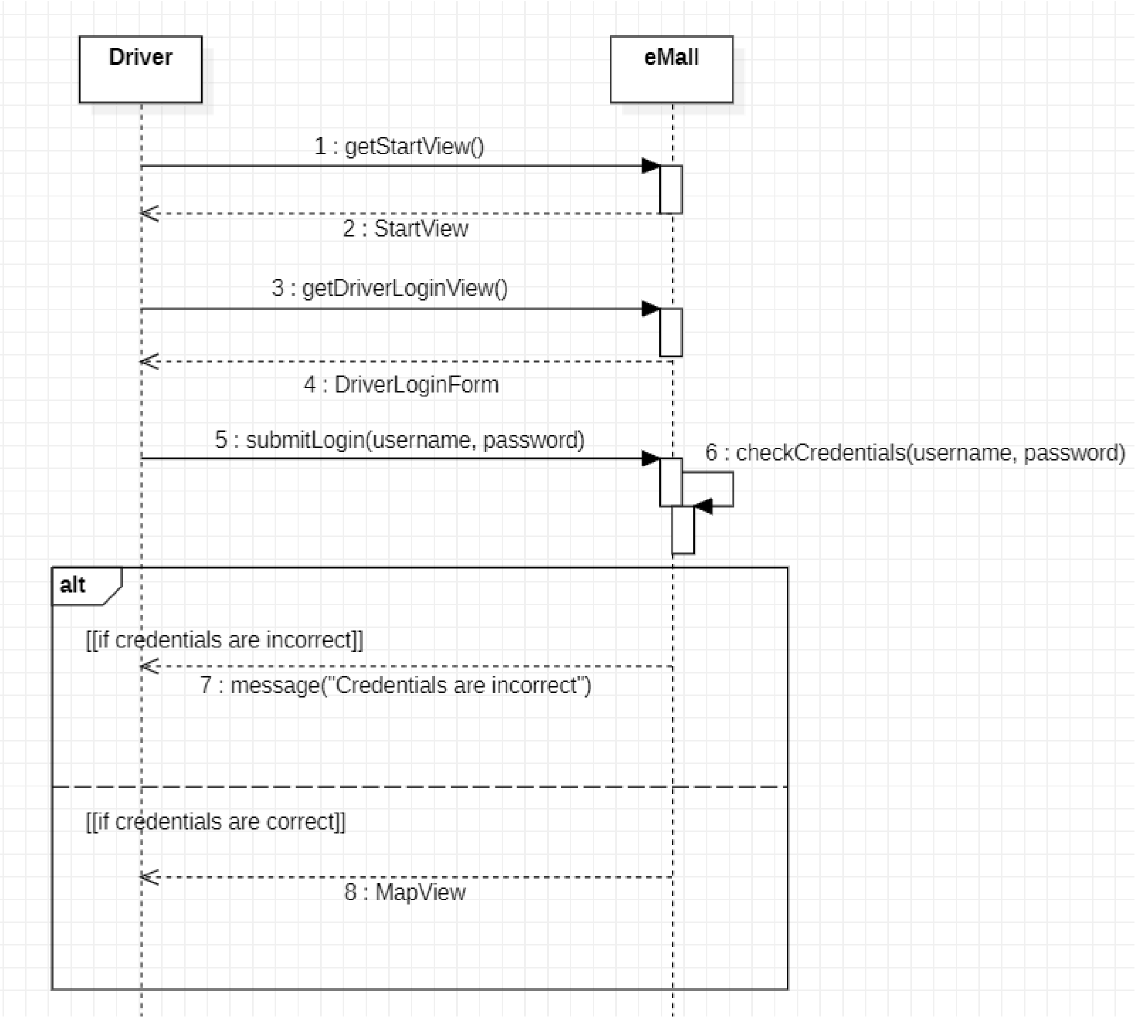

Driver
eMall
1 : getStartView()
2 : StartView
3 : getDriverLoginView()
4 : DriverLoginForm
5 : submitLogin(username, password)
6 : checkCredentials(username, password)
alt
[[if credentials are incorrect]]
7 : message("Credentials are incorrect")
[[if credentials are correct]]
8 : MapView

Figure 8: Driver login sequence diagram

[UC - 2] Driver consults the information of a station

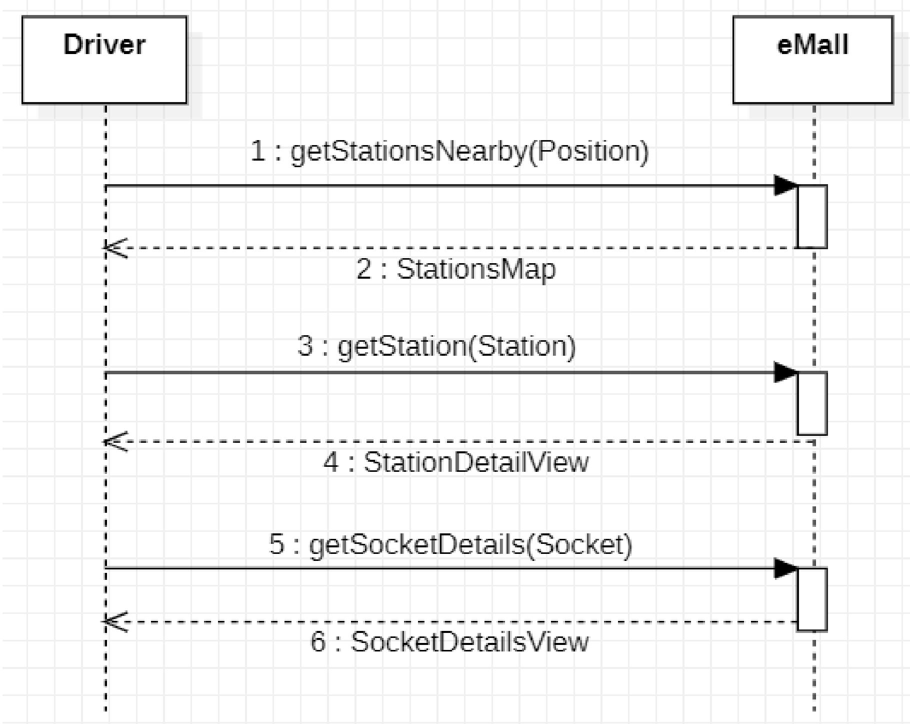


Figure 9: Get station information sequence diagram

[UC - 3] Driver books a charge

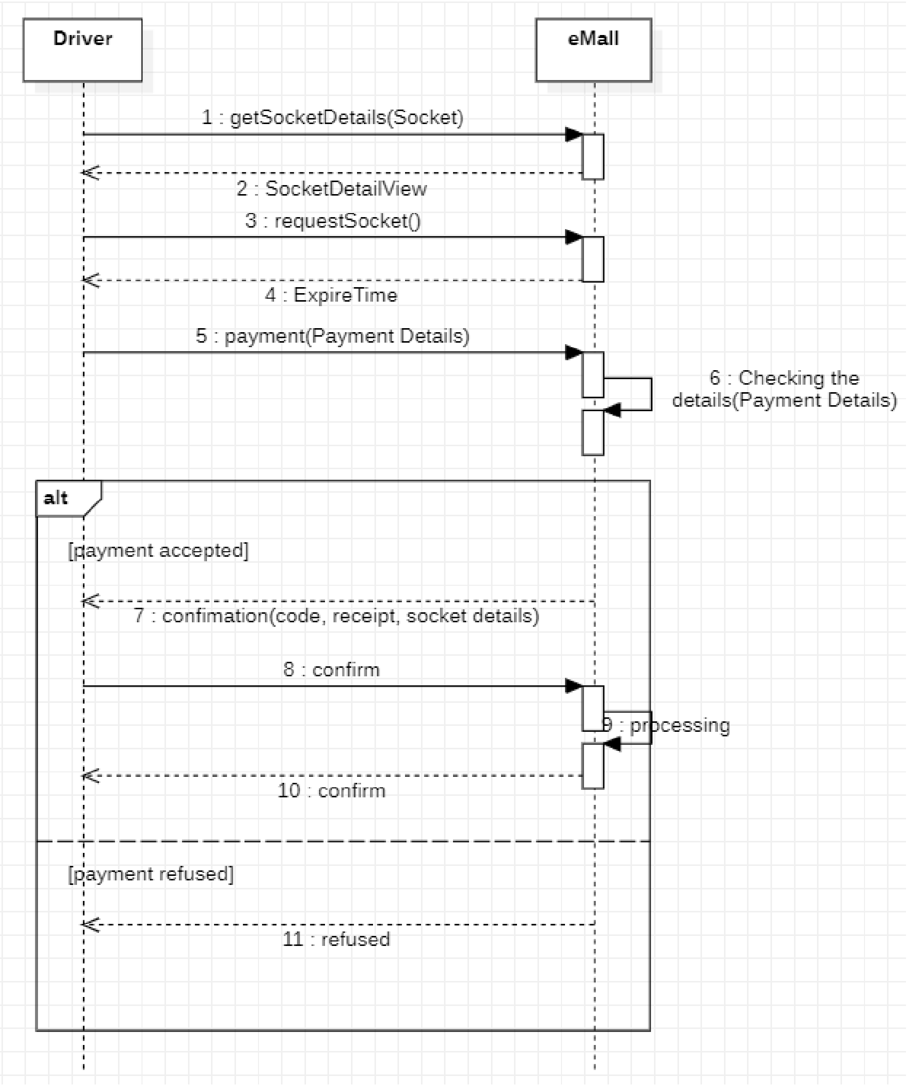


Figure 10: Book a charge sequence diagram

[UC - 4] Driver consult the details of a booking

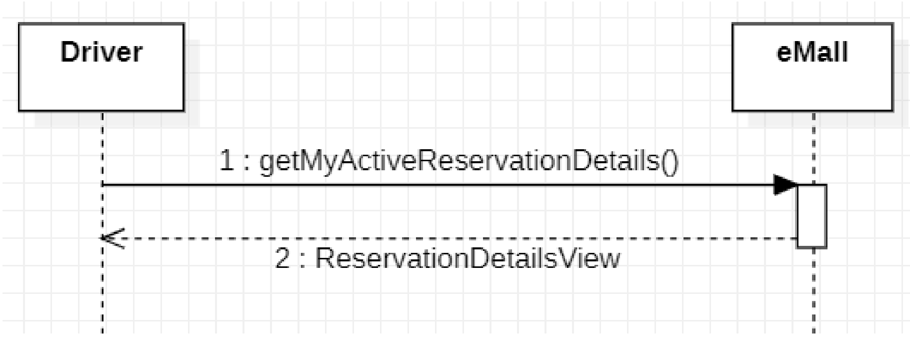

Driver
eMall
1 : getMyActiveReservationDetails()
2 : ReservationDetailsView

Figure 11: Get reservation details sequence diagram

[UC - 5] Driver cancels a reservation

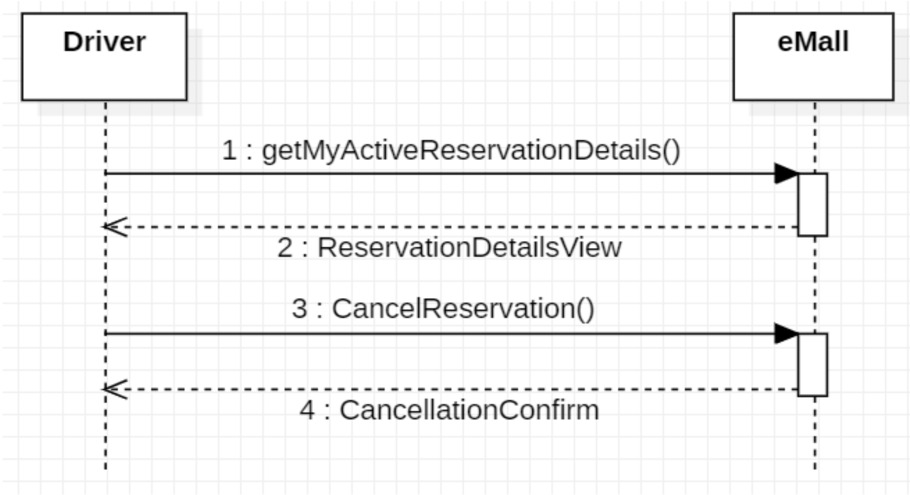

Driver
eMall
1 : getMyActiveReservationDetails()
2 : ReservationDetailsView
3 : CancelReservation()
4 : CancellationConfirm

Figure 12: Cancel a reservation sequence diagram

[UC - 6] Driver starts a charging process in a station

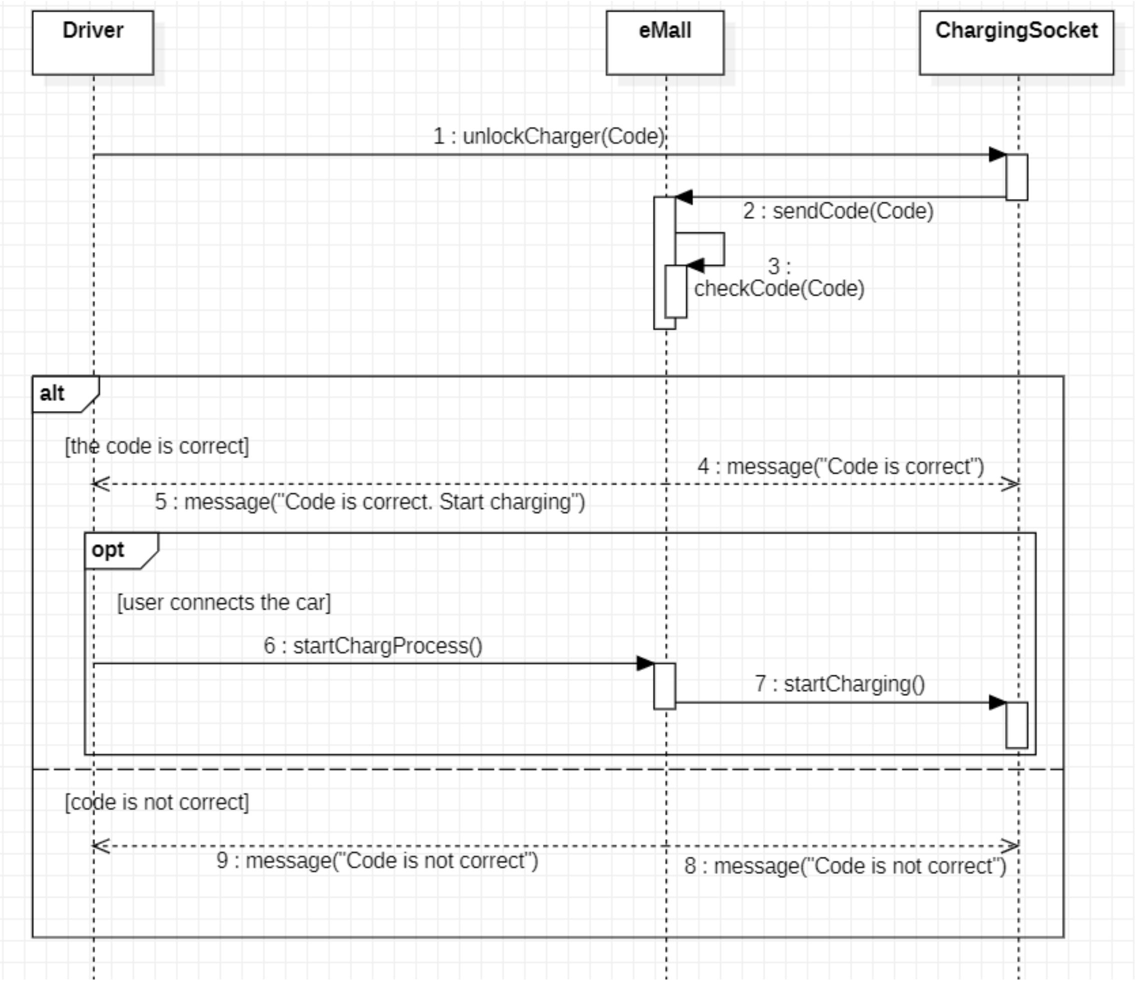


Figure 13: Start charging process sequence diagram

[UC - 7] Driver ends an ongoing charging process in a station

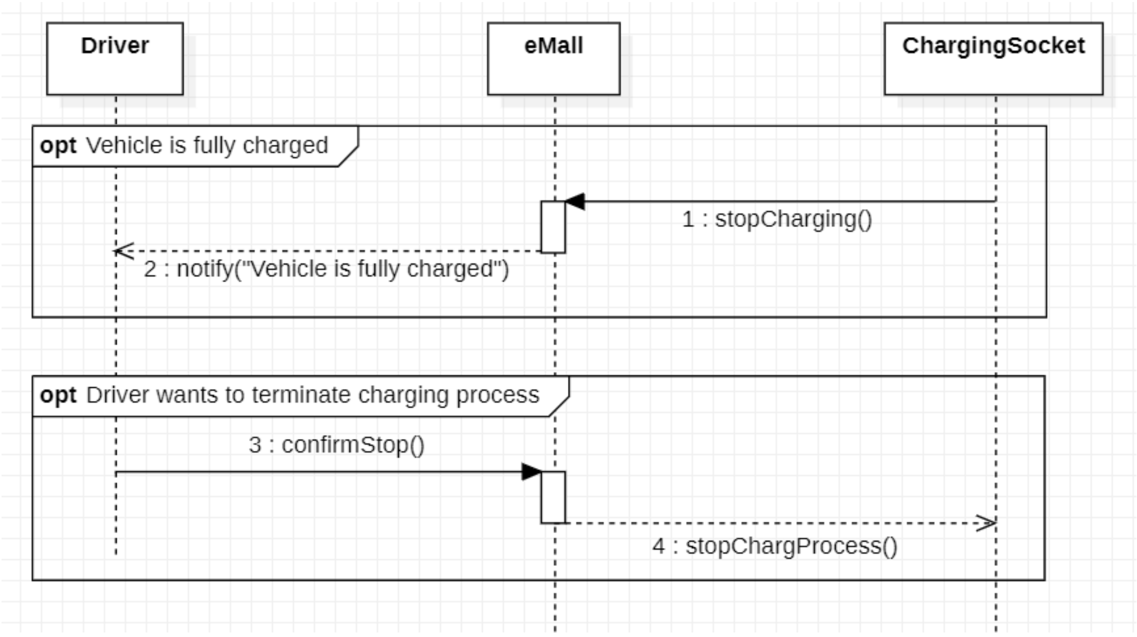

Driver
eMall
ChargingSocket
opt Vehicle is fully charged
1 : stopCharging()
2 : notify("Vehicle is fully charged")
opt Driver wants to terminate charging process
3 : confirmStop()
4 : stopChargProcess()

Figure 14: Stop charging process sequence diagram

[UC - 8] User-operator login to eMall

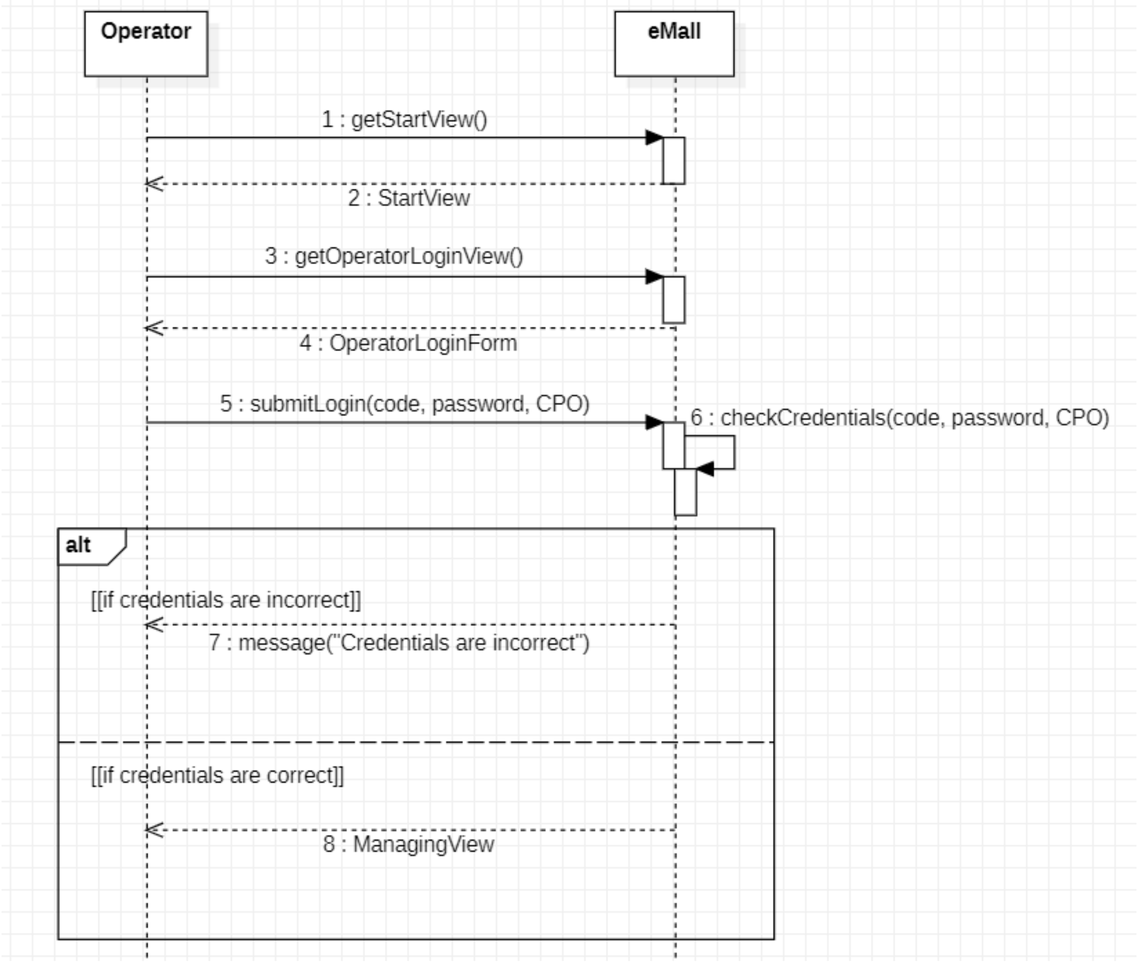

Operator
eMall
1 : getStartView()
2 : StartView
3 : getOperatorLoginView()
4 : OperatorLoginForm
5 : submitLogin(code, password, CPO)
6 : checkCredentials(code, password, CPO)
alt
[[if credentials are incorrect]]
7 : message("Credentials are incorrect")
[[if credentials are correct]]
8 : ManagingView

Figure 15: Operator login sequence diagram

[UC - 9] Operator consults the energy price of a DSO

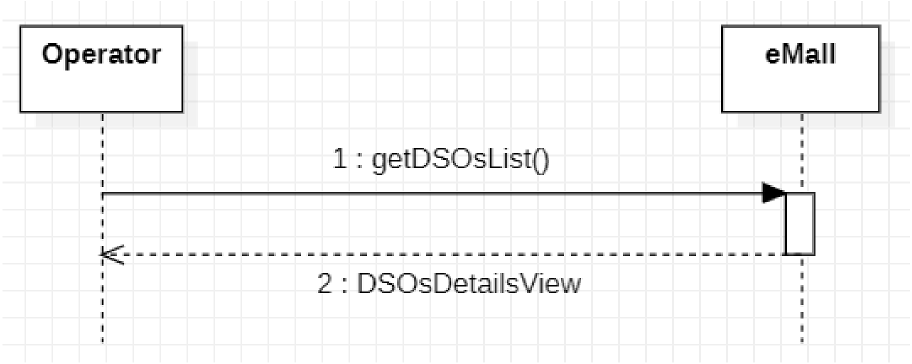


Figure 16: Get energy price from a DSO sequence diagram

[UC - 10] Operator accesses the management panel of a station

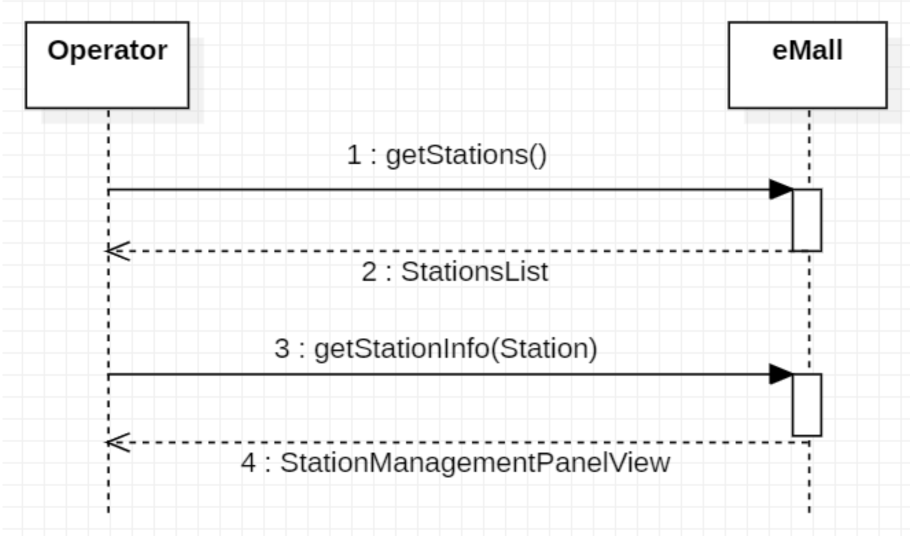


Figure 17: Select charging station sequence diagram

[UC - 11] Operator checks a socket status

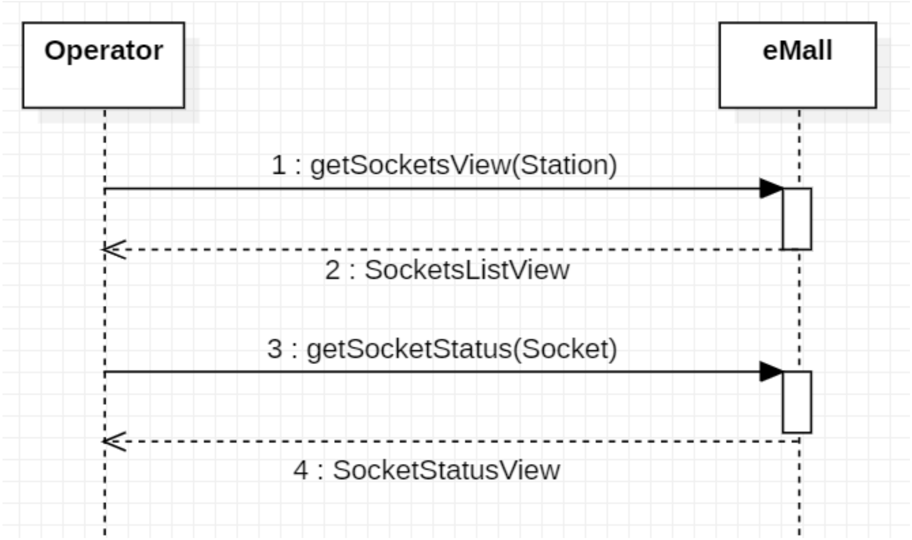


Figure 18: Get socket status sequence diagram

[UC - 12] Operator checks the batteries status of a station

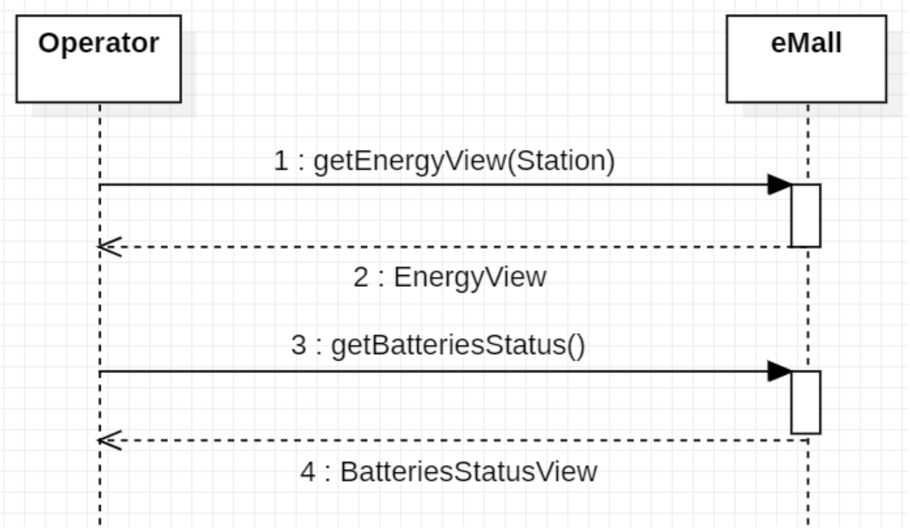


Figure 19: Get socket status sequence diagram

[UC - 13] Operator changes the energy source settings of a station

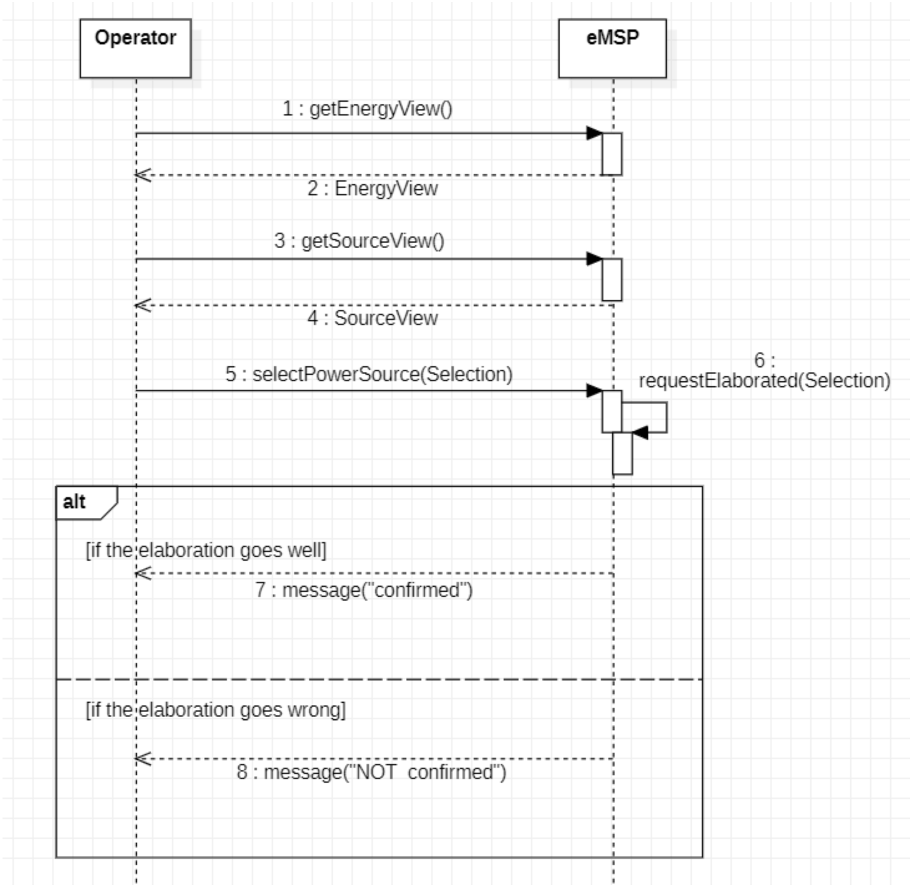

Operator
eMSP
1 : getEnergyView()
2 : EnergyView
3 : getSourceView()
4 : SourceView
5 : selectPowerSource(Selection)
6 :
requestElaborated(Selection)
alt
[if the elaboration goes well]
7 : message("confirmed")
[if the elaboration goes wrong]
8 : message("NOT confirmed")

Figure 20: Manage energy source sequence diagram

3.2.4 Mapping on requirements

[G1] Allow drivers to know information about charging stations nearby

Requirements	Domain Assumptions
R[1] - R[2] - R[3] - R[4] - R[5]	D[3] - D[9]

[G2] Allow drivers to book a charge for a vehicle in a specific charging station

Requirements	Domain Assumptions
R[1] - R[2] - R[3] - R[4] - R[6]	D[1] - D[3] - D[4] - D[9]

[G3] Allow drivers to manage and consult their bookings in progress

Requirements	Domain Assumptions
R[1] - R[6] - R[7] - R[8] - R[22]	D[7] - D[9]

[G4] Allow drivers to control a charging process

Requirements	Domain Assumptions
R[1] - R[6] - R[9] - R[10] - R[11] - R[12] - R[21] - R[22] - R[23]	D[1] - D[2] - D[5] - D[7] - D[8]

[G5] Allow charging station operators to monitor the status of a station

Requirements	Domain Assumptions
R[13] - R[14] - R[15] - R[16]	D[6] - D[9]

[G6] Allow charging station operators to manage a station's energy sources

Requirements	Domain Assumptions
R[13] - R[14] - R[17] - R[18] - R[19] - R[20]	D[6] - D[10] - D[11] - D[12] - D[13]

3.3 Performance Requirements

The system has to guarantee good performance in order to be useful for the users. This means that the delay between an interaction with the system and its response should be no more than a couple of seconds, given that the internet connection works properly. If a customer has a slow internet connection, the response time can increase because it depends on factors out of the control of the system. In order to avoid this phenomenon as much as possible, the size of the data exchanged has to be as small as possible.

3.4 Design Constraints

3.4.1 Standards compliance

The application interacts with external financial APIs, Since this is a very sensible matter, is crucial that the application respects the standards set by these APIs; in order to avoid any problems for the Users.

3.4.2 Hardware limitations

The application should run on most of the smartphones, so the application should be light enough and should be portable so that ven if there are different operating systems the users still have the possibility to use it.

3.5 Software System Attributes

In this section are further described the non-functional requirements of the project.

3.5.1 Reliability

It is necessary that the system always perform the intended functions, otherwise the customers may not get the charge they booked, or a customer whose car has a low battery percentage cannot book a charge.
It is crucial also for the CPO operators that the system is reliable so that when stations are using the batteries instead of the current from a DSO, they can monitor when the battery is going to run out and buy current from a DSO.
Furthermore, it is critical that during the rush hours the system doesn't slow down to give everyone the possibility to book a charge as soon as it is available.

3.5.2 Availability

The availability is a crucial factor. The users and the CPO operators may need to access at any hour and at any day so no denial of service can be tolerated. The services provided by the app must be available 24/7. Users may need to urgently charge their car because it has a low battery percentage. CPO may need to start buying current from a DSO because the batteries are running out.

3.5.3 Security

The system stores all the drivers' and CPO operators' information. To value their privacy, such information must be encrypted and can be accessed only by them. It is crucial to security that only the registered CPO operators have access to stations' information.
They have access only to the data of their CPO stations and can manage only these stations and no other one.
The system also handles financial transactions from CPO to DSO and transactions from drivers to the CPMS of the respective CPO; it is important that the data relative to the transactions remain undisclosed and no third party person access those data.

3.5.4 Maintainability

The system, as specified above, requires high levels of reliability, availability and security. These attributes are achieved by keeping the code always up to date. So, there will be periodic updates. Since the updates will be frequent and constant, the software must be highly maintainable. The code of the software must be written in a way that is easy to search, reuse and easy to modify and extend.

3.5.5 Portability

Since the system will be used by various drivers and operators, each person can have a different device with a different operating system. The portability of the code plays a crucial role because a portable app means more devices can use the application. Hence more clients and no need to rewrite the code for different platforms.

4 Formal analysis

4.1 Alloy code

4.1.1 Signatures

```
// Name is used to represent a string containing the name of users
sig Name {}

// Posizione is used to define longitude and latitued of a Station
sig Position {}

// Type is used to specify which type of socket is a certain charging socket
sig Type {}

// a boolean to represent if a charging station is free to be used or not
sig Bool {}

// a code used to unlock the charging socket before start the charging
sig Code {}

// an int between 0 and 100
sig Percentage {}

// date and time for the reservation
sig Time {}


sig User {
        name: one Name,
}

sig Operator  extends User {
        cpo: one CPO, // Ogni operatore ha una CPO
}

sig CPO {
        name: one Name,
        operators: some Operator,
} {
        #operators > 0  // Every CPO has at least one
}

sig Driver extends User {}

sig Station {
        name: one Name,
        position: one Position,
        sockets: some ChargingSocket,
        batteries: some Battery,
} {
        #sockets >=  1
}
```

```
sig ChargingSocket {
        type: one Type,
        available: one Bool,
        unlocked: one Bool,
        inUse: one Bool,
        nextCode: lone Code,
}

sig DSO {
        supply: Station,
}

sig Battery {
        percentage: one Percentage,
}

sig Reservation {
        user: one User,
        socket: one ChargingSocket,
        station: one Station,
        code: one Code,
        startingTime: one Time,
}
```

4.1.2 Facts

In the following are stated in Alloy the facts that must hold for the domain modeled in order to maintain consistency with the real world.

```
// All stations are located in a different position
fact stationLocations {
        ( all disj s1, s2: Station | s1.position != s2.position )
        and
        (#Position = #Station)
}

// All stations have a battery, the number of batteries
// in the system is greate than the number of stations
fact stationsHaveBatteries {
        ( all s: Station | #s.batteries >= 1 )
        and
        (#Station =< #Battery)
}

// All stations have batteries that no other station has,
//there can't be shared batteries
fact noSharedBatteries {
        (all disj s1, s2: Station | #(s1.batteries & s2.batteries) = 0)
}

// Socket in use -> socket not available
fact usingIffNotFree {
        (all c: ChargingSocket |        c.available != c.inUse)
}

// Every user is either a Driver or an Operator
fact userDisjointed {
        (#User = #Operator + #Driver)
}

// The Socket of each reservation belongs to the reported Station
fact socketBelongsToStation {
        (all r: Reservation  | #(r.socket - (r.station).sockets) = 0)
}

// Reservation codes are different and unique,
// there can't be any duplicate. The code of a reservation has
// always a socket with the corresponding code
fact codeIsUnique {
        (all disj r1, r2: Reservation | r1.code != r2.code)
        and
        (#Reservation < #ChargingSocket)
        and
        (all r: Reservation |
        one c: ChargingSocket |
        r.code = c.nextCode and r != none and c != none)
}

// same cardinality for time and reservations
```

```
fact sameNumTimeReserv {
        ( #Time = #Reservation )
}

// Each Driver has at most one reservation
fact driverHas1ReservAtMost {
        all disjoint r1, r2: Reservation | r1.user != r2.user
}

// a station can't be supplied by two DSO
fact oneDSO {
        all disj d1, d2: DSO | (d1.supply & d2.supply) = none
}
```

4.1.3 Assertion check

```
// a Driver has a reservation but no socket has a matching code
assert reservationIsMatched {
        all r: Reservation, c :ChargingSocket |
        ( r != none and c != none ) implies c.nextCode = r.code
}
check reservationIsMatched

// a station is supplied by two DSO
assert oneDSO {
        all d1, d2: DSO |
        d1 != d2 implies (d1.supply & d2.supply) = none
}
check oneDSO
```

All the three checks on the assertions return “No counterexample found. Assertion may be valid.”. This implies that the domain modeled may be consistent with how the system’s behaviour has been described in this document.

5 Effort spent

The time tables written below represent just an approximation of the time spent for the writing and discussions the team had for each specific chapter of this document. These times have not been measured while producing this document and are just based on the personal perception the team members have of the time spent.

Filippo Cinfrignini

Chapter	**Effort (in hours)**
1	18
2	23
3	13
4	10

Davide Esposito

Chapter	**Effort (in hours)**
1	19
2	24
3	16
4	6

6 References

[1] assignment Assignment RDD AY 2022-2023_v3.pdf.

[2] assignment Lectures slides of the Software Engineering II course on https://beep.metid.polimi.it/