eMall - e-Mobility for All

Requirement Analysis and Specification Document

23/12/2022

Version 1

Authors:

Brugnano Matilde

Buttiglieri Giorgio Natale

Academic Year 2022/2023

Software Engineering 2 Project

Table of Contents

1 INTRODUCTION

1.1 Purpose

Nowadays electric mobility (e-mobility) is one of the greatest challenges of our planet.

All sectors of contemporary industry actually want to be "green": public opinion is increasingly aware of the importance of climate change and the very high risks that this implies, both for present and future generations.

According to this objective the electric vehicle market is constantly expanding, and, in parallel with this, more and more business opportunities were revealed in the electric vehicles charging-service companies, which sees eMSPs and CPOs as the main characters.

eMSP, which stands for e-Mobility Service Providers, is a company offering an electric vehicle charging service to the drivers by providing them access to multiple charging points around a geographic area.

CPO, which stands for Charge Point Operator, is a company managing a pool of charging points which creates value by providing smart charging features to eMSPs.

eMSPs and CPOs are two crucial and interdependent elements of a charging network. Together, they install and manage electric vehicle charging stations capable of supporting a fast, easy-to-use, largely automated charging system.

This document aims at providing the requirements analysis and specifications needed for the development of e-Mall (e-Mobility for All), a system made of both eMSP and CPMS needed to manage electric vehicle charging from the energy sources acquisition to the final customers service.

1.2 Acronyms

Term	Definition
eMSP	e-Mobility Service Providers: company offering a charging service to drivers of electric vehicles. Here, eMSP is intended as the software system used by the company to manage charging processes.
eMSP app	Mobile app or web app through which drivers of electric vehicles can interact with the services offered by eMSP.
CPO	Charge Point Operator: company that owns and manages a network of charging stations.
CPMS	Charge Point Management System: software system that manages the charge point infrastructure of a CPO company through its connection with the charging equipment.
CPMS app	Mobile app or web app through which CPO operators can interact with the services offered by CPMS.
DSO	Distribution System Operators: external company responsible for distributing and managing energy from the generation sources to charging stations.

1.3 Definitions

Term	Definition
Customer	End user of eMSP app, generally intended as the electric vehicle owner.
eMSP admin	eMSP employee responsible for configuring the CPOs network (so their CPMSs' one) with which their eMSP collaborates.
CPO admin	CPO employee responsible for registering CPO operators, DSO partnerships and charging stations in the CPMS app. They also need to configure the eMSP network with which their CPMS collaborates. All the operations managed by CPO operators can be performed by CPO admins as well.
CPO operator	CPO employee which can manage CPO operations through CPMS app. They lack the permissions due to the CPO admins.
CPO employee	CPO admin or operator.
Charging station	Infrastructure made up of one or more charging points for plug-in electric vehicles.
Charging point	Piece of equipment that supplies electrical power for plug-in electric vehicles made up of one or more charging sockets. We refer to charging points making a distinction on the basis of their "mode": "in advance" or "on the fly". Charging points whose mode is "in advance" can be reserved with bookings "in advance" only; in the same way, charging points whose mode is "on the fly" can be reserved with bookings "on the fly" only. Without losing any generality, we can both refer to the mode of sockets and the mode of charging point, assuming that all its sockets have the same mode as it.
Booking a charge in advance	Booking made when a customer wants to reserve themselves a spot in a certain charging station to recharge their vehicle in a selected future timeframe. Only charging points whose mode is "in advance" will be bookable by bookings "in advance".
Booking a charge on the fly	Booking made when a customer wants to recharge their vehicle at the same moment they reach a charging station. Only charging points whose mode is "on the fly" will be bookable by bookings "on the fly".
Booking	Booking made in advance or on the fly.
Booking code	Identifier code associated with a booking by CPMS.
Natural identifier	Something that is used in the real world as an identifier.

External status of a charging station	Characteristics of a charging station evaluated by CPOs to communicate its current availability to customers: number of charging points and their mode (in advance/on the fly), socket's number and type (slow/fast/rapid), their cost, and, if all sockets of a certain type are occupied, the estimated amount of time until the first socket of that type is freed.
Internal status of a charging station	Characteristics of a charging station evaluated by CPOs to manage and monitor the energy sources used for charging: amount of energy available in its battery, if present, number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the charge.
Pricing policy	CPO's approach to determining the price set by the company in a charging station. Each created policy contains some parameters that have to be met when prices are automatically set by CPMS (e.g. minimum/maximum percentage of cost-plus pricing added onto DSO's prices).
Default pricing policy	Default global pricing policy applied to all managed charging stations. Pricing policies can be also customised and associated to a subset of charging stations by CPO employees.
Payment gateway	Merchant service provided by an e-commerce application service provider that authorises credit card or direct payments processing. Here, payment gateway is intended as the third party service that handles payments between CPMS company, eMSP company and customers.
Energy switcher	Third party service that physically handles the use of charging stations' batteries and the energy acquisition from DSOs given a specific mix of energy sources by CPMS. Every charging station is managed by an energy switcher, which is able to disconnect a battery from the grid, use it as a source of energy, store energy in it, or to switch on/off the energy acquisition from a specific DSO.
Map service	Third party service that provides the map view of a certain geographical area. eMSP exploits this service to show charging stations nearby the customer.
Mail service	Third party service that provides companies (here, CPMS and eMSP) with tools to send emails.

1.4 Abbreviations

Term	Definition
G	Goal
WP	World Phenomena
SPW	Shared Phenomena, World-controlled

SPM	Shared Phenomena, Machine-controlled
SC	Scenario
STD	State Diagram
DA	Domain Assumption
R	Functional Requirement
UC	Use Case
SQD	Sequence Diagram
UCD	Use Case Diagram

1.5 Goals

#	Description
G.1	Customers want to know the current availability of charging stations nearby, their location, the current price and any offer they have.
G.2	Customers want to book a charge in a specific charging station for a certain timeframe and manage its bookings.
G.3	Customers want to charge their electric vehicles at a certain station and monitor the charging process.
G.4	CPOs want to automatically set prices and manually set offers for each charging station.
G.5	CPOs want to manage (both manually and automatically) for each charging station the used mix of energy sources (station batteries, DSOs) and optionally decide to store energy in batteries.
G.6	CPOs want to know the location and the status of a charging station, both internal and external.
G.7	CPOs want to offer the customers the possibility to charge their electric vehicles.
G.8	CPOs want to offer the customers the possibility to book a charge.
G.9	CPOs want to keep track of their energy consumption.

A comment on the payments required for the system to operate may be needed.

Payments from customers to the eMSP company and the ones from the eMSP company to CPOs are implicitly included in G.3 and G.7, as payments are intended as the means through which the charging service takes place.

Payments towards DSOs for the energy consumption of CPOs were instead omitted from the scope of the project, as they were assumed to be carried out through external services according to long-term specific commercial agreements. One of the proposed goals for this

analysis aims at keeping track of CPOs energy consumption, so the companies can have a total overview of their costs over the suppliers, but the payments themselves are omitted here.

1.6 Scope

The analysis carried out within this document focuses on the requirements and specifications of both eMSP and CPMS softwares. The two systems are thought of as separate, but their cooperation is needed to accomplish the commissioned goals.

In the following analysis we will refer to a single CPMS partnered with a single eMSP, but the same specifications hold for the communication between an eMSP and multiple CPMSs compliant with the same standards. In particular, an eMSP can interact with multiple CPMSs each of which is owned by a different CPO. Based on our assumptions, the network of CPMSs in communication with the eMSP is configured by the eMSP admins, while the network of eMSPs in communication with each CPMS is configured by the CPO admins.

Referring to the terminology used in “The World and the Machine” (Jackson M., 1995), the whole system made of eMSP and CPMS(s) is here considered as the “machine”, while the “world” is intended as the portion of the real-world affected by this system, for example the customers, the charging points, the electric vehicles, the DSOs and the CPO operators which interact with the CPMS.

1.6.1 World phenomena

#	Description
WP.1	Customer decides to charge their electric vehicle
WP.2	Customer stops by a charging point
WP.3	Customer decides to book a charge in advance
WP.4	Customer is billed for a charge
WP.5	Electric vehicle of a customer runs low in battery
WP.6	The socket of a charging point delivers energy to the vehicle
WP.7	eMSP company receives money for its customers’ charges
WP.8	CPO signs commercial deals with a DSO
WP.9	Charging stations receive energy from DSOs

1.6.2 Shared phenomena World-controlled

#	Description
SPW.1	Customer registers into the eMSP
SPW.2	Customer logs into the eMSP

SPW.3	Customer queries the eMSP for charging stations nearby
SPW.4	Customer sends a request for booking a charge in advance to the eMSP
SPW.5	Customer sends a request for booking a charge on the fly to the eMSP
SPW.6	Customer sends a request for cancelling one of their bookings to the eMSP
SPW.7	Customer plugs a socket into their electric vehicle
SPW.8	Customer plugs a socket out from their electric vehicle
SPW.9	Customer sends a request for starting the charging process to the eMSP
SPW.10	The electric vehicle informs the charging point that its battery is fully charged
SPW.11	The charging point informs the CPMS that its own socket is ready to deliver energy
SPW.12	The charging point informs the CPMS that its own socket has started delivering energy
SPW.13	The charging point informs the CPMS that its own socket has stopped delivering energy
SPW.14	The charging point informs the CPMS that its own socket has been plugged into a vehicle
SPW.15	The charging point informs the CPMS that its own socket has been plugged out from a vehicle
SPW.16	The charging point informs the CPMS that the battery of a vehicle plugged to its own socket is fully charged
SPW.17	The charging point informs the CPMS that its own socket is available for other charging processes
SPW.18	CPO admin registers a CPO operator into the CPMS
SPW.19	CPO operator logs into the CPMS
SPW.20	CPO admin manages (register, edit, delete) a DSO partnership through the CPMS
SPW.21	CPO admin manages (register, edit, delete) a charging station through the CPMS
SPW.22	CPO admin configures the eMSPs' network through the CPMS
SPW.23	eMSP admin configures the CPMSs' network through the eMSP
SPW.24	CPO employee manually changes the mix of energy sources (DSOs, station's battery) through the CPMS app
SPW.25	CPO employee manages a pricing policy (create, edit, delete, flag it as default pricing policy)

SPW.26	CPO employee manually assigns a pricing policy to a charging station
SPW.27	CPO employee manually sets offers for a charging station through the CPMS app

1.6.3 Shared phenomena Machine-controlled

#	Description
SPM.1	eMSP sends to the customer the charging stations close to them
SPM.2	eMSP sends information about a charging station to the customer
SPM.3	eMSP sends to the customer a confirmation code for a booking the customer requested to reserve (booking code)
SPM.4	eMSP sends a confirmation message to the customer to inform them that the charging point is ready to deliver energy to the vehicle
SPM.5	eMSP sends a notification to the customer to inform them that one of their bookings has expired
SPM.6	eMSP sends a confirmation message to the customer to inform them that the selected booking has been successfully cancelled
SPM.7	eMSP sends a notification to the customer to inform them that their booking is completed
SPM.8	eMSP sends a request to the payment gateway to bill the customer for a charge
SPM.9	eMSP sends a request to the map service to render the map in a certain geographical area
SPM.10	CPMS sends a request to the charging point to get its own socket ready for delivering energy
SPM.11	CPMS sends a request to the charging point to deliver energy to the plugged vehicle from its own socket
SPM.12	CPMS sends a request to the charging point to stop delivering energy to the plugged vehicle from its own socket
SPM.13	CPMS sends a request to the charging point to make its own socket available for other charging processes
SPM.14	CPMS requests to DSOs to get their pricings
SPM.15	CPMS sends a request to the energy switcher of a charging station to physically apply a specific mix of energy sources.

1.7 Revision History

- 23/12/2022: Version 1

1.8 Reference Documents

- “Assignment RDD AY 2022-2023_v3”
- Jackson, M., “The World and the Machine”, 1995

1.9 Document Structure

This document is composed of five sections, detailed below.

In the first section the application domain is introduced together with the associated goals of the project. It also provides an analysis of the world and the shared phenomena defining the scope of the project.

The second section contains an overall description of the system, including the most important product functions and the domain assumptions necessary to reach the goals described in the first section. Moreover, in this section you can find several scenarios, user characteristics, the class diagram and the main state diagrams which could help the understanding of the system.

The third section details the requirements the project aims to fulfil: functional and non-functional requirements, external interface requirements and performance requirements. Use cases are also presented, associated with use case diagrams and sequence diagrams.

The fourth section contains a formal analysis of a selected portion of the model using Alloy.

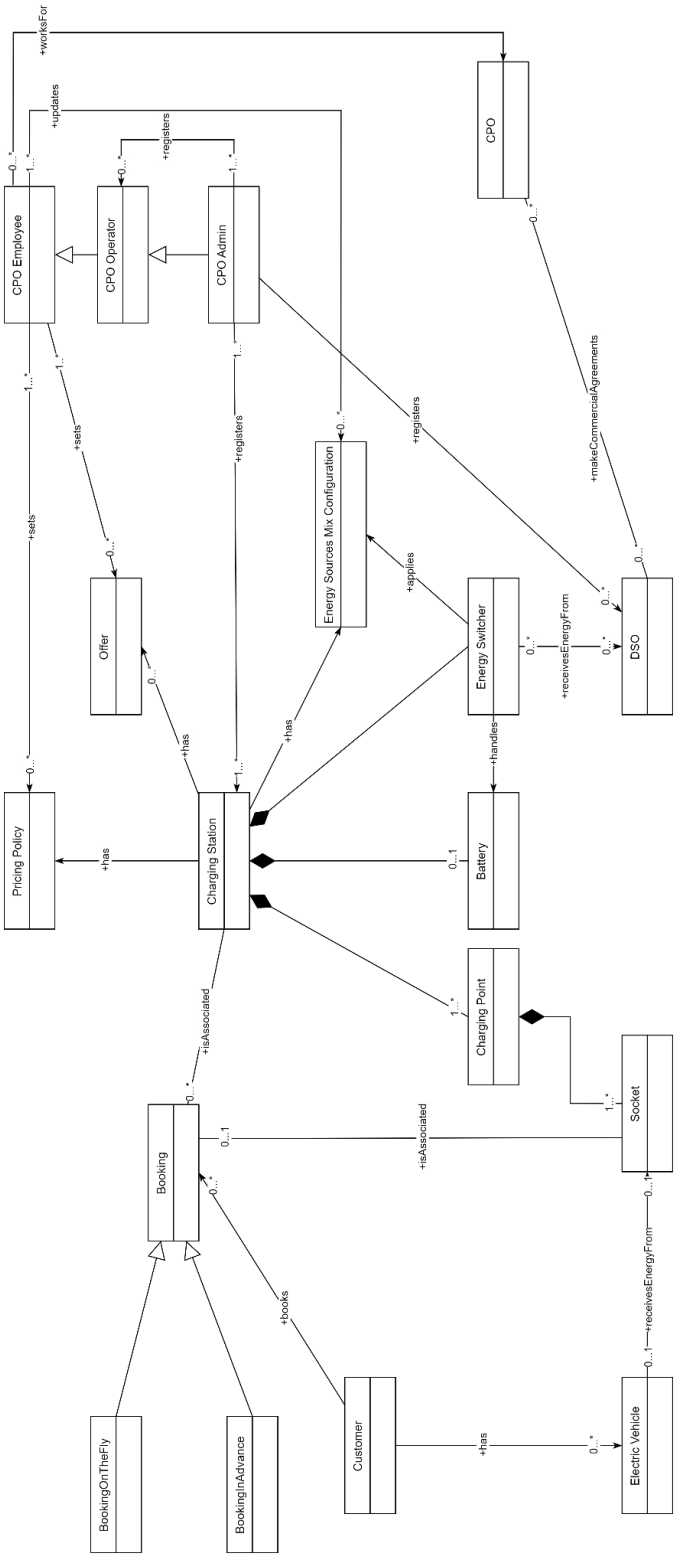
The fifth section reports each project member’s contribution to this document.

2 OVERALL DESCRIPTION

2.1 Product perspective

2.1.1 Class diagram

The following class diagram depicts the relationships between the main concepts of our model.



2.1.2 Scenarios

[SC.1] Customer needs to find a suitable charging station

Alice is driving towards her workplace when she realises that her electric vehicle is low in battery. She does not have much time, so she would like to find a charging station with fast sockets. Moreover she would like to find a charging station that is approximately on the same route she usually takes to go to work.

She decides to open her favourite eMSP app. The app offers her a section in which stations nearby are shown in a map. Alice realises that a charging station near her workplace offers a good price and lots of available fast sockets. She feels lucky and decides to go forward to the charging station without spending time booking a charge in advance through the app.

[SC.2] Customer books a charge in advance

Bob and his family have planned to have a picnic at the park for the weekend. His daughter reminds him that the last time they went to that park, they spent almost an hour trying to find a parking slot. Indeed the park is quite crowded, especially on the weekends.

Moreover their new electric vehicle is at 25% of battery, so they would need to recharge it before the beginning of next week.

Bob decides to solve both problems at the same time, leaving his car at a charging station next to the park for the duration of the picnic.

He opens his eMSP app and lands on a page showing a map with charging stations nearby. He finds a charging station next to the park, selects it, and decides to book a charge in advance there. He fills and submit a form selecting Sunday 23th of June from 12:30 pm to 2:30 pm as the desidered charging timeframe and he chooses a fast socket.

The app finally shows him a success message containing a code associated with the booking.

[SC.3] Customer starts charging his vehicle by booking a charge on the fly

Carl is driving home from work. On his way to home, he encounters a charging station and he decides to stop and charge his car.

He stops by the closest charging point available for charges “on the fly”, opens his eMSP app and lands on a page showing a map with charging stations nearby. He selects the charging station where he stopped. The app offers him a form to book a charge on the fly in that station. Carl fills the form with the identifier of the socket in the charging point next to his car (printed on the charging point).

The app displays a confirmation message: Carl can plug in the socket and start charging the car.

[SC.4] Customer charges his vehicle with a booking in advance

David arrives with his electric car at a charging station for which he had booked a charge in advance.

He opens the eMSP app, which offers him a section in which he can find all the charges he had booked in advance. David finds the booking he previously made and selects it. The app offers him the possibility to start the charge, which David does.

He reads through the eMSP app the identifier of the socket associated with his booking, so he stops by the specific charging point where that code is printed on.

The app displays a confirmation message: David can plug in the socket and start charging the car.

[SC.5] Customer is notified when the charging process is finished

In the afternoon Eveline will have a walk in her favourite park. She had already planned to do this walk, so she took this opportunity to charge her car and had booked a charge in advance for the occasion.

The booking was planned between 3 pm and 4 pm. She arrives at the charging station at 3:10 pm, starts charging the car and goes for the walk.

At 4 pm Eveline receives a notification that reminds her that the slot of her booking is expired. She hurries up and arrives at the charging station in a few minutes, she removes the plug from her car and leaves.

[SC.6] Customer is charging until the vehicle battery is full

Frank needs to charge his electric car. He drives to a charging station next to him and finds an available socket. He starts the charging process and goes back to his car. While the car is charging, he decides to read his favourite book.

Immersed in his reading, he loses track of time, but fortunately he receives a notification from his eMSP app and realises that his car is fully charged. He then plugs out his car and leaves the charging station.

Once he arrived home, he also received a notification from the eMSP regarding the success of the payment for the charge just done.

[SC.7] Customer registers through the app and enters his payment credentials

George has just bought a new electric car. He was searching for applications he could use to help him with the charging process scheduling, when he discovered the eMSP app. He reads very good reviews about the offered service, so he decides to register.

He is asked to fill in a form with his personal information and his payment credentials, so that the corresponding money could be automatically withdrawn from the bank account at the end of each future charge.

After a few hours, he notices from his bank account that eMSP service has verified his payment credentials by debiting him and crediting back 1 euro.

[SC.8] CPO employee creates a new default pricing policy

Juliet is employed in a CPO company and is responsible for managing pricing policies, in order to let the CPMS automatically calculate prices for charging stations assuring certain profit margins for the company.

She decides to add a new pricing policy to the system, so she fills in a form with the name of the new pricing policy and the two percentages related to the desired minimum and maximum cost-plus pricings: 5% and 25%. She also decides to flag this pricing policy as the new default one, then she submits the form and the CPMS app answers with a successful message: from now on, until other changes in the pricing policies, all the prices automatically

calculated by CPMS (and shown in every charging station to the customers) will comply with her new default pricing policy.

[SC.9] CPO employee sets pricing policy and offers for a charging station

Harry is employed in a CPO company and takes care of pricing and special offers for each managed charging station. Thanks to a brand-new algorithm, every 30 minutes CPMS calculates a proposal of pricing, based on the current prices of DSOs and the pricing policies associated with each charging station.

At 6 am Harry arrives at the office and he looks at the suggestion shown by the CPMS app after logging in. He finds them not suitable for a recently opened charging station in the city centre, so he decides to associate it with a different pricing policy from the existing ones through the CPMS app.

Moreover, he decides to add a special offer of 10% discount for that charging station.

The CPMS app confirms the success of his operation through a message on screen.

[SC.10] CPO employee notices an automatic change in the mix of energy sources operated by CPMS

Ilary is employed in a CPO company and is responsible for monitoring the handling of energy sources in all the managed charging stations. Thanks to some advanced algorithms which compare DSOs current prices and the current mix of energy sources in charging stations, CPMS offers an automatic system to check every 30 minutes whether a better mix of energy sources is available and eventually apply it.

At the beginning of her work shift, Ilary logs in the CPMS app and notices that the CPMS automatic algorithm decided to suspend the use of batteries in a certain charging station and, on the contrary, use energy from external suppliers only.

[SC.11] CPO employee handles manually a change in the mix of energy sources

Laurence is employed in a CPO company and is responsible for business relationships with DSOs which is partnered of. He is in charge of managing the energy furniture from third parties based on their current prices and the mix of energy sources currently used in charging stations.

Thanks to the CPMS app, he can monitor the current consumption of energy in the charging station he is analysing and he notices that the quantity of energy stored in the station batteries is under the recommended thresholds to meet the needs of the next few days.

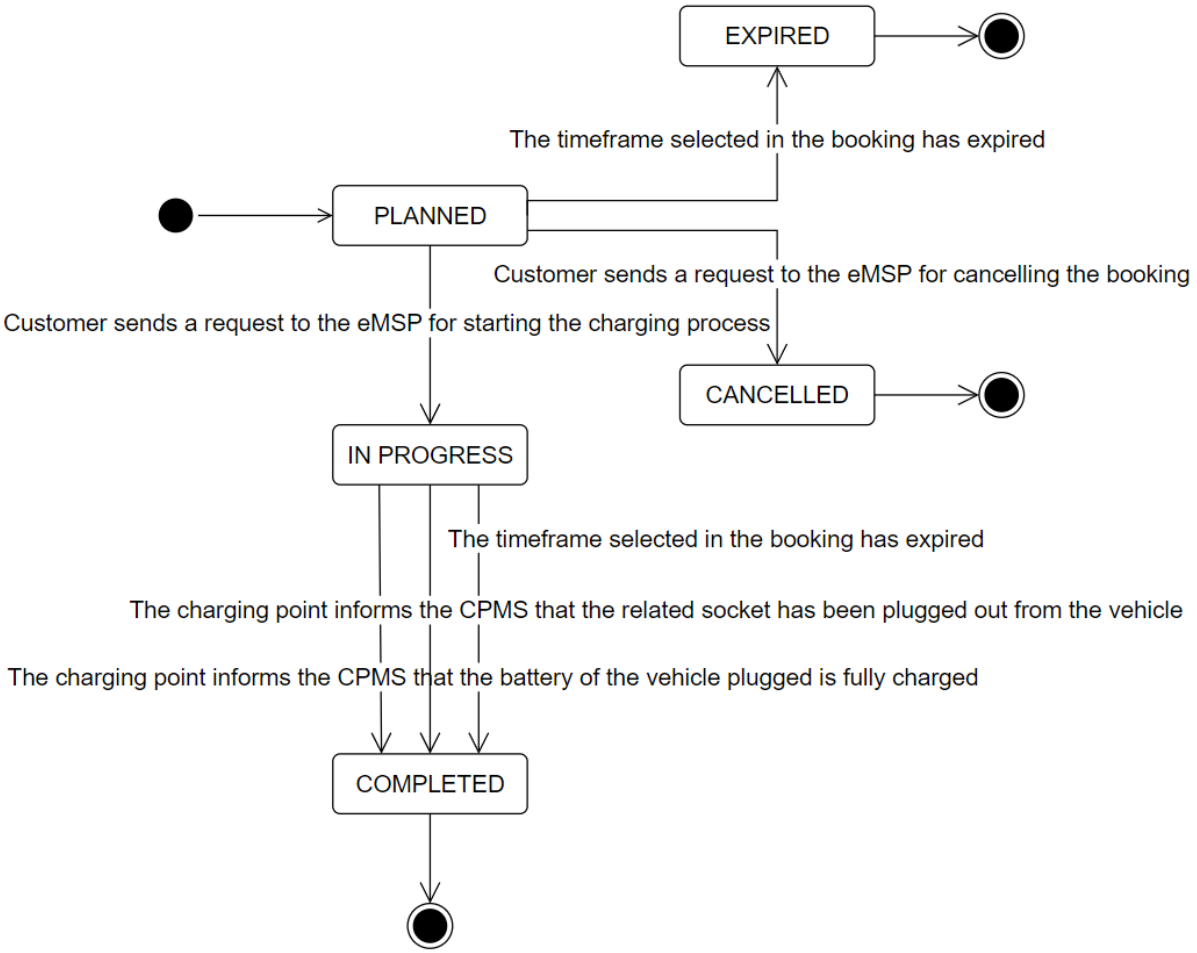
For this reason, Laurence decides to switch off the automatic handling of the mix of energy sources for that station, and on the contrary set it manually. He picks out the two DSOs which have the most competitive prices among all and includes them through the CPMS in the used mix of energy sources. Moreover, he decides to recharge the station batteries through the CPMS app until their maximum capacity.

2.1.3 State diagrams

The following state diagrams track the different states of the most relevant objects in the system: the transitions from one state to another are represented by shared phenomena described in section 1.

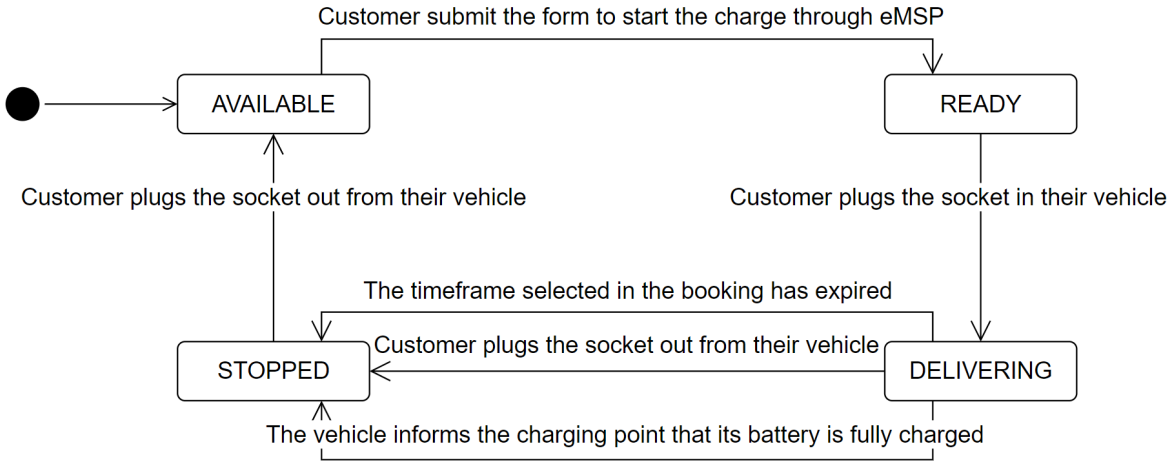
[STD.1] Booking of a charge

This diagram shows the states of a booking made by a customer.



[STD.2] Charging point's socket

This diagram shows the states associated with the tuple made of a charging point and one of its sockets.



2.2 Product functions

The main product functions are described in this section and will be detailed in the following ones.

Book a charge

The system allows customers to book a charge through the eMSP app. In particular, two types of booking are available:

- Booking in advance: a customer reserves themselves a spot in a certain charging station to recharge their vehicle in a selected future timeframe;
- Booking on the fly: a customer reserves themselves a socket in a certain charging point where to recharge their vehicle at the same moment they reach the charging station.

The two types of bookings are assigned only to the charging points who have the matching mode: by doing so, bookings in advance can be scheduled by CPMS on the specific subset of charging points with mode “in advance”, while the rest of them (mode “on the fly”) can be occupied at any time by customers without interfering with the bookings already scheduled.

The customer needs to submit a form in which they are asked to specify the desired charging station, timeframe and type of socket (slow/fast/rapid), in the case of booking in advance, or the charging point they stop by, in the case of booking on the fly.

In case of success in the operation of booking, the eMSP app finally shows to the customer the code associated with their booking.

Whenever a customer wants, they can cancel bookings made in advance through the app.

Start and manage the charging process

The system allows customers to start a charge through the eMSP app.

In case the customer had previously booked a charge in advance, they are asked to select the specific booking they want to activate and then they can start the charge. If the customer doesn’t start the charge in the timeframe specified in the booking, the booking expires and it is not possible for the customer to charge their vehicle unless they make another booking.

In case the customer submits a booking on the fly, the charge is automatically considered as started, so they can plug in the vehicle as soon as they receive the successful message by the eMSP app.

The charge has to be considered finished just after one of the following scenarios occurs:

- The battery of the electric vehicle reaches full capacity;
- The customer interrupts the charge by plugging the socket out from their vehicle;
- The timeframe selected in case of booking in advance has expired.

The system will notify the customer when the charging process is finished and will then send a request to the payment gateway to bill the customer for the charge.

Manage external and internal status of charging stations

CPMS allows CPO employees to monitor both external and internal status of charging stations, while eMSP, thanks to its connection to CPMSs, allows customers to know about

charging stations (location, current prices, offers, charging points' mode and type of sockets currently available).

Moreover, CPMS offers the two following functionalities:

- Setting prices and offers for the managed charging stations, which will also be shown by eMSP to the customers. Every 30 minutes prices are handled automatically by CPMS, which sets a price for each managed charging station based on its pricing policy associated. Indeed, CPO employees can create pricing policies and associate them to each charging station, so as to granularly manage pricings. Moreover, CPO employees can manually set offers for charging stations.
- Handling the mix of energy sources used for the managed charging stations, both automatically and manually depending on the preferences expressed for each charging station by CPO employees. This allows CPO to manage the different DSOs' offers in the most convenient way and exploit the batteries owned by charging stations to regulate energy flow during the whole day. The energy switcher collaborates in the implementation of these operations controlled by CPMS.

2.3 User characteristics

The system will interact with the following types of users:

- Customers;
- eMSP admins;
- CPO admins;
- CPO operators.

Customers need to register and log into the eMSP app to access all the offered functionalities: searching for charging stations in their area, booking and starting a charge and paying for the service.

The network of CPMSs and eMSPs communicating with each other is configured respectively by eMSP admins through eMSP and CPO admins through CPMS.

Admin credentials are assumed to be provided separately to the CPO and eMSP companies the moment the software is delivered and their privileges are unique within the company.

CPO admins also use their own credentials to register CPO operators, DSO partnerships and charging stations in the CPMS app. CPO operators need to log into the CPMS after a CPO admin has registered them into the system. They are charged with all the operational functionalities of CPMS, for example monitoring the status of charging stations, setting prices and handling the mix of sources used in charging stations.

2.4 External systems overview

Many of the functionalities offered by the system eMall relies on communication with external systems. In this section it is given an overview of their role in the interaction with eMall.

Map service

The map service provides the eMSP with the information needed to display to customers a map showing the charging stations nearby.

In order to do so, the eMSP sends to the map service the centre of the area that will be shown to the customer, represented by the same customer's GPS coordinates, and the desired radius, here assumed to be equal to 10 km.

Mail service

The mail service is used by both eMSP and CPMS to notify the users of the system through emails when needed. An example is the procedure of registration of a CPO operator managed by a CPO admin: the mail service allows the CPMS to send an email to the email address specified by the CPO Admin with a temporary password for the new account to be accessed.

Energy switcher

The energy switcher, owned by each charging station, is requested by the CPMS to physically handle the acquisition of energy by DSOs and the use of the eventual battery in the station.

The operations it can handle are listed below:

- Disconnect the battery from the grid;
- Use the battery as source of energy;
- Use the battery to store energy;
- Switch on the energy acquisition from a specific DSO;
- Switch off the energy acquisition from a specific DSO.

The energy switcher can be activated by both human and automatic operations managed by CPMS, which sends to it the desired mix of energy sources to apply to the charging station.

Charging point

Each CPMS communicates with a set of charging points managed by a specific charging station registered in the system.

The communication between CPMS and each charging point concerns the following objectives:

- Make one of the charging point’s socket ready to deliver energy;
- Start the charge of a vehicle;
- Stop the charge of a vehicle;
- Make one of the charging point’s socket available for future charges;
- Notify when one of the charging point’s socket is plugged or unplugged from a vehicle;
- Notify when a plugged vehicle has its battery full;
- Fetch the current status of the charging point.

Payment gateway

The payment gateway is used by the eMSP company to bill the customers for their charges and by the CPMS companies to bill eMSP for the corresponding charges.

Customers’ payment credentials are asked at the moment of the customer registration in the eMSP app, which sends a request to the payment gateway to verify the given credentials by debiting him and crediting back 1 euro.

Electric vehicle

The electric vehicle is able to inform the charging point to which it is plugged that its battery is fully charged. This information is then transmitted by the charging point to the CPMS to handle the vehicle's charge process.

DSO

DSOs are the external companies with which CPMSs communicate to manage energy acquisition. Each DSO exposes its price so that each CPMS can evaluate whether to switch on or off the acquisition of energy.

2.5 Domain Assumptions

#	Domain assumption
DA.1	The map service correctly renders the map in the asked geographical area.
DA.2	Customers allow the eMSP to collect data about their GPS coordinates.
DA.3	The GPS on the customer's device returns the right position of the customer.
DA.4	The configurations made by CPO admins about charging stations correspond to reality (location, presence of batteries, number and mode of charging points, number and type of sockets…).
DA.5	The configurations made by CPO admins about DSOs correspond to real potential sources of energy.
DA.6	The configurations made by CPO admins about eMSPs' network correspond to real agreements between the parties.
DA.7	The configurations made by eMSP admins about CPMSs' network correspond to real agreements between the parties.
DA.8	Every charging station has a natural identifier.
DA.9	Customers can distinguish charging points in a charging station between "in advance" and "on the fly".
DA.10	Every socket of a charging point has a natural identifier, unique in the charging station.
DA.11	Every DSO has a natural identifier.
DA.12	Charging point's features (energy delivering, sensors which measure the presence/absence of a plugged vehicle) work as expected.
DA.13	Charging point communicates to the CPMS its real status (type and number of charging sockets available, amount of power absorbed by each connected vehicle…).
DA.14	Charging points implement a security measure to stop delivering energy if the battery of the plugged vehicle is full or if its sockets are plugged out from any vehicle.

DA.15	The vehicle communicates to the plugged charging point when its battery is full.
DA.16	The energy switcher handles the use of stations' batteries and the acquisition of energy from DSOs as expected by the commands of CPMS.
DA.17	The emails sent by the mail service are always supposed to be received.
DA.18	DSOs need to always expose their prices.
DA.19	Permanent communication or hardware failure with any external system will be eventually solved (i.e. it is not possible for any external system to remain unavailable forever).

3 SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

Thinking about user interfaces, we can imagine that customers and eMSP/CPO employees have different needs in terms of usability of the system.

CPO employees are expected to be highly specialised and experienced figures, surely with a technical or managerial background, so the CPMS app interfaces will be designed to be suitable for technical operators. The same considerations can hold for eMSP admins, which will access an administrator interface for the configuration of the CPMSs' network.

Due to the variety of backgrounds of customers, it is preferable to design very user-friendly interfaces for the eMSP app, and take special care to the user experience, thought to be the most intuitive, immediate and captivating as possible.

3.1.2 Hardware Interfaces and limitations

We can identify two types of hardware requested:

- device or cluster of devices on which the system shall run;
- devices through which the users (customers, eMSP admins, CPO employees) can access the system.

For what concerns the users' devices, for accessing the web application it is necessary a device (mobile phone, personal computer or tablet) connected to the Internet and equipped with a modern web browser capable of rendering. For accessing the mobile app, it is instead required to have a smartphone running on iOS or Android operating system.

Devices used by customers to access eMSP also need to be able to retrieve the customer's position through GPS.

In order to allow as many users as possible to access the system, the hardware limitations proposed below shall be the most inclusive as possible.

iOS mobile app hardware requirements (compatible with e.g. iPhone 7):

- Display size: 4.7''
- Display resolution: 750x1334
- Chipset architecture: ARM64
- Chip: A10 Fusion chip
- RAM: 2GB

Android mobile app hardware requirements:

- Display size: 5"
- Display resolution: 1920x1080
- Chipset architecture: 64-bit
- Chipset: Snapdragon 835 or equivalent
- RAM: 4GB

Web application hardware requirements:

- Display size: 13"

- Display resolution: 1920x1080
- Chipset architecture: 64-bit
- Chipset: Intel i3 Skylake 3 GHz or equivalent
- RAM: 4GB

3.1.3 Software Interfaces and limitations

The system shall run over an operating system.

The web application is thought to be supported on the main modern browsers, with references to the following minimum versions:

- Google Chrome 90;
- Chrome for Android 106;
- Edge 85;
- Safari 15;
- Safari on iOS 15.

The mobile application shall be available on iOS and Android operating systems, in particular on the minimum versions iOS 12.4 and Android 9 Pie (API version 28).

3.1.4 Communication Interfaces

Both the communication intra-cluster and with the outside it is based on Internet protocol suite TCP/IP. The system shall be HTTPS compliant.

3.2 Functional Requirements

3.2.1 Requirements

#	Functional requirement
R.1	eMSP allows customers to register
R.2	eMSP allows customers to login
R.3	CPMS allows CPO admins to login
R.4	CPMS allows CPO admins to register CPO operators
R.5	CPMS allows CPO operators to login
R.6	CPMS allows CPO admins to configure the eMSPs' network
R.7	eMSP allows eMSP admins to configure the CPMSs' network
R.8	eMSP shall fetch information about charging stations (location, currently available sockets' types, current price, current offers) from CPMSs
R.9	CPMS allows CPO admins to submit the configuration of a charging station (location, number and mode of charging points, number and type of sockets, potential sources of energy)

R.10	CPMS allows CPO employees to obtain information about the configured charging stations (location, number and mode of charging points, number and type of sockets, potential sources of energy)
R.11	CPMS shall fetch current status of the charging points (for each of its socket: amount of power absorbed by each connected vehicle, if it is ready to deliver energy, if it has started delivering energy, if it has stopped delivering energy, if it is available for other charging processes)
R.12	CPMS shall compute the external status of a charging station based on the information retrieved by charging points
R.13	CPMS shall compute the internal status of a charging station based on the information retrieved by charging points
R.14	eMSP allows customers to obtain information about charging stations in their desired geographical area (location, currently available sockets' types, current price, current offers)
R.15	eMSP shall render a map of a geographical area through the map service given its centre's coordinates and radius
R.16	CPMS allows CPO employees to obtain information about the external status of a charging station
R.17	CPMS allows CPO employees to obtain information about the internal status of a charging station
R.18	eMSP allows customers to submit booking requests in advance for a charge in a specific charging station for a certain timeframe
R.19	eMSP allows customers to submit booking requests on the fly for a charge in a specific socket in a charging station
R.20	eMSP allows customers to cancel bookings made for a charge
R.21	eMSP shall send customers' booking requests to the corresponding CPMS
R.22	CPMS shall confirm a booking request received by eMSP if a socket (of the same type and mode as specified) is available at the desired charging station for the timeframe specified in the booking, otherwise it shall deny it
R.23	eMSP shall confirm/deny a booking request to the associated customer depending on the CPMS's response
R.24	CPMS shall keep track of the status of bookings
R.25	CPMS shall notify eMSP for changes in the status of bookings
R.26	eMSP should fetch the status of bookings when notified by CPMS
R.27	eMSP allows customers to obtain information about their bookings (charging station, type and identifier of socket, status, timeframe in case of booking in advance)
R.28	eMSP shall notify customers for the expiration of their bookings

R.29	eMSP allows customers to start a booked charge
R.30	eMSP shall send customers' requests to start a charge to CPMS
R.31	CPMS shall send a request to the charging point to get its own socket ready for delivering energy
R.32	CPMS shall send a request to the charging point to deliver energy to the plugged vehicle from its own socket
R.33	CPMS shall send a request to the charging point to stop delivering energy to the plugged vehicle from its own socket
R.34	CPMS shall send a request to the charging point to make its own socket available for other charging processes
R.35	CPMS shall notify eMSP for the completion of a charging process
R.36	eMSP shall notify customers for the completion of their charging process
R.37	CPMS allows CPO admins to manage a DSO partnership (registration, deletion)
R.38	CPMS shall fetch current prices by DSOs
R.39	CPMS allows CPO employees to manage pricing policies (creation, edit, deletion, flagging it as default)
R.40	CPMS allows CPO employees to associate a pricing policy to a charging station
R.41	CPMS shall automatically compute prices for each charging station based on its associated pricing policy
R.42	CPMS allows CPO employees to set manually offers for each charging station
R.43	CPMS shall automatically compute the mix of energy sources for each charging station
R.44	CPMS allows CPO employees to set manually the mix of energy sources for each charging station
R.45	CPMS shall send a request to the energy switcher of a charging station to physically apply a specific mix of energy sources.
R.46	eMSP shall bill customers for their charges through the payment gateway
R.47	CPMS shall bill eMSP for the corresponding charges through the payment gateway
R.48	CPMS shall keep track of the energy acquired from each registered DSO
R.49	CPMS shall aggregate the data on acquired energy for each registered DSO
R.50	CPMS shall estimate the cost due to DSOs for the acquired energy

3.2.2 Mapping Goals - Requirements - Domain Assumptions

Goal	Requirements	Domain Assumptions
G.1	R.1, R.2, R.7, R.8, R.14, R.15	DA.1, DA.2, DA.3, DA.4, DA.7, DA.8, DA.19
G.2	R.1, R.2, R.7, R.18, R.19, R.20, R.21, R.23, R.26, R.27, R.28	DA.4, DA.7, DA.8, DA.10, DA.19
G.3	R.1, R.2, R.7, R.29, R.30, R.36, R.46	DA.4, DA.7, DA.8, DA.9, DA.10, DA.12, DA.13, DA.14, DA.15, DA.19
G.4	R.3, R.4, R.5, R.9, R.37, R.38, R.39, R.40, R.41, R.42	DA.11, DA.17, DA.18, DA.19
G.5	R.3, R.4, R.5, R.9, R.10, R.13, R.17, R.37, R.38, R.43, R.44, R.45	DA.4, DA.5, DA.11, DA.13, DA.16, DA.17, DA.18, DA.19
G.6	R.3, R.4, R.5, R.9, R.10, R.11, R.12, R.13, R.16, R.17	DA.4, DA.8, DA.17, DA.19
G.7	R.3, R.6, R.9, R.11, R.31, R.32, R.33, R.34, R.35, R.47	DA.4, DA.6, DA.8, DA.10, DA.12, DA.13, DA.14, DA.15, DA.19
G.8	R.3, R.6, R.9, R.11, R.12, R.22, R.24, R.25	DA.4, DA.6, DA.8, DA.10, DA.19
G.9	R.3, R.37, R.38, R.48, R.49, R.50	DA.5, DA.11, DA.16, DA.18, DA.19

3.3 Use cases

The following use cases depict the most important interactions within eMall.
Along with information about the participating actors, entry and exit conditions, each use case describes the main flow of events of the interaction.

Exceptional flows are described separately, providing an indication of the causes that induce a divergence from the main flow. The listed exceptional flows are related to business-related causes. Unrecoverable technological issues such as permanent communication or hardware failure will result in the abortion of the interaction are excluded by DA.19.

Some use cases include other use cases. Exceptions of included use cases are present in the use cases that include them only if they can not be reconciled with the main event flow.

3.3.1 Sequence Diagrams

Each use case is accompanied by a sequence diagram which depicts the flow of the interaction.

To avoid verbosity, we introduced some conventions:

- With the exception of the Login and Signup use cases, all customers and CPO employees have already been authenticated;
- All asynchronous messages do not have an explicit response. We assume that in case of asynchronous messages the only immediate response is a confirmation of the received message. The answer will be provided by sending another asynchronous message upon the completion of the execution.

3.3.2 Use Case Diagrams

The main relationships between actors and use cases are depicted by the use case diagrams (UC).

Actors on the left are primary actors, the ones actively interested in the UC. Actors on the right are secondary actors, the ones used by the system in order to accomplish the UC.

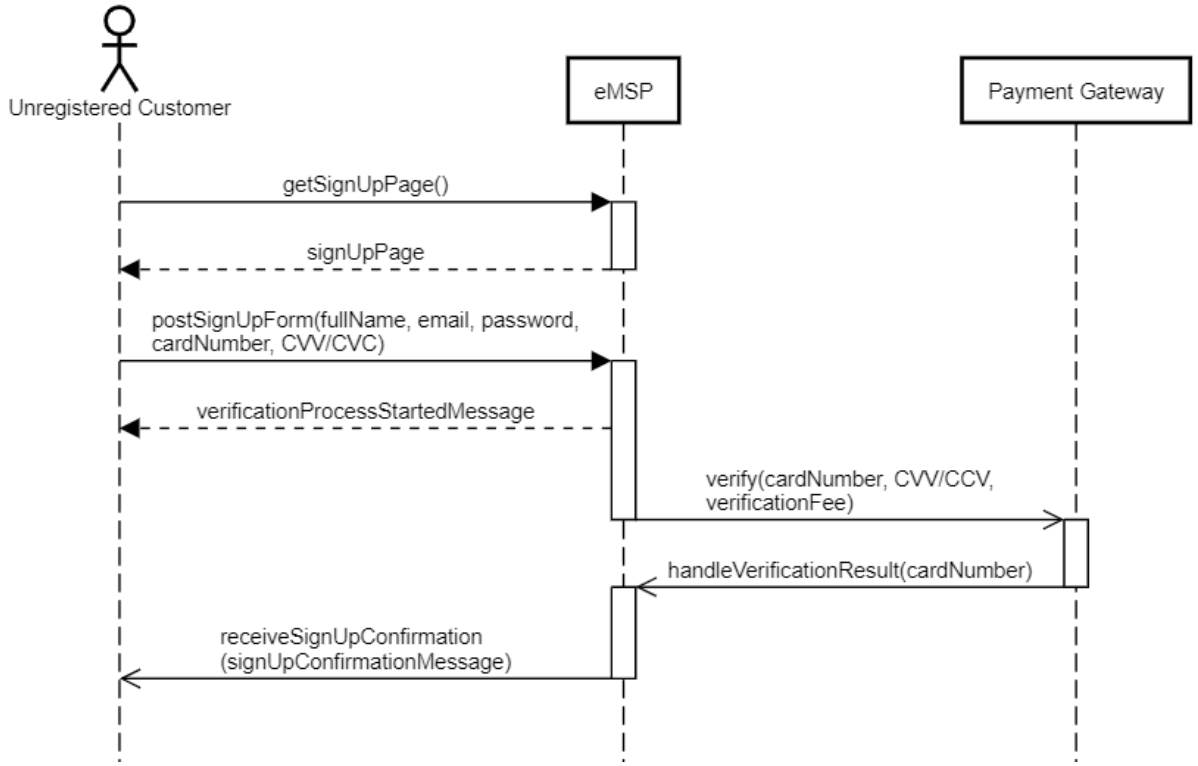
Some UCs include other UCs. In our case, when a UC includes another UC, all actors of the first UC become also the actors of the included UC.

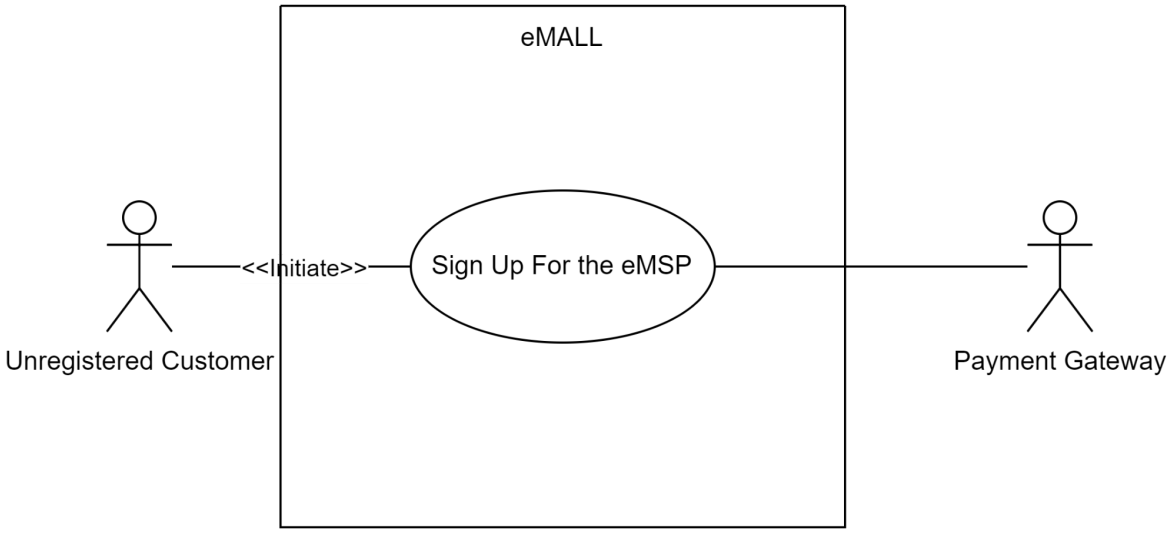
Note: the name of each of the following paragraphs is followed by the references to the use case, the related sequence diagram and eventually the use case diagram. In case of inclusion of use cases, the associated use case diagram (including a set of use cases) is reported where the included use case is explained.

User signs up for the eMSP [UC.1, SQD.1, UCD.1]

#	UC.1
Name	Sign up for the eMSP
Actors	Unregistered Customer, Payment Gateway
Entry Conditions	- The Unregistered Customer is not logged into the eMSP - The Unregistered Customer has a payment card - The Unregistered Customer knows their personal information and the details of their payment card
Event Flow	1. Unregistered Customer requests to the eMSP the sign-up page 2. eMSP shows the Unregistered Customer a form 3. The Unregistered Customer fills the form with their personal information (full name, email, password) and payment information (card number and CVV/CVC) 4. The Unregistered Customer submits the form 5. The eMSP informs the Unregistered Customer that the previously inserted information have been correctly acquired and that the verification process has begun 6. eMSP verifies the payment card through the Payment Gateway by charging and refunding a small fee of 1€ 7. eMSP informs the Unregistered Customer that the sign-up procedure has terminated successfully
Exit Conditions	- A new Customer with their personal information and their payment information is inserted in the eMSP

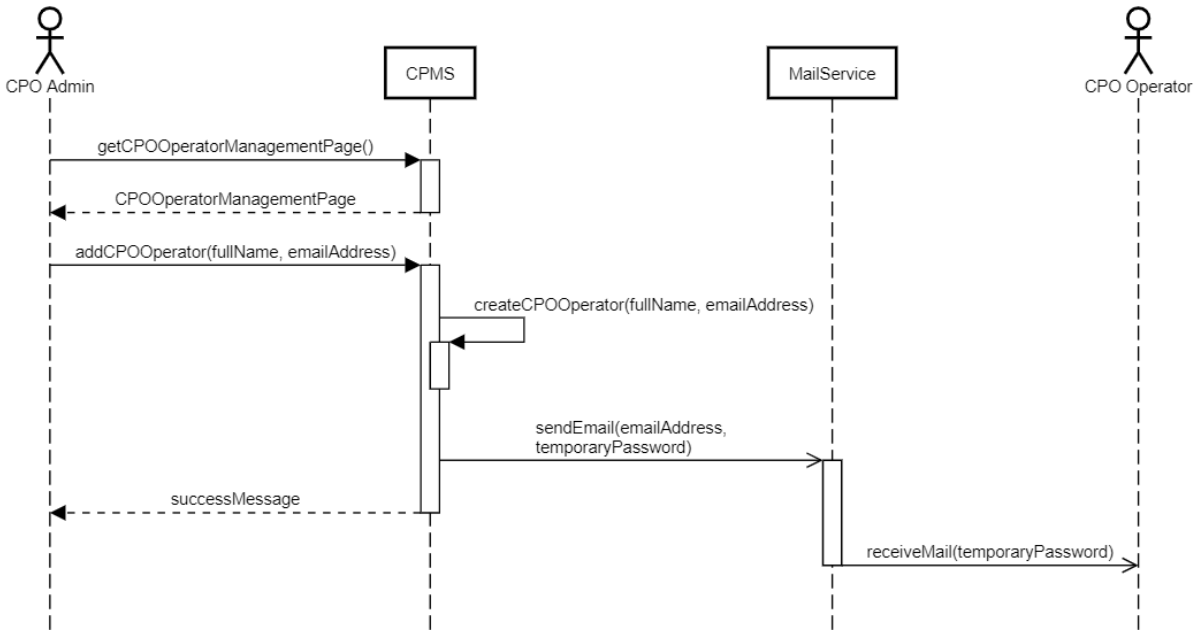
Exceptions	- One or more fields inserted in the form are not in a valid format. → A message indicating the format problem is displayed by the eMSP and the eMSP doesn't allow the Unregistered User to submit the configuration until the condition is met - A customer with the same email address is already registered in the eMSP. → The registration fails and a message indicating the problem is displayed by the eMSP - The Payment Gateway cannot perform the money transactions to verify the payment method. → The registration fails and a message indicating the problem is displayed by the eMSP

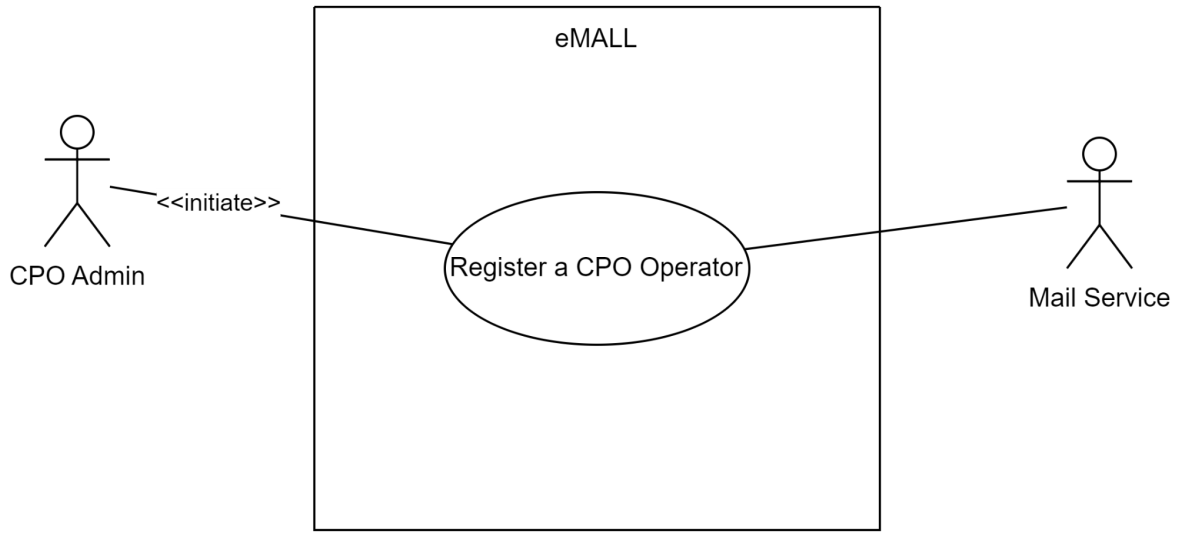




CPO Admin registers a CPO Operator [UC.2, SQD.2, UCD.2]

#	UC.2
Name	Register a CPO Operator
Actors	CPO Admin, CPO Operator, Mail Service
Entry Conditions	- The CPO Admin is logged into the CPMS - The CPO Admin owns the information needed to register the new CPO Operator
Event Flow	1. The CPO Admin requests the page to manage CPO Operator accounts to the CPMS 2. The CPMS sends the page to manage CPO Operator accounts to the CPO Admin 3. The page includes a form to register a new CPO Operator, which the CPO Admin fills with the full name and the email address of the new CPO Operator 4. The CPO Admin submits the form to the CPMS 5. The CPMS creates a new account for the CPO Operator 6. The CPMS, using the Mail Service, sends an email to the email address specified by the CPO Admin with a temporary password for the new account 7. The CPMS sends to the CPO Admin a success message
Exit Conditions	- There is a new CPO Operator account with the correct information registered in the CPMS
Exceptions	- The CPO Admin tries to add a new CPO Operator, but the specified email address is already associated with a CPO Operator account. → The new CPO Operator account is not inserted in the system and an explanatory message is sent to the CPO Admin

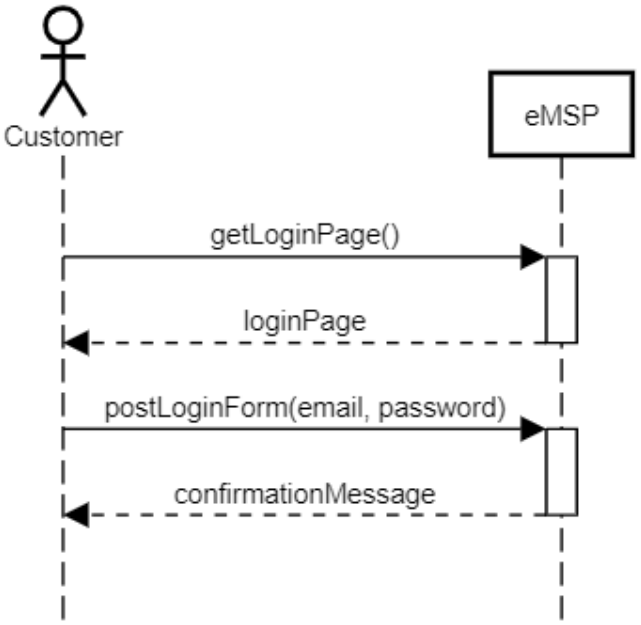


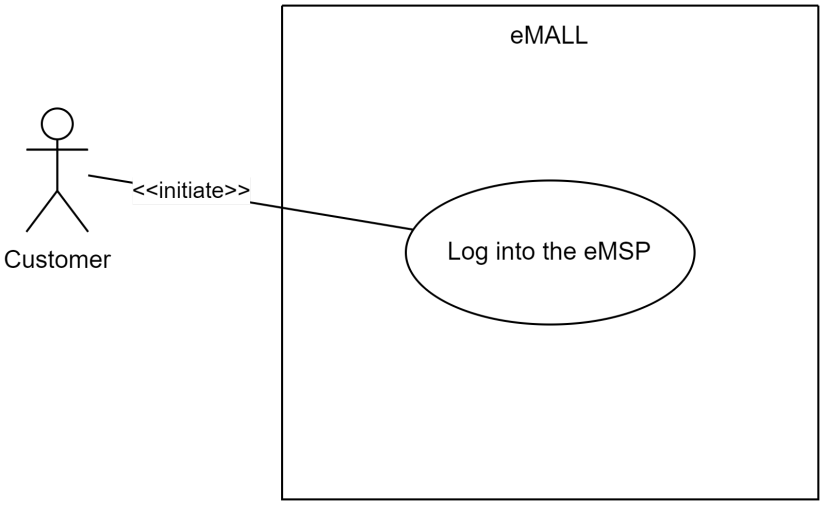


Customer logs into the eMSP [UC.3, SQD.3, UCD.3]

#	UC.3
Name	Log into the eMSP
Actors	Customer
Entry Conditions	- The Customer is not logged into the eMSP - The Customer remembers their credentials
Event Flow	1. The Customer requests to the eMSP the login page 2. eMSP shows the Customer a form

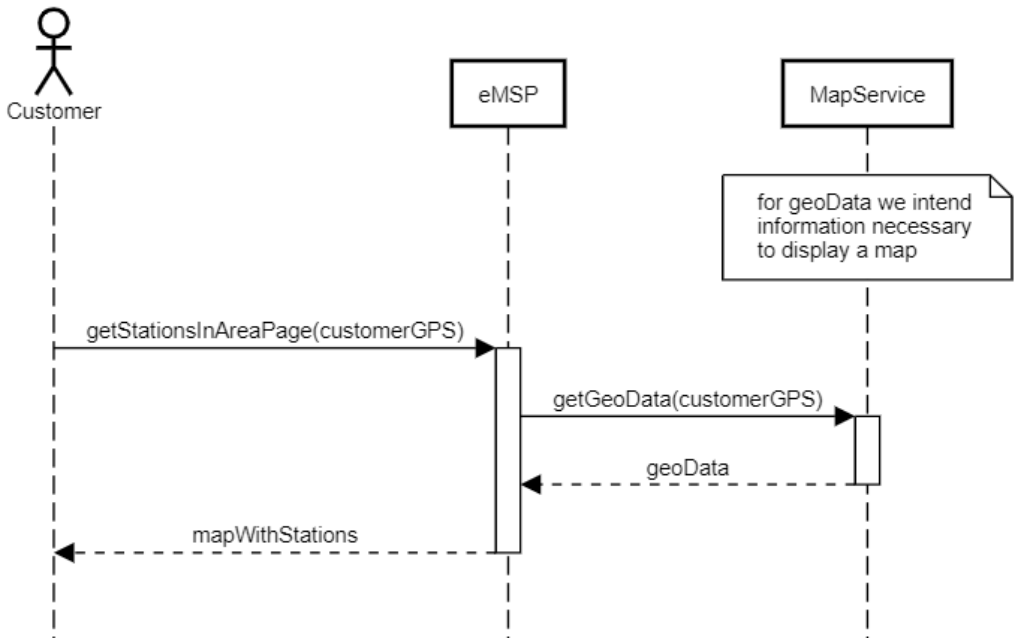
	3. The Customer fills the form with their email and password 4. The Customer submits the form 5. eMSP sends to the Customer a confirmation message
Exit Conditions	- The Customer is logged into the eMSP
Exceptions	- Email and/or password fields are not in a valid format. → The login fails, a message indicating the format problem is displayed by the eMSP and the eMSP doesn't allow the Customer to submit the configuration until the condition is met

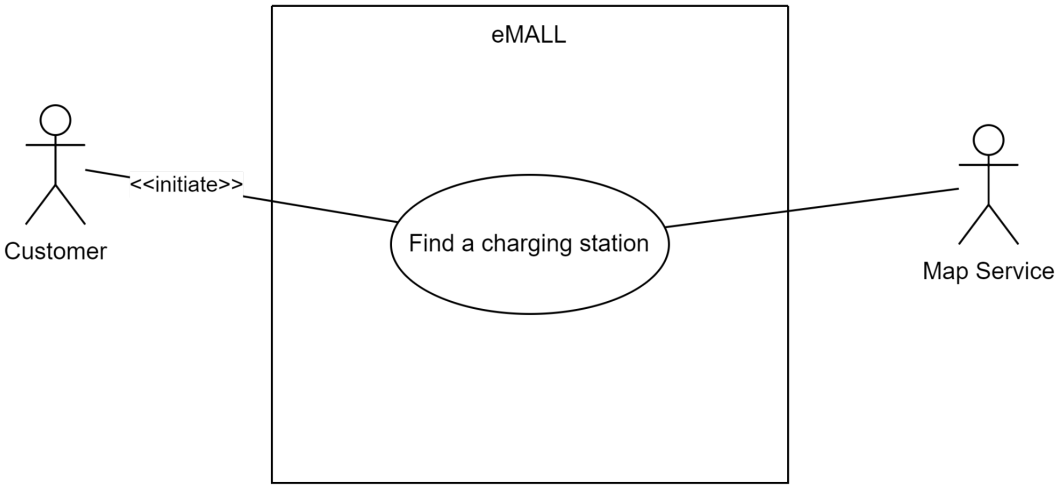




Customer finds a charging station [UC.4, SQD.4, UCD.4]

#	UC.4
Name	Find a charging station
Actors	Customer, Map Service
Entry Conditions	- The Customer is logged into the eMSP - The Customer allows the eMSP to collect data about their GPS coordinates
Event Flow	1. The Customer requests to the eMSP the page that shows the charging stations in the area, along with their GPS coordinates 2. eMSP selects charging stations within 10 Km from the Customer GPS coordinates 3. eMSP fetches from the Map Service the information needed to display a map showing the charging stations in the area
Exit Conditions	- The Customer can see charging stations in the area on a map
Exceptions	/

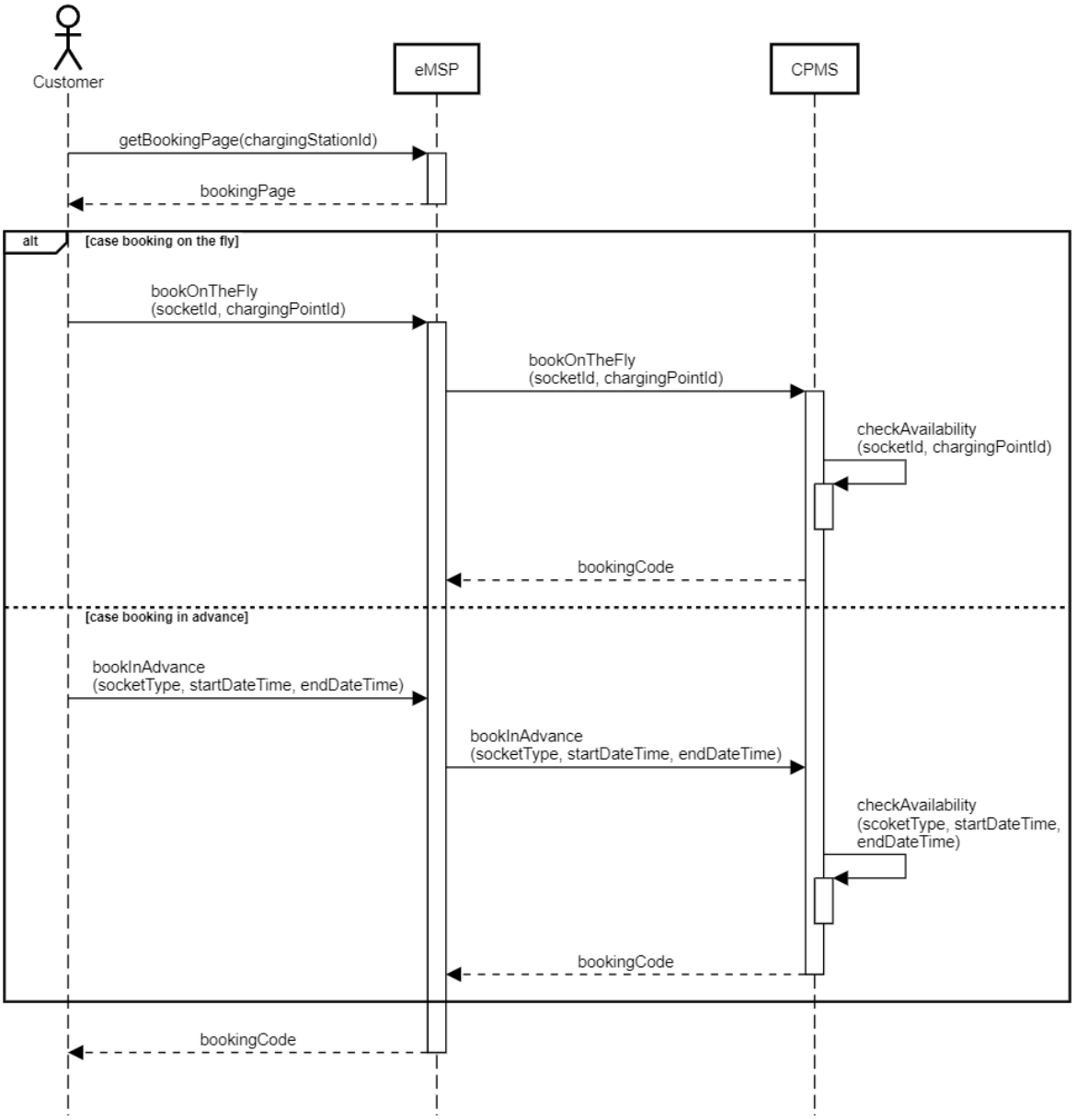


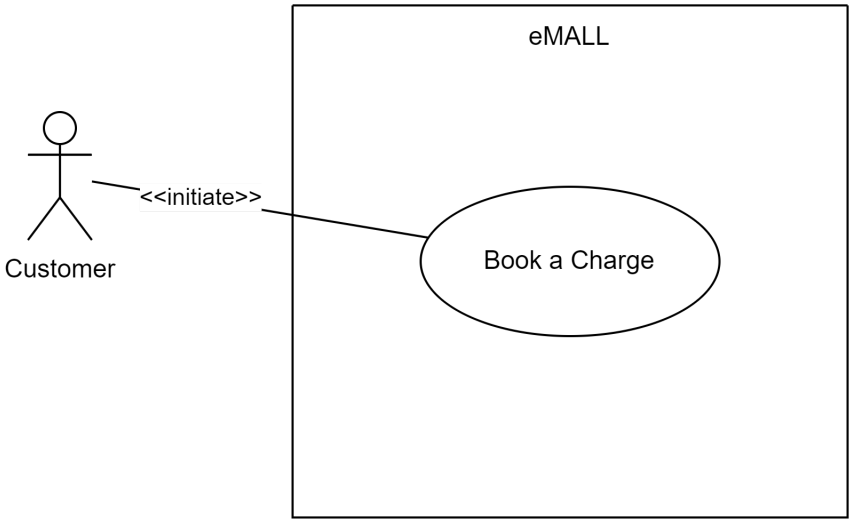


Customer books a charge [UC.5, SQD.5, UCD.5]

#	UC.5
Name	Book a charge
Actors	Customer
Entry Conditions	- The Customer is logged into the eMSP - The Customer knows the charging station they want to book for - The Customer does not have any unpaid completed booking
Event Flow	1. The Customer requests to the eMSP the page for booking a charge specifying the desired charging station 2. eMSP shows the Customer a form 3. The Customer selects the desired type of booking between the following options: “on the fly” or “in advance” 4. If the selected type is “on the fly” then the Customer selects the identifier of the socket the Customer wants to charge their vehicle with 5. If the selected type is “in advance” then the Customer selects the desired timeframe for the booking (start date-time, end date-time) and the desired type of socket between the following speed options: “slow”, “fast”, “rapid” 6. The Customer submits the form to the eMSP 7. The eMSP sends the request for the booking to the CPMS associated to the desired charging station 8. The CPMS checks for availability of an available socket according to the booking request’s parameters 9. The CPMS sends to the eMSP the booking code 10. The eMSP sends to the Customer the booking code along with a success message
Exit Conditions	- A new booking is stored into the CPMS
Exceptions	- The selected booking type is “on the fly” and there are no sockets of the selected type and mode available in the selected charging

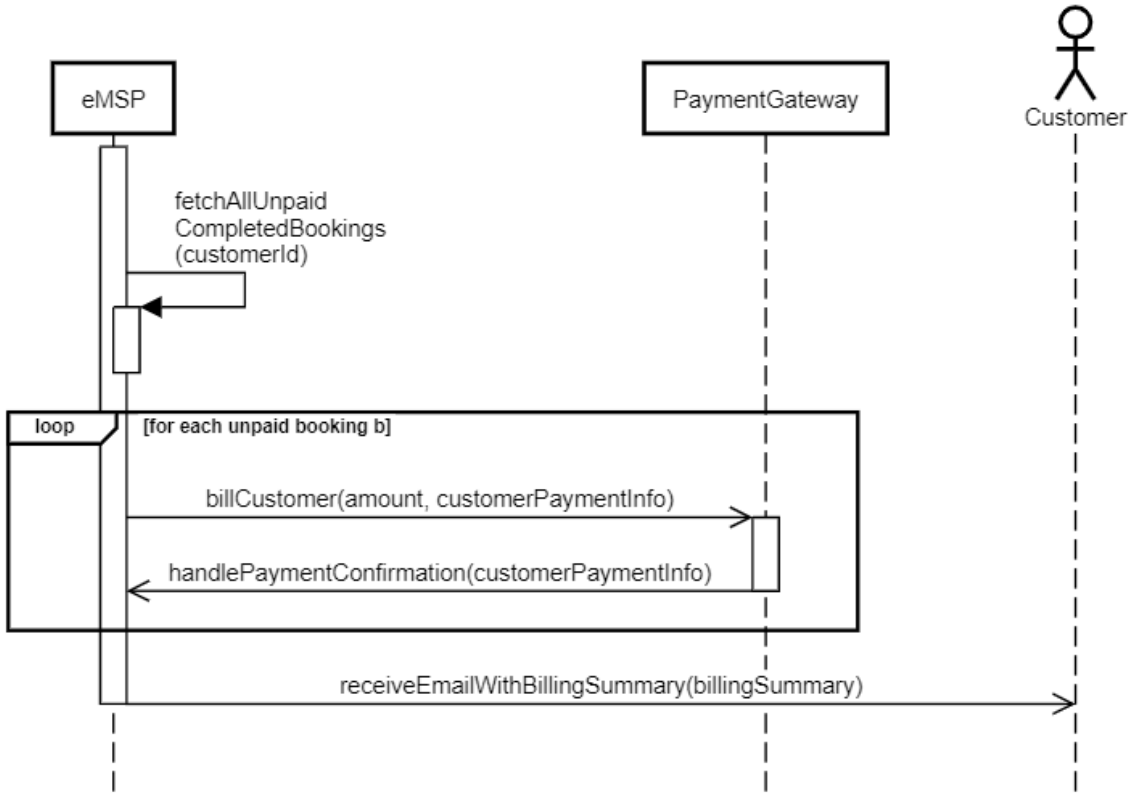
	point. → The booking fails and a message is shown by the eMSP to the Customer - The selected booking type is “in advance” and there are no available sockets of the specified type and mode for the specified timeframe. → The booking fails and a message is shown by the eMSP to the Customer

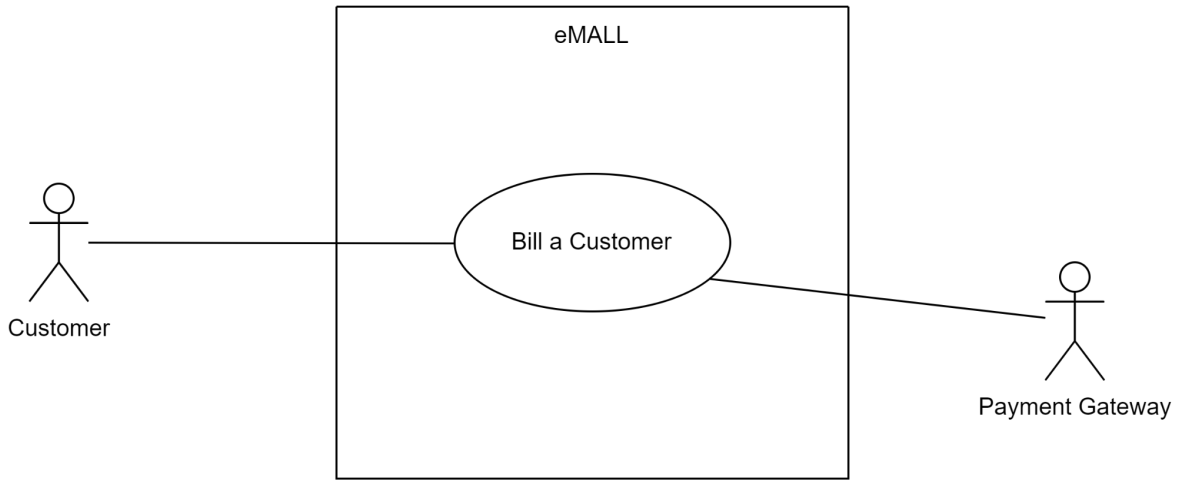




eMSP bills a customer [UC.6, SQD.6, UCD.6]

#	UC.6
Name	Bill a Customer
Actors	Payment Gateway, Customer
Entry Conditions	- The Customer has at least one completed booking not paid
Event Flow	1. The eMSP fetches all the non paid completed booking associated to the Customer 2. For each fetched booking the eMSP sends a request to the Payment Gateway to bill the Customer using the payment information provided by the Customer upon signing up 3. The eMSP sends an email to the Customer notifying them for the billing
Exit Conditions	- The Customer does not have any non paid completed booking - The correct amount of money is transferred from the credit card provided from the Customer to the eMSP bank account
Exceptions	- The Payment Gateway is not able to bill the Customer due to incorrect payment credentials. → The Customer has the same non paid completed booking they had before

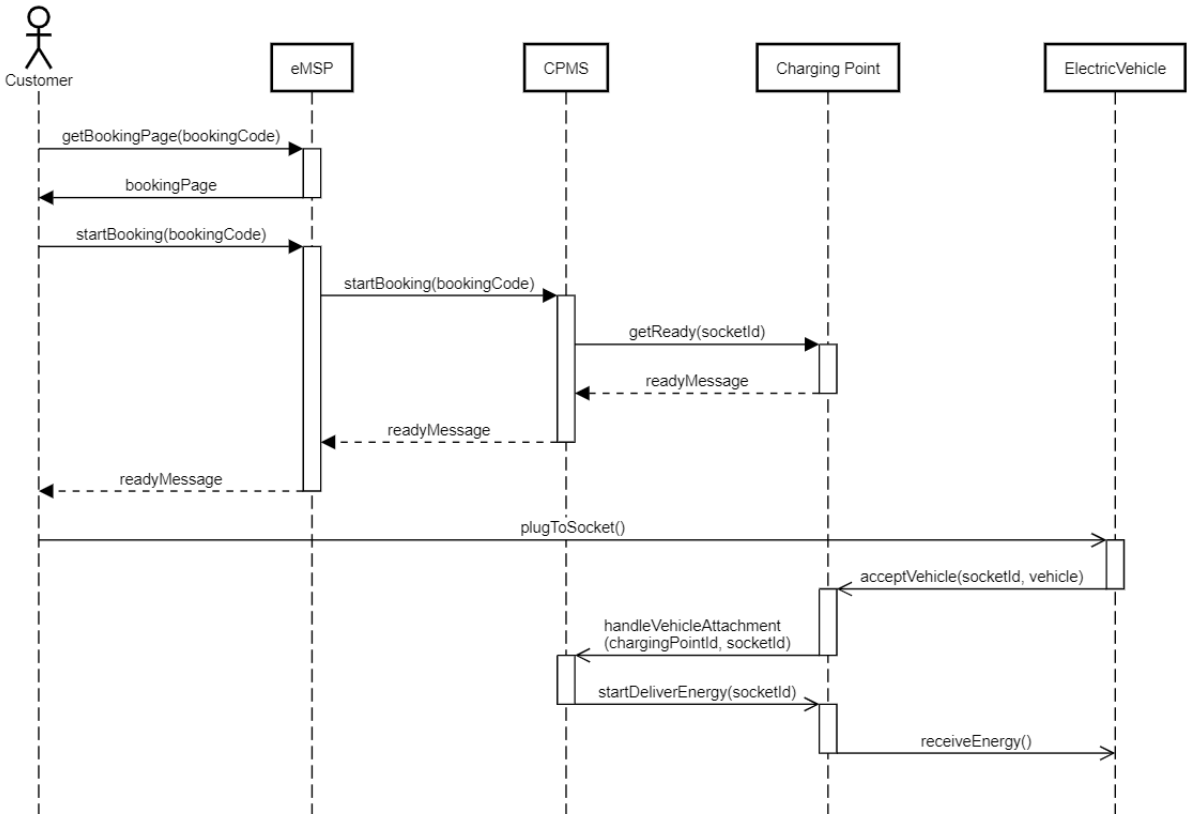


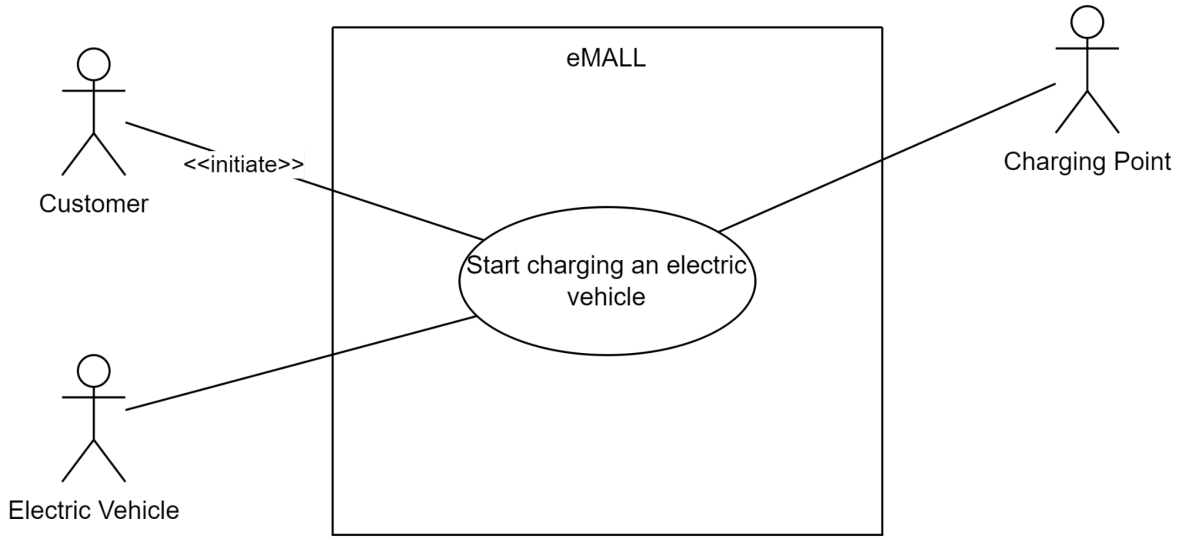


Customer starts charging their vehicle [UC.7, SQD.7, UCD.7]

#	UC.7
Name	Start charging an electric vehicle
Actors	Customer, Electric Vehicle, Charging Point

Entry Conditions	- The Customer already has a planned booking on the fly or in advance, and the current instant is within the timeframe specified by their booking - The Customer and their Electric Vehicle are in the surrounding of the charging station specified in the booking they are interested in - The Customer is logged into the eMSP
Event Flow	1. The Customer requests to the eMSP the page of the booking they are interested in activating 2. The Customer reads on the app the identifier of the socket associated with the booking 3. The Customer stops their Electric Vehicle by the assigned Charging Point 4. The Customer presses a button on the app to inform the eMSP that they are ready to start the charge 5. The eMSP informs the CPMS that the booking needs to start 6. The CPMS informs the corresponding Charging Point that a customer wants to start a charge with the specified socket 7. The Charging Point informs the CPMS that it is ready to deliver energy 8. The CPMS informs the eMSP that Charging Point is ready to deliver energy 9. The eMSP informs the Customer that the Charging Point is ready to deliver energy 10. The Customer plugs the specified socket into their Electric Vehicle 11. The Charging Point informs the CPMS that an Electric Vehicle has been attached to one of its sockets 12. The CPMS sends to the Charging Point the command to start delivering energy 13. The Charging Point starts delivering energy through the specified socket
Exit Conditions	- The Charging Point delivers energy through the correct socket to the plugged Electric Vehicle - The Booking associated with the charge is in progress
Exceptions	- The type of the booking is “in advance” and before the Customer has informed the eMSP that they are ready to start the charge, the timeframe specified by the booking expires. → The eMSP informs the Customer that the booking is expired and the booking is in the expired state - The Charging Point is not available due to a malfunction. → The eMSP informs the Customer that there is a failure in the Charging Point. The booking remains planned, but the identifier of the Charging Point and of the socket is changed. If no other Charge Point is available for the booking, the booking gets cancelled - The Customer attaches the Electric Vehicle to one of the sockets of the Charging Point that was not indicated in any booking → The socket does not deliver energy and the charging process does not continue until the Customer plugs a socket associated with a booking into their Electric Vehicle

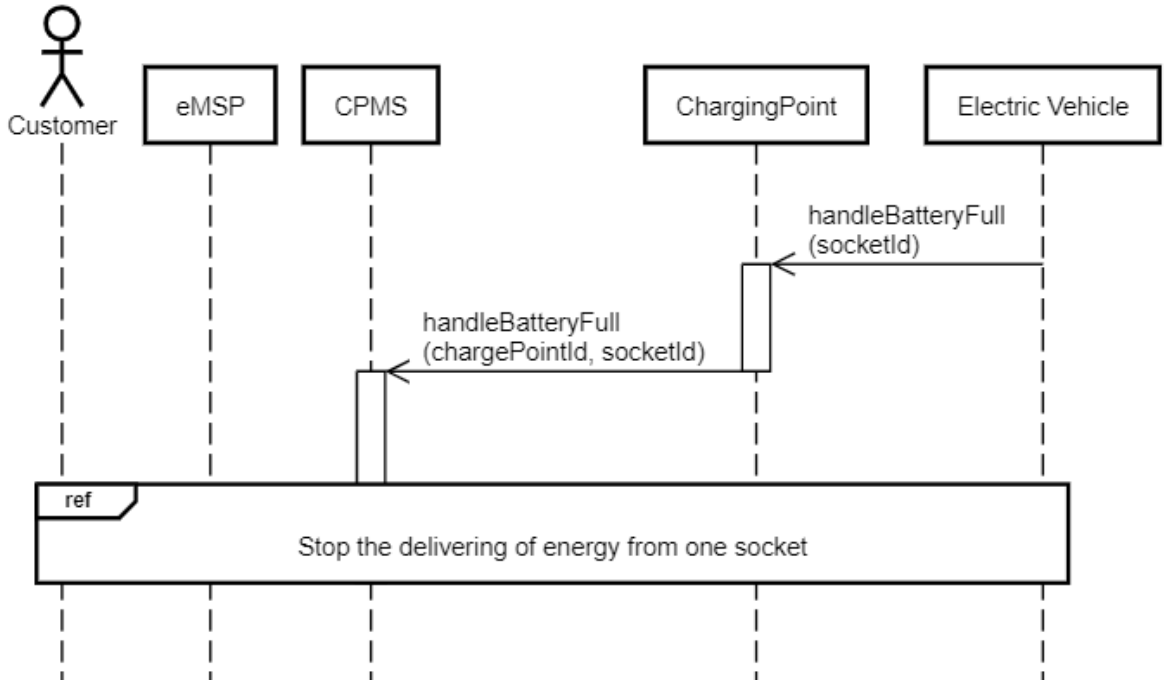




The delivery of energy to a vehicle stops due to full battery [UC.8, SQD.8]

#	UC.8
Name	Stop the delivery of energy due to full battery
Actors	Customer, Electric Vehicle, Charging Point

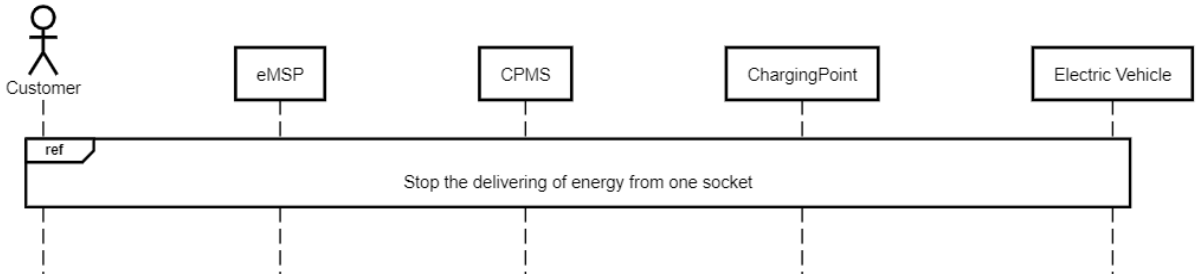
Entry Conditions	- There is a booking in progress associated to the charging - The socket of the Charging Point is delivering energy - The battery of the Electric Vehicle is full
Event Flow	1. The Electric Vehicle informs the Charging Point that its battery is full 2. The Charging Point informs the CPMS that the Electric Vehicle is fully charged 3. The CPMS stops the delivery of energy from the socket that was charging the Electric Vehicle [UC.11]
Exit Conditions	- The Charging Point does not deliver energy through the socket - The booking associated with the charge is completed - The Charging Point and the socket are available
Exceptions	- The Charging Point cannot cease to deliver energy due to a malfunction. The process is aborted and a timer of 3 seconds is set in the CPMS. Upon the expiration of the timer the process is tried again - The Charging Point can not become available due to a malfunction → The Charging Point does not deliver energy through the socket. The booking associated with the charge is completed, the Charging Point and the socket are stopped (so not available yet)



The delivery of energy to a vehicle stops due to booking timeframe ending [UC.9, SQD.9]

#	UC.9
Name	Stop the delivery of energy due to booking timeframe ending

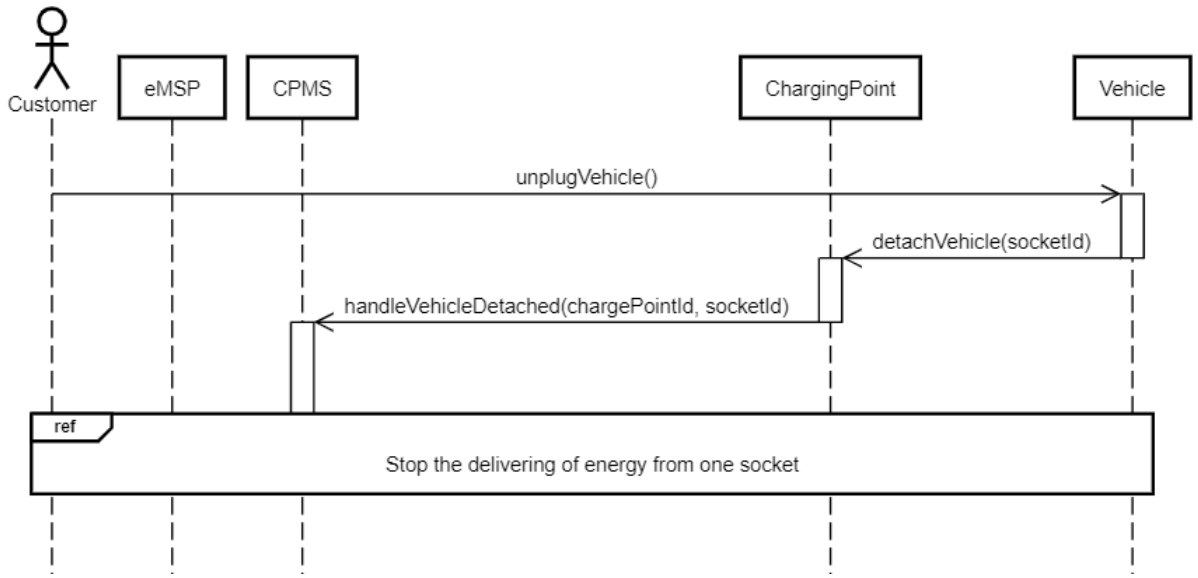
Actors	Customer, Electric Vehicle, Charging Point
Entry Conditions	- There is a booking in progress associated to the charging - The socket of the Charging Point is delivering energy - The timeframe of the booking has ended
Event Flow	1. The CPMS stops the delivery of energy from the socket that was charging the Electric Vehicle [UC.11]
Exit Conditions	- The Charging Point does not deliver energy through the socket - The booking associated with the charge is completed - The Charging Point and the socket are available
Exceptions	- The Charging Point cannot cease to deliver energy due to a malfunction. → The process is aborted and a timer of 3 seconds is set in the CPMS. Upon the expiration of the timer the process is tried again - The Charging Point can not become available due to a malfunction → The Charging Point does not deliver energy through the socket. The booking associated with the charge is completed, the Charging Point and the socket are stopped (so not available yet)



The delivery of energy to a vehicle stops due to the unplug of the socket from the vehicle [UC.10, SQD.10]

#	UC.10
Name	Stop the delivery of energy due to the unplug of the socket from an Electric Vehicle
Actors	Customer, Electric Vehicle, Charging Point
Entry Conditions	- There is a booking in progress associated to the charging - The socket of the Charging Point is delivering energy - The battery of the Electric Vehicle is full
Event Flow	1. The Customer unplugs the socket from their Electric Vehicle 2. The Charging Point informs the CPMS that an Electric Vehicle that was being charged has been unplugged from the socket 3. The CPMS stops the delivery of energy from the socket that was charging the Electric Vehicle [UC.11]

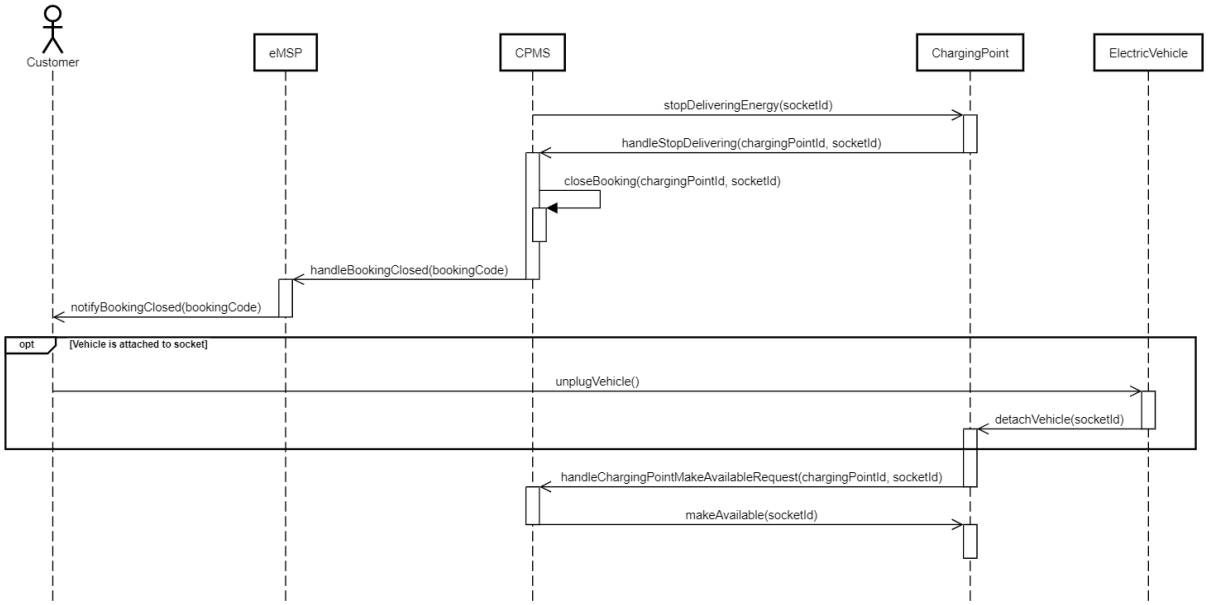
Exit Conditions	- The Charging Point does not deliver energy through the socket - The booking associated with the charge is completed - The Charging Point and the socket are available
Exceptions	- The Charging Point cannot cease to deliver energy due to a malfunction. → The process is aborted and a timer of 3 seconds is set in the CPMS. Upon the expiration of the timer the process is tried again - The Charging Point can not become available due to a malfunction. → The Charging Point does not deliver energy through the socket. The booking associated with the charge is completed, the Charging Point and the socket are stopped (so not available yet)

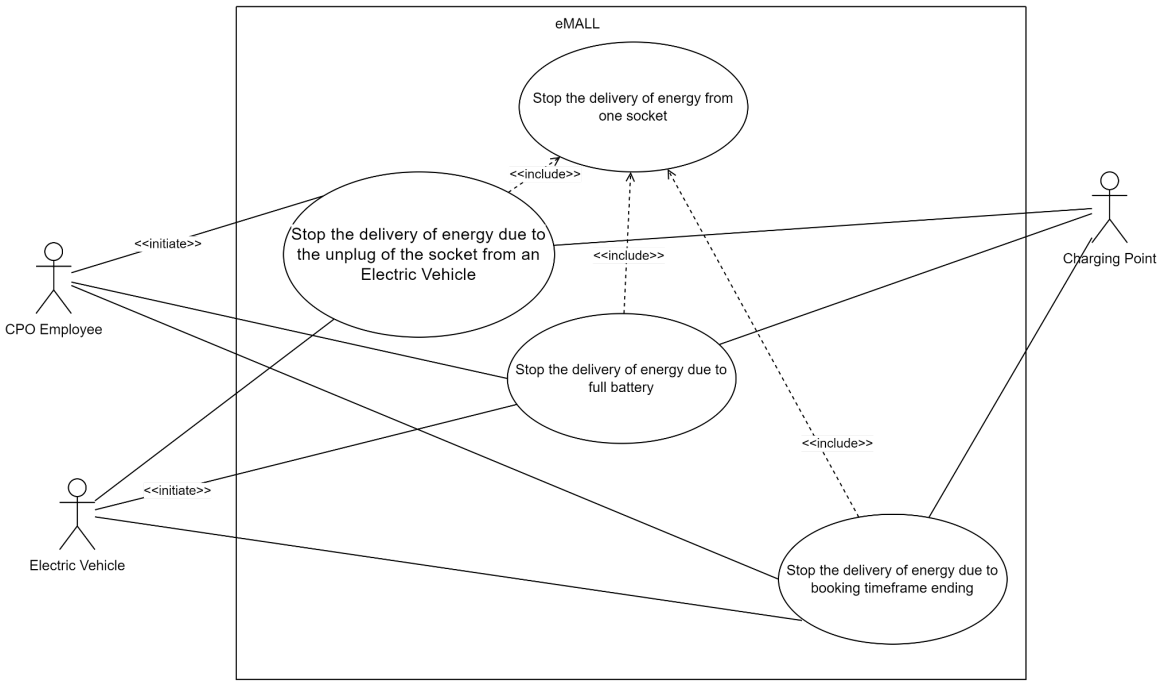


The delivery of energy to a vehicle stops [UC.11, SQD.11, UCD.8]

#	UC.11
Name	Stop the delivery of energy from one socket
Actors	Customer, Electric Vehicle, Charging Point
Entry Conditions	- There is a booking in progress associated to the charging - The socket of the Charging Point is delivering energy
Event Flow	1. The CPMS sends a message to the Charging Point to stop the delivery of energy 2. The Charging Point ceases to deliver energy 3. The Charging Point informs the CPMS that it has ceased to deliver energy 4. The CPMS closes the booking associated with the charge 5. The CPMS informs the Customer through the eMSP of the completion of the task

	6. If the socket of the Charging Point is plugged into an Electric Vehicle: a. The Customer plugs off the socket from the Electric Vehicle 7. The Charging Point informs the CPMS that the socket is not plugged in any Electric Vehicle 8. The CPMS sends a message to the Charging Point to make it available for future charges
Exit Conditions	- The Charging Point does not deliver energy through the socket - The booking associated with the charge is completed - The Charging Point and the socket are available
Exceptions	- The Charging Point cannot cease to deliver energy due to a malfunction. → The process is aborted and a timer of 3 seconds is set in the CPMS. Upon the expiration of the timer the process is tried again - The Charging Point can not become available due to a malfunction. → The Charging Point does not deliver energy through the socket. The booking associated with the charge is completed, the Charging Point and the socket are stopped (so not available yet)

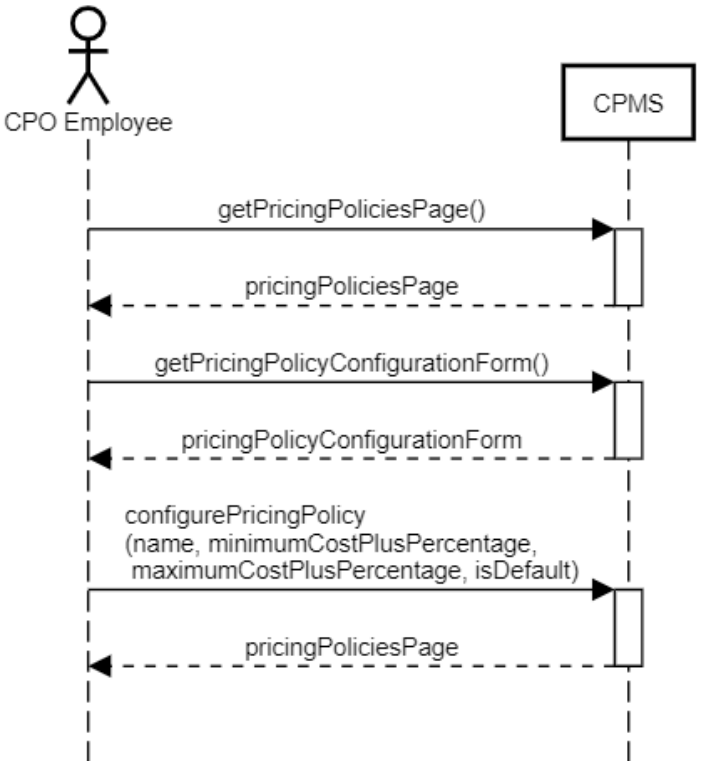


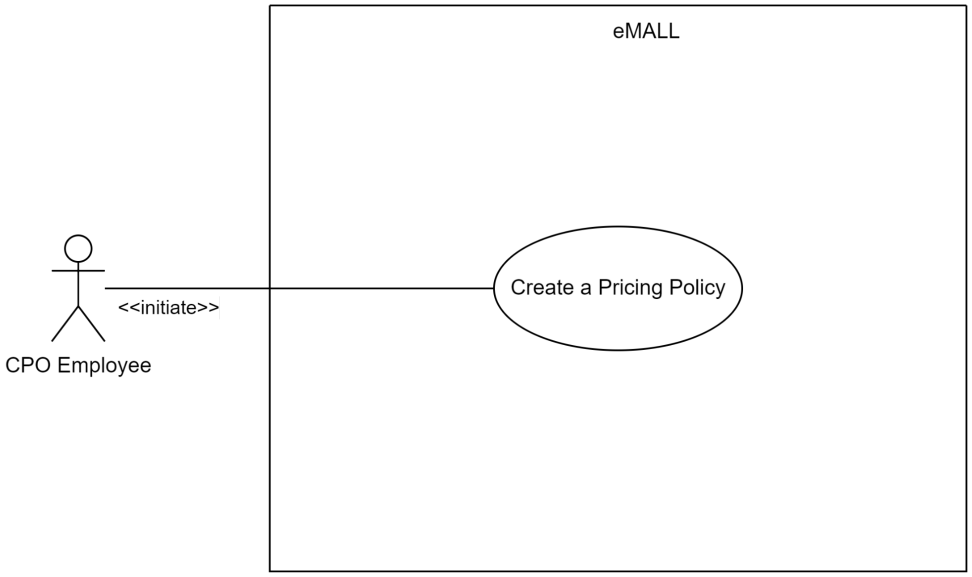


CPO employee creates a pricing policy [UC.12, SQD.12, UCD.9]

#	UC.12
Name	Create a pricing policy
Actors	CPO Employee
Entry Conditions	- The CPO Employee is logged into the CPMS
Event Flow	1. The CPO Employee requests the page to manage pricing policies to the CPMS 2. The CPMS sends a page with the list of configured pricing policies to the CPO Employee 3. The CPO Employee requests to add a new pricing policy 4. The CPMS sends a form to the CPO Employee asking for the configuration of the pricing policy 5. The CPO Employee fills the form with the desired configuration (name, minimum and maximum percentage of cost-plus pricing and if the pricing policy is the new default one) 6. The CPO Employee submits the form 7. The CPMS sends to the CPO Employee the updated list of the Pricing Policies configured in the CPMS
Exit Conditions	- The CPO Employee tries to input values smaller than 0 in the percentage fields. → The CPMS displays a message reminding the CPO Employee about the correct range for those fields and the CPMS doesn't allow the CPO Employee to submit the configuration until the condition is met

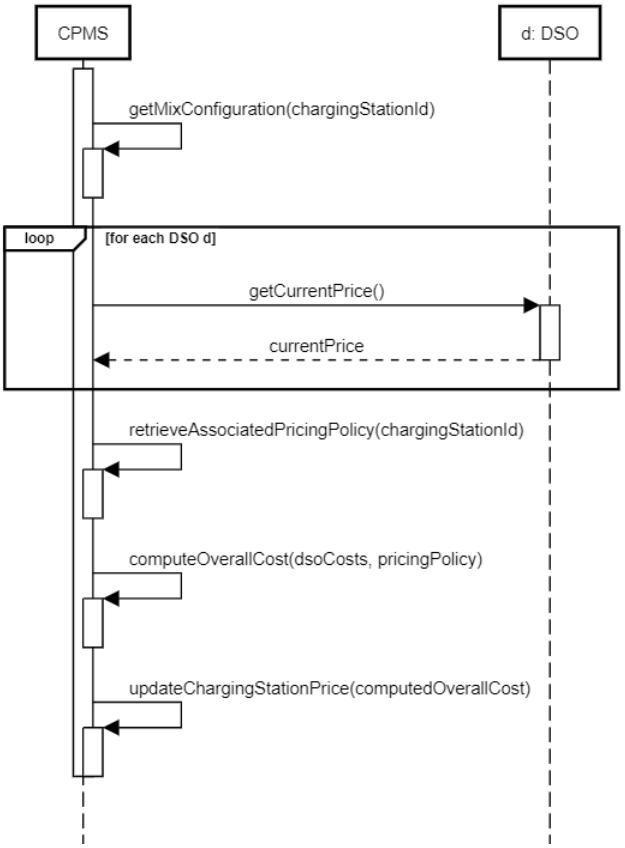
	- The new pricing policy is added to the list of configured pricing policy with the same configuration provided by the CPO Employee
Exceptions	- The CPO Employee tries to create a new pricing policy having the same name of another pricing policy already configured. → The operation fails and the CPMS informs the CPO Employee that another pricing policy with the same name is already present in the CPMS

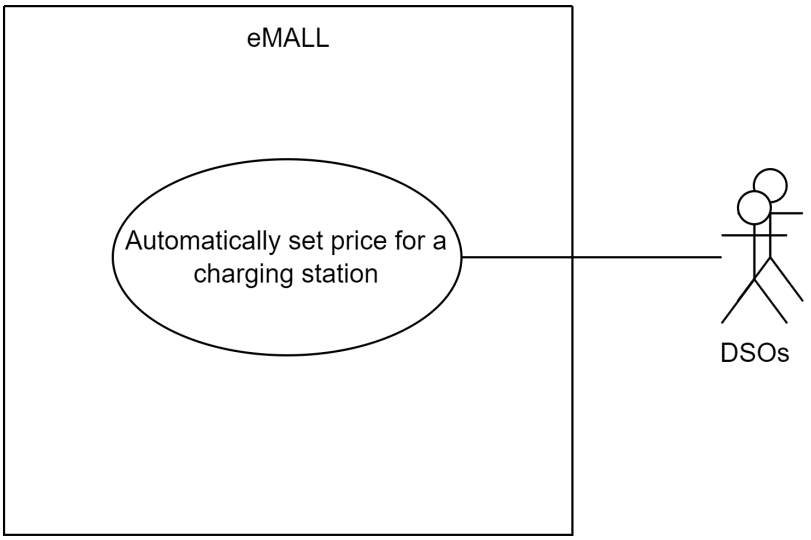




CPMS automatically sets price for a charging station [UC.13, SQD.13, UCD.10]

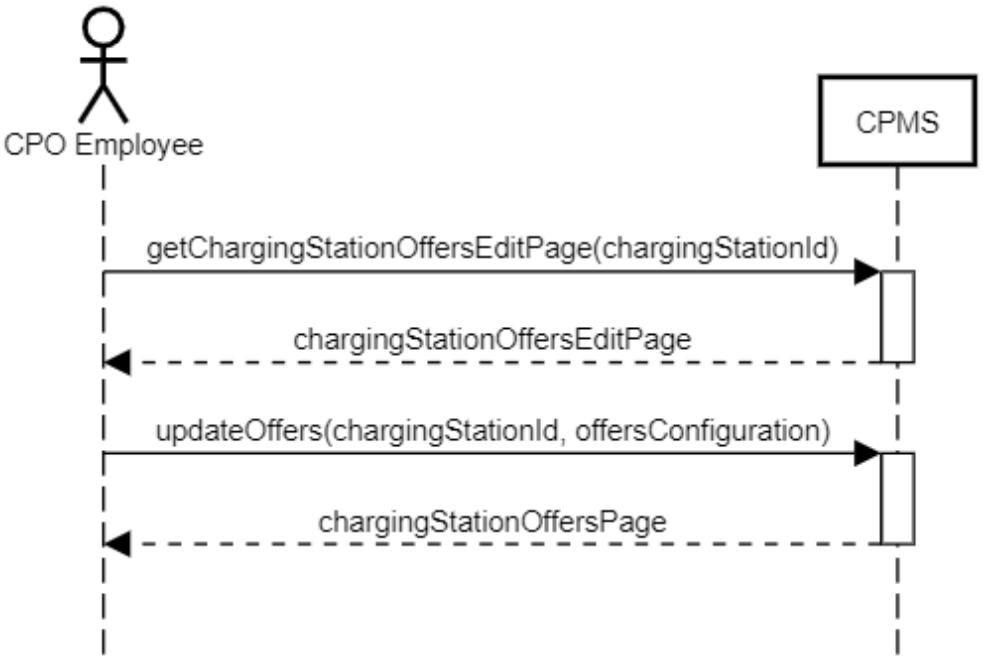
#	UC.13
Name	Automatically set price for a charging station
Actors	DSOs
Entry Conditions	- This use case has not been performed in the last 30 minutes for the same charging station
Event Flow	1. The CPMS retrieves the current configuration of the mix of energy sources of the charging station 2. The CPMS retrieves the current cost of energy from each DSO configured in the mix of energy sources 3. The CPMS computes an aggregate cost of energy based on the retrieved information 4. The CPMS sets a price for the charging station that satisfies the pricing policy associated to the charging station 5. The CPMS updates the price of the charging station with the computed one
Exit Conditions	- The price of the charging station is the one computed by the CPMS
Exceptions	/

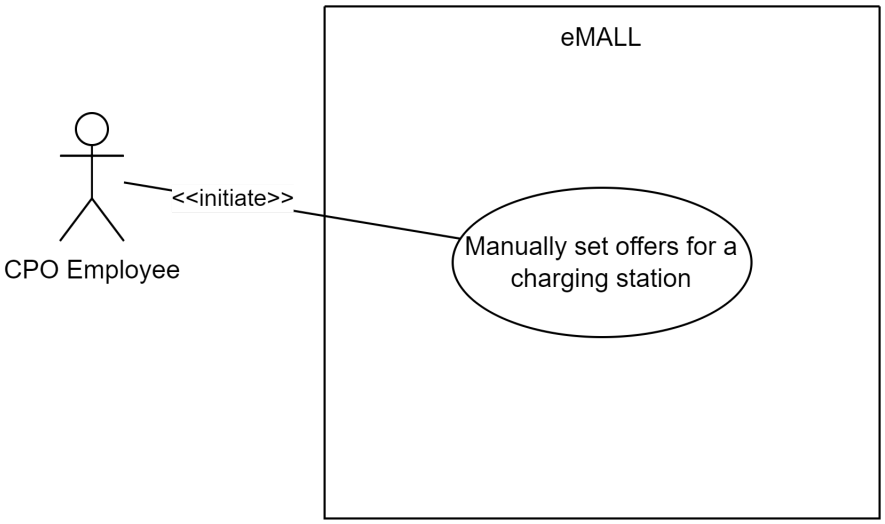




CPO employee manually sets offers for a charging station [UC.14, SQD.14, UCD.11]

#	UC.14
Name	Manually set offers for a charging station
Actors	CPO Employee
Entry Conditions	- The CPO Employee is logged into the CPMS
Event Flow	1. The CPO Employee requests to the CPMS the page to edit the offers for the specified charging station 2. The CPMS sends to the CPO Employee a form showing the active offers (if any) of the charging station. Each offer consists of the percentage of discount. 3. The CPO Employee edits the form with the desired offers for the charging station 4. The CPO Employee submits the form 5. The CPMS updates the offers for the charging station 6. The CPMS sends to the CPO Employee the page showing the updated offers for the specified charging station
Exit Conditions	- The offers of the charging station are the ones set by the CPO Employee
Exceptions	- The CPO Employee tries to insert an offer specifying a discount which is less than 0. → The CPMS displays a message reminding the CPO Employee about the correct range for those fields and doesn’t allow the CPO Employee to submit the configuration until the condition is met

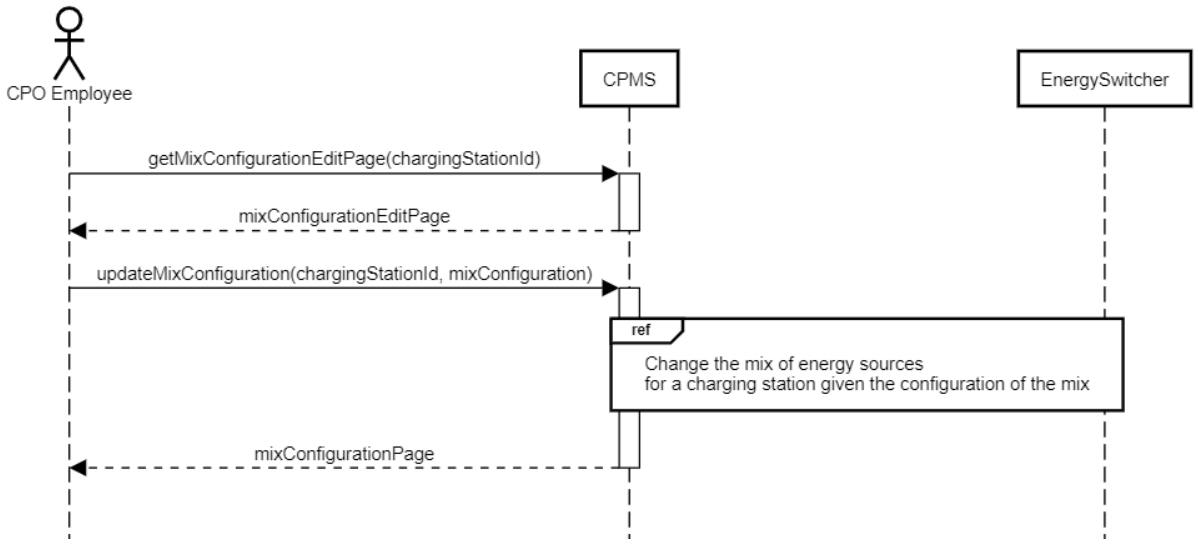




CPO employee manually changes the mix of energy sources for a charging station [UC.15, SQD.15]

#	UC.15
Name	Manually change configuration of the mix of energy sources for a charging station
Actors	CPO Employee, Energy Switcher
Entry Conditions	- The CPO Employee is logged into the CPMS - The CPO Employee knows the charging station they want to change the mix of energy sources for

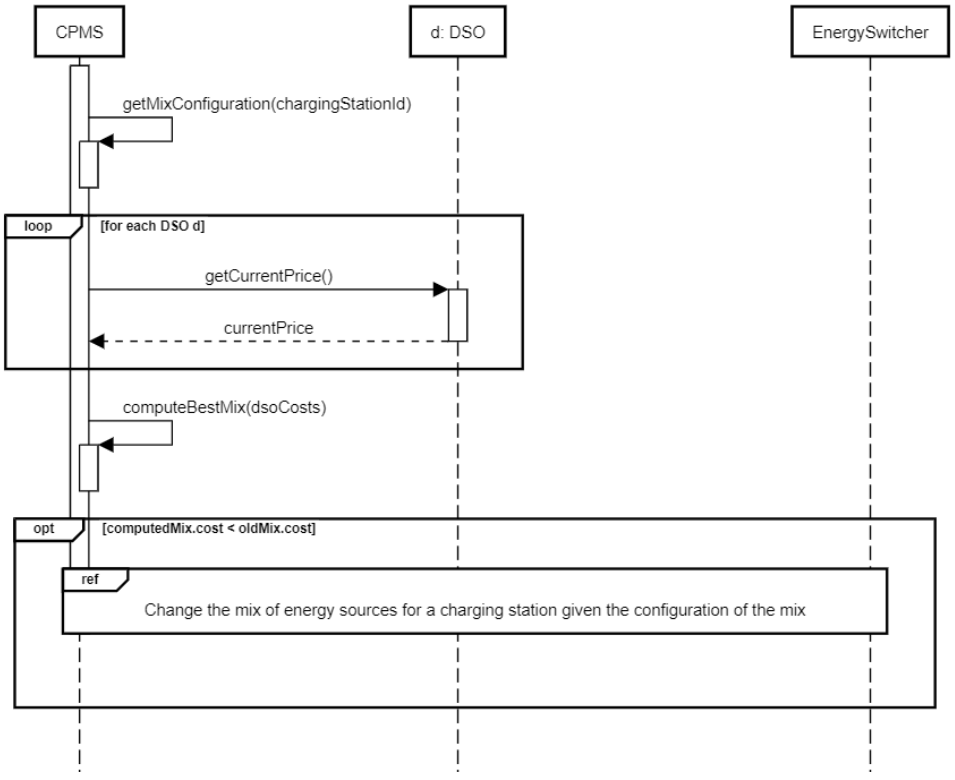
Event Flow	1. The CPO Employee requests to the CPMS the page to edit the mix of energy sources of the charging station 2. The CPMS sends to the CPO Employee the requested page that contains the list of energy sources currently used in the charging station with the related proportion of the total energy demand and, if the charging station has the battery, whether it is used as a source of energy or if it needs to be charged 3. The CPO Employee edits the list of energy sources 4. If the battery is present in the charging station, the CPO Employee chooses whether to use it as a source or if it needs to be charged 5. The CPO Employee submits to the CPMS the new configuration of the mix of energy sources 6. The CPMS changes the mix of energy sources for a charging station [UC.17] 7. The CPMS sends to the CPO Employee the page with the updated configuration of the mix for the charging station
Exit Conditions	- The new mix is correctly applied to the charging station - The new mix configuration is correctly stored in the CPMS
Exceptions	- The sum of the proportions of the energy sources is different from 100%. → The CPMS app shows the CPO Employee a descriptive message and doesn't allow the CPO Employee to submit the configuration until the condition is met



CPMS automatically change the mix of energy sources for a charging station [UC.16, SQD.16]

#	UC.16
Name	Automatically change configuration of the mix of energy sources for a charging station

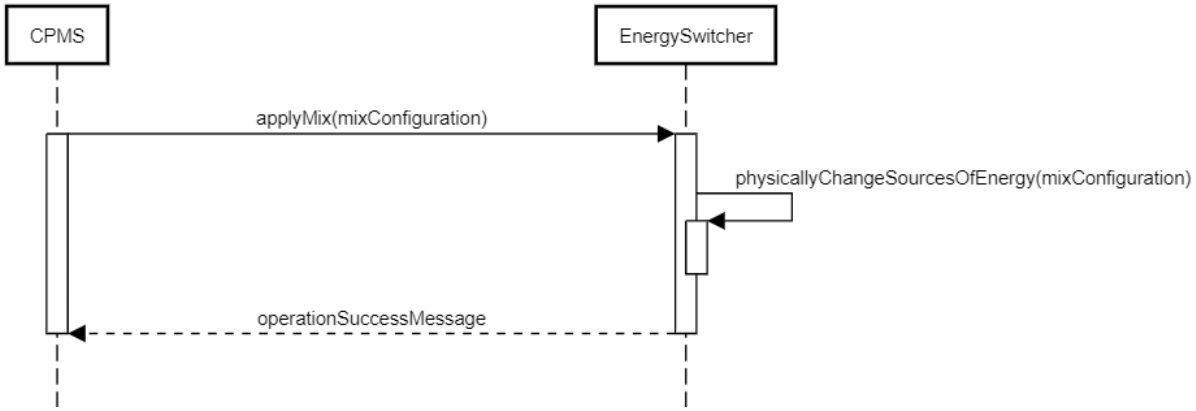
Actors	Energy Switcher, DSOs
Entry Conditions	- This use case has not been performed in the last 30 minutes for the same charging station - The charging station allows the automatic configuration of the mix of energy sources
Event Flow	1. The CPMS retrieves the current configuration of the mix of energy sources of the charging station 2. The CPMS retrieves the current cost of energy from the DSOs registered in the system 3. The CPMS computes the best potential mix for the station 4. If the computed mix costs less than the current mix, the CPMS replaces the old mix configuration with the new one 5. The CPMS changes the mix of energy sources for a charging station [UC.17]
Exit Conditions	- The new mix is correctly applied to the charging station - The new mix configuration is correctly stored in the CPMS
Exceptions	/

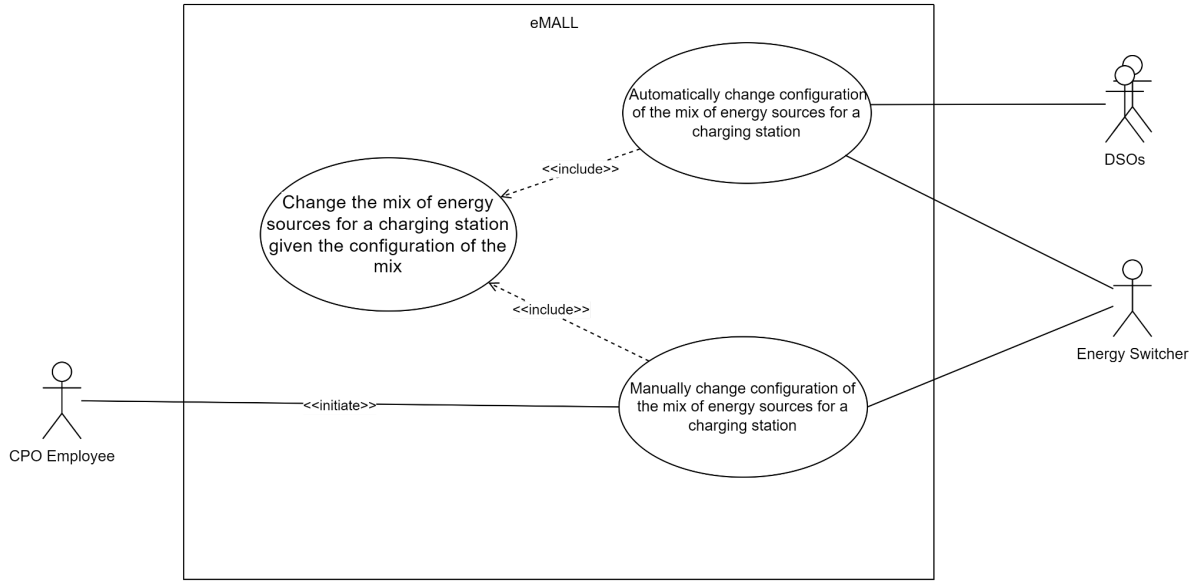


CPMS operates a change in the mix of energy sources for a charging station through the Energy Switcher [UC.17, SQD.17, UCD.12]

#	UC.17
Name	Change the mix of energy sources for a charging station given the configuration of the mix

Actors	Energy Switcher
Entry Conditions	True
Event Flow	1. The CPMS sends to the Energy Switcher of the specified charging station the new configuration of the mix of energy sources 2. The Energy Switcher physically apply the new configuration of energy sources to the charging station 3. The Energy Switcher informs the CPMS that the new mix is correctly applied
Exit Conditions	- The new mix is correctly applied to the charging station
Exceptions	/





3.3.3 Mapping Use Case - Requirements

Use Case	Requirements
UC.1	R.1
UC.2	R.4
UC.3	R.2
UC.4	R.14, R.15
UC.5	R.18, R.19, R.21, R.22, R.23
UC.6	R.46
UC.7	R.24, R.25, R.26, R.27, R.29, R.30, R.31, R.32
UC.8	R.24, R.25, R.26, R.28
UC.9	R.24, R.25, R.26, R.28
UC.10	R.24, R.25, R.26, R.28
UC.11	R.24, R.25, R.26, R.28, R.33, R.34, R.35, R.36
UC.12	R.39
UC.13	R.38, R.41
UC.14	R.42
UC.15	R.44
UC.16	R.38, R.43
UC.17	R.45

3.4 Performance Requirements

The average number of users accessing eMSP is expected to be very high due to its potential use by customers in whole regions or nations: if the service is well advertised, the app could reach millions of users.

Generally, CPMS is expected to manage a lower number of users, because the number of CPO employees accessing the application will be around a dozen for each charging station. However, CPMS workload depends on the number of charging stations managed by CPO.

Both eMSP and CPMS need to communicate with several external systems. Particular attention is due to the real time connection between CPMS and the charging points, because of the CPMS execution logic which manages the functioning of all the simultaneous charging processes happening in the charging station.

In order to support a good experience of use of the system to customers and CPO employees, the system shall have a response time of 3 seconds.

3.5 Software System Attributes

3.5.1 Reliability

The system shall prevent downtime in order to let customers manage their bookings and charges without disservices, and CPO employees to responsively administrate all their charging stations.

The system shall have MTTF (Mean Time To Failure) equal to 10^4 hours (approximately equals to a year).

3.5.2 Availability

The system shall be available 24/7. The overall workload is expected to be significant throughout the day, especially in the morning and at the end of the working hours, while a low affluence is expected during the night.

Assuming MTTF (Mean Time To Failure) = 10^4 hours and MTTR (Mean Time To Repair) = 1 hour, we can estimate an availability of 4-nines, meaning a down-time of about an hour in a year, which is considered acceptable for the application in examination.

3.5.3 Security

Security is a crucial requirement for the system, especially considering the confidential information of customers transmitted through it and stored (personal information, payment credentials…).

Moreover, access control is very critical for CPMS: e.g. CPO admins' authentication is necessary to access configuration features which are forbidden to CPO operators.

Communication between parties shall be encrypted and transmitted on a secure channel through TLS protocol.

All sensitive data, such as customer's payment credential and their position, shall be confidentially stored and encrypted with high-security encryption.

3.5.4 Maintainability

The system shall be designed thinking about maintainability as a high priority, in order to allow for future addition of functionalities with minimum effort. Design techniques which encourage high modularity and reusability of components will be followed.

Moreover, granular testing routines shall be provided.

3.5.5 Portability

A relevant aspect for the system is its cross platform potentiality, which would allow as many customers as possible to exploit the offered service on the most common platforms.

The web application shall be supported on the principal operating systems and browsers, either on mobile phones and computers. The mobile app shall be supported by iOS and Android operating systems.

4 ALLOY

4.1 Introduction

For this section of the document, we have selected a subset of requirements and a portion of the corresponding goal about the submission of bookings by customers.

In particular, the subset of requirements is:

- [R.18] eMSP allows customers to submit booking requests in advance for a charge in a specific charging station for a certain timeframe;
- [R.21] eMSP shall send customers' booking requests to the corresponding CPMS;
- [R.23] eMSP shall confirm/deny a booking request to the associated customer depending on the CPMS's response;

and the sub-goal (portion of goal G.2) is:

- A Customer wants to book a charge in a specific charging station for a certain timeframe.

We used the Alloy specification language to formally prove that if the given subset of requirements is satisfied, then the specified goal is satisfied as well. To provide a proof for this implication, we have modelled just the interested parts of the system. It follows the list of the main signatures used, along with their attributes:

- Customer
    - id: natural identifier for the Customer. It is helpful to maintain the identity of a Customer throughout the various instances that represent state changes
    - bookingRequests: the set of BookingRequest sent to the eMSP
    - observedBookings: the set of Booking that have been observed by the Customer

- BookingRequest
    - customerId: natural identifier of the Customer that issued this BookingRequest
    - chargingStationId: natural identifier of the ChargingStation where the Customer requests a charge
    - timeFrame: timeframe requested by the Customer for the charge
- Booking:
    - customerId: natural identifier of the Customer associated to this Booking
    - chargingStationId: natural identifier of the ChargingStation associated to this Booking
    - timeFrame: timeframe in which the charge requested by the Customer will be consumed
    - socket: the socket associated to the charge
- ChargingStation:
    - id: natural identifier for the ChargingStation. It is helpful to maintain the identity of a Customer throughout the various instances that represent state changes
    - acceptedBookings: the set of Booking registered for this ChargingStation

The verification of the goal is represented as the assertion "goalAchieved".

4.2 Alloy Code

```
sig Socket { }

sig CustomerId { }
fact {
	all cId: CustomerId | some c: Customer | c.id = cId
}
sig Customer {
	id: one CustomerId,
	bookingRequests: set BookingRequest,
	observedBookings: set Booking
} {
	//for all observed bookings there must be one request with the same fields
	all observedBooking: observedBookings |
	#{bookingRequest: bookingRequests |
	isBookingRequestRelativeToBooking[bookingRequest, observedBooking]} = 1

	//there are not two equal observed bookings
	all b1, b2: observedBookings | equalBookings[b1, b2] implies b1 = b2

	//there are not two equal booking requests
	all br1, br2: bookingRequests | equalBookingRequests[br1, br2] implies br1 = br2

	//all observedBookings need to be associated to this customer
	all b: observedBookings | b.customerId = id

	//all booking requests need to be associated to this customer
	all br: bookingRequests | br.customerId = id
}

sig ChargingStationId { }
fact {
	all csId: ChargingStationId | some cs: ChargingStation | cs.id = csId
}

sig ChargingStation {
	id: one ChargingStationId,
	acceptedBookings: set Booking,
	sockets: some Socket
}{
	//there are not two equal accepted bookings
	all b1, b2: acceptedBookings | equalBookings[b1, b2] implies b1 = b2

	//bookings with the same socket do not have intersected timeframes
	all b1, b2: acceptedBookings | b1.socket = b2.socket and
	areTimeFramesIntersected[b1.timeFrame, b2.timeFrame] implies b1 = b2
```

```
        //all accepted bookings need to refer to this charging station
        all b: acceptedBookings | b.chargingStationId = id

        //all accepted bookings need to refer to a socket of this charging station
        all b: acceptedBookings | b.socket in sockets
}

sig DateTime {
        value: one Int
}

pred isBefore(t1: DateTime, t2: DateTime) {
        t1.value < t2.value
}

pred isAfter(t1: DateTime, t2: DateTime) {
        t1.value > t2.value
}

sig TimeFrame {
        startDateTime: one DateTime,
        endDateTime: one DateTime
} {
        isBefore[startDateTime, endDateTime]
}

pred areTimeFramesIntersected(t1: TimeFrame, t2: TimeFrame) {
        ! isBefore[t2.endDateTime, t1.startDateTime] and
        ! isAfter[t2.startDateTime, t1.endDateTime]
}

pred equalTimeFrames(t1: TimeFrame, t2: TimeFrame) {
        t1.startDateTime = t2.startDateTime
        t1.endDateTime = t2.endDateTime
}

sig BookingRequest {
        customerId: one CustomerId,
        chargingStationId: one ChargingStationId,
        timeFrame: one TimeFrame
}

fact {
        all br: BookingRequest | some c: Customer | br in c.bookingRequests
}
```

```
pred equalBookingRequests(b1: BookingRequest, b2: BookingRequest) {
        b1.customerId = b2.customerId and
        b1.chargingStationId = b2.chargingStationId and
        equalTimeFrames[b1.timeFrame, b2.timeFrame]
}

pred isBookingRequestRelativeToBooking(br1: BookingRequest, b2: Booking) {
        br1.customerId = b2.customerId and
        br1.chargingStationId = b2.chargingStationId and
        equalTimeFrames[br1.timeFrame, b2.timeFrame]
}

sig Booking {
        customerId: one CustomerId,
        chargingStationId: one ChargingStationId,
        timeFrame: one TimeFrame,
        socket: one Socket
}

pred equalBookings(b1: Booking, b2: Booking) {
        b1.customerId = b2.customerId and
        b1.chargingStationId = b2.chargingStationId and
        equalTimeFrames[b1.timeFrame, b2.timeFrame] and
        b1.socket = b2.socket
}

fact {
        all booking: Booking | one bookingRequest: BookingRequest |
        isBookingRequestRelativeToBooking[bookingRequest, booking]
        all bookingRequest: BookingRequest | lone booking: Booking |
        isBookingRequestRelativeToBooking[bookingRequest, booking]
}

// requirement: eMSP allows customers to submit booking requests in advance for a charge
in a specific charging station for a certain timeframe
pred createBookingRequest(requiredCustomerId: CustomerId, requiredChargingStationId:
ChargingStationId, requiredTimeFrame: TimeFrame) {
        one c: Customer |
        one c': Customer |
        one newBr: BookingRequest |
        c.id = requiredCustomerId and
        c'.id = c.id and
        newBr.customerId = requiredCustomerId and
        newBr.chargingStationId = requiredChargingStationId and
        newBr.timeFrame = requiredTimeFrame and
        #{br: c.bookingRequests | equalBookingRequests[br, newBr]} = 0 and
        c'.bookingRequests = c.bookingRequests + newBr and
        c'.observedBookings = c.observedBookings
```

```
}

// requirement: eMSP shall send customers' booking requests to the corresponding CPMS
pred createBooking(requiredCustomerId: CustomerId, requiredChargingStationId:
ChargingStationId, requiredTimeFrame: TimeFrame) {
        one newBooking: Booking |
        one cs: ChargingStation |
        one cs': ChargingStation |
        cs.id = requiredChargingStationId and
        cs'.id = cs.id and
        cs'.sockets = cs.sockets and
        newBooking.customerId = requiredCustomerId and
        newBooking.chargingStationId = requiredChargingStationId and
        newBooking.timeFrame = requiredTimeFrame and
        #{booking: cs.acceptedBookings | equalBookings[booking, newBooking]} = 0 and
        cs'.acceptedBookings = cs.acceptedBookings + newBooking
}

//requirement: eMSP shall confirm/deny a booking request to the associated customer
pred informTheCustomer(requiredCustomerId: CustomerId, requiredChargingStationId:
ChargingStationId, requiredTimeFrame: TimeFrame){
        one c: Customer |
        one c': Customer |
        one newObservedBooking: Booking |
        one bookingRequest: BookingRequest |

        //pick the correct customer based on the booking request
        bookingRequest.customerId = requiredCustomerId and
        bookingRequest.chargingStationId = requiredChargingStationId and
        bookingRequest.timeFrame = requiredTimeFrame and
        #{br: c.bookingRequests | equalBookingRequests[br, bookingRequest]} = 1 and

        //define the right new booking
        newObservedBooking.customerId = requiredCustomerId and
        newObservedBooking.chargingStationId = requiredChargingStationId and
        newObservedBooking.timeFrame = requiredTimeFrame and

        //c' has same attributes as c, but a new observed booking
        c'.id = c.id and
        c'.bookingRequests = c.bookingRequests and
        #{booking: c.observedBookings | equalBookings[booking, newObservedBooking]} = 0
        and c'.observedBookings = c.observedBookings + newObservedBooking
}

//partial goal: a customer wants to book a charge in a specific charging station for a certain
timeframe
pred book(requiredCustomerId: CustomerId, requiredChargingStationId: ChargingStationId,
requiredTimeFrame: TimeFrame) {
```

```
        some c: Customer |
        c.id = requiredCustomerId and
        one booking: c.observedBookings |
        booking.chargingStationId = requiredChargingStationId and
        booking.timeFrame = requiredTimeFrame
}

assert goalAchieved {
        all requiredCustomerId: CustomerId, requiredChargingStationId: ChargingStationId,
requiredTimeFrame: TimeFrame |
        (createBookingRequest[requiredCustomerId, requiredChargingStationId,
requiredTimeFrame]
        and
        createBooking[requiredCustomerId, requiredChargingStationId, requiredTimeFrame]
        and
        informTheCustomer[requiredCustomerId, requiredChargingStationId,
requiredTimeFrame])
        implies
        book[requiredCustomerId, requiredChargingStationId, requiredTimeFrame]
}
pred showGoalAchieved(requiredCustomerId: CustomerId, requiredChargingStationId:
ChargingStationId, requiredTimeFrame: TimeFrame) {
        createBookingRequest[requiredCustomerId, requiredChargingStationId,
requiredTimeFrame]
        and
        createBooking[requiredCustomerId, requiredChargingStationId, requiredTimeFrame]
        and
        informTheCustomer[requiredCustomerId, requiredChargingStationId,
requiredTimeFrame]
}

check goalAchieved for 5
run showGoalAchieved for 5
```

4.3 Commented results

One instance for this model produced these results:

this/Customer	id	bookingRequests	observedBookings
Customer$0	CustomerId$0	BookingRequest$1	Booking$0
Customer$1	CustomerId$0	BookingRequest$0	Booking$0
		BookingRequest$1	
Customer$2	CustomerId$0	BookingRequest$0	Booking$0
		BookingRequest$1	Booking$1

Customer$0 represents a customer that had already requested a booking (BookingRequest$1) and had already received a confirmation for the booking (Booking$0 is in its observedBookings set).

Customer$1 represents the same customer after they requested a new booking (BookingRequest$0).

Customer$2 represents the same customer after they received the confirmation for the booking they previously requested (Booking$1).

this/ChargingStation	id	acceptedBookings	sockets
ChargingStation$0	ChargingStationId$0	Booking$0	Socket$0
			Socket$1
ChargingStation$1	ChargingStationId$0	Booking$0	Socket$0
		Booking$1	Socket$1

ChargingStation$0 represents a charging station with an accepted booking (Booking$0).

ChargingStation$1 represents the same charging station after the customer requested the new booking (Booking$1).

this/Booking	customerId	chargingStationId	timeFrame	socket
Booking$0	CustomerId$0	ChargingStationId$0	TimeFrame$0	Socket$1
Booking$1	CustomerId$0	ChargingStationId$0	TimeFrame$1	Socket$0

this/BookingRequest	customerId	chargingStationId	timeFrame
BookingRequest$0	CustomerId$0	ChargingStationId$0	TimeFrame$1
BookingRequest$1	CustomerId$0	ChargingStationId$0	TimeFrame$0

BookingRequest$0 is the request for Booking$1 and BookingRequest$1 is the request for Booking$0.

As we can see, the customer with natural id CustomerId$0 successfully booked a charge (Booking$1) in a charging station (charging station with natural id ChargingStationId$0) for the time frame TimeFrame$1.

5 EFFORT SPENT

Section	Total effort spent
Section 1 - Introduction	40 hours
Section 2 - Overall description	20 hours
Section 3 - Specific requirements	60 hours
Section 4 - Alloy	20 hours

Team member’s contributions:

Brugnano Matilde 70 hours

Buttiglieri Giorgio Natale 70 hours