RASD
Requirements Analysis and Specification Document

Software Engineering 2 - Matteo Rossi

Niccolò Nicolosi, Francesco Negri, Marcos Pietrucci

23/12/2022

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe in details e-Mobility for all (eMall), the system to be. This document aims to provide the functional and non-functional requirements needed to develop the system, the most common use cases, a list of interactions between the system and the user and more useful information in order to guide the developers of eMall throughout the design and the implementation of the system.

1.2 Abbreviations and definitions

- eMall: e-Mobility for all (the system to be)
- eMSPs: eMall Service Providers
- eMSS: eMall Service System (subsystem 1)
- end-users: eMSS users
- CPMS: Charging Point Management System (subsystem 2)
- CPOs: Charging Point Operators (CPMS users)
- DSOs: Distribution System Operators
- users: generic users of the system (both end-users and CPOs)
- external status of a charging station:
  - types of charging sockets offered (slow/fast/rapid) and their cost
  - number of charging sockets available and their type
  - estimated amount of time until the first socket of a specific type is freed
  - special offers
- internal status of a charging station:
  - current DSO supplying it
  - amount of energy available in its batteries (if any)
  - number of vehicles being charged
  - amount of power absorbed by each charge
  - time left to the end of each charge

1.3 Goals

Below, we present the main goals of the system. We consider two main subsystems of eMall: the eMSS, responsible to offer services to the end-users, and the CPMS, that offers services to the CPOs. The subsystems communicate with each other in order to exchange the information needed to provide their services.

1.3.1 Subsystem 1: eMSS

Goal	Description
G1	Allow the end-user to know the location of charging stations nearby and their external status
G2	Allow the end-user to book a charge in a specific charging station for a certain time frame
G3	Allow the end-user to start the charging process at a certain station
G4	Notify the end-user when the charging process is finished
G5	Allow the end-user to pay for the obtained service
G6	Proactively suggest the end-user to go and charge his vehicle(s)

1.3.2 Subsystem 2: CPMS

Goal	Description
G7	Allow the CPO to know the location, the external status and the internal status of proprietary charging stations
G8	Allow a charging point to start charging a vehicle according to the amount of power supplied by the socket, and monitor the charging process to infer when the battery is full
G9	Allow the CPO to acquire information about the current price of energy and mix of energy sources used by the DSOs
G10	Allow the CPO to dynamically decide from which DSO(s) to acquire energy
G11	Allow the CPO to dynamically decide whether to store energy in stations storage batteries
G12	Allow the CPO to dynamically decide where to get energy for charging from (stations storage batteries or DSOs)
G13	Allow the CPO to dynamically decide the prices of charging and set special offers
G14	Automatize some of the decisions that can be taken by the CPO

1.4 Scope

The eMall system has two types of users: end-users and CPOs.

The end-user has access to the following main functionalities:

- view and select charging stations and start a charge from the stations page
- view the details of a charging station, book a charge and ask for directions to reach the station from the details page
- manage their vehicles from the vehicles page

- view the charges and payments history and active reservations and cancel reservations from his personal page
- link his calendar cloud service from the settings page

The CPO has access the following main functionalities:

- view and select proprietary charging stations from the stations page
- view the details of proprietary charging stations and manage them from the details section in the stations page
- view the available and active DSOs from the DSO page
- view the details of DSOs from the details section in the DSO page

1.4.1 World Phenomena

Phenomenon	Description
W1	The end-user decides to charge its vehicle and wants to find a charging station
W2	The end-user moves from a location to another
W3	The end-user plugs his vehicle into the charging station
W4	The end-user waits for the charging process to finish
W5	The end-user unplugs the vehicle from the charging station
W6	The CPO contacts the administrators of eMall to ask them to be registered to the system
W7	The CPO opens a new station
W8	The CPO dismantles a station
W9	The DSO transfers the energy to the stations that request it
W10	The DSO changes the price of energy
W11	The CPO pays the DSO(s) for the amount of energy consumed during the last period

1.4.2 Shared Phenomena

Phenomenon	Description	Controller
S1	The end-user registers to the system	end-user
S2	The end-user unregisters from the system	end-user
S3	The user logs into the system	user
S4	The system shows information about nearby charging stations to the end-user	system
S5	The end-user selects a charging station to view its details	end-user

S6	The end-user asks for directions to reach a specific station	end-user
S7	The end-user books a charge in a specific station for a specific time frame	end-user
S8	The end-user cancels a reservation he previously requested	end-user
S9	The system sends the reminder notification of a reservation to the end-user	system
S10	The system sends a suggestion notification to the end-user	system
S11	The end-user starts the charging process at the chosen station	end-user
S12	The system notifies the end-user of the end of the charging process	system
S13	The end-user checks his charges and payments history	end-user
S14	The end-user adds a vehicle	end-user
S15	The end-user removes a vehicle	end-user
S16	The end-user connects its calendar to the system	end-user
S17	The end-user disconnects its calendar from the system	end-user
S18	The end-user changes its payment method	end-user
S19	The system shows information about proprietary stations to the CPO	system
S20	The CPO selects a charging station (or group of stations) to view its details	CPO
S21	The CPO sets prices of charging stations	CPO
S22	The CPO sets special offers of charging stations	CPO
S23	The CPO starts the process of recharging storage batteries	CPO
S24	The CPO sets the mix of energy sources of charging stations (batteries and DSO energy)	CPO
S25	The CPO sets a charging station to auto-mode	CPO
S26	The CPO selects the DSO(s) to acquire the energy from	CPO
S27	The CPO checks available and active DSOs	CPO
S28	The CPO selects a DSO to view its details	CPO

2 Description of the system

2.1 User characteristics

As already specified, the eMall system has two types of users: end-users and CPOs.

2.1.1 End-user

The end-user can be any owner of one or more electric vehicles, interested in knowing where to charge them. For this reason the application should be easy to use for end-users and, being that they are probably often moving, it should be possible for them to access it from mobile devices.

2.1.2 CPO

The CPO is a user that represents a company and has to manage charging stations in the interests of his company. Thus, the application should provide the CPO with all the functionalities he needs in such a way that they are well organized, quick and easy to use.

2.2 Product functions

2.2.1 Stations search

The end-user can search for nearby stations consulting a map. He can select a station to view the details about its external status, book a charge or ask for directions to reach it.

2.2.2 Charging

The end-user can start the charging process of one of his vehicles at a charging station by pressing the *Start charge* button and scanning the socket QR code or manually inserting the socket identifier code. Then, the system will show the charge details, like the socket speed, the expected time of charging and the price per unit of measure and it will wait for a confirmation to start. The vehicle must be correctly plugged into the chosen socket for the process to start. The end-user can simply unplug his vehicle from the socket to stop the charging process.

2.2.3 Booking charges

The end-user can book a charge in a station he chose by pressing the *Book a charge* button. The end-user has to specify the time of the reservation, the requested socket type and then press the *Confirm* button.

2.2.4 Vehicle management

The end-user can manage his vehicles, adding new ones by pressing the *Connect a vehicle* button or removing the ones already connected by pressing the *Disconnect* button next to each.

2.2.5 Suggestions

The system proactively sends a suggestion notification to the end-user whenever the battery level of one of his vehicles is low. The notification recommends the end-user to charge his vehicle in a specific nearby station, chosen by taking into account the distance from the user, the availability of charging sockets and any current special offers. The notification also proposes a specific time frame that does not overlap with any of the tasks in the user's calendar (if connected to the system).

2.2.6 Stations management

The CPO can visualize proprietary stations in a map or in a list and can select one or more of them to view the details about their external and internal status and manage them. The CPO can:

- set prices by pressing the *Set prices* button, specifying the price of every type of socket available in the group of stations and then pressing the *Confirm* button

- set special offers by pressing the *Set special offers* button and managing the list of active offers (adding or removing offers). For each offer added, the CPO has to specify the period of the offer and the discounts provided and confirm

- charge storage batteries by pressing the *Charge batteries* button and confirming

- set the mix of energy provided during the charges by pressing the *Set energy mix* button, specifying the percentages of DSO energy and battery energy to use and then pressing the *Confirm* button

- set the DSO that supplies the group of stations by pressing the *Set DSO* button and selecting one of the available ones

- set the station to auto-mode so that the system decides DSO, prices, battery charging and the energy mix to use automatically by pressing the *Auto-mode* button, specifying the desired profits per unit of measure for each socket type and confirming

2.2.7 DSO view

The CPO can view the list of available and active DSOs and can select one of them to view the price of their energy, the energy sources they use and the stations they currently supply (if active).

2.3 Scenarios

1. **End-user registers to the system**

   Giulia has just installed the eMall app on her smartphone because she wants an easy way to charge her new electric vehicle. To access the application she has to register an account. So, she opens the app and presses the *Sign up* button. Then she is redirected to the registration page, in which she has to insert her personal information. She inserts the requested data and confirms. Now the registration process is completed and Giulia can use the eMall app just by logging in with her email and password.

2. **End-user books a charge**

   Giuseppe is planning a trip from Milan to Turin on the next weekend. He has an electric car and he uses it every day to go to work. Giuseppe needs to make sure that the car's battery will be fully charged by the time he leaves for the trip, so that he doesn't have to worry about recharging it during the trip. Thus, Giuseppe decides to use eMall to find a charging station near to his office and charge his car during work. He opens the app and searches the map, zooming on his office's neighbourhood. He finds out there are 2 stations just 5 minutes on foot from his office. By quickly looking at each one, he notices that station 1 only offers slow charging sockets that cost 0.45€/kWh while station 2 only offers rapid charging sockets and they cost 0.60€/kWh. Knowing he has to work for several hours, he chooses to go with station 1 and to book a charge for Friday morning. He selects station 1 and presses on the *Book a charge* button. Then he selects the day and time and confirms the booking. Finally he makes sure that the reservation is active by looking at his personal page and then he closes the app.

3. **End-user receives a notification when the charging is complete**

   Chiara leaves her electric motorbike charging at a charging station while she goes to a nearby cafe with her friends. She has a great time, but while everyone is leaving she knows her motorbike is still charging as she has not received any notification yet. So, she asks one of her friends to stay a bit more and have an ice cream with her while she waits. 30 minutes later Chiara receives a notification from eMall on her phone that says her motorbike's charging has completed. She goes back to the station and during the trip she checks her personal page on the eMall app and her credit card balance on her banking app to verify that the right amount of money was billed. Once she arrives, she unplugs the vehicle from the charging socket and leaves on her motorbike.

4. **eMall suggest end-user to go charge his vehicle**

Gigi has to go to an important meeting to the other side of the city. He is in a bit of a hurry, so he jumps in the car and doesn't notice that its batteries are low. Once he arrives, he receives a notification on his phone that suggests him to charge the vehicle. The notification also proposes the charging station nearest to the office and shows that a special offer is available. Gigi realizes he managed to arrive early and so he has the time to reach the charging station and leave his car in charge without risking to get late for the meeting. This way Gigi will not have to waste time waiting for his car to charge after the meeting.

5. **CPO selects a specific DSO**

   Antonio works for Charge&Go, a company that provides charging stations for electric vehicles in Italy. Charge&Go has recently decided to only rely on renewable energy sources for its charging stations. So, Antonio is in charge of interacting with the eMall system to select BeYourEnergy (BYE), a clean energy producing company, as DSO of Charge&Go. Antonio logs into the system with the CPO credentials and selects all the proprietary stations. Antonio presses the *Set DSO* button and selects BYE from the list of available DSOs. So, BYE gets correctly inserted as the DSO of every station of the company.

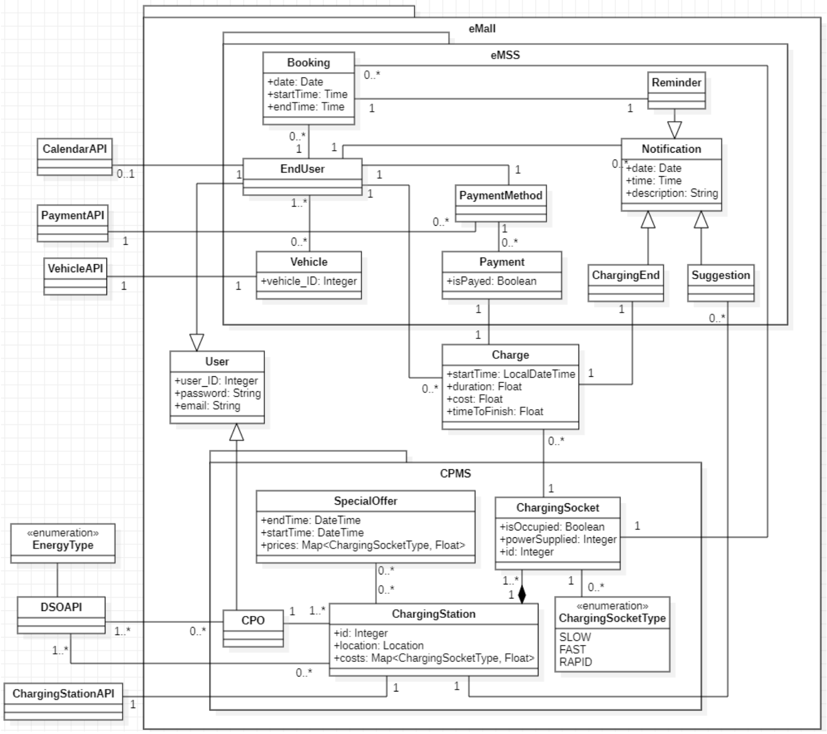
6. **CPO recharges the storage batteries of a group of stations**

   Marcello is an employee of YellowStream, a company in the business of vehicle charging stations in Rome and Milan. Given the low energy prices of the last few days, YellowStream tasked Marcello to buy and store energy in stations' batteries where possible. Marcello studies the case and decides to fully charge the batteries of all the stations equipped with them in Rome. Thus, he logs into the eMall system with the CPO credentials and he's showed the company's stations view, from which he selects all the stations in Rome. Then Marcello presses the *Charge batteries* button and the system confirms the success of the operation.

7. **CPO sets auto-mode and a special offer**

   Anna works for LombarGreen, a recently created company of fast charging in the Lombardy region in Italy. A new charging station of LombarGreen has recently been opened in Bergamo, so, Anna needs to manage it. She opens the eMall app and selects it. Then she sets the station to work in auto-mode: she specifies the profit of 0.05, 0.10 and 0.15 €/kWh respectively on the standard prices of slow, fast and rapid sockets and sets the threshold for charging batteries at 0.20€/kWh. This way, the station will periodically update its DSO, set its prices to reach the desired profit, charge batteries when the price of DSO energy goes below the given threshold and choose the energy mix to use automatically. To foster new customers Anna decides to set a special offer in the new station: for the first week after the inauguration, the energy will be sold at cost, beating by far any competitors. Thus, she presses the *Set offers* button and specifies the start time, the end time and prices of the offer. Then she confirms and the offer is accepted by the system.

2.4 Class Diagram



2.5 Domain Assumptions and dependencies

Assumption	Description
D1	The end-user's vehicle has a navigation system that can be connected with a smartphone and exchange information about the vehicle
D2	The end-user's payments will always succeed at some point
D3	The end-user will unplug his vehicle and free the socket at some point
D4	The charging process will always succeed at some point
D5	Charging stations have an internal system that regulates the station and that is connected to the internet
D6	Each charging station can be supplied by every available DSO
D7	The infrastructure between charging stations and DSOs is always in function
D8	Every charging socket of every station is identified by a unique QR code and a unique identifier code
D9	Charging stations are all equipped with the necessary sensors for the CPO to properly monitor charging processes
D10	CPOs that want to register to or unregister from the system make arrangements with the administrators of eMall in order to get or remove their account

3 Systems requirements

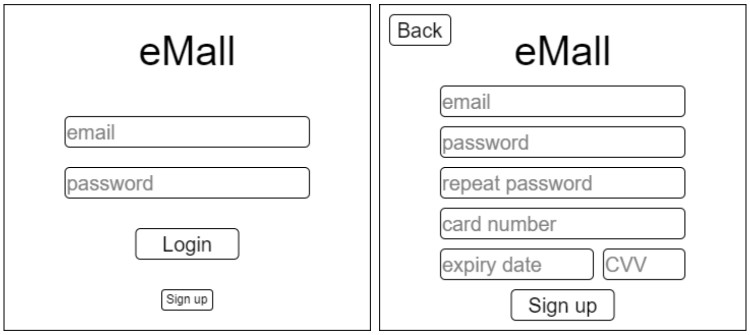
3.1 External interface requirements

3.1.1 User interfaces

In this section we will present early-phase user interfaces for the main functions of the system.

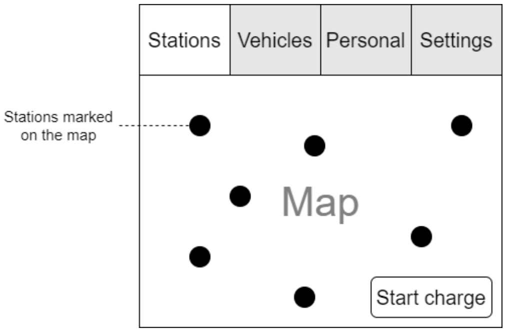
End-user UI

When the end-user launches the app, he can log into the system or register using the following pages

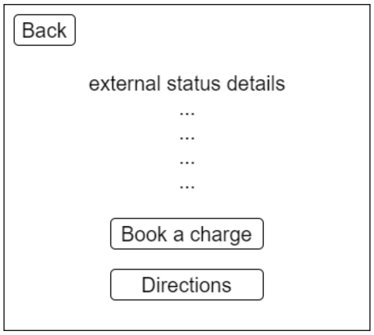


The interface is then divided into four different main pages:

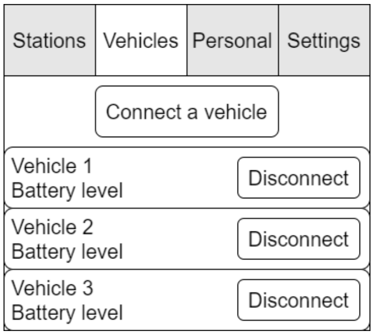
1. The stations page, that shows a map of nearby stations to the end-user. In this page, the user can navigate the map and press on a station to view its details in a dedicated page or start a charge



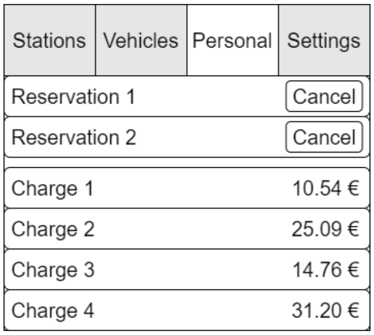
From the details page of a station, the user can also book charges and ask for directions to reach the station



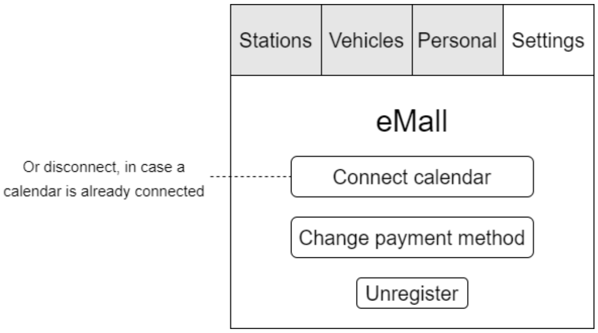
2. The vehicles page, in which the end-user can manage his vehicles, adding new ones or removing the ones already connected to the system



3. The personal page, that lists all the user's active reservations and the history of charges and payments. From this page, the end-user can cancel active reservations and select charges to view their details in a dedicated page

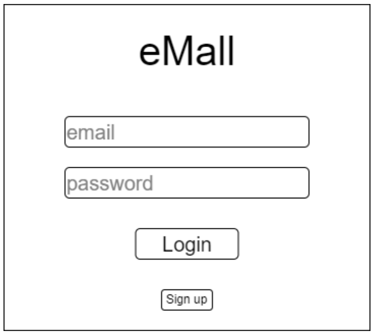


4. The settings page. From this page, the end-user can connect or disconnect his calendar cloud service and unregister from the system



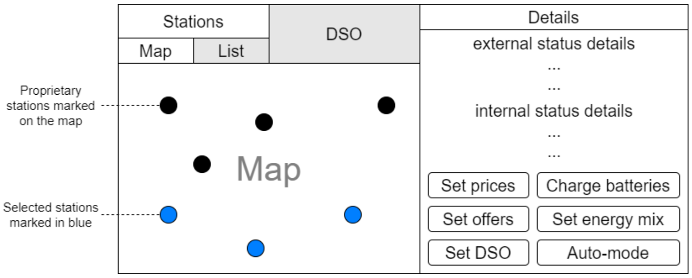
CPO UI

When the CPO launches the application, he can log into the system using the following page:

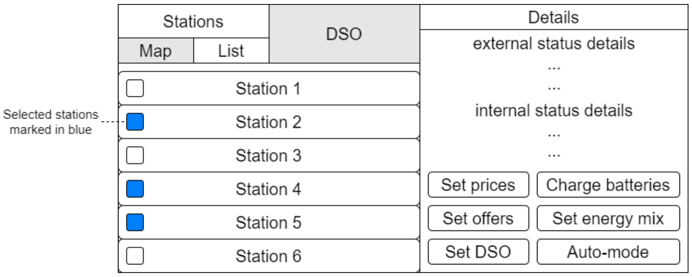


The interface is then divided into two main pages:

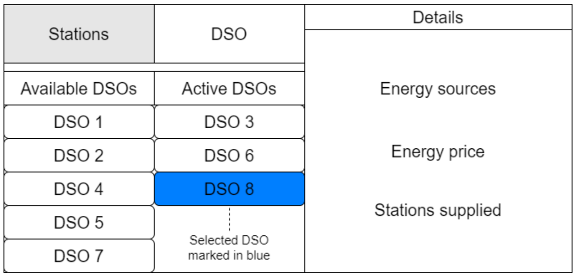
1. The stations page, that shows a map of proprietary stations to the CPO. In this page, the CPO can navigate the map and select one or more stations to view their details in the dedicated section. From the details section, the CPO can also manage the selected stations (E.g. setting prices and special offers, recharging the batteries, etc...)

   The CPO can also choose to view a list of the stations rather then a map, by clicking on the list option:

2. The DSO page, from which the CPO can view all the available and active DSOs. The CPO can select a specific DSO to view its details



3.1.2 Hardware Interfaces

Considering the context of the application, end-users are required to have a smartphone device capable of connecting to the internet, equipped with the Bluetooth technology in order to connect to vehicles' navigation systems, with GPS to track the user's location and optionally with a camera to scan QR codes. CPO users, given the need of accessing the application in specific locations only, like charging stations or CPO offices, are not required to have a mobile device. Instead, they could use any device capable of connecting to the internet, with a web browser installed. Preferably a PC or a laptop due to the complexity of the CPO UI.

3.1.3 Software Interfaces

The application will interface with different external APIs:

- Vehicles' navigation system API: to know the battery level of vehicles
- Map API: to get the navigable map to show to the user and redirect the end-user to the map application itself when he asks for directions
- Bank API: to verify payment methods and manage the payments of end-users
- Calendar API: to know the end-user's tasks in order to send suggestions accordingly
- Station API: to get information about the station and apply changes to it
- DSO API: to get information like the price of energy and the energy sources of the DSO

3.1.4 Communication Interfaces

In order to grant all the functionalities of the application, an internet connection is always required for the end-user.
A CPO, instead, could also temporarily use the application offline if the device he is using is locally connected to a specific charging station's system to access the functionalities related to the management of that station.

3.2 Functional requirements

Requirement	Description
R1	The system shall allow an unregistered end-user to register
R2	The system shall allow a registered end-user to unregister
R3	The system shall allow a registered user to login
R4	The system shall display a navigable map of the user's surrounding area in the end-user's main page, representing charging stations at their location
R5	The end-user should be able to connect his vehicle(s) to the system from the vehicles page
R6	The end-user should be able to disconnect his vehicle(s) from the system from the vehicles page
R7	The end-user should be able to view the details of the external status of a specific charging station in a dedicated details page
R8	The end-user should be able to book a charge in a specific charging station for a certain (available) time frame from its details page
R9	The system shall send a phone notification and an email to the end-user 1 hour before his reservation starts, to remind him not to miss the appointment
R10	The end-user should be able to start the charging process at a specific charging socket from the stations page if and only if a vehicle is correctly plugged into that socket and the socket is not reserved for someone else
R11	The system shall control the charging processes, starting them when requested, deciding the amount of power to use for each and monitoring them to know when they have ended
R12	The system shall send a phone notification and an email to the end-user whenever one of its current charging processes ends
R13	The system shall charge the cost of the service to the end-user, using the chosen payment method, whenever a charge ends
R14	The end-user should be able to view and cancel his current reservations from his personal page
R15	The end-user should be able to view the history of charges and payments from his personal page
R16	The end-user should be able to change his payment method from the settings page
R17	The end-user should be able to connect his calendar cloud service to the system (if supported) from the settings page
R18	The end-user should be able to disconnect his calendar cloud service from the system from the settings page

R19	The system shall periodically inspect the battery level of the end-user's vehicle(s), the end-user's position and calendar (if linked to the system) to send a phone notification and an email suggesting a charge if the battery level is under 30%. The suggestion must recommend a station near to the user, taking into account the distance from the user, the availability of charging sockets and any current special offers, and it must propose a time that does not overlap with any of the tasks in the user's calendar
R20	The system shall display a navigable map in the CPO's main page representing proprietary charging stations at their location
R21	The CPO should be able to view the details of the external and internal status of a proprietary charging station (or group of stations) in a dedicated details section
R22	The CPO should be able to set the prices of the charging types (slow/fast/rapid) of a specific proprietary station (or group of stations) from its details section
R23	The CPO should be able to set special offers on a specific proprietary station (or group of stations) from its details section
R24	The CPO should be able to set the DSO of a specific proprietary station (or group of stations) from its details section
R25	The CPO should be able to recharge the storage batteries of a specific proprietary station (or group of stations) from its details section
R26	The CPO should be able to decide the mix of energy sources for charging (storage batteries and DSO energy) in a specific proprietary station (or group of stations) from its details section
R27	The CPO should be able to set a specific proprietary station (or group of stations) in auto-mode from its details section
R28	The CPO should be able to overwrite the auto-mode for a specific charging station (or group of stations) by manually setting prices, the DSO or the energy mix or by charging batteries
R29	The CPO should be able to view the list of all available DSOs from the DSO page
R30	The CPO should be able to view the list of all the DSOs supplying proprietary stations from the DSO page
R31	The CPO should be able to view the details of a specific DSO in a dedicated details section

3.3 Mapping requirements on goals

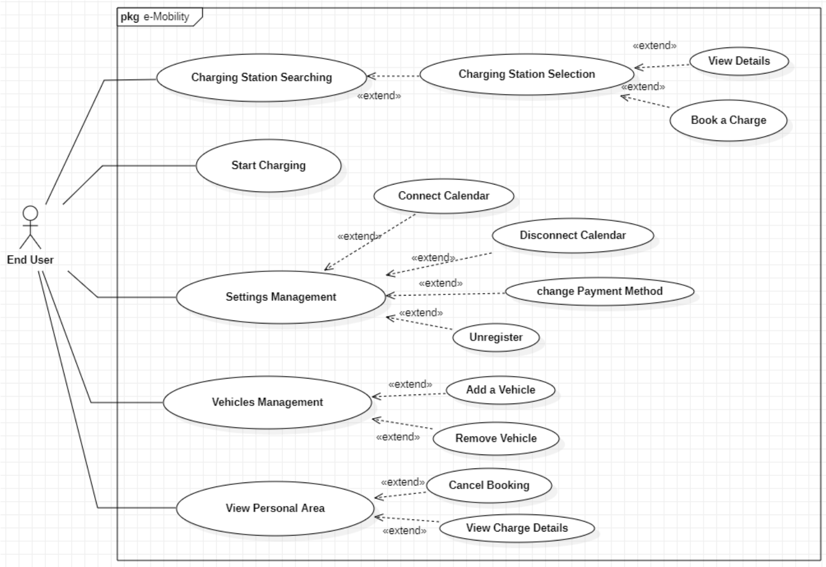
Goal	Assumptions	Requirements
G1	D5	R4, R7
G2	D3, D5	R7, R8, R9, R14
G3	D4, D5, D7, D8, D9	R10, R11
G4	D4, D9	R11, R12
G5	D2	R13, R15, R16
G6	D1, D5, D9	R5, R6, R17, R18, R19
G7	D5, D10	R20, R21
G8	D5, D7, D8, D9	R11
G9	D7	R29, R30, R31
G10	D5, D6, D7	R21, R24, R28
G11	D5	R21, R25, R28
G12	D5	R21, R26, R28
G13	D5	R21, R22, R23, R28
G14	D5	R21, R27

NOTE: Requirements R1, R2 and R3 are left out of the mapping because they are basically needed to reach every goal

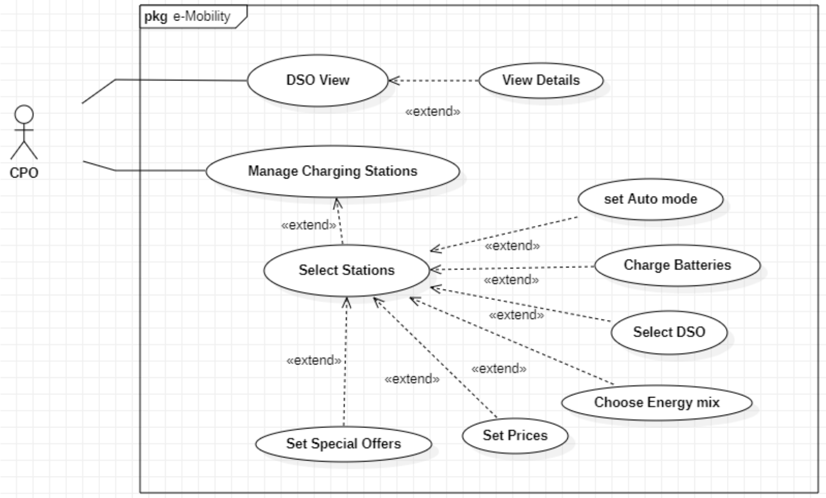
3.4 Use case diagrams

The following section will include use case diagrams depicting all the actions and operations offered to end-users and CPOs.

3.4.1 Registered end-user use case diagram



3.4.2 CPO use case diagram

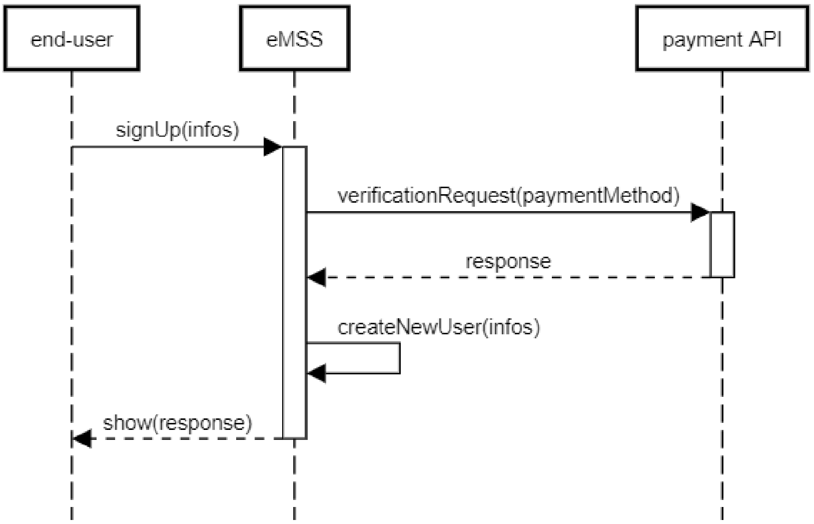


3.5 Sequence Diagrams

The following section will include all the sequence diagrams related with actions and operations described in the previous sections.
With the CPO and the end-user participants in the diagrams we intend to

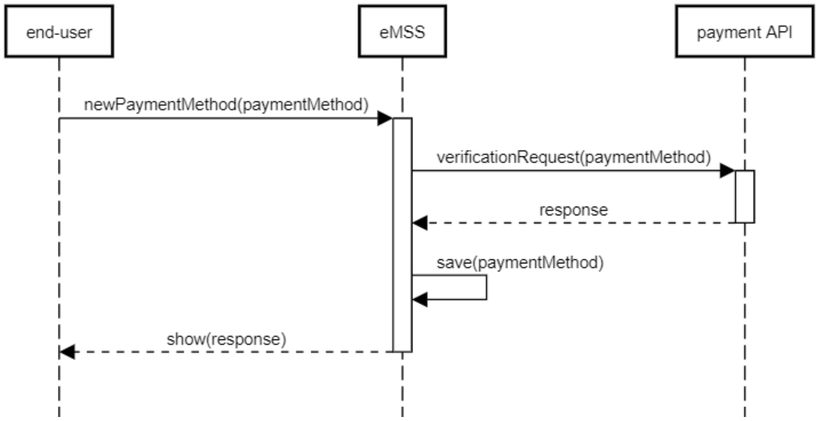
represent the clients of the application: the smartphone app for the end-users and the web page for the CPOs.

3.5.1 End-user registers



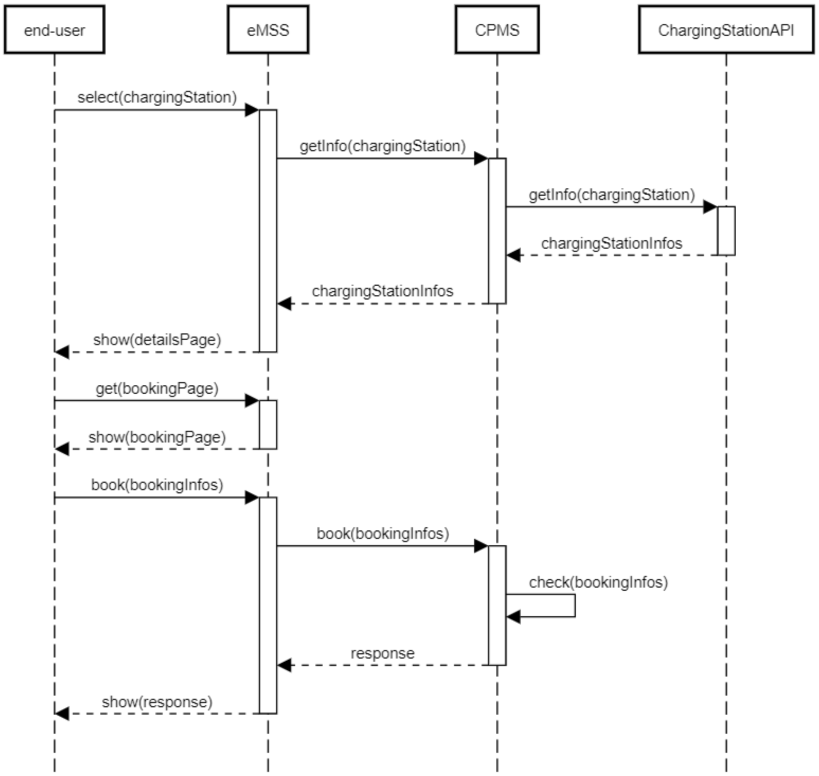
Name	End-user registers
Actors	End-user
Entry conditions	• The end-user is not yet registered into the system
Flow of events	• The end-user presses the *Sign Up* button • The end-user submits the requested information
Exit conditions	The system creates a new account for the user and the user is logged into the application
Exceptions	• if the end-user closes the application in the middle of the process, the system should allow the end-user to continue the operation once he re-opens the application • if the end-user insert incompatible data, the system should prevent the user to complete the operation and should warn the user with a message • if the payment method verification results in an error, the system should prevent the user to complete the operation and should warn the user with a message

3.5.2 End-user changes payment method


end-user
eMSS
payment API
newPaymentMethod(paymentMethod)
verificationRequest(paymentMethod)
response
save(paymentMethod)
show(response)

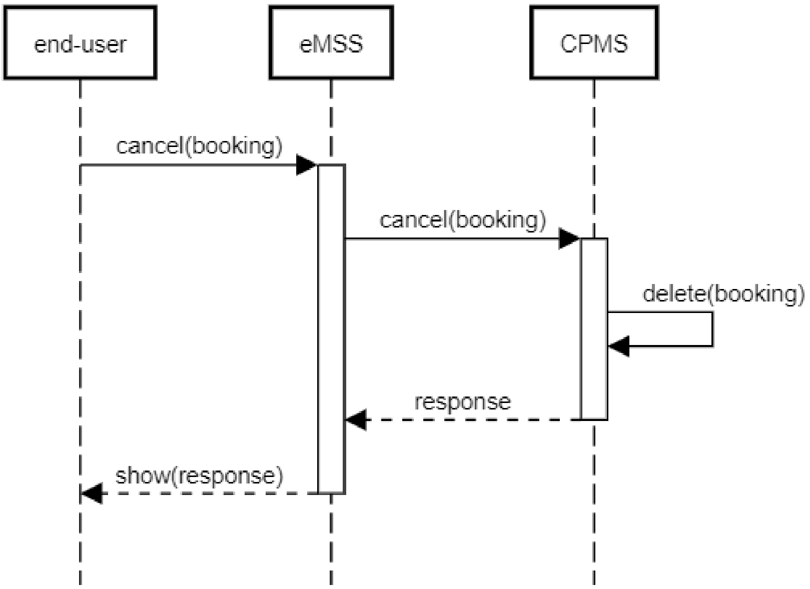
Name	End-user changes payment method
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the settings page
Flow of events	• The end-user presses the *Change payment method* button • The end-user submits the requested information
Exit conditions	The payment method of the user is changed to the new one
Exceptions	• if the end-user inserts incompatible data, the system should prevent the user to complete the operation and should warn the user with a message • if the payment method verification results in an error, the system should prevent the user to complete the operation and should warn the user with a message

3.5.3 End-user books a charge



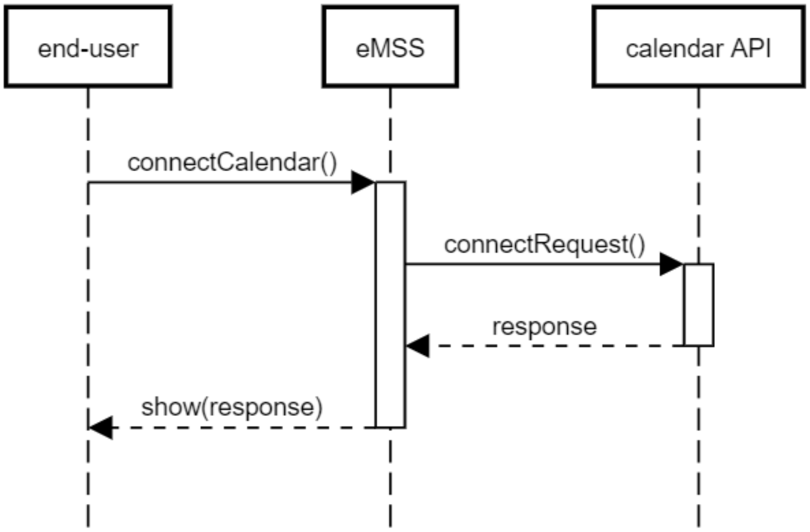
Name	End-user books a charge
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the stations page
Flow of events	• The end-user selects a charging station • The end-user presses the *Book a charge* button • The end-user selects an available time frame and an available socket type • The end-user presses the *Confirm* button
Exit conditions	The system saves the reservation
Exceptions	• if the end-user submits an unavailable time frame or an unavailable socket type, the system should prevent the user to complete the operation and should warn the user • if the end-user presses the *Confirm* button without selecting an available time frame or an available socket type, the system should prevent the user to complete the operation and should warn the user

3.5.4 End-user cancels a booking



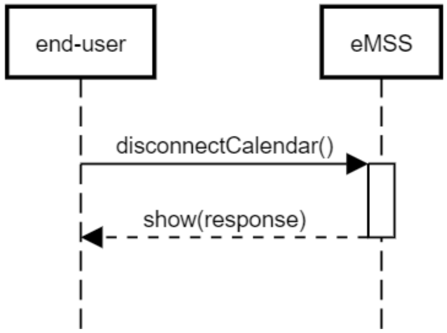
Name	End-user cancels a booking
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the personal page • The end-user has a booking that wants to cancel
Flow of events	• The end-user presses the *Cancel* button near the reservation he wants to cancel
Exit conditions	The system cancels the reservation and it will no longer be visible in the personal page
Exceptions	-

3.5.5 End-user connects calendar



Name	End-user connects calendar
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the settings page
Flow of events	• The end-user presses the *Connect Calendar* button • The end-user selects his calendar cloud service and logs in
Exit conditions	The system connects with the end-user's calendar API
Exceptions	-

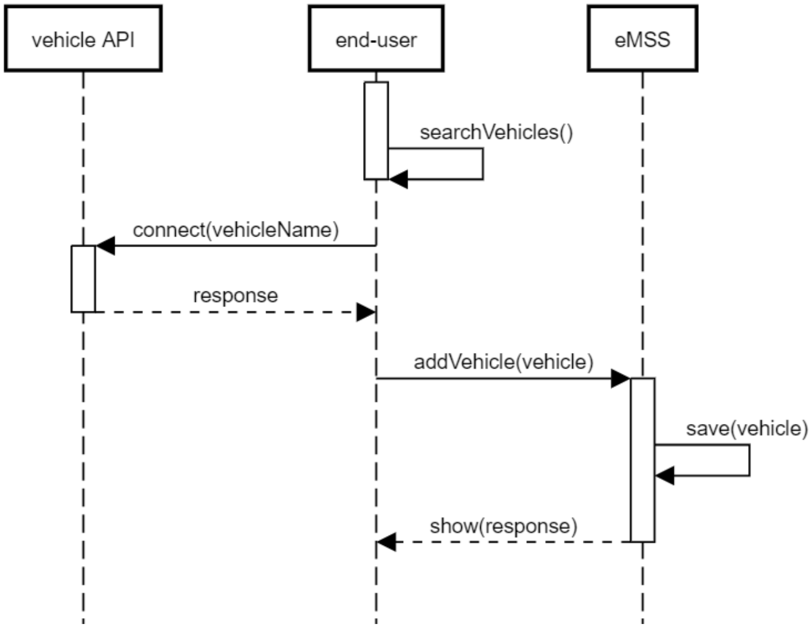
3.5.6 End-user disconnects calendar



Name	End-user disconnects calendar
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the settings page • The end-user has already connected his calendar to the system
Flow of events	• The end-user presses the *Disconnect Calendar* button
Exit conditions	The system is no longer connected with the end-user's calendar API
Exceptions	-

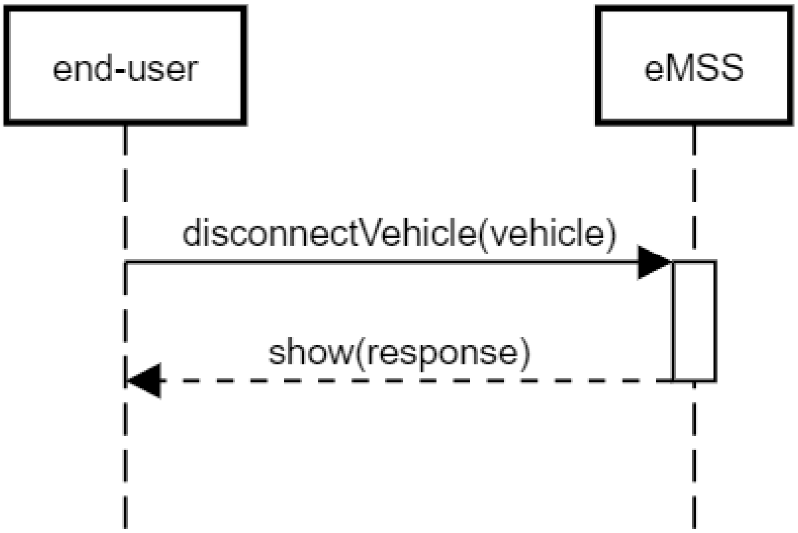
NOTE: The *Disconnect Calendar* button only appears in place of the *Connect Calendar* button when a calendar is connected.

3.5.7 End-user connects a vehicle



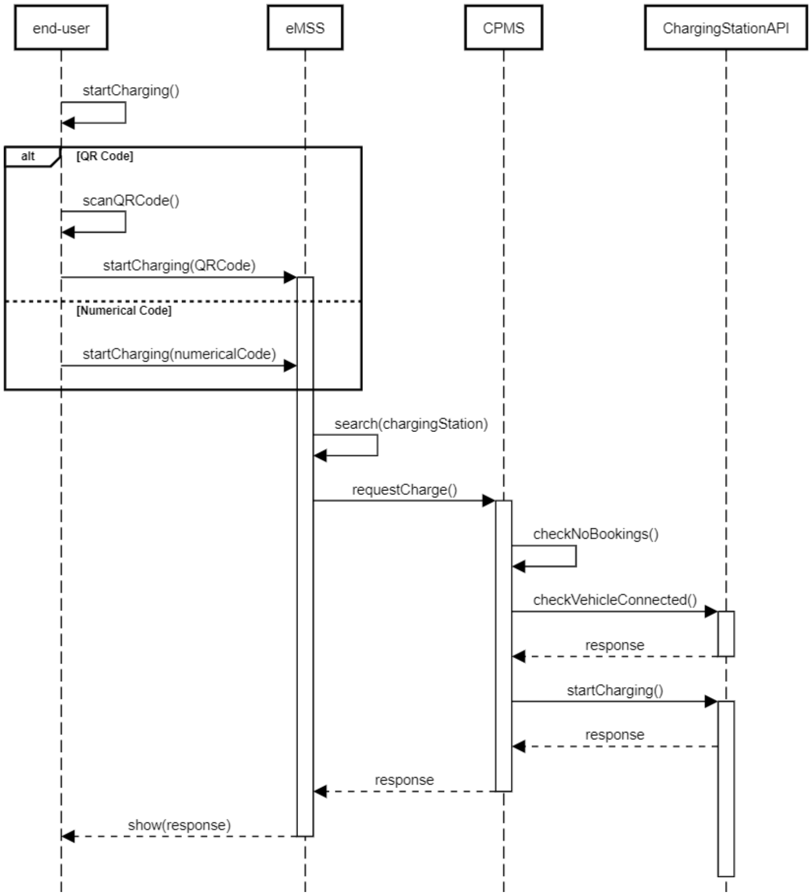
Name	End-user register a vehicle
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the vehicles page
Flow of events	• The end-user presses the *Connect a vehicle* button • The end-user selects the vehicle he wants to add
Exit conditions	The system registers the vehicle
Exceptions	• if there is no vehicle to connect with, the system should warn the end-user with a *no vehicles found* message • if the connection fails, the system should warn the end-user with an error message • if the selected vehicle is already saved, the system should warn the user with a massage

3.5.8 End-user disconnects a vehicle



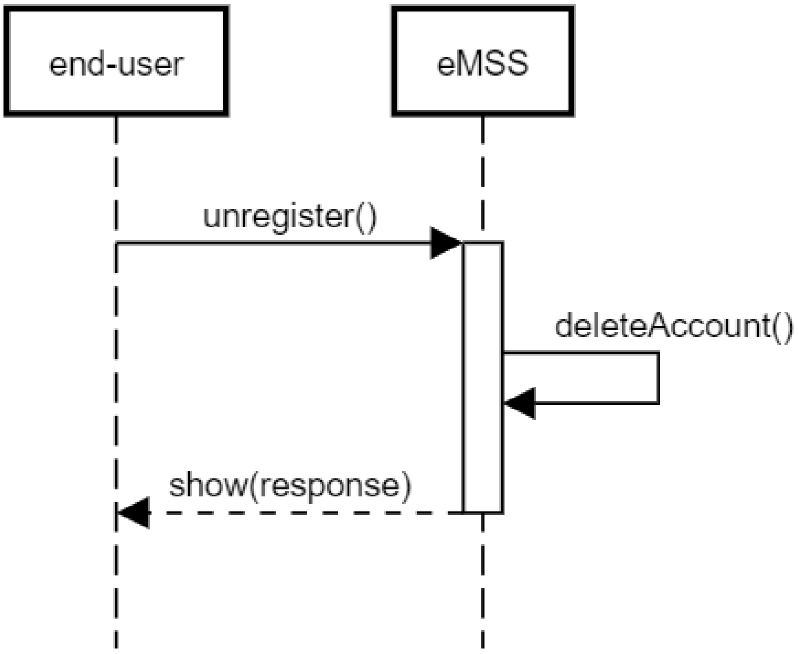
Name	End-user disconnects a vehicle
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the vehicles page • The end-user has connected at least one vehicle
Flow of events	• The end-user presses the *Disconnect* button near the vehicle that wants to disconnect
Exit conditions	The application is no longer connected with the vehicle
Exceptions	-

3.5.9 End-user starts charging a vehicle



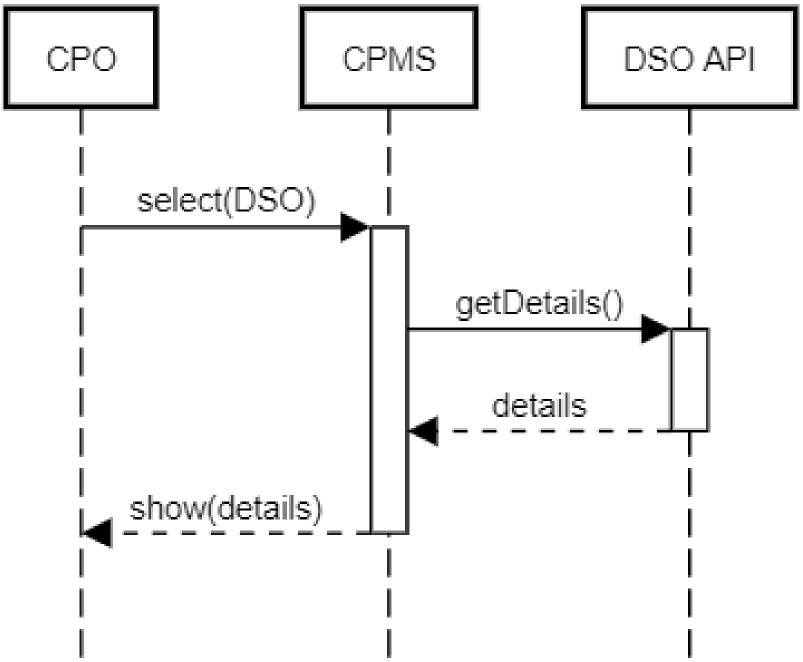
Name	End-user starts charging a vehicle
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the stations page • The end-user has his vehicle plugged into a charging socket
Flow of events	• The end-user presses the *Start Charging* button • The end-user chooses one method to identify the socket (QR code or numerical code)
Exit conditions	The system starts to charge the vehicle
Exceptions	• if the end-user has connected his vehicle to a reserved charging station, the system must prevent the end-user from starting the charge and should warn the end-user with an error message • if the end-user tries to start the charging process but no car is connected to the socket, the system must prevent the end-user from starting the charge and should warn the end-user with an error message

3.5.10 End-user unregisters



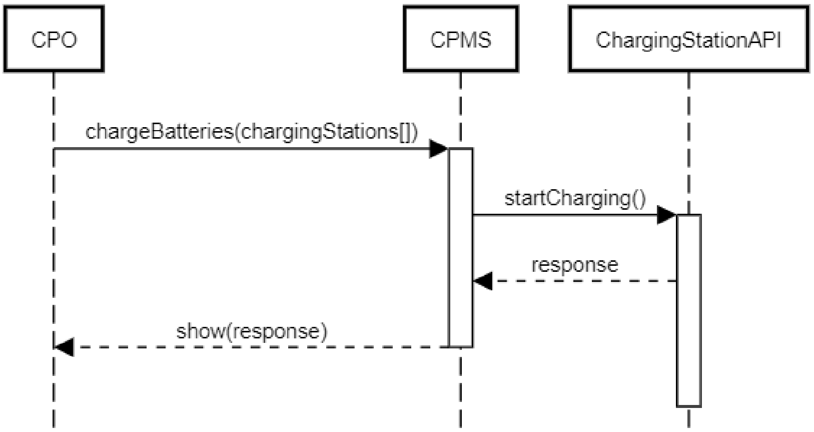
Name	End-user unregisters
Actors	End-user
Entry conditions	• The end-user is logged into the system • The end-user is on the settings page
Flow of events	• The end-user presses the *Unregister* button • The end-user confirms his/her decision
Exit conditions	The end-user account and all data relative to it are deleted from the system
Exceptions	-

3.5.11 CPO looks at the details of a DSO



Name	CPO looks at the details of a DSO
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the DSO page
Flow of events	• The CPO selects the DSO to view the details of
Exit conditions	The system shows the details of the selected DSO
Exceptions	-

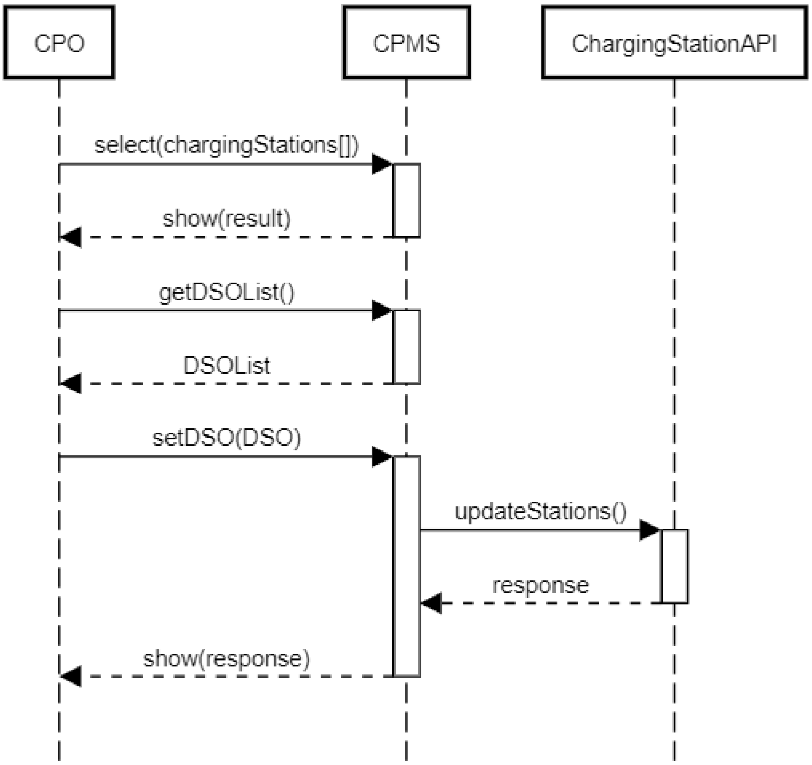
3.5.12 CPO charges batteries



Name	CPO charges batteries
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Charge Batteries* button • The CPO presses on the *Confirm* button
Exit conditions	The batteries of the selected stations start charging
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if the CPO selects some station that does not have storage batteries, the system should execute the action for all the other stations ignoring them • if the CPO selects only stations that do not have storage batteries, the system should warn the user with a message

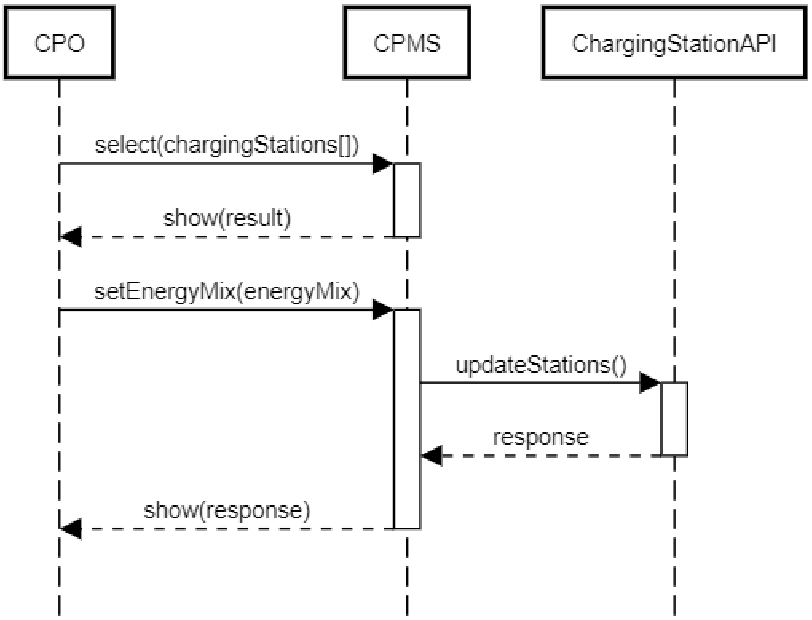
NOTE: A notification will be sent to the CPO application once the charging is completed

3.5.13 CPO selects DSO



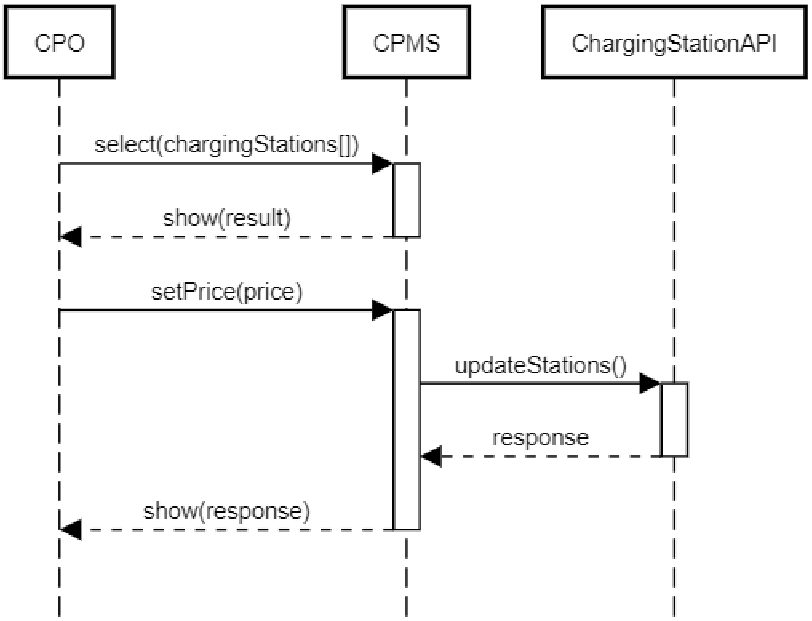
Name	CPO selects DSO
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Set DSO* button • The CPO select a DSO among the ones that are available • The CPO presses on the *Confirm* button
Exit conditions	The selected stations change their DSO
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if the CPO goes back before confirming, the system should not make changes • if the CPO closes the application in the middle of the process, the system should allow the CPO to continue the operation once he re-opens the application

3.5.14 CPO selects energy mix



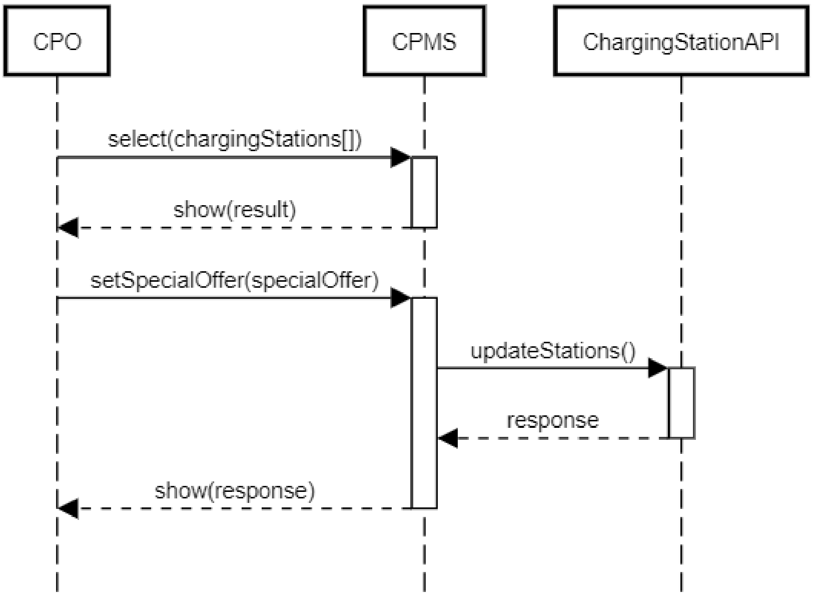
Name	CPO selects energy mix
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Set Energy Mix* button • The CPO sets the preferred mix • The CPO presses on the *Confirm* button
Exit conditions	The energy mix of the selected station is changed
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if the CPO goes back before selecting a mix, the system should not make changes • if the CPO closes the application in the middle of the process, the system should allow the CPO to continue the operation once he re-opens the application

3.5.15 CPO set prices



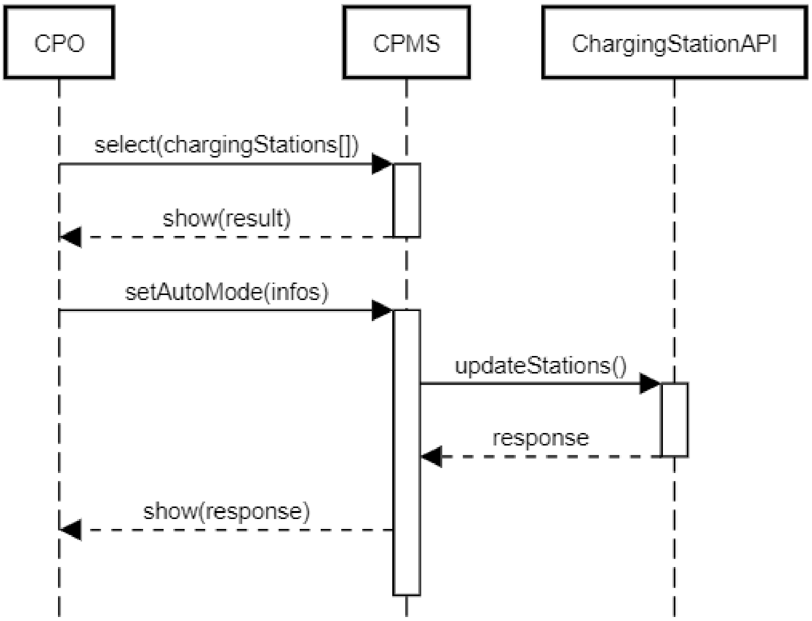
Name	CPO set prices
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Set Prices* button • The CPO inserts new prices • The CPO presses on the *Confirm* button
Exit conditions	The prices in the selected stations are updated
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if the CPO submits illegal values, the application should warn the user with a message and prevent any change • if the CPO closes the application in the middle of the process, the system should allow the CPO to continue the operation once he re-opens the application

3.5.16 CPO set special offers



Name	CPO set special offers
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Set special offers* button • The CPO submit the requested infos about the special offer • The CPO presses on the *Confirm* button
Exit conditions	A new Special Offer is added, and is visible in the details of the associated stations
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if some of the required infos are missing, the application should warn the user with a message and prevent the submission until all fields are filled with legal values • if the CPO closes the application in the middle of the process, the system should allow the CPO to continue the operation once he re-opens the application

3.5.17 CPO set Auto Mode



Name	CPO set Auto Mode
Actors	CPO
Entry conditions	• The CPO is logged into the system • The CPO is on the stations page
Flow of events	• The CPO selects one or more stations • The CPO presses on the *Set Auto Mode* button • The CPO inserts the requested infos • The CPO presses on the *Confirm* button
Exit conditions	The selected stations are set to auto-mode
Exceptions	• if the CPO goes back before confirming, the system should not make changes • if some of the required infos are missing, the application should warn the user with a message and prevent the submission until all fields are filled with legal values

3.5.18 Mapping on Requirements

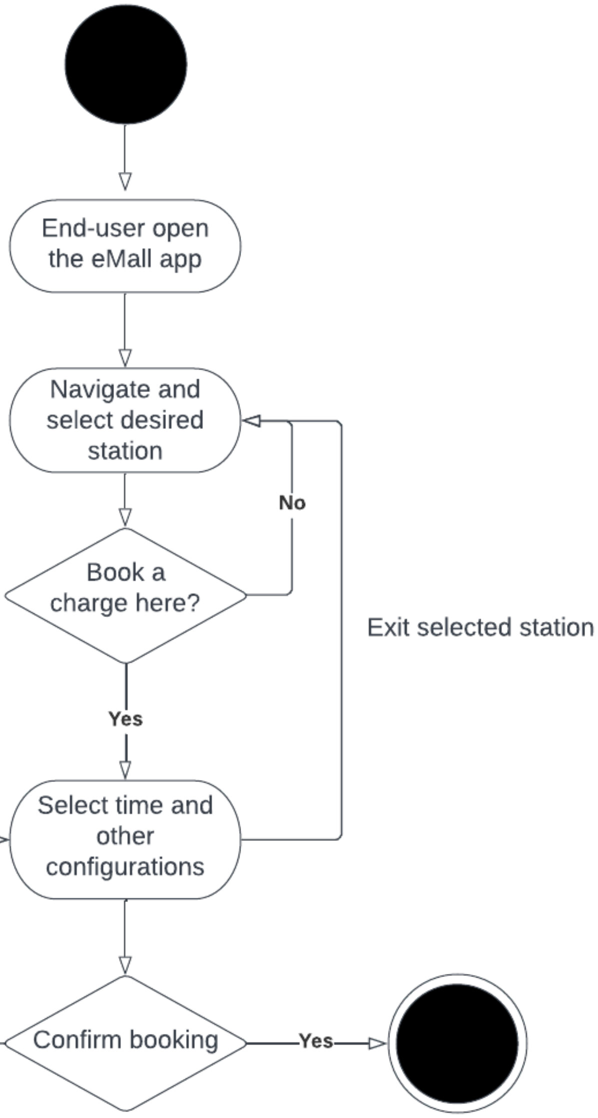
Use Case	Requirements
End-user registers	R1, R3
End-user changes payment method	R16
End-user books a charge	R4, R8, R17
End-user cancels booking	R14
End-user connects calendar	R17
End-user disconnects calendar	R18
End-user connects a vehicle	R5
End-user disconnects vehicle	R6
End-user pays for a charge	R13
End-user starts charging a vehicle	R7, R10
End-user unregisters	R2
CPO looks at the details of a DSO	R29, R31
CPO charges batteries	R25, R28
CPO selects DSO	R26, R28
CPO selects energy mix	R26
CPO set prices	R22
CPO set special offers	R23
CPO set Auto Mode	R27

3.6 Activity diagrams

In this section we will show the activity diagrams of two of the main features

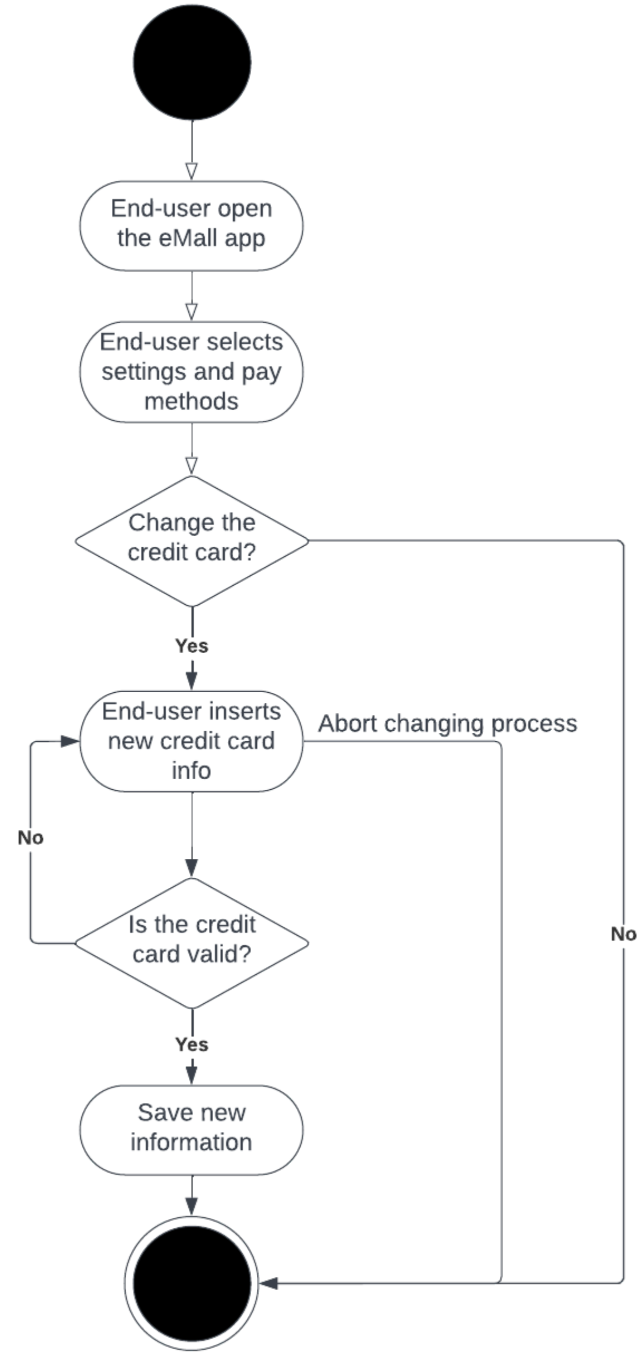
3.6.1 End-user book a charge

The following diagram describes the process of the user booking a charge in a specific station


End-user open the eMall app
Navigate and select desired station
No
Book a charge here?
Exit selected station
Yes
Select time and other configurations
No
Confirm booking
Yes

3.6.2 End-user change payment method

The following diagram describes the process of the end-user changing his payment method


End-user open the eMall app
End-user selects settings and pay methods
Change the credit card?
Yes
No
End-user inserts new credit card info
Abort changing process
No
Is the credit card valid?
Yes
Save new information

3.7 Performance requirements

The system should at least guarantee a proper experience with a minimum download speed of 5Mb/s and a minimum upload speed of 1Mb/s.

The heaviest processing in the system will be managed by the DBMS, that needs to handle a lot of data. The most obvious case in which the system needs to retrieve much data from the database is the loading of the stations map. This shouldn't take longer than 5 seconds to ensure proper usability.

Other minor operations, like changing the payment method, confirming a booking or changing the settings of a station should be handled in even shorter times, as they are simpler and their main time obstacle is passing data over the internet. Thus, we consider an upper bound of 3 seconds for the other activities.

3.8 Design constraints

3.8.1 Standard compliance

For the proper functioning of the system, it is essential to control specific restricted functionalities of their smartphone, like GPS and push notifications. For this reason, the end-user needs to grant these privileges to the eMall app.

CPOs, on the other side, do not need to grant any permissions for the web application to work.

3.8.2 Hardware and software constraints

The end-user's smartphone needs to have:

- Access to a stable internet connection of at least of 5Mb/s of download speed and 1Mb of upload speed
- Enough memory to download and run the application. The actual constraint will depend on the final app size and the size of its cache files.
- An appropriate OS version:
  - Android 7 - Marshmallow or newer
  - IOS 15 or newer

The CPO's device needs to have:

- Access to a stable internet connection of at least of 5Mb/s of download speed and 1Mb of upload speed
- One of the supported browsers installed:
  - Google Chrome

  - Safari
  - Microsoft Edge
  - Mozilla Firefox

The server needs to have:

- Access to a stable internet connection of at least of 100Mb/s of download speed and 100Mb of upload speed
- An OS on which to install the DBMS
- Enough memory and computing power to manage a load of concurrent requests in the order of ten thousands
- Enough disk space to store the database. This depends on the volume of users, CPOs, stations etc...

3.9 Software system attributes

3.9.1 Reliability

Since we want to guarantee a good user experience, the system should be at least 99% reliable, so that, on average, only 1 operation over 100 fails. Therefore, the system should be fault tolerant, as failures may occur. The server should also be replicated and should take periodic snapshots to prevent data loss.

3.9.2 Availability

The system should offer high availability as users may need the service at any time. We should guarantee an availability of at least 99.9% in order to have at most 8.77 hours of downtime per year.

3.9.3 Security

Security is a major pillar of the system, as it deals with sensitive data, like the end-users' position or their payment credentials. The following security measures should be adopted:

- Require strong passwords (at least 8 characters, at least a lower-case letter, an upper-case letter and a number)
- Use double-verification for logins on new devices by requesting a verification code sent by email
- Encrypt all the data relative to end-users and CPOs
- Use technologies such as firewalls and Intrusion Prevention Systems to secure the server

3.9.4 Maintainability

The system should divide the different functionalities into modules that communicate with each other in order to simplify testing, bug fixing and the development of updates. Error logs should also be collected and stored automatically to know the most common and critical errors and to be able to provide fixes in short times.

3.9.5 Portability

The system will be released as a smartphone app dedicated to end-users on Android and IOS and as a web application dedicated to CPOs that will be supported by the most common browsers. The web application should work on every device, regardless of the its screen size, even though it is mostly intended to be used in PC environments.

4 Formal analysis

In this section, we will present a formal model of the system realized with Alloy. The following model is based on the class diagram and represents the core of the system and the main constraints.

4.1 Alloy model

```
//----------Signatures----------//

//------------USERS-------------//
abstract sig User {}

sig EndUser extends User {
    paymentMethod: one PaymentMethod,
    calendar: lone Calendar,
    vehicles: set Vehicle,
    bookings: set Booking,
    charges: set Charge,
    suggestion: lone Suggestion
}

sig CPO extends User {
    chargingStations: some ChargingStation,
    listDSO: some DSO
}

//-------------eMSS-------------//
sig Vehicle {}

abstract sig Notification {
    dateTime: one DateTime,
}
```

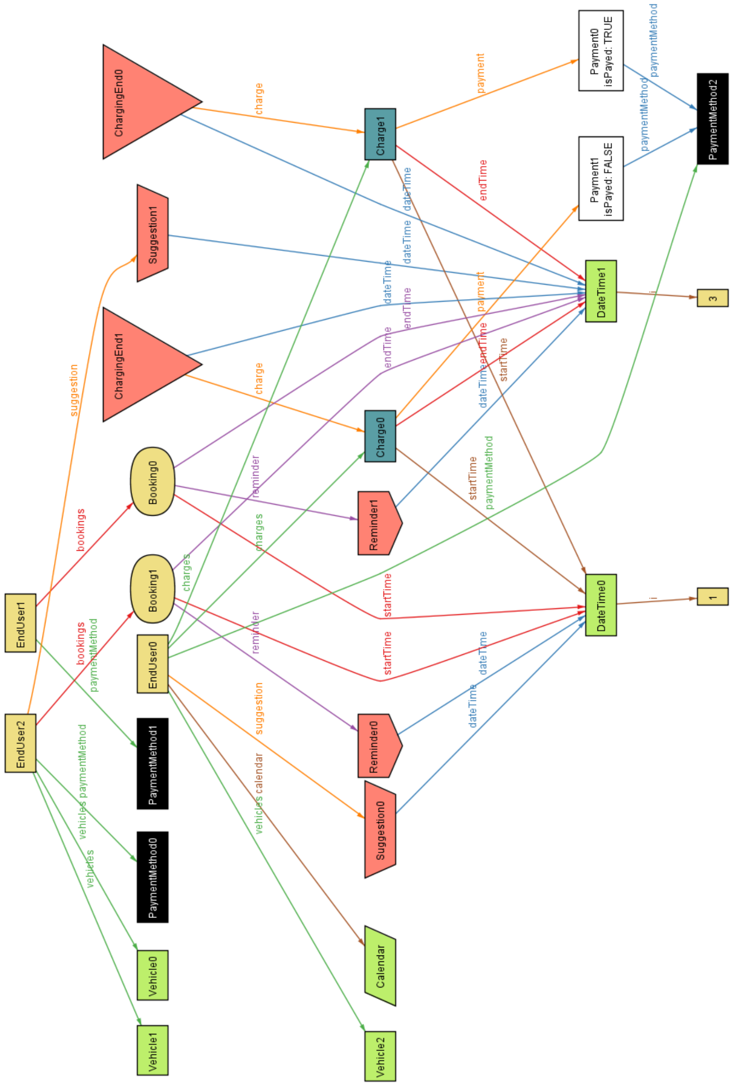
```
sig Reminder extends Notification {}

sig ChargingEnd extends Notification {
    charge: one Charge
}

sig Suggestion extends Notification {
    chargingStation: some ChargingStation
}

sig Booking {
    startTime: one DateTime,
    endTime: one DateTime,
    chargingSocket: one ChargingSocket,
    reminder: one Reminder
}

sig PaymentMethod {}

sig Payment {
    isPayed: one Boolean,
    paymentMethod: one PaymentMethod
}

//-------------CPMS-------------//
sig ChargingStation {
    location: one Location,
    cost: one CostTable,
    chargingSockets: some ChargingSocket,
    connectedDSO: one DSO,
    listSpecialOffers: some SpecialOffer
}

sig ChargingSocket {
    isOccupied: one Boolean,
    type: one ChargingSocketType
}

sig SpecialOffer {
    startTime: one DateTime,
    endTime: one DateTime,
    prices: one CostTable
}

//--------shared classes--------//
sig Charge {
    startTime: one DateTime,
    endTime: one DateTime,,
    payment: one Payment,
    chargingSocket: one ChargingSocket
}

//-------external classes-------//
sig Calendar {}
sig DSO {}

//------Defining new types------//
abstract sig ChargingSocketType {}
one sig SLOW extends ChargingSocketType {}
one sig FAST extends ChargingSocketType {}
one sig RAPID extends ChargingSocketType {}

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

sig DateTime {
    i: one Int
}{i > 0}

sig Location {}

sig CostTable {}

sig Float {}

//------------Facts-------------//

//-------eMSS constraints-------//
fact eachEndserHasOnePaymentMethod {
    all e: EndUser | one p: PaymentMethod |
        e.paymentMethod = p
}

fact eachPaymentMethodIsOwnedByOneEndUser {
    all p: PaymentMethod | one e: EndUser |
        e.paymentMethod = p
}

fact eachVehicleOwnedByOneEndUser {
    all v: Vehicle | one e: EndUser |
        v in e.vehicles
}

fact eachReminderHasDateTime {
    all r: Reminder | one dt: DateTime |
        r.dateTime = dt
}

fact noSharingReminders {
    all b1, b2: Booking |
        b1 != b2 implies
        b1.reminder != b2.reminder
}

fact eachBookingOwnedByOneEndUser {
    all b: Booking | one e: EndUser |
        b in e.bookings
}

fact eachChargeAssociatedToOneEndUser {
    all c: Charge | one e: EndUser |
        c in e.charges
}

fact eachPaymentAssociatedToOneCharge {
    all p: Payment | one c: Charge |
        c.payment = p
}

fact eachChargeAssociatedToOnePayment {
    all c: Charge | one p: Payment |
        c.payment = p
}

fact eachPaymentAssociatedToOnePaymentMethod {
    all p: Payment | one pm: PaymentMethod |
        p.paymentMethod = pm
}

fact oneUserForCalendar {
    all c: Calendar | one e: EndUser |
        e.calendar = c
}

fact eachChargePayedByProperUser {
    all c: Charge | all e: EndUser |
        c in e.charges implies
        c.payment.paymentMethod in e.paymentMethod
}

fact noSharingSuggestion {
    all e1, e2: EndUser |
        e1 != e2 implies
        (e1.suggestion != e2.suggestion)
}

fact oneChargeToOneChargingEnd {
    all c: ChargingEnd | one ch: Charge|
        c.charge = ch
}

fact noSharingCharge {
    all ch1, ch2: ChargingEnd |
        ch1 != ch2 implies
        (ch1.charge != ch2.charge)
}

fact rightTimeChargingEnd {
    all ce: ChargingEnd | all c: Charge |
        ce.charge = c implies ce.dateTime = c.endTime
}
```

```
fact everyPaymentMethodIsDifferent {
    all e1, e2: EndUser |
        e1 != e2 implies e1.paymentMethod != e2.
           paymentMethod
}

//----------CPMS constraints----------//
fact eachStationIsOwnedByOneCPO {
    all s: ChargingStation | one c: CPO |
        s in c.chargingStations
}

fact eachStationConnectedToALegalDSO {
    all s: ChargingStation | all c: CPO | all d: DSO |
        (d = s.connectedDSO and s in c.
           chargingStations)
        implies (d in c.listDSO)
}

fact eachStationHasDifferentLocation {
    all l : Location | one c : ChargingStation |
        c.location = l
}

fact eachSocketInOneStation {
    all so: ChargingSocket | one st: ChargingStation |
        so in st.chargingSockets
}

fact eachSocketHasType {
    all s: ChargingSocket | one t: ChargingSocketType|
        s.type = t
}

fact eachSpecialOfferBelongsToOneChargingStation {
    all spo: SpecialOffer | one ch: ChargingStation |
        spo in ch.listSpecialOffers
}

fact noSharingSpecialOffers {
    all ch1, ch2: ChargingStation |
        ch1 != ch2 implies
        ((ch1.listSpecialOffers) not in ch2.
           listSpecialOffers)
}

fact noWrongDateTimeSpecialOfffer {
    all sp: SpecialOffer |
        sp.startTime != sp.endTime
}

fact noSameCostTableChargingStations {
    all cs1, cs2: ChargingStation |
        (cs1 != cs2 implies
        cs1.cost != cs2.cost)
```

```
}

fact noSameCostTableSpecialOffer {
    all sp1, sp2: SpecialOffer |
        (sp1 != sp2 and !(sp1.startTime.i > sp2.
           endTime.i or sp1.endTime.i < sp2.startTime.
           i))
        implies sp1.prices != sp2.prices
}

fact noSameCostTableBetween {
    all sp: SpecialOffer | all cs: ChargingStation |
        sp.prices != cs.cost
}

//-----DateTime Consistence-----//
fact uniqueDateTime {
    all d1, d2: DateTime | d1 != d2 implies d1.i != d2
       .i
}

fact specialOffersTimeConsistence {
    all s: SpecialOffer | s.startTime.i < s.endTime.i
}

fact bookingsTimeConsistence {
    all b: Booking | b.startTime.i < b.endTime.i
}

fact chargingsTimeConsistence {
    all c: Charge | c.startTime.i < c.endTime.i
}

fact noChargingInSameTimespace{
    all c1,c2: Charge |
        (c1 != c2 and c1.chargingSocket = c2.
           chargingSocket)
        implies (c1.startTime.i > c2.endTime.i or c1.
           endTime.i < c2.startTime.i)
}

fact noBookingInSameTimespace{
    all b1,b2: Booking |
        (b1 != b2 and b1.chargingSocket = b2.
           chargingSocket)
        implies (b1.startTime.i > b2.endTime.i or b1.
           endTime.i < b2.startTime.i)
}

//the only case in which a booking can overlap a
   charge is that it is done by same user
fact noBookingInSameTimespaceAsCharge {
    all b: Booking | all c: Charge | all e: EndUser |
        (b.chargingSocket = c.chargingSocket and
        !(b.startTime.i > c.endTime.i or b.endTime.i <
            c.startTime.i))
        implies (b in e.bookings and c in e.charges)
}

//user cannot do a charge and a booking in same
   timespace unless is the charge associated with that
    booking
fact noOverlappingChargesOrBookingsOfUser{
    all b: Booking | all c: Charge | all e: EndUser |
        (b in e.bookings and c in e.charges and b.
           chargingSocket != c.chargingSocket)
        implies (b.startTime.i > c.endTime.i or b.
           endTime.i < c.startTime.i)
}

//------Redundant instances-----//
fact noRedundantLocations {
    all l: Location | one c: ChargingStation |
        c.location = l
}

fact noRedundantDSO {
    all d: DSO | some c: CPO |
        d in c.listDSO
}

//-------------Show-------------//

pred show {
    #CPO = 3
    #ChargingStation = 3
    #EndUser = 3
    #DSO = 4
    #ChargingSocket = 4
    #Booking = 2
    #Reminder = 2
    #Charge = 2
    #ChargingEnd = 2
    #Suggestion = 2
}

run show for 10
```

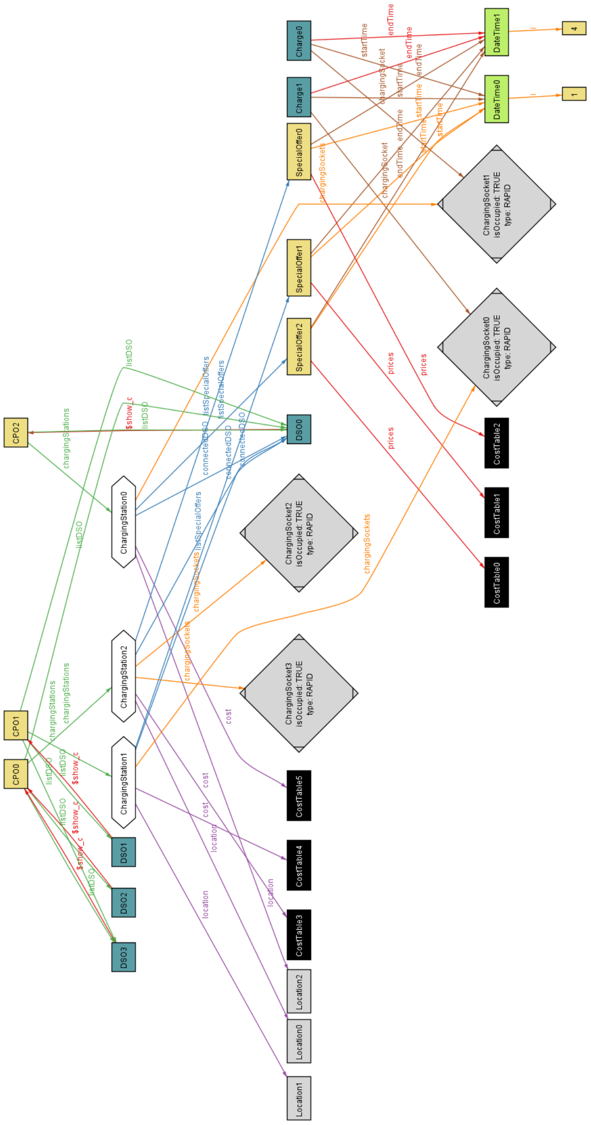
4.2 World generated - eMSS

The following image contains the world generated by alloy visualizing only the eMSS system.



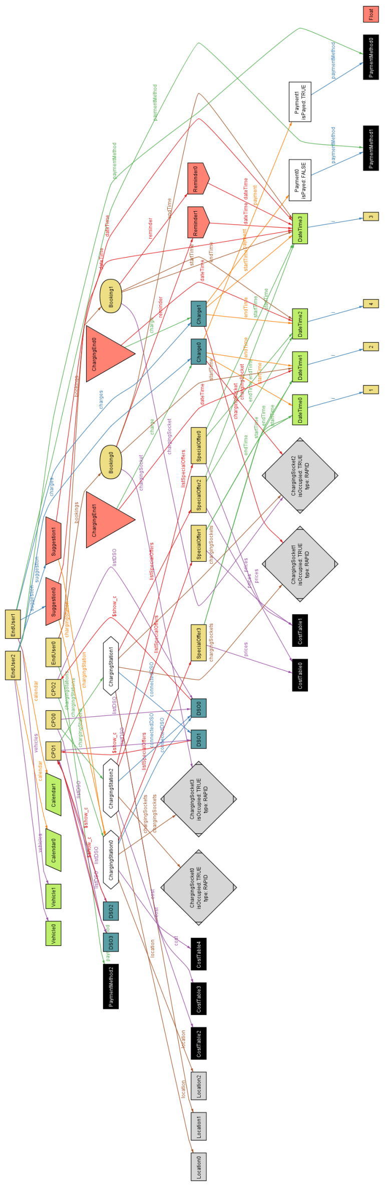
4.3 World generated - CMPS

The following image contains the world generated by alloy visualizing only the CMPS system.



4.4 World generated - eMall

The following image contains the world generated by alloy visualizing the entire eMall system.



5 Effort spent

5.1 Niccolò Nicolosi

Task	Hours
Latex document setup	1
Goals definition	3
Phenomena identification	2:30
Scenarios	2:30
Product functions	2
User characteristics	0:30
External interface requirements	4
Functional requirements	5
Requirements mapping	1
Use cases and sequence diagrams	2
Software system attributes	2
Alloy	2
Revision and adjustments	11
Total	38:30

5.2 Francesco Negri

Task	Hours
Goals definition	1:30
Phenomena identification	2:30
Scenarios	0:30
Class diagram	3:30
Use cases and sequence diagrams	15
External interface requirements	1
Alloy	12
Revision and adjustments	4
Total	40:00

5.3 Marcos Pietrucci

Task	Hours
Latex document setup	1
Goals definition	2:30
Phenomena identification	3
Scenarios	1:30
Domain assumptions	2:30
Use cases and sequence diagrams	2
Use cases mapping	0:30
Activity Diagrams	4
Design Constraints	2
Alloy	13
Revision and adjustments	3:30
Total	33:30