

# **e-Mobility for All**

## Requirement Analysis & Specification Document

Abbondanza Alessia - 995959  
Campo Marco Lorenzo - 103213  
De Luca Alessandro - 103542

November 2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Definition, Acronyms & Abbreviations . . . . .	3
1.3	Scope . . . . .	4
1.3.1	Actors . . . . .	5
1.3.2	World Phenomena . . . . .	5
1.3.3	Shared Phenomena . . . . .	6
1.3.4	Goals . . . . .	7
1.4	Revision History . . . . .	8
1.5	Reference Documents . . . . .	8
1.6	Document Structure . . . . .	8
<b>2</b>	<b>Overall Description</b>	<b>10</b>
2.1	Product Perspective . . . . .	10
2.1.1	Scenarios . . . . .	10
2.1.2	Class Diagrams . . . . .	13
2.1.3	Activity Diagrams . . . . .	16
2.2	Product Functions . . . . .	18
2.2.1	Users Related Functions . . . . .	18
2.2.2	CPO Related Functions . . . . .	18
2.3	User Characteristics . . . . .	19
2.4	Assumptions, Dependencies & Constraints . . . . .	19
<b>3</b>	<b>Specific Requirements</b>	<b>21</b>
3.1	External Interface Requirements . . . . .	21
3.1.1	User Interfaces . . . . .	21
3.1.2	CPMS interfaces . . . . .	21
3.1.3	Hardware Interfaces . . . . .	21
3.1.4	Software Interfaces . . . . .	22
3.1.5	Communication Interfaces . . . . .	23
3.2	Functional Requirements . . . . .	26
3.2.1	Requirements . . . . .	26

3.2.2	Mapping of Requirements . . . . .	28
3.2.3	Use Cases Description . . . . .	32
3.2.4	Sequence Diagrams . . . . .	41
3.3	Performance Requirements . . . . .	48
3.4	Design Constraints . . . . .	48
3.4.1	Standards Compliance . . . . .	48
3.4.2	Hardware Limitations . . . . .	48
3.5	Software System Attributes . . . . .	48
3.5.1	Reliability & Availability . . . . .	48
3.5.2	Security . . . . .	49
3.5.3	Maintainability . . . . .	49
3.5.4	Portability . . . . .	49
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>50</b>
4.1	Alloy Model . . . . .	50
4.1.1	Model Description . . . . .	50
4.1.2	Alloy Code . . . . .	51
4.2	Metamodels . . . . .	56
4.2.1	World 1 - General case . . . . .	56
4.2.2	World 2 - No overlapping bookings for a station . . . . .	56
4.2.3	World 3 - No overlapping bookings for a user . . . . .	57
4.2.4	World 4 - Charging station with no bookings . . . . .	57
4.2.5	World 5 - User with no bookings . . . . .	58
4.3	Result of Predicates . . . . .	59
<b>5</b>	<b>Effort Spent</b>	<b>60</b>
5.1	Team Effort . . . . .	60
5.2	Alessia Abbondanza . . . . .	60
5.3	Marco Lorenzo Campo . . . . .	61
5.4	Alessandro De Luca . . . . .	61

# Chapter 1

## Introduction

### 1.1 Purpose

This is a Requirement Analysis and Specification Document (RASD). The aim of this document is to describe the e-Mall system in terms of functional and non-functional requirements, analyzing the real needs of the users in order to purposely design the system, taking into consideration constraints and limitations. Use cases for the different scenarios, user interfaces and user experience will be addressed with respect to the different actors involved in such a system.

E-Mall stands for e-Mobility for All, the aim is to provide a technology to make bookings, managing them and charge electric vehicles more easily, this will also allow to limit the carbon footprint caused by urban and suburban mobility. Additionally, e-Mall will plan the charging process of the electric vehicle in such a way to introduce minimal constraints on the user's schedule.

### 1.2 Definition, Acronyms & Abbreviations

Here we provide a list of useful abbreviations that will be used throughout the document.

Definitions:

- **e-Mall:** e-Mobility for All, software that will be analyzed in detail in this document
- **eMSP:** e-Mobility Service Providers, software to manage the interaction between users and charging stations
- **CPOs:** Charge Point Operators, who physically sets up the station and operates it. They interact with a CPMS.

- **CPMS:** Charge Point Management System, software used by CPOs to manage the charging station and the operations that take place there.
- **DSO:** Distribution System Operator, outside organization that provides the energy.

Acronyms:

- **API:** Application Programming Interface
- **GPS:** Global Positioning System

Abbreviations:

- **WPx:** World Phenomena number x
- **SPx:** Shared Phenomena number x
- **Gx:** Goal number x
- **Rx:** Requirement number x
- **DAx:** Domain Assumption number x
- **Fx:** Function number x
- **Sx:** Scenario number x
- **UCx:** Use case number x

## 1.3 Scope

E-Mall is a complex system which will help users and charging stations manage the whole charging process of an electric vehicle. For this reason, this document will focus on e-Mall as two communicating subsystems. This will allow for an easier understanding of each part and of the system as a whole unit.

The two separate and communicating modules are:

- **e-Mall eMSP:** downloadable application module for users to book, manage and pay for charges

- **e-Mall CPMS:** software implemented by CPOs to manage charging stations

Unfortunately, especially in the south-west region of Europe, coverage for electric charges is very scarce. For this reason, e-Mall will also interact with other providers' CPMSs, in order to always provide the user with the most comprehensive and optimal charging option.

This RASD document will focus on the interaction between a user and the e-Mall eMSP, as well as the interaction between said application and CPMSs. The CPMSs taken into consideration are both e-Mall CPMS and external CPMSs provided by third parties, each one owned by a different CPO. The interaction approach between eMSP and CPMSs will be a focus of the project as well.

This document will also describe a smarter eMSP feature consisting in the possibility for the eMSP to proactively suggest the user a custom charging plan, tailored for their needs. This plan will depend on the status of the battery of the vehicle, the schedule of the user, the special offers made available by some CPOs, and the availability of charging slots at the chosen stations.

### 1.3.1 Actors

The actors that are outside of our control and thus part of the outside world are:

**Users:** People who interact with the e-Mall eMSP, they actively perform all the main functions that the software needs to provide;

**CPMS:** Third party software installed to manage charging stations.

**DSO:** Energy provider, acts according to the charging station needs;

### 1.3.2 World Phenomena

World phenomena: events occurring in the outside world that the system described in this document cannot observe.

**WP1.** The charge status of the vehicle changes

**WP2.** Prices provided by the DSOs change

**WP3.** CPOs operators decide what special offers to provide

**WP4.** CPOs operators decide to use locally stored energy or acquire it from a DSO

**WP5.** CPOs operators decide what DSO to acquire energy from

**WP6.** User decides to end the charging process before it's due

### 1.3.3 Shared Phenomena

Shared phenomena: Shared Phenomena are events happening in the World, which the software system described in this document is involved in.

- SP1.** User asks for a list of nearby charging stations
- SP2.** User asks for price for a selected station
- SP3.** User asks for offers for a selected station
- SP4.** User asks for information for a selected station
- SP5.** User books a charge in a specific charging station and time frame
- SP6.** User deletes a booking of a charge in a specific charging station and time frame
- SP7.** User starts the charging process
- SP8.** User receives information about time left to end the charge
- SP9.** User receives a notification when the charge is complete
- SP10.** User ends the charging process before charge is complete
- SP11.** User inputs their banking credentials to pay for a charge
- SP12.** User is notified by eMSP that the vehicle should be charged based on the battery status
- SP13.** User is notified by eMSP that the vehicle should be charged based on their daily calendar
- SP14.** User is notified by eMSP that the vehicle should be charged based on the available offers
- SP15.** eMSP retrieves information on special offers of CPOs
- SP16.** eMSP retrieves information on the costs provided by CPOs
- SP17.** eMSP retrieves information on the available power outlets
- SP18.** eMSP retrieves information on the battery status of the vehicle
- SP19.** eMSP retrieves information on the user's calendar
- SP20.** Information is collected and made available through API for the eMSPs to check. Such as: number of charging sockets available, charge type (slow, fast or rapid), the cost and estimated time until free (if currently in use)

- SP21.** eMSP books an available socket
- SP22.** An eMSP books a socket for a specific user and said socket is reserved for that user
- SP23.** eMSP deletes a previously made booking
- SP24.** eMSP starts the charging process
- SP25.** eMSP monitors the charging process
- SP26.** eMSP ends the charging process
- SP27.** The user registers to the e-Mall network
- SP28.** The user logs into the e-Mall network

#### **1.3.4 Goals**

In order for e-Mall to satisfy all that is written in the specification document, these Goals have been defined as the ones that need to be achieved in such a way to have a functioning product.

##### **eMSP goals**

- G1.** Allow users to know the location, status, prices and special offers provided by charging stations
- G2.** Allow users to manage bookings of a charge in a specific location and time frame
- G3.** Allow users to start, monitor and end the charging process
- G4.** Allow users to pay for the obtained service
- G5.** Suggest users to charge their vehicle based on the charge status, the schedule of the users, special offers provided by the CPOs and availability of the charging slots at the identified charging stations

##### **CPMS goals**

- G6.** Provide a complete “external” overview of the charging station (location, availability, charging options. . . )
- G7.** Manage and correctly execute a charge for a specific time frame



- G8.** Know the “internal” status of a charging station
- G9.** Handle the payment of a charge
- G10.** Provide the eMSP with the most convenient charging options by choosing to use locally stored energy or acquire it from DSOs

## 1.4 Revision History

- Version 1.0 - 12/23/2022

## 1.5 Reference Documents

References useful to understand the requirement and what contained in the document are:

- Requirement Engineering and Design Project: goal, schedule and rules, Software Engineering 2, Politecnico di Milano, A.Y. 2022-2023;
- IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, ISO/IEC/IEEE 29149 dated 2018;
- M. Jackson and P. Zave, "Deriving Specifications from Requirements: an Example," 1995 17th International Conference on Software Engineering, 1995, pp. 15-15, doi; 10.1145/225014.225016.

## 1.6 Document Structure

This document is composed of four sections:

1. **Introduction:** this section introduces the concept of the e-Mall system to the reader, highlighting the scope of the project, its goals, but also the terms and acronyms that will be used throughout the report, and the reference documents
2. **Overall Description:** the main goal of this section is to make the reader understand the constraints and limitations of the system, in addition to providing assumptions made or given

3. **Specific Requirements:** here both functional and non-functional requirements are precisely disclosed, listing all the requirements and attributes of the system; in this section a list of scenarios is provided
4. **Formal Analysis using Alloy:** contains an alloy model of the system and its description.

# Chapter 2

## Overall Description

### 2.1 Product Perspective

This section provides details on the main functionalities offered by the system.

#### 2.1.1 Scenarios

A scenario is a scene that illustrates a possible interaction between an Actor and the e-Mall system. Scenarios capture the system, as viewed from the outside, by a user, using specific examples.

- **S1.** User checks charging stations
- **S2.** User manages a booking of a charge in a specific charging station and time frame
- **S3.** User decides to delete a charge booked previously
- **S4.** User charges the vehicle
- **S5.** User pays for the charge
- **S6.** User is notified by the e-Mall eMSP that the vehicle should be charged
- **S7.** e-Mall eMSP generates a charging plan based on user's habits
- **S8.** e-Mall CPMS requests information about energy cost and source mix to the DSO
- **S9.** e-Mall eMSP asks to the external CPMSs information regarding the external status of a charging station (number of charging sockets available, their type, their cost, and the estimated amount of time until the first socket of the needed type is freed)

- **S10.** e-Mall eMSP sends payment notification to external CPMS
- **S11.** e-Mall eMSP receives update about charge from external CPMS
- **S12.** e-Mall eMSP books a charge from an external CPMS
- **S13.** e-Mall eMSP deletes a charge from an external CPMS
- **S14.** e-Mall CPMS decides which source to acquire energy from

**Scenario 1:** *User checks charging stations*

It's 12:00, Marco, noticing his car's battery is running low, decides he needs to charge his vehicle. He opens the e-Mall mobile applications and decides to look for charging stations in his proximity. The application provides him a map view of his surroundings marking registered charging stations he can choose from.

**Scenario 2:** *User manages a booking of a charge in a specific charging station and time frame*

Marco decides to pick a specific station, e-Mall shows him all the information regarding costs, availability and special offers provided by said station. Unfortunately there are no rapid charge sockets available, but Marco needs one as soon as possible, so he chooses another station and books a charge.

**Scenario 3:** *User decides to delete a previous booking*

Alessia booked a charge in her favorite station, but remembers she forgot the doctor appointment that she had to attend in the next 20 minutes. She decides to check her bookings and delete her most recent one. She will go to the doctor and eventually make a new booking depending on stations availability.

**Scenario 4:** *User charges the vehicle*

Alessia finally managed to book a charge. She gets to the charging station and plugs the fast charge plug into her car's outlet. She opens the e-Mall application and starts the charge, she's met with a notification stating the charge has started and an estimated time until completion. She checks the battery's status from time to time through the e-Mall application.

**Scenario 5:** *User pays for the charge*

After a while, Alessia receives another notification from e-Mall stating that the charge is completed. A new page pops up on her smartphone with the cost of the charge to pay digitally (through e-Mall's transactions system).

**Scenario 6:** *User is notified by the eMSP that the vehicle should be charged*

Alessandro enabled e-Mall's smart features and is going on about his day. All of a sudden he receives a notification from e-Mall, reminding him to charge his vehicle because the battery is low. This notification also provides him with suggestions on what charging station to choose based on his usual routes, an estimate of costs of the charge and special offers taken into consideration.

**Scenario 7:** *eMSP generates a charging plan based on user's habits*

Matteo enables the smart scheduling feature on his device. The e-Mall application generates a customised charging plan. This new plan is based on his usual daily schedule. He checks the new plan and considers when to charge the vehicle next.

**Scenario 8:** *CPMS requests information to the DSO*

Sandra is the owner of a charging station. She accesses the CPMS portal to get information on the energy available from the different DSOs. This is done via the CPMS through the use of APIs. The information contain: energy cost and means of production of that energy. Sandra then uses this information to set up the prices and the offers.

**Scenario 9:** *eMSP asks to the external CPMSs information regarding the external status of a charging station*

Maria accesses the e-Mall application to get information on the available charging station. Unfortunately she is in an area where no e-Mall charging stations are present. In order to provide users with the necessary information to book a charge in a different charging station, the eMSP connects to third party CPMSs through the use of APIs to gather information regarding the external status of each charging station.

**Scenario 10:** *eMSP sends payment notification to external CPMS*

Once the charging process has ended, the user pays for the charge with their digital credit. The eMSP sends all the payment details to the external CPMS through APIs which will register the transaction.

**Scenario 11:** *eMSP receives updates about charge from external CPMS*

The external CPMS periodically sends updates about the state of the charging process to the eMSP in order to display them to the user. This is possible through the use of standardized APIs.

**Scenario 12:** *eMSP books a charge from an external CPMS*

The eMSP shows the user all the possible charging stations available. Once the user selects one that belongs to an external organization, the eMSP sends a booking request to the associated external CPMS in order to book a socket. The CPMS

responds with the approval or denial of the booking request.

**Scenario 13:** *eMSP deletes a charge from an external CPMS*

Andrea decides to delete a booking made from an external charging station. As in Scenario 3 the user clicks on their eMSP application the button to delete a charge, then the eMSP sends a deletion request to the external CPMS. The CPMS acknowledges the request and responds with the confirmation and deletes the reservation.

**Scenario 14:** *CPMS decides which source to dispense energy from*

Due to a fault in the dispensing system, the charging station managed by Mario lacks energy. Mario doesn't want to lose his clients (or at least the majority of them) and thus decides to provide energy from the batteries that are present in the station, knowing that the fault will be fixed soon. This can happen also automatically if the system knows that a fault is happening and switches to battery dispensing mode.

## 2.1.2 Class Diagrams

The following UML class diagram provides a high level overview of the system. It contains the representation of all the elements of the application domain and how they interact with each other. This diagram does not include all the classes needed for the implementation of the architecture, since its aim is to describe the system's domain. The diagrams will focus on e-Mall CPMS and e-Mall e-MSP as two separate subsystems interacting with each other, as well as the outside world.

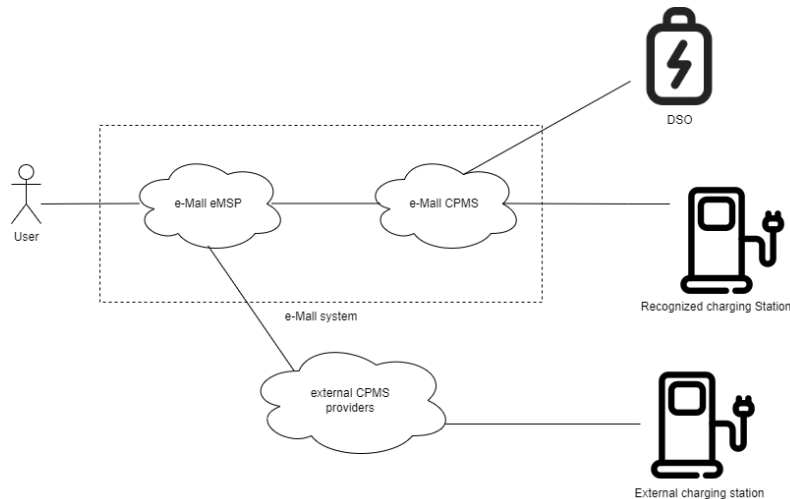


Figure 2.1: High level overview of the e-Mall system

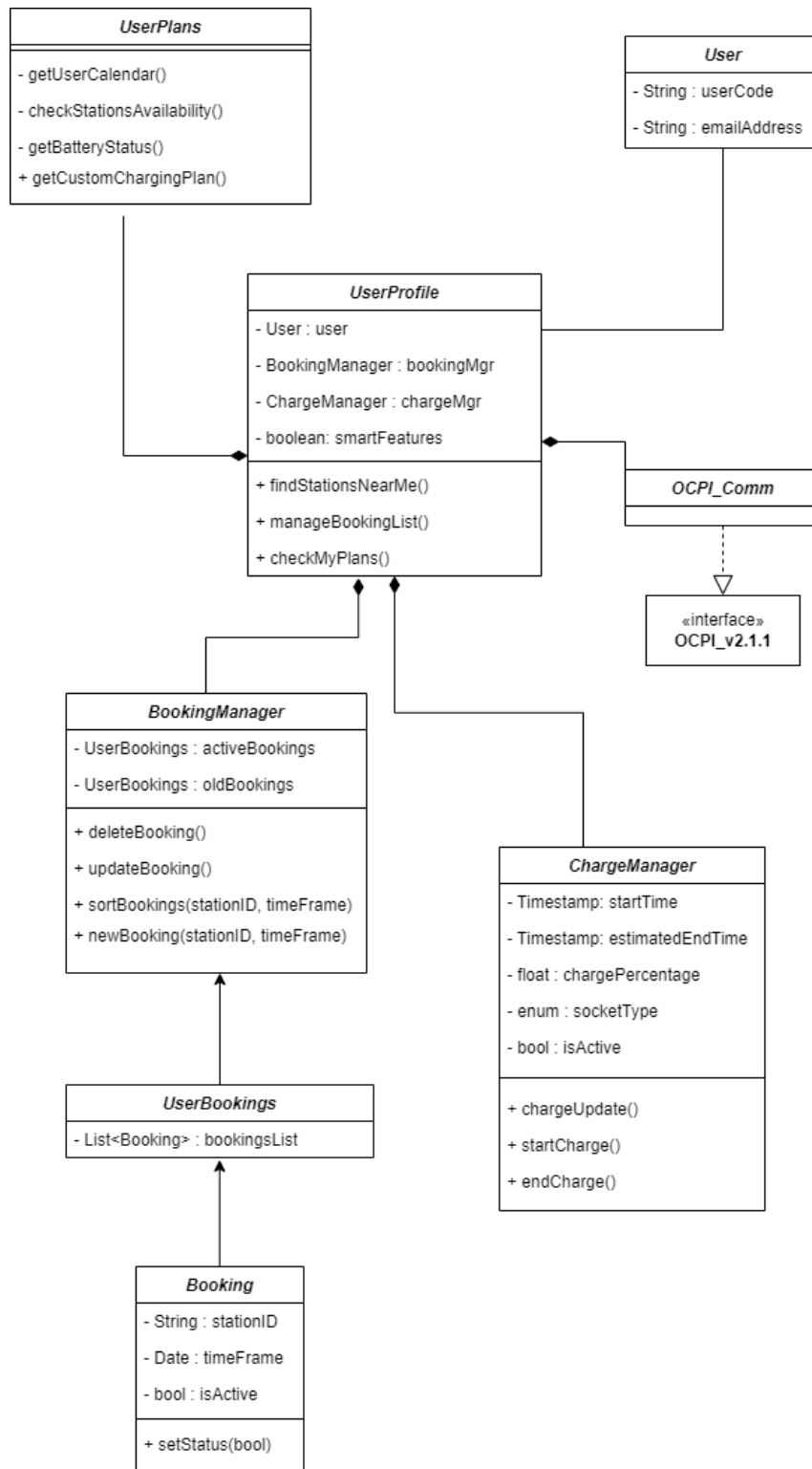


Figure 2.2: Class diagram of the e-Mall eMSP main components

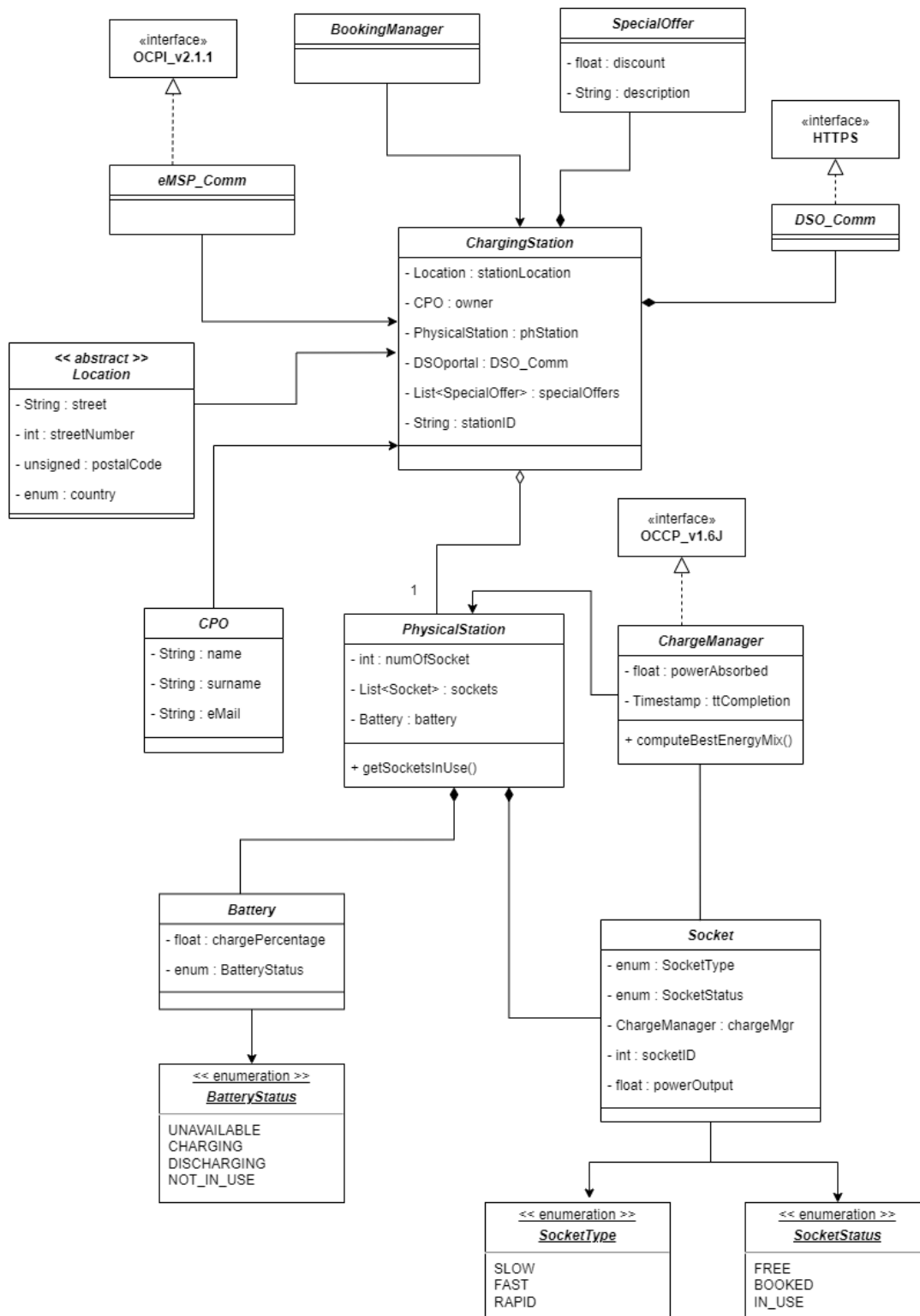


Figure 2.3: Class diagram of the e-Mall CPMS main components



### 2.1.3 Activity Diagrams

This section focuses on the evolution of the system during its interaction with the actors.

The activity diagram below shows the interaction between the user and the eMSP, highlighting how the user can book a charge, manage it and see its custom plans based on the daily schedule.

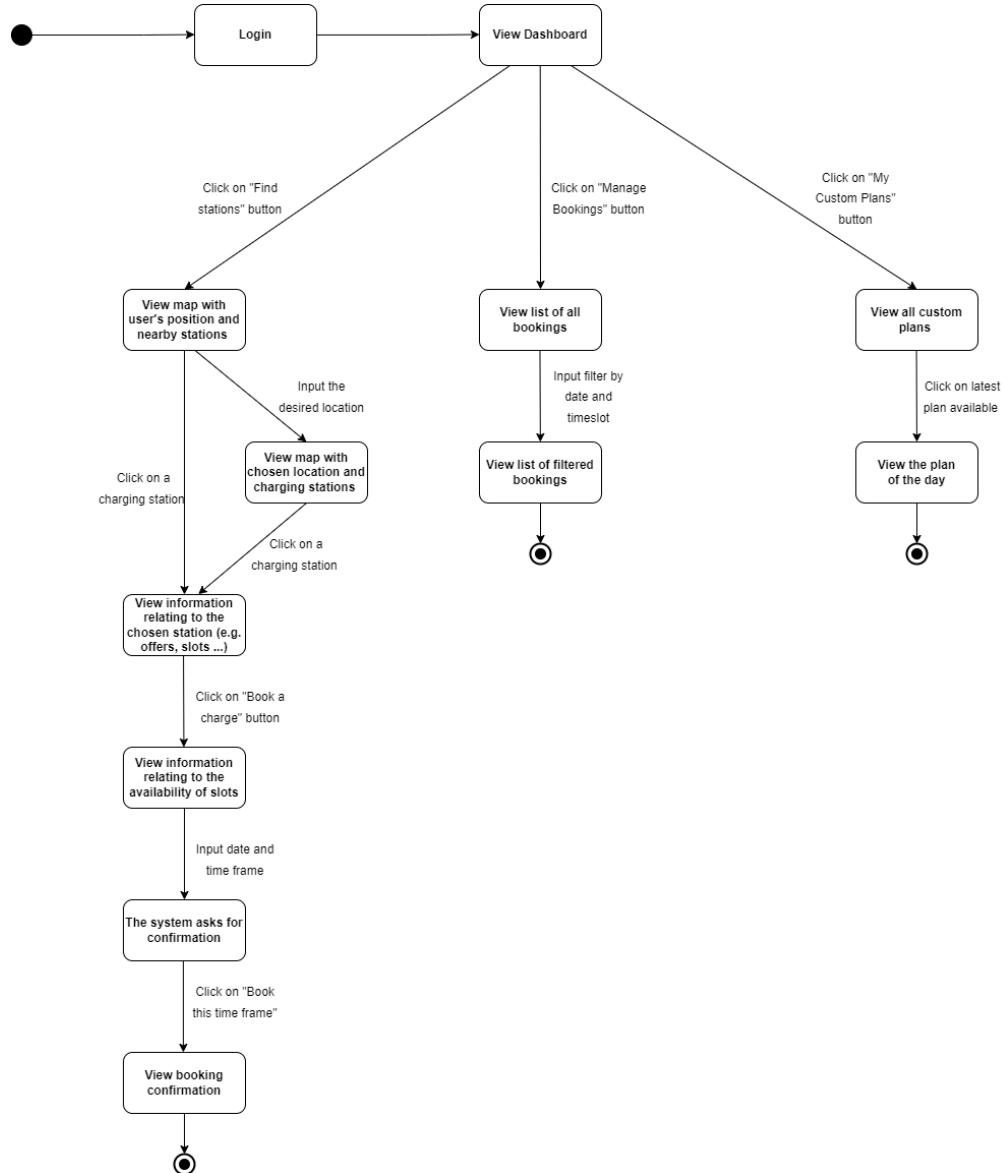


Figure 2.4: User's activity diagram

The activity diagram below shows the interaction between the CPO operator and their Charging Station (through the use of e-Mall CPMS), external DSOs and e-Mall eMSP.

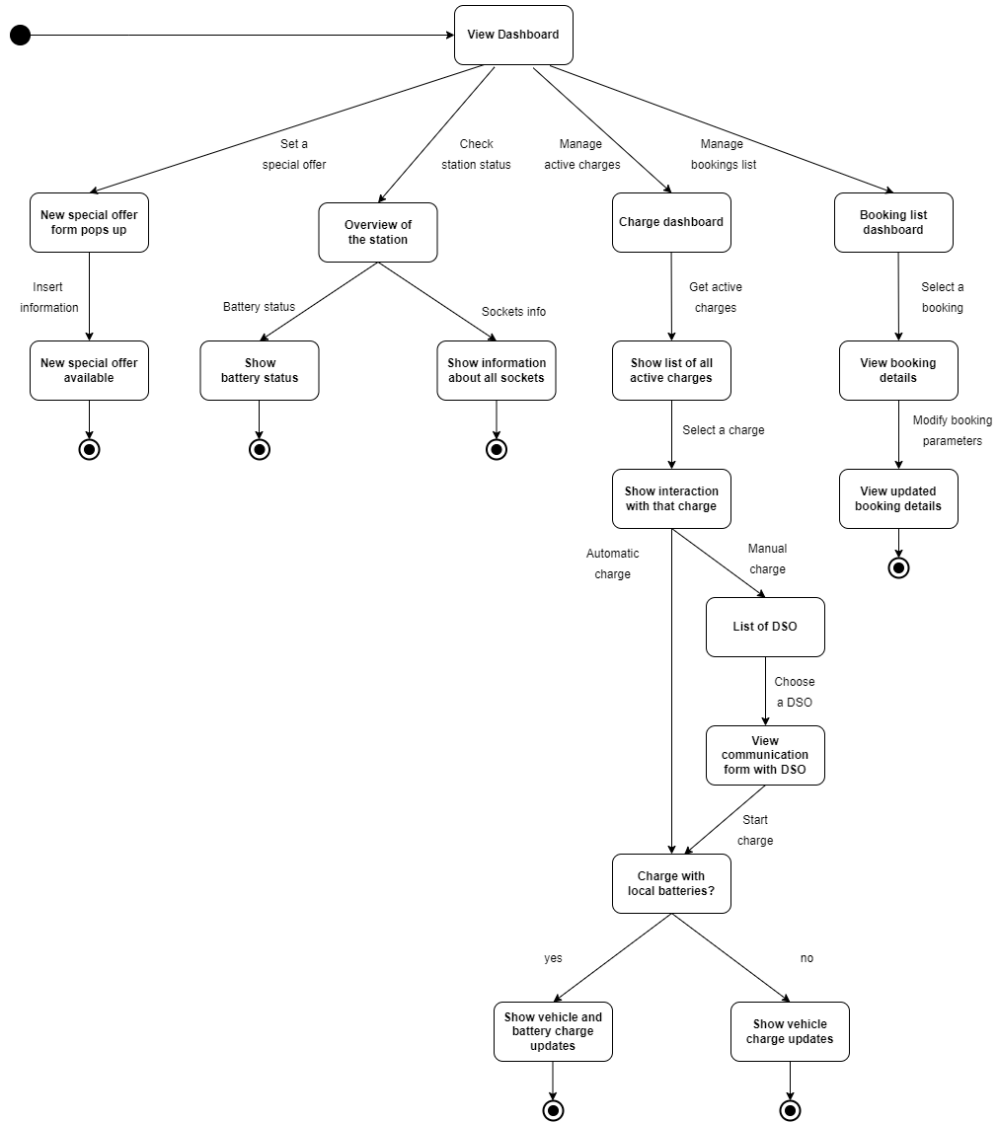


Figure 2.5: CPO's activity diagram

## 2.2 Product Functions

The aim of the e-Mall system is to provide a platform to combine useful data, involving stations prices, locations, special offers and energy prices for Users, as well as provide CPOs the easiest and most reliable Charging Station handling software. This section presents the main functionalities provided by the e-Mall system. The two main actors are Users and Charging Point Operators.

### 2.2.1 Users Related Functions

- F1: e-Mall eMSP can make a custom charging plan, tailored for the specific user. This is made possible by a smart feature that takes into account the user's habits, their charge status, charging station availability and (if possible) special offers available.
- F2: e-Mall eMSP provides the user an up-to-date map of their surroundings (acquired via GPS), highlighting charging stations and special offers they may provide.
- F3: e-Mall eMSP provides the user a search bar, to look for charging stations in a specific location.
- F4: e-Mall eMSP gives the possibility to show availability in a specific charging station, making it possible to book a specific time slot starting within 15 minutes from the moment of the booking
- F5: e-Mall eMSP gives the possibility to consult previous bookings, sort them and, if necessary, delete an active one.
- F6: e-Mall eMSP makes it possible to manage the charging process, giving the start and end signal to a charge, monitoring the charging process and eventually pay for the charge.
- F7: e-Mall eMSP provides a registration and login platform, so that every action needs to be authenticated before execution.

### 2.2.2 CPO Related Functions

- F1: e-Mall CPMS gives operators the possibility to look at their station internal status. So, availability of the charging slots, their type, their energy output and their local batteries status.
- F2: e-Mall CPMS provides the operators a way to promote new special offers to be seen by eMSPs users.

- F3: e-Mall CPMS provides operators a section to manually interact with DSOs to decide the optimal energy mix to be used and/or be stored in local batteries.
- F4: e-Mall CPMS provides operators a section to look at bookings for each charging socket.
- F5: e-Mall CPMS provides operators a way to manually interact with each socket, in order to provide all the functions supported by the DIN (or ISO) protocol implemented.

## 2.3 User Characteristics

e-Mall will support these actors in their interaction with the system:

- **Users:** People who interact with the e-Mall eMSP, they actively perform all the main functions that the software needs to provide;
- **External CPMSs:** Software installed by third parties to manage charging stations. E-Mall eMSP will interact with it;
- **CPO operators:** They can work for a CPO and manually interact with DSOs through e-Mall CPMS.

## 2.4 Assumptions, Dependencies & Constraints

- DA1.** An internet connection is available at all times during the interaction with the system
- DA2.** Every user has an Internet connected device
- DA3.** Every device used by the users has an integrated GPS sensor
- DA4.** GPS is active when the web application is running
- DA5.** APIs of external CPMSs provide reliable information
- DA6.** APIs of DSOs provide reliable information
- DA7.** Every vehicle is correctly paired with the user's device at all times
- DA8.** External CPMSs use the same communication protocol as e-Mall (OCCP)
- DA9.** E-Mall can read vehicle's and user's data
- DA10.** Every station has an Internet connection

**DA11.** CPMS sensors installed by the CPO are always present and reliable

**DA12.** When the time slot for a charge is active, the charging slot is always accessible

# Chapter 3

## Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The functionalities of the e-Mall eMSP will be provided through a downloadable web application, this implementation allows for easy access and implementation and will be available for both android and iOS.

Users will always use their phone to book and manage the charges through the application, optimized for a more “on the go” use.

#### 3.1.2 CPMS interfaces

The functionalities of the e-Mall CPMS will be provided with a downloadable and installable desktop only web application. This will provide CPOs with all the necessary tools to manage the charging station, as well as manual interfaces to interact directly with DSOs and the charging sockets.

#### 3.1.3 Hardware Interfaces

A user of the system must have access to a smartphone with a stable internet connection and localization provided via GPS in order to interact with the system. A CPO must have access to the dedicated computer of the charging station, which needs to be properly hooked to all the charging station’s sensors in order to provide the functionalities already discussed.

### 3.1.4 Software Interfaces

E-Mall system relies on external interfaces to provide its functionalities. We will now analyze both of the subsystems.

- **Region Map:** The system uses a public API to provide the different users with a map of the geographical region of interest, with respect to their position and their needs. It will also provide special markers in order to show the position of charging stations;
- **External CPMS:** e-Mall needs to face CPMSs of other providers in order to provide the best and most comprehensive charging option available. This communication is made possible by the communication protocol that will be discussed later;
- **DSOs interfaces:** The system will interact through APIs with energy providers (DSOs) and exchange information about charging prices.

### 3.1.5 Communication Interfaces

The two subsystems communicate via the internet with each other and external CPMS, making this distinction invisible to the user.

The three main communication protocols in use in the e-Mall system: OCPI, OCCP, DIN/ISO. Each component of the system implements the corresponding protocol and uses it for communication with the environment/other subsystems (both internal and external).

e-Mall CPMS is also capable of communicating with DSOs through standardized APIs. The communication protocol implemented is HTTPS.

**OCPI (v 2.1.1)** : The Open Charge Point Interface (OCPI) enables a scalable, automated EV roaming setup between CPOs and eMSPs. It supports authorization, charge point information exchange (including live status updates and transaction events), charge detail record exchange, remote charge point commands and, finally, the exchange of smart-charging commands between parties.

It also handles the authorization of the charging session and the handling of the bookings.



Figure 3.1: OCPI communication protocol

**OCCP (v 1.6J)** : Communication protocol implemented by CPMS (e-Mall CPMS and external ones). Built on the client-server paradigm, the protocol is used to handle:

- Charging session (Start/Stop session);
- Exchange of session data (battery percentage, time left to end the charge, local energy storage status etc.);
- Diagnostic of the charging point (reboot, firmware updates, logs);
- Updating of the charging point status.



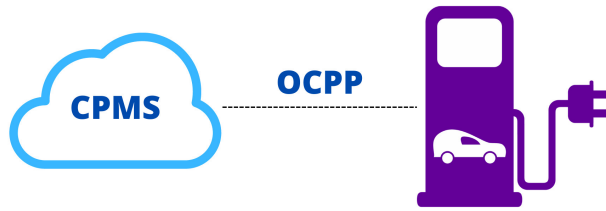


Figure 3.2: OCPP communication protocol

**DIN (SPEC 70121) (or ISO 15118)** : Communication protocol implemented by the CPMSs and some electric vehicles. It handles the exchange of charging information from the vehicle to the CPMS such as:

- Electric Vehicle information;
- Charging control signal (state of the charge, max power input).

In case ISO 15118 is implemented, it can also provide the Plug & Charge functionality that enables an electric vehicle to automatically identify and authorize itself to a compatible charging station on behalf of the driver, to receive energy for recharging its battery.



Figure 3.3: DIN/ISO communication protocols

### Communication Approach

The communication style implemented depends on the communication scenario/interface.

- **e-Mall eMSP  $\longleftrightarrow$  CPMS:** This scenario requires a mixed communication style.

The data requests from eMSP to CPMS are performed whenever an user needs for example a rendering of information about stations. In this case a query is performed to nearby stations in order to provide all necessary information. For this reason the system needs to be responsive to asynchronous requests from outside.

When a charge is in progress and updates are provided to the user's eMSP, a series of synchronous messages are exchanged between the parties, as long as the charge is underway. In this case a preferred refresh rate would be 1 update each 3 seconds, taking into consideration the nature of the information that's being exchanged.

- **e-Mall CPMS  $\longleftrightarrow$  DSO:** This is a different communication interface, however the communication choice implemented is still asynchronous.

APIs of DSOs are queried when a charge needs to take place and energy prices and sources are required.

## 3.2 Functional Requirements

### 3.2.1 Requirements

- R1.** The e-Mall application should allow users to register by providing their email address
- R2.** The e-Mall application should allow users to login using the credentials input at registration time
- R3.** The e-Mall application should retrieve locations of charging stations from their CPMS
- R4.** The e-Mall application should show availability of charging stations from their CPMS
- R5.** The e-Mall application should retrieve prices of charging stations from their CPMS
- R6.** The e-Mall application should retrieve special offers of charging stations from their CPMS
- R7.** The e-Mall application should acquire locations of charging stations from their CPMS
- R8.** The e-Mall application should allow the user to book a charging station for a specific location and time frame
- R9.** The e-Mall application should allow the user to delete a previously booked charging station
- R10.** The e-Mall application should register a start signal when the “start charge” button is pressed
- R11.** The e-Mall application should forward the start signal of the charging to the CPMS
- R12.** The e-Mall application should register an end signal when the “end charge” button is pressed
- R13.** The e-Mall application should forward the end signal of the charging to the CPMS
- R14.** The e-Mall application should allow users to insert their banking credentials
- R15.** The e-Mall application should be able to communicate with the user’s bank

- R16.** The e-Mall application should connect to external CPMSs through API
- R17.** The e-Mall application should know the charge status of the user's vehicle
- R18.** The e-Mall application should know the daily schedule of the user
- R19.** The e-Mall application should be aware of special offers provided by the CPO
- R20.** The e-Mall application should be aware of the availability of charging slots at the identified station
- R21.** The CPMS should collect real time information about available charging sockets
- R22.** The CPMS should collect real time information about the charge speed provided by each charging socket (classified as slow, fast or rapid)
- R23.** The CPMS should collect real time information about the cost of a charge
- R24.** The CPMS should collect real time information about the estimated time to completion of a charge currently happening
- R25.** The CPMS should set an "available" socket as "unavailable" for 15 minutes after it receives a booking
- R26.** The CPMS should set a booked socket as "available" if no charge has been started for 15 minutes
- R27.** The CPMS should set a booked socket as "available" if the eMSP deletes the relative booking
- R28.** The CPMS should set a socket as "unavailable" as long as a charge is in process
- R29.** The CPMS should stop dispensing power when the plugged battery is fully charged
- R30.** The CPMS should provide real time expenses of a charge
- R31.** The CPMS should receive payments from the eMSP
- R32.** The CPMS should communicate with DSOs' API to retrieve the cost of energy
- R33.** The CPMS should choose the most convenient DSO based on the eMSPs' charge requests
- R34.** The CPMS should know the CPOs' local energy availability
- R35.** The CPMS should know if batteries are available in the charging station

- R36.** The CPMS should compute where to dispense energy from (local batteries, DSOs or a mixture of the two), to provide the most convenient options
- R37.** The CPMS should allow operators to manually handle the energy acquisition and dispense procedure
- R38.** The CPMS should communicate the current special offers to the e-Mall application
- R39.** The e-Mall application should know the charge status of the user's vehicle
- R40.** The e-Mall application should know the daily schedule of the user
- R41.** The e-Mall application should be aware of special offers provided by the CPO
- R42.** The e-Mall application should be aware of the availability of charging slots at the identified station

### 3.2.2 Mapping of Requirements

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
DA1	X	X	X	X	X					
DA2	X	X								
DA3	X	X	X	X						
DA4	X									
DA5					X					X
DA6										X
DA7						X				
DA8			X							X
DA9					X					
DA10						X			X	
DA11							X	X		
DA12			X	X		X	X			

Table 3.1: Goal-Domain Assumption mapping

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
R1	X	X	X	X	X					
R2	X	X	X	X	X					
R3	X									
R4	X									
R5	X									
R6	X									
R7	X									
R8		X								
R9		X								
R10			X							
R11			X							
R12			X							
R13			X							
R14				X						
R15				X						
R16						X	X	X	X	X
R17					X					
R18					X					
R19					X					
R20					X					
R21						X	X			
R22						X	X			
R23						X	X			
R24						X	X			
R25							X			
R26							X			
R27							X			
R28							X			
R29							X			
R30			X						X	
R31									X	
R32										X
R33						X				
R34								X		
R35								X		
R36										X
R37										X
R38	X					X				
R39			X		X					
R40					X					
R41	X					X				
R42	X				29	X				

Table 3.2: Goal-Requirements mapping

Goal	Domain Assumptions	Requirements
G1	DA1, DA2, DA3, DA4	R1, R2, R3, R4, R5, R6, R7, R38, R41, R42
G2	DA1, DA2, DA3	R1, R2, R8, R9
G3	DA1, DA3, DA8, DA12	R1, R2, R10, R11, R12, R13, R30, R 39
G4	DA1, DA3, DA12	R1, R2, R14, R15
G5	DA1, DA5, DA9	R1, R2, R17, R18, R19, R20, R 39, R40
G6	DA7, DA10, DA12	R16, R21, R22, R23, R24, R33, R38, R41, R42
G7	DA11, DA12	R16, R21, R22, R23, R24, R25, R26, R27, R28, R29
G8	DA11	R16, R34, R35
G9	DA10	R16, R30, R31
G10	DA5, DA6, DA8	R16, R32, R36, R37

Table 3.3: Goal-Domain Assumptions-Requirements mapping

To simplify the reading, there is a sum up of the domain assumptions and goals used in the tables of the mappings. Domain assumptions:

- DA1.** An internet connection is available at all times during the interaction with the system
- DA2.** Every user has an Internet connected device
- DA3.** Every device (smartphone or computer) used by the users has an integrated GPS sensor
- DA4.** GPS is active when the web application is running
- DA5.** APIs of external CPMSs provide reliable information
- DA6.** APIs of DSOs provide reliable information
- DA7.** Every vehicle is correctly paired with the user's device at all times
- DA8.** External CPMSs use the same communication protocol as e-Mall (OCCP, OCPI)
- DA9.** E-Mall can read vehicle's and user's data
- DA10.** Every station has an Internet connection
- DA11.** CPMS sensors installed by the CPO are always present and reliable
- DA12.** When the time slot for a charge is active, the charging slot is always accessible

Goals:

- G1.** Allow users to know the location, status, prices and special offers provided by charging stations
- G2.** Allow users to manage bookings of a charge in a specific location and time frame
- G3.** Allow users to start, monitor and end the charging process
- G4.** Allow users to pay for the obtained service
- G5.** Suggest users to charge their vehicle based on the charge status, the schedule of the users, special offers provided by the CPOs and availability of the charging slots at the identified charging stations
- G6.** Provide a complete “external” overview of the charging station (location, availability, charging options. . .)
- G7.** Manage and correctly execute a charge for a specific time frame
- G8.** Know the “internal” status of a charging station
- G9.** Handle the payment of a charge
- G10.** Provide the eMSP with the most convenient charging options by choosing to use locally stored energy or acquire it from DSOs



### 3.2.3 Use Cases Description

Use cases capture functional requirements of a system from the outside perspective.

Platform Login	
<b>ID</b>	UC.1
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• The user has access to the e-Mall application</li><li>• The application is running on his device</li></ul>
<b>Event Flow</b>	<ol style="list-style-type: none"><li>1. The user launches the e-Mall application</li><li>2. The user clicks the “Sign Up” button</li><li>3. The user inserts eMail and password</li><li>4. The user clicks the “submit” button</li></ol>
<b>Exit Conditions</b>	The user is correctly logged in the system
<b>Exceptions</b>	<ul style="list-style-type: none"><li>• If the userCode is not recognized by the system, the credentials are not registered or the userCode is incorrect. The system notifies the actor and the procedure is aborted.</li><li>• If the inserted password is wrong, the system notifies the actor and the procedure is aborted</li></ul>

Table 3.4: Platform Login use case

<b>Stations Check In A Specified Location</b>	
<b>ID</b>	UC.2
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user has accessed the e-Mall application</li> <li>• The application is running on his device</li> <li>• The GPS is properly set up and working</li> <li>• The internet connection works properly</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The user launches the e-Mall application</li> <li>2. The user selects the "Find Stations" section</li> <li>3. The user inputs the desired location</li> <li>4. The system show a map marking the charging stations in the area</li> </ol>
<b>Exit Conditions</b>	The user is met with a map marking the charging stations in the desired location
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• There are no available station in the specified area</li> <li>• The internet connection is disrupted</li> <li>• GPS position cannot be acquired</li> </ul>

Table 3.5: Stations Check use case

<b>Book a Charge</b>	
<b>ID</b>	UC.3
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user has accessed the e-Mall application</li> <li>• The application is running on his device</li> <li>• The GPS is properly set up and working</li> <li>• The internet connection works properly</li> <li>• The user has a view of the stations to choose from</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The user selects a specific station</li> <li>2. The system shows all details regarding that specific station</li> <li>3. The user goes to the “Book a Charge” page in said station</li> <li>4. The user selects a date and a time frame</li> <li>5. The user submits the form</li> <li>6. System asks for confirmation</li> <li>7. The user clicks on “Book this time frame”</li> <li>8. The system shows the booking to the user</li> <li>9. The system notifies the user using the provided contacts</li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• The user has correctly booked a charge in a specific station and time frame</li> <li>• The charge has been added to the section “My Bookings” in the user’s profile</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• The internet connection is disrupted</li> <li>• The time frame selected has been taken by another user before the system receives a confirmation from current user</li> <li>• GPS position cannot be acquired</li> </ul>

Table 3.6: Book a Charge use case

<b>Monitor Booking</b>	
<b>ID</b>	UC.4
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user has accessed the e-Mall application</li> <li>• The application is running on his device</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The user goes to the "Manage bookings list" section</li> <li>2. The user can filter bookings by date and time slot</li> <li>3. The system provides the user a list with all the bookings that match the filter</li> </ol>
<b>Exit Conditions</b>	The user views the bookings
<b>Exceptions</b>	If no bookings are available, an error is prompted

Table 3.7: Monitor Bookings use case

<b>Delete Booking</b>	
<b>ID</b>	UC.5
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user has accessed the e-Mall application</li> <li>• The application is running on his device</li> <li>• The user has accessed the “Manage My Booking list” function</li> <li>• The internet connection works properly</li> <li>• The charging process for the reservation the user wants to delete didn’t started yet</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The user sees the reservation they want to delete</li> <li>2. System asks for confirmation</li> <li>3. The user clicks the “Delete” button</li> <li>4. The user confirms the deletion</li> <li>5. The system notifies the user that the deletion process has been successful</li> </ol>
<b>Exit Conditions</b>	The selected reservation is deleted from the user’s booking list
<b>Exceptions</b>	The internet connection is disrupted

Table 3.8: Delete Booking use case

Executing a Charge	
<b>ID</b>	UC.6
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user has accessed the e-Mall application</li> <li>• The application is running on his device</li> <li>• The GPS is properly set up and working</li> <li>• The internet connection works properly</li> <li>• The vehicle is at the charging station during a booked timeframe</li> <li>• The plug is inserted in the vehicle</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The system sends a notification to the user stating that the booked time frame has started</li> <li>2. The user presses the start button from the application</li> <li>3. The system presents the user with the charging page with the charging percentage and the estimated time to completion, providing updates periodically</li> <li>4. System confirms the end of the charge at the end of time frame or the user decides to stop</li> <li>5. If the user decides to end the charge the system asks for confirmation to end the charge</li> </ol>
<b>Exit Conditions</b>	The user has terminated the charging process
<b>Exceptions</b>	

Table 3.9: Executing a Charge use case

New Custom Plan	
<b>ID</b>	UC.7
<b>Actors</b>	User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• The user is logged into the application</li> <li>• e-Mall advanced features are on</li> <li>• e-Mall has access to user's calendar</li> <li>• e-Mall and user's vehicle are correctly paired</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. System sends a notification informing the user a new charging plan is available</li> <li>2. User accesses the "My Custom Plans" section in his profile where a list of all custom plans can be checked</li> <li>3. User clicks on the latest one</li> </ol>
<b>Exit Conditions</b>	A custom charging plan is available to be read by the users
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>• Permissions have been revoked, so data can't be collected</li> <li>• User's device and vehicle aren't paired, data can't be collected</li> <li>• GPS position cannot be acquired</li> </ul>

Table 3.10: New Custom Plan use case

Request to DSO	
<b>ID</b>	UC.8
<b>Actors</b>	DSO
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Internet connection is available</li> <li>• DSOs' API are reachable</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. e-Mall CPMS connects to the DSO</li> <li>2. e-Mall CPMS requests information about the energy cost and energy mix to the DSO</li> <li>3. DSO provides all information needed</li> </ol>
<b>Exit Conditions</b>	Information are available to the e-Mall CPMS
<b>Exceptions</b>	Internet connection is disrupted

Table 3.11: Request to DSO use case



<b>Exchange Charge Details to DSO</b>	
<b>ID</b>	UC.9
<b>Actors</b>	DSO
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• Internet connection is available</li> <li>• Connection with the DSO is open and running</li> </ul>
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. CPMS requests the DSO to start dispensing energy</li> <li>2. DSO answers with a “start charge” message</li> <li>3. DSO forwards information regarding the charge</li> <li>4. CPMS requests the DSO to stop dispensing energy</li> <li>5. DSO answers with a “stop charge” message</li> </ol>
<b>Exit Conditions</b>	Vehicle is fully charged and CPMS has ended the charge process
<b>Exceptions</b>	Internet connection is disrupted

Table 3.12: Exchange Charge Details to DSO use case

### 3.2.4 Sequence Diagrams

In this section a series of sequence diagrams is presented, to understand the functioning of the use cases analyzed above.

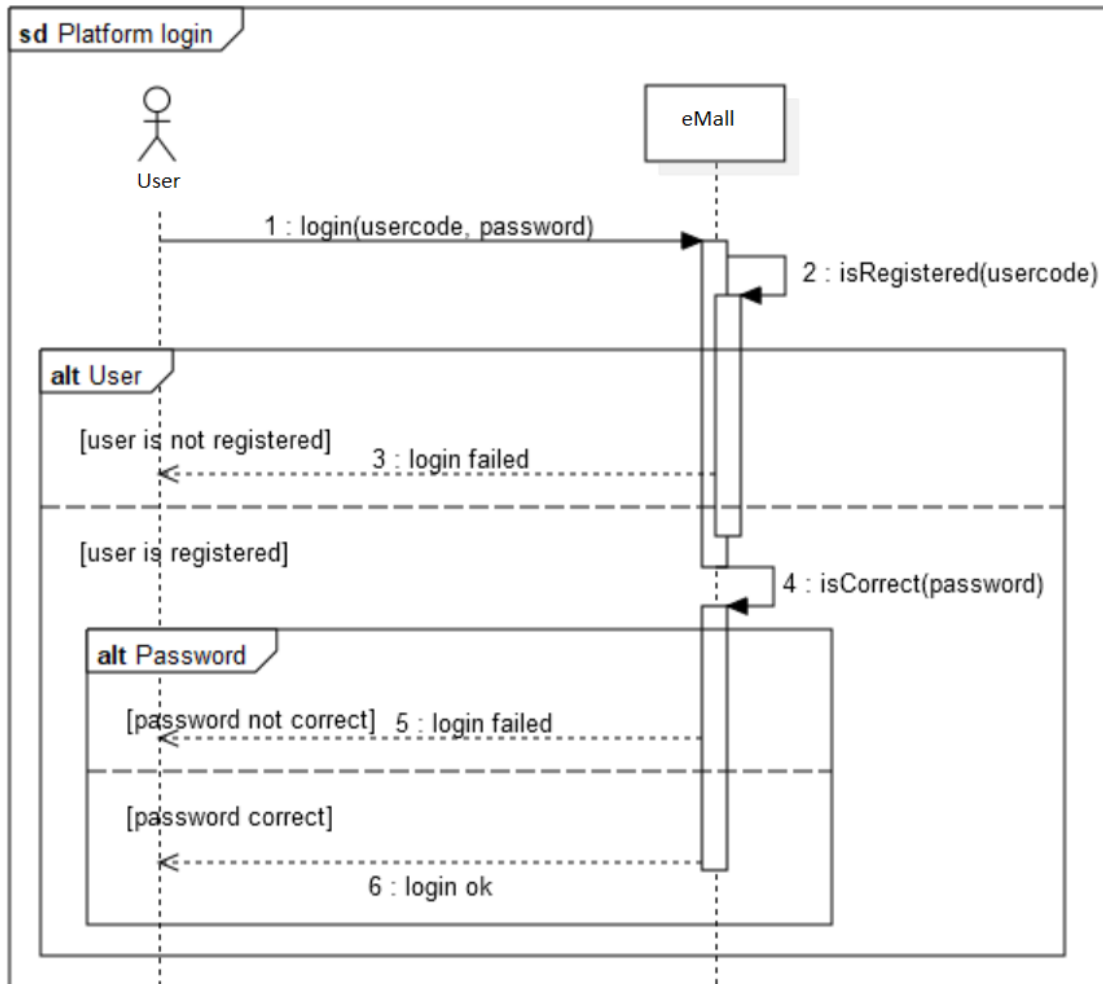


Figure 3.4: Platform Login sequence diagram.

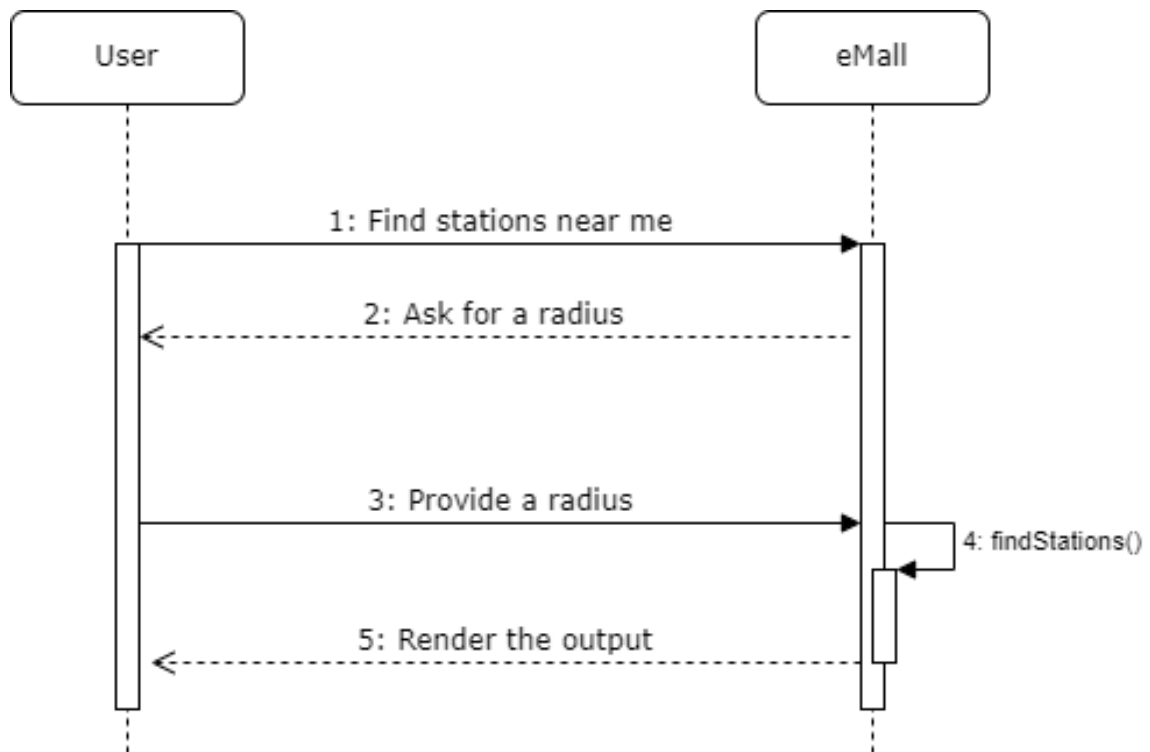


Figure 3.5: Stations Check sequence diagram

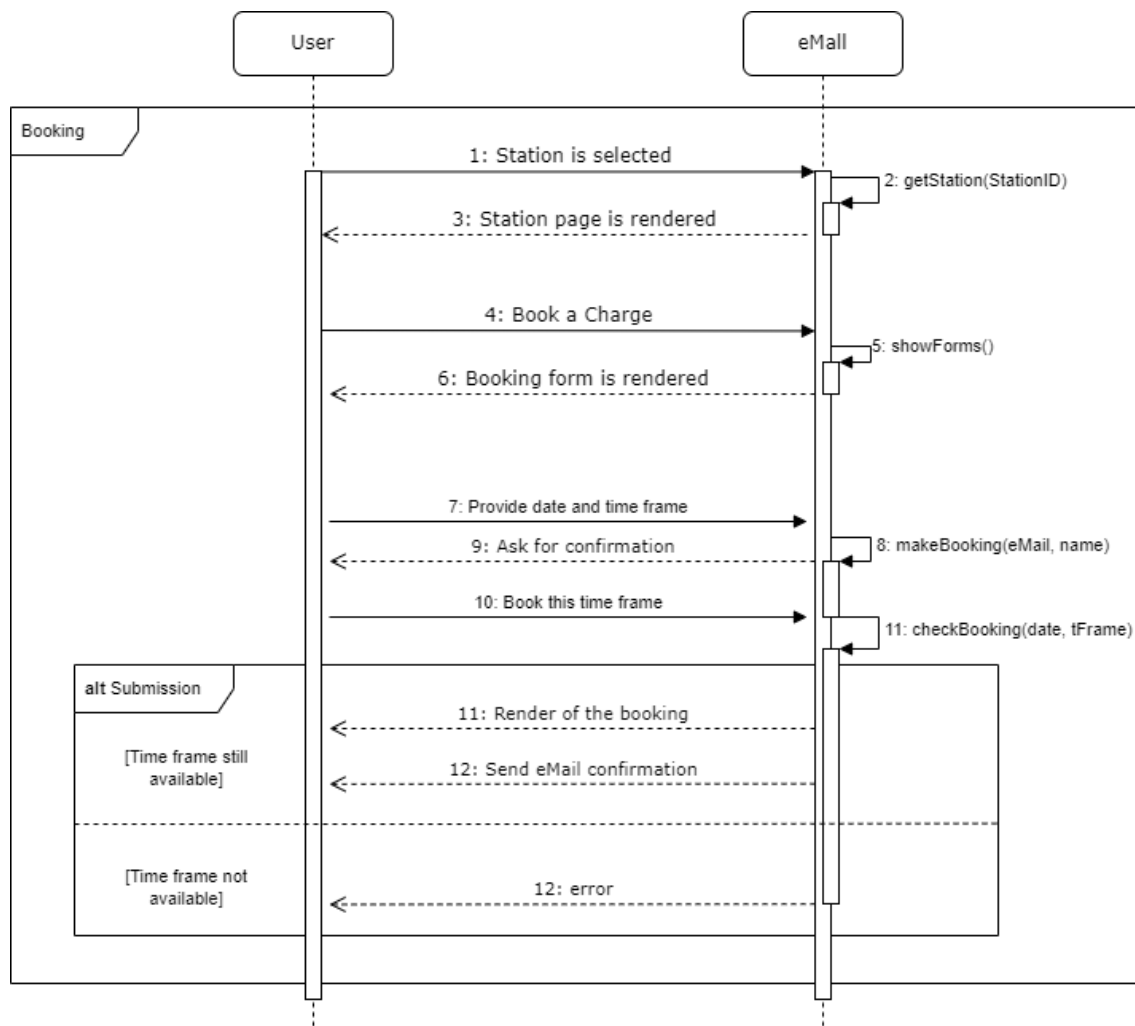


Figure 3.6: Book a Charge sequence diagram

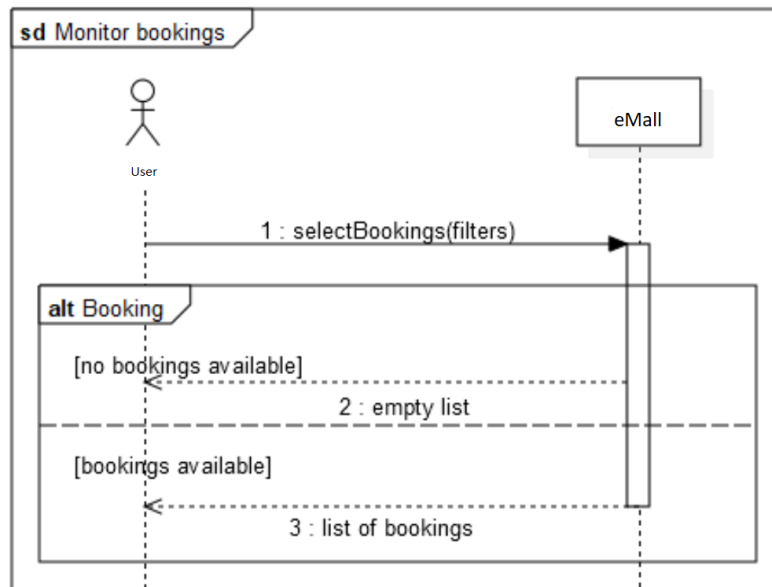


Figure 3.7: Monitor Bookings sequence diagram

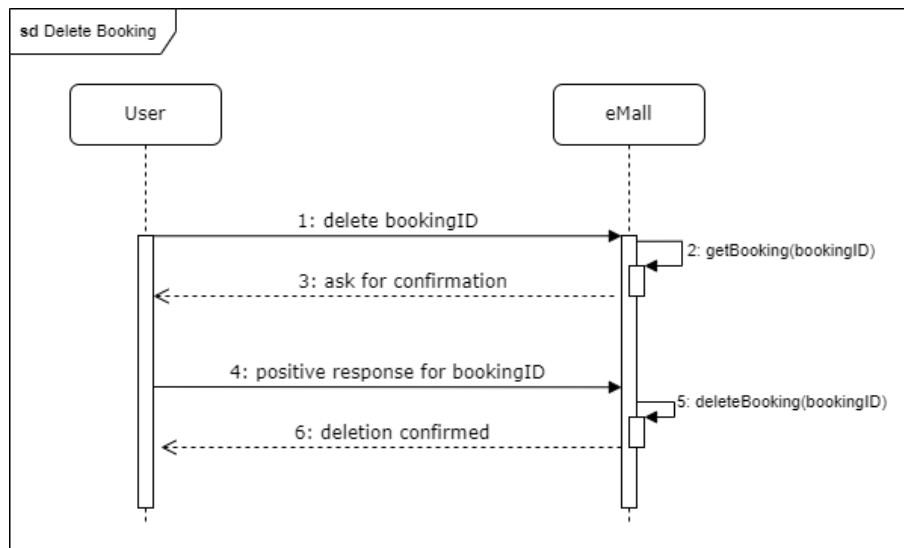


Figure 3.8: Delete a Booking sequence diagram

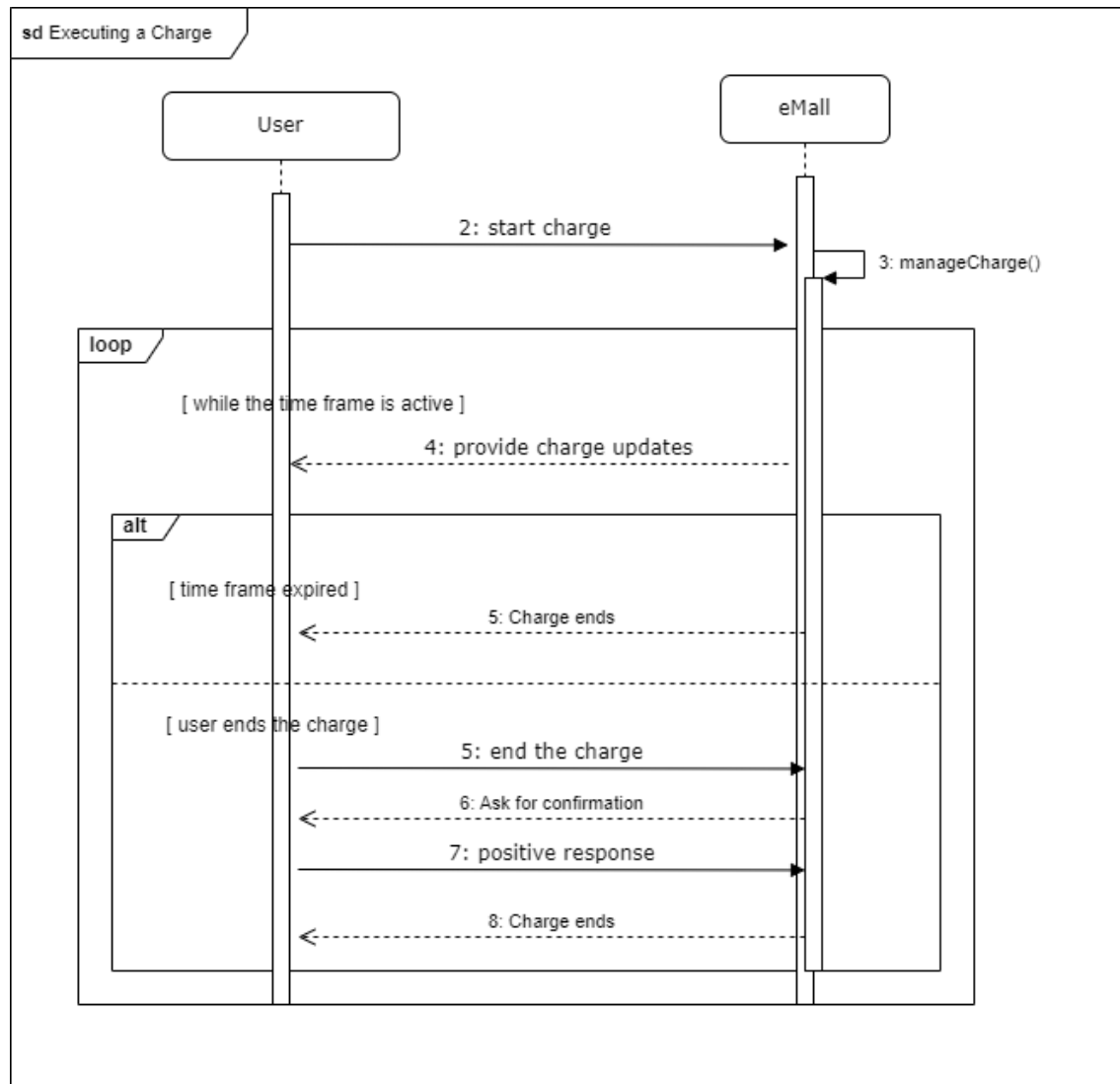


Figure 3.9: Executing a Charge sequence diagram

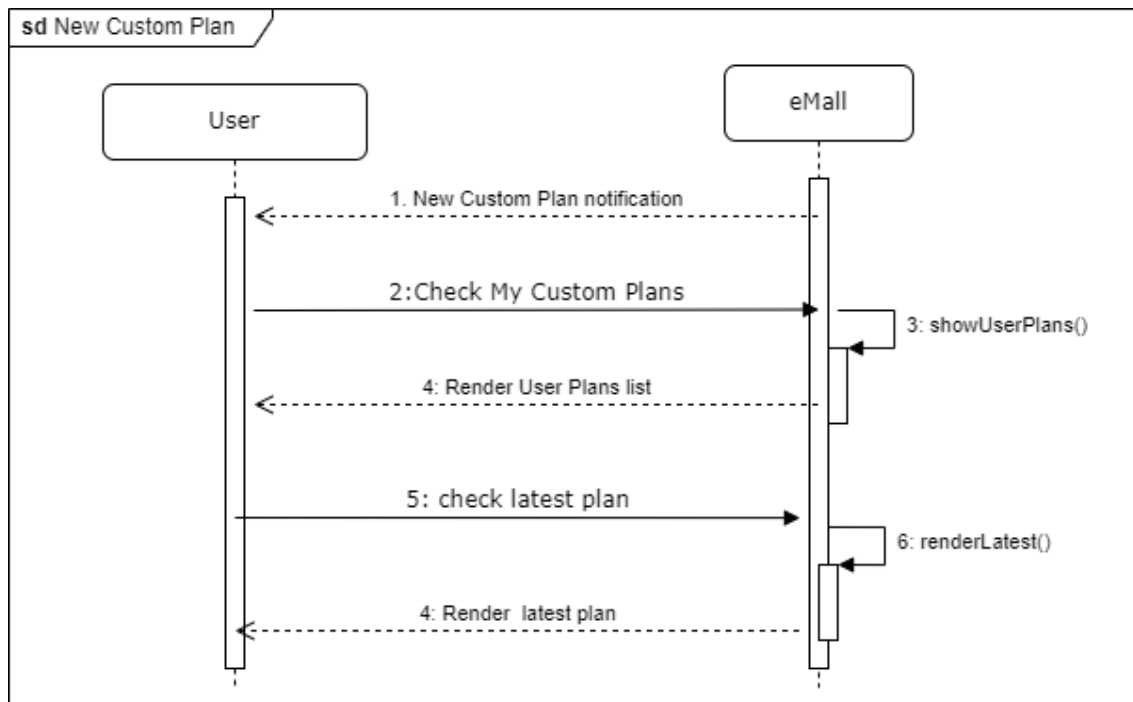


Figure 3.10: New Custom Plan sequence diagram

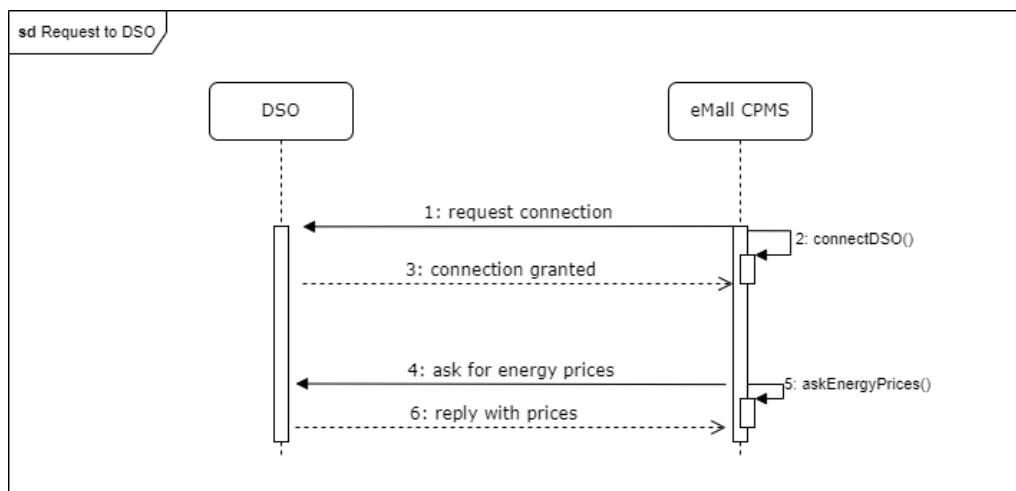


Figure 3.11: Request to DSO sequence diagram

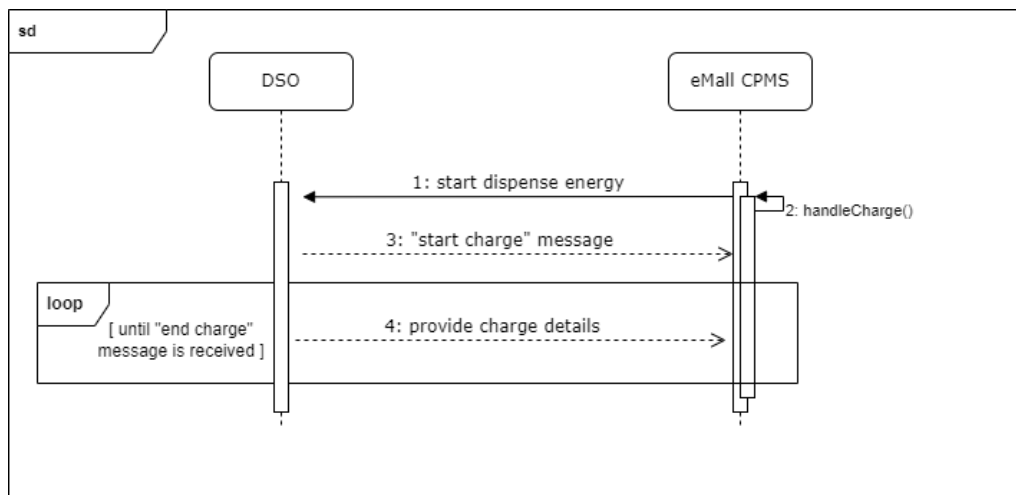


Figure 3.12: Exchange Charge Details to DSO sequence diagram



### **3.3 Performance Requirements**

The e-Mall application must be able to store data for up to 25.000.000 different users and should be able to support up to 5.000.000 simultaneous connections. This takes into consideration the current number of electric vehicles worldwide, as well as the staggering increase in popularity of renewable energy sources.

The e-Mall CPMS module must be able to handle at most 500 simultaneous connections. Of course the maximum number of simultaneous charges is equal to the number of sockets provided by the CPO, we estimate it to be by far lower than 500.

E-Mall eMSP should be able to provide lightweight pages, with a maximum size of 1 MB and a maximum loading time of 5 seconds, in order to overcome possible bottlenecks created by the communication infrastructure.

### **3.4 Design Constraints**

#### **3.4.1 Standards Compliance**

The implementation of the system should follow the requirements stated in this document. It should be well documented in order to facilitate maintenance. Following, the system must manage the procured user data with respect to laws.

#### **3.4.2 Hardware Limitations**

The e-Mall application requires the user to own a device capable of connecting to the Internet which also needs GPS which the system must be able to access. CPOs need to own a desktop computer with the e-Mall eMSP module installed and correctly hooked to the sensors of the station.

Moreover, the communication protocols aforementioned need to be correctly implemented.

### **3.5 Software System Attributes**

#### **3.5.1 Reliability & Availability**

The system should be the most reliable as possible, this choice is dictated by the frenetic usage of the system that, by nature, is designed for a “on the go” usage. As mentioned above, the e-Mall system should be a distributed one, in order to provide data consistency and replication. Distributions also allow lower latency for users dislocated in various parts of Europe (main target of the system). Each component

of the system should be connected to the Internet 24/7, with sufficient bandwidth to serve all the connected users and sufficient storage to handle all the data involved.

### **3.5.2 Security**

All user information needs to be encrypted, this includes username, password, position and any other information that might be sensitive. The system should be protected against any unauthorized access.

### **3.5.3 Maintainability**

The code of the web application should be written in Java to allow for multi-platform support and agile development of the distributed environment. The code should be well commented and being open source to allow external developers a full comprehension of the internal components.

### **3.5.4 Portability**

Considering the nature of a web application, the e-Mall system should be able to be accessed by mobile users or desktop CPOs.

# Chapter 4

## Formal Analysis Using Alloy

### 4.1 Alloy Model

Alloy has been used to describe and formally analyze the relationships between the main entities that compose the e-Mall system.

#### 4.1.1 Model Description

The main concepts that were modelled in alloy are: Users, Bookings, Charging Stations, CPOs and CPMSs and all the entities that are useful to specify them. Then the following constraints were taken into consideration:

- Email, Password and UserCode exist only if a User exists
- Date exists only if a Booking exists
- A TimeSlot exists only if a Booking exists and the start time must be lower than the end time
- A Booking exists only if a CPMS exists
- An Address exists only if a ChargingStation exists
- A ChargingSlot exists only if a ChargingStation exists
- A ChargingStation exists only if a CPO exists and it has to have at least one ChargingSlot
- A CPMS exists only if a CPO exists
- Each User has a unique Email and a unique UserCode
- A User can't have two Bookings for the same Date at the same TimeSlot

- A Booking belongs to a single CPMS and the ChargingSlot it reserves has to be in a ChargingStation owned by the CPO that has the CPMS to which the Booking belongs
- Two Bookings for the same ChargingSlot can't have same Date and overlap in TimeSlots
- All CPMS belong to different CPOs

### 4.1.2 Alloy Code

*//SIGNATURES*

```
sig Email {} { this in User.email}
sig Password {} {this in User.password}
sig UserCode {} {this in User.userCode}

sig User {
    email : one Email,
    password : one Password,
    userCode : one UserCode
}

sig Date{} {this in Booking.date}

sig TimeSlot{
    start : one Int,
    end : one Int
}{start < end and start ≥ 0 and this in Booking.timeSlot}

sig Booking {
    user : one User,
    date : one Date,
    timeSlot : one TimeSlot,
    chargingSlot : one ChargingSlot
}{this in CPMS.bookings}

sig Address {}{this in ChargingStation.address}

sig ChargingSlot{}{this in ChargingStation.chargingSlots}

sig ChargingStation{
    address : one Address,
    chargingSlots : some ChargingSlot
```

```

}{#chargingSlots > 0 and this in CP0.chargingStations}

sig CPMS{
    bookings : set Booking
}{this in CP0.cpms}

sig CP0 {
    chargingStations : some ChargingStation,
    cpms : one CPMS
}

//PREDICATE FOR FACTS

//Predicate for overlapping time slots
pred timesOverlap [ts1, ts2 : TimeSlot] {
    (ts2.start < ts1.start and ts1.start < ts2.end) or
    ( ts1.start < ts2.start and ts2.start < ts1.end) or
    (ts1.start = ts2.start)
}

//FACTS

//USER

//One email per user
fact oneEmailPerUser {
    no disj u1, u2 : User |
        u1.email = u2.email
}

//One usercode per user
fact oneCodePerUser {
    no disj u1, u2 : User |
        u1.userCode = u2.userCode
}

//No overlaps for user
fact userIsNotUbiquitous{
    all u : User |
        no disj b1, b2 : Booking |
            b1.user = u and
            b2.user = u and
            b1.date = b2.date and

```

```

        timesOverlap[b1.timeSlot , b2.timeSlot
        ↪ ]
    }

//BOOKINGS

//The slot of a booking needs to belong to the correct CPO
fact bookingsAreCoherent{
    no b : Booking |
        some cpo : CPO |
            b in cpo.cpms.bookings and
            b.chargingSlot not in cpo.
            ↪ chargingStations.chargingSlots
}

//A booking belongs to a single CPMS
fact {
    no disj cpms1, cpms2 : CPMS |
        one b : Booking |
            b in cpms1.bookings and b in cpms2.
            ↪ bookings
}

//No overlaps on date, timeslot, charging station and charging
    ↪ slot
fact noOverlapsInBooking {
    no disj b1, b2 : Booking |
        b1.date = b2.date and
        timesOverlap[b1.timeSlot, b2.timeSlot] and
        b1.chargingSlot = b2.chargingSlot
}

//CHARGING STATIONS & CHARGING SLOTS

//All stations belong to different CPOs
fact allStationsToDifferentCPO{
    no disj cpo1, cpo2 : CPO |
        some chargingStation : ChargingStation |
            chargingStation in cpo1.
            ↪ chargingStations and
            chargingStation in cpo2.
            ↪ chargingStations
}

```

```

//All charging slots belong to a charging station
fact noSlotWithoutChargingStation{
    all slot : ChargingSlot |
        one cs : ChargingStation |
            slot in cs.chargingSlots
            ↪
}

//Each charging station has a unique address
fact noSameAddress {
    no disj cs1, cs2 : ChargingStation |
        cs1.address = cs2.address
}

//The same slot cannot belong to two different charging
    ↪ stations
fact noSameSlots {
    no disj cs1, cs2 : ChargingStation |
        some slot : ChargingSlot |
            slot in cs1.chargingSlots and slot in
            ↪ cs2.chargingSlots
}

//CPO & CPMS

//All CPMS belong to different CPOs
fact allCPMSToDifferentCPO{
    no disj cpo1, cpo2 : CPO |
        some cp: CPMS |
            cp in cpo1.cpms and
            cp in cpo2.cpms
}

//PREDICATES TO GENERATE WORLDS

//General World
pred generalWorld {
    #ChargingStation > 3
    #CPO > 2
    #Booking > 4
    #User > 1
}

```

```

run generalWorld for 10

//World where booking for same socket in same day has
  ↪ different timeslot
pred noOverbooking {
    #ChargingSlot = 1
    #Booking = 2
    #User = 2
    #Date = 1
}
run noOverbooking for 10

//World where a user can't have the same date for two
  ↪ bookings in the same timeslot
pred noSamePlace {
    #TimeSlot = 1
    #User = 1
    #Booking = 3
}
run noSamePlace for 10

//World where there's a ChargingStation without any booking
pred emptyChargingStation{
    #Booking = 0
    #CPMS = 1
}
run emptyChargingStation for 10

//World where a user has no bookings
pred userWithNoBookings[u :User]{
    no b : Booking | b.user = u
    #Booking > 1
}
run userWithNoBookings for 10

```



## 4.2 Metamodels

In this section, the Metamodels generated to validate some system properties are presented

### 4.2.1 World 1 - General case

This model shows the entirety of the system and all the complex relations among its part. It's useful to understand how the single parts interact with one another.

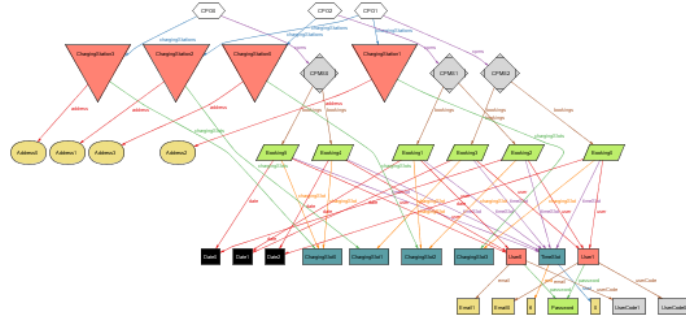


Figure 4.1: General World

### 4.2.2 World 2 - No overlapping bookings for a station

For the model generated by the NoOverbooking predicate, it shows the essential property of guaranteeing that no Charging Slot can be booked for more than a customer at the same time. It is clear by the different time slots generated connected to the same date.



Figure 4.2: No overlapping bookings for a station

### 4.2.3 World 3 - No overlapping bookings for a user

The diagram shows another fundamental property of the system which is that no user can book more than one charge for the same date and time slot. This is made so to prevent unused reservation, so that the system can be as optimized as possible.



Figure 4.3: No overlapping bookings for a user

### 4.2.4 World 4 - Charging station with no bookings

This world shows that a charging station managed by the system developed might not have any bookings. This world was generated to show that the system is compliant even for corner cases such as no bookings for a charging station.

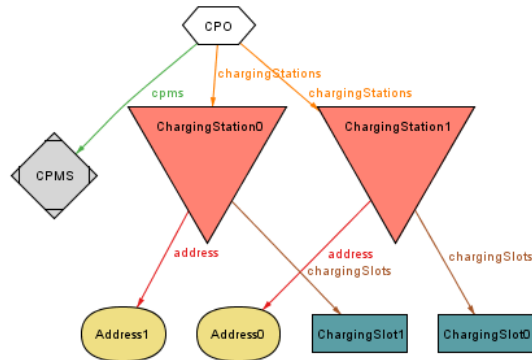


Figure 4.4: World with empty charging station

#### 4.2.5 World 5 - User with no bookings

This world shows that a user might not send any booking request but still be registered in the system. This world was generated to show that the system is compliant even for corner cases such as a registered user that doesn't book any charge.



Figure 4.5: User with no bookings

## 4.3 Result of Predicates

In this section the results of the predicates are shown after running the different worlds of the Alloy model. The result is shown in the Alloy graphical user interface after running the predicates.

<b>Executing "Run generalWorld for 10"</b> Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 48246 vars. 1640 primary vars. 104583 clauses. 659ms. <b>Instance</b> found. Predicate is consistent. 381ms.
<b>Executing "Run noOverbooking for 10"</b> Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 48221 vars. 1640 primary vars. 104541 clauses. 488ms. <b>Instance</b> found. Predicate is consistent. 108ms.
<b>Executing "Run noSamePlace for 10"</b> Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 48165 vars. 1640 primary vars. 104317 clauses. 416ms. <b>Instance</b> found. Predicate is consistent. 57ms.
<b>Executing "Run emptyChargingStation for 10"</b> Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 48109 vars. 1640 primary vars. 104093 clauses. 244ms. <b>Instance</b> found. Predicate is consistent. 23ms.
<b>Executing "Run userWithNoBookings for 10"</b> Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 48379 vars. 1650 primary vars. 104462 clauses. 253ms. <b>Instance</b> found. Predicate is consistent. 56ms.

Figure 4.6: Result of predicates

# Chapter 5

## Effort Spent

### 5.1 Team Effort

Task	Time
Project setup and initial meeting	3h
Introduction	1h
Domain Assumptions, Goals and Requirements	9h
Document Revision	9h

### 5.2 Alessia Abbondanza

Task	Time
Scenarios	1h
Activity Diagrams	3h
Product Functions	3h
Mapping of Requirements	3h
Use Case Description	2h
Sequence diagrams	3h
Design Constraints	2h
Software System Attributes	1h
Formal Analysis using Alloy	6h

### 5.3 Marco Lorenzo Campo

Task	Time
Scenarios	3h
Class Diagrams	3h
Product Functions	2h
External Interface Requirements	4h
Mapping of Requirements	2h
Use Case Description	4h
Sequence diagrams	3h
Performance Requirements	2h
Design Constraints	1h
Software System Attributes	1h

### 5.4 Alessandro De Luca

Task	Time
Scenarios	1h
Product Functions	2h
Mapping of Requirements	2h
Use Case Description	1h
Performance Requirements	1h
Design Constraints	2h
Software System Attributes	1h
Formal Analysis using Alloy	12h