



## eMall – e-Mobility for All

### ABSTRACT

[Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you're ready to add your content, just click here and start typing.]

Sara Limooee  
Parham Ebadi

# Table of Contents

1. INTRODUCTION.....	1
1.1. PURPOSE.....	2
1.1.1. <i>Goals</i> .....	3
1.2. SCOPE.....	2
1.2.1. <i>World phenomena</i> .....	3
1.2.2. <i>Shared phenomena</i> .....	3
1.3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	2
1.3.1. <i>Definitions</i> .....	3
1.3.2. <i>Acronyms</i> .....	3
1.3.3. <i>Abbreviations</i> .....	3
1.4. REVISION HISTORY .....	2
1.5. REFERENCE DOCUMENTS.....	2
1.6. DOCUMENT STRUCTURE .....	2
 2. OVERALL DESCRIPTION.....	 1
2.1 PRODUCT SERVICE .....	2
2.1.1. <i>Scenarios</i> .....	3
2.1.2. <i>CLASS DIAGRAM</i> .....	2
2.1.3. <i>State charts</i> .....	3
2.2. <i>Product Functions</i> .....	3
2.3. USER CHARACTERISTICS.....	2
2.4. <i>Assumptions, Dependencies and Constraints</i> .....	
 3. SPECIFIC REQUIREMENTS .....	 1
3.1. EXTERNAL INTERFACES REQUIREMENTS .....	2
3.1.1. <i>User Interfaces</i> .....	3
3.1.2. <i>HARDWARE INTERFACES</i> .....	2
3.1.3. <i>Software Interfaces</i> .....	3
3.1.4. <i>Communication Interfaces</i> .....	3
3.2. FUNCTIONAL REQUIREMENTS .....	2
3.2.1. <i>Use Case Diagram</i> .....	3
3.2.2. <i>Use Cases</i> .....	3
3.2.3. <i>Sequence Diagrams</i> .....	3
3.2.4. MAPPING ON REQUIREMENTS .....	2
3.3. PERFORMANCE REQUIREMENTS .....	2
3.4. DESIGN CONSTRAINTS.....	2
3.4.1. <i>Standards Compliance</i> .....	3
3.4.2. <i>HARDWARE LIMITATIONS</i> .....	2
3.4.3. <i>ANY OTHER CONSTRAINTS</i> .....	2
3.5. SOFTWARE SYSTEM ATTRIBUTES .....	2
3.5.1. <i>RELIABILITY</i> .....	2
3.5.2. <i>Availability</i> .....	3
3.5.3. <i>SECURITY</i> .....	2
3.5.4. <i>MAINTAINABILITY</i> .....	2
3.5.5. <i>PORTABILITY</i> .....	2
 4. FORMAL ANALYSIS.....	 1
4.1. CODE.....	2
4.2. <i>Result</i> .....	3

4.3. GENERATED INSTANCES.....	2
5. EFFORT SPENT .....	1
6. REFERENCES.....	1

# 1. Introduction

## 1.1. Purpose

In recent years, due to the global warming and the increase in the number of pollution in the air produced by fuel vehicles because of using petrol, gasoline etc, there is a huge need for models of cars which must be less disastrous for the environment. Hybrid electric vehicles (HEVs) and electric vehicles (EVs) use less or even no fuels so they have much less effect on the environment. However, these kinds of cars need to be charged whenever their battery is low. So, there must be some Charging Point Stations so that the EV drivers can charge their car, thus generating new business opportunities such as EV charging business which lets the EV drivers understand what options they have.

These Charging Point Stations are handled by Charging Point Operators (CPOs). While, each CPO has its own IT infrastructure administrated through the so-called Charge Point Management System (CPMS), they can decide to work with different Distribution System Operators (DSOs) which produce and provides energy to charging point stations.

The idea is to develop an electric Mobility Service Provider(eMSP) so that users who are EV drivers can easily make a decision among various charging points and book a place for charging their electric vehicles based on some factors e.g., distance, price, etc.

This document focuses on the Requirements Analysis and Specification Document (RASD) of the system and describes the main goals, the domain assumptions, the scenarios which may happen, the uses cases, the list of functional and non-functional requirements which system should fulfill and finally the diagrams to visualize the interactions between components and performance of the system.

### 1.1.1. Goals

Goals	Descriptions
G1	eMSP allows EV-drivers to search for charging stations nearby, their cost and any special offer.
G2	eMSP allows EV-drivers to book a charge in a specific charging station 15 minutes before connection.
G3	eMSP allows EV-drivers to sort the available charging point stations based on the selected filter.
G4	eMSP allows EV-drivers to rate charging station they received the service from.
G5	eMSP allows EV-drivers to add credit to the wallet inside their account for paying later.
G6	eMSP allows EV drivers to pay for the obtained service either using internal wallet or by credit card.
G7	CPMS allows CPOs to know about the amount of energy to be used for each vehicle.
G8	CPMS allows CPOs to know the “external” status OF charging stations, e.g. location, #of charging sockets available, type, cost
G9	CPMS allow CPOs to monitor the charging process to infer when the battery is full
G10	CPMS allow CPOs to know the “internal” status of a charging station
G11	CPMS allow CPOs to know the current price of energy acquired from the DSOs

## 1.2. Scope

In order to help EV-drivers to find nearby charging stations for their electric car and know about any special offer that either the eMSP or the CPMS (means any offer or suggestions that the CPOs give to the customers), The application has the following parts:

1. EV-driver login to the eMSP system and inserts data about his/her car. According to this information, he can search for charging stations by various filters e.g., distance, price, type of sockets, etc. He can also pay for the obtained service through the interface provided by the eMSP.
2. CPOs log in to the CMPS system to control the general internal and external status of their charging stations, the current price of energy obtained from each DSO that the charging station is working with and to control some other functionalities that can be also controlled by humans manually.
3. eMSP system interacts with the CPMS system to start the charging process, shows the charging process to the end user (EV-driver), giving information about the available charging stations and free sockets of any type of socket (slow, fast, rapid) and their prices and generally any information that must be changed between eMSP and CPMS.

EV-drivers might receive some special offers either from eMSP system for using the application or from CPMSs. Moreover, some suggestions can be given to EV-drivers about when to charge their car based on the previous charging and the distance that the car has passed (this information can be obtained by an API between the CPMS and the navigation system of the car in order to estimate how much distance a car can go at most).

Since this project has been done by a group of two students, the last point mentioned above, the suggestions given by CPMS to the EV-drivers have not been considered in the project.

### 1.2.1. World Phenomena

World Phenomena	Descriptions
WP1	User finds out that his/her car needs charging
WP2	User drives at the station
WP3	User plug in the car
WP4	User shows the QR code of booking confirmation to the station

### 1.2.2. Shared Phenomena

Shared Phenomena	Descriptions	Control
SP1	User searches for nearby charging stations	World
SP2	eMPS shows the nearby stations by default	System
SP3	User sort the available stations	World
SP4	eMPS shows the occupied and free sockets of that station (type, cost, etc.)	System
SP5	eMPS shows the amount of time for the first occupied socket to be freed	System
SP6	User books one of the free sockets	World
SP7	eMPS provides a receipt (QR code) for booking confirmation	System
SP8	eMPS shows the remaining time of charging to the user	System
SP9	Users can monitor the charging process of his/her car	World
SP10	eMPS notifies the user when the charging process is finished	System
SP11	CPMS finish the process of charging****	System
SP12	User pay for the bill	World

### 1.3. Definitions, Acronyms and Abbreviations

#### 1.3.1. Definitions

Definitions	Descriptions
External Status	number of charging sockets available, their type, cost, and estimated amount of time until the first occupied socket is freed
Internal Status	amount of energy available in its batteries, number of vehicles being charged, amount of power absorbed, and remaining time of the charge of each vehicle
Notification	A message shown to the user by the system when he/she must be notified about something (ex: when a new offer is available, or when the charging process starts or finishes)
API	Stands for Application Programming Interface. APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

#### 1.3.2. Acronyms

Acronyms	Descriptions
eMPS	e-Mobility Service Provider
CPO	Charging Point Operator
CPMS	Charge Point Management System
DSO	Distribution System Operators
API	Application Programming Interface



### 1.3.3. Abbreviations

Abbreviations	Descriptions
G	Goal
WP	World Phenomena
SP	System Phenomena
D	Domain Assumption
R	Requirement

### 1.4. Revision History

Version	Date	Modifications
1.0		
2.0		

### 1.5. Reference Documents

- Specification Document: “Assignment RDD AY 2022-2023\_v2.pdf”
- Course slides

### 1.6. Document Structure

#### ▪ Section 1

Contains an overview of the purpose of the project and defines the scope of the system. Describe the specifications such as the definitions, acronyms, abbreviations, revision history, and references. As well as introducing the goals, world and shared phenomena of the application.

#### ▪ Section 2

Defines the main scenarios and explains the main features in the application by class diagram and state charts. Explaining actors of the system in user characteristics. The subsection product function defines the functionalities of the application. Finally, the domain assumptions are defined.

- Section 3

The main part of the project introduces interface requirements such as user interface, hardware interface, software interface, and communication interfaces. Presenting the functional requirements that are shown by use case diagrams and sequence diagrams. In the end, the mapping of requirements to use cases is written.

- Section 4

Analyzing the system using Alloy Language and brief comments for clarifying the Alloy codes.

- Section 5

Amount of time spent by each member of the group for the project.

- Section 6

Contains the references used to make the project done.

## **2. Overall Description**

### **2.1. Product Perspective**

#### **2.1.1. Scenarios**

##### **1- Book a charge:**

Mike is an employee in a big city. Every day he needs to use his electric car to go to his office. he has heard about the new system called eMSP in his town. The new startup provides fantastic offers to customers so He was motivated to use this service to charge his car. He registered himself in the application as a customer. He searches for nearby stations and filters the results by price and distance. The eMSP also shows him the specific types of sockets available in each station. He books a charging socket in the specific charging stations. After the booking process is completed, the eMSP provides an electronic receipt containing a QR code to the user, which is required to start the charging process.

##### **2- Charging process:**

Monica is an old customer of the eMSP system. One day when she feels that her car needs to be charged, she logged in to her account as a customer and she has reserved a place in a charging station near her house. When she arrives there, she parked her car in the reserved area for her and plugged in her car, and uses the digital QR code provided after booking, to start the charging process. The CPMS(Charge Point Management System) automatically scans the QR code and validates her personal and car information and begins to charge the car. The CPMS shows the charging process and the remaining time for the charging to be finished.

### 3- Payment Process:

David is an EV driver. One day he attended one of the charging stations to charge his electronic car. After the charging process of his car, Based on the type of socket and the current price of energy that was used for charging his car, a bill is issued for him by eMSP. So, In order to reach that, first he logs in to his personal account and after the charging process is finished, he will be notified by the eMSP, so He can choose whether to pay it with cash or by credit card. Then he chooses to use his credit card to pay the bill so he clicks on the credit card icon and pays the bill. After ending the payment process, the receipt of payment is sent to him. He can download the file of that receipt via the share button.

Also, after each charge, the system will give some tokens(the amount of tokens is in relation to the amount of money that is paid every time), which can be stored in the virtual wallet and be used as a discount for the next time of charging. They can be also used as a discount in some other markets, hotels, restaurants, etc. which have a contract with our system.

### 4- Making decisions about the way of using energy:

Mattia is a manager of a charging station in a crowded city. As there is hard competition among all CPOs in the city, organizing the price and the energy consumption of the whole station is an important matter in order to increase the efficiency of the station and to get a higher rate from all consumers. Alex is a manager in one DSO, an external 3<sup>rd</sup> party company that provides energy to the charging stations. with the help of an interaction throw specific API between CPOs and DSOs. As usual of every day, Mattia comes to his office and logs in to his specific account which is designed for CPOs managers. On the main page, there are available functions for him. So he can enter the station's status page, where he can monitor the situation of his station and then make decisions about whether to use batteries or get energy from DSO or mix of energy and when to store energy in the batteries of the station.

\*\*\*\*\*

#### 5- Monitoring the external status

Sarah is the head manager of one of the CPMSs in a big city. One of his responsibility is to check the external status of the charging station that is working with his company. In order to achieve that goal, he has to log in to her personal account which is designed for CPMSs managers. On the main page, there is a button named (external status), she clicks it and transferred to the page related to the station's external status. The system has provided different facilities for them to check the external status of a station which contains different things such as charging sockets and their types(fast ,slow ,rapid) and their daily costs. She can easily determine the types of sockets available in the specific CPO. Moreover, she can enter each CPO section and specify the remaining time for each socket to be freed. All this information is being shown to users through CPOs by eMSP.

#### 6- Monitor the internal process:

Alice as an internal manager of CPMS has a lot of duties in the company. One of her main duties is updating the internal status of the CPO who is working with her company. Every day she should log in to her personal area in the application. And then there are some available functions for her to do. She clicks on the internal process button. On the next page there is a screen showing the status of each battery(amount of energy available in them), and immediately can monitor the number of cars being charged in the station. Furthermore, she is able to check the amount of power absorbed by each car in the station and the remaining time of charging of each of them. She should update this information and send them to CPO.

#### 7- Rate process

Daniel is an old user of eMSP and prefer to use it to charge his electronic car. One day when he is driving to his work, he feels that his car needs to be charged. So he login into his personal account on eMSP application. After entering the main page, he searches for nearest station and goes there for charging. After the payment process is finished, he is notified to rate the CPO that he has used. When he clicks on the rate button he is transferred to the next page where he can enter his comment about the service he received, also he can tick (Anonymous) button. In the end, he clicks on the (submit) button and is notified that his comment is posted. Moreover, he is able to see his previous comments on other stations by clicking on (comment) button and he is able to edit them.

### 2.1.2. Class Diagram

### 2.1.3. State Charts

## 2.2. Product Functions

### 2.3. User Characteristics

- EV drivers:  
A person who registers in the eMSP system as a user and has an electric car. They can search for charging stations and any available charge based on different factors e.g., distance, price, and type of sockets. They can also book a charge in a specific charging station 15 minutes before connection and pay for the obtained service through the app after the charging process is finished.
- CPO:  
A person who can control the external and internal status of charging stations through CPMS system and can control some functionalities of the CPMS system. CPOs are operators of the CMPS system who works for the owner company of charging stations and they do not need to register in the system. The only action for authorization is the login.

### 2.4. Assumptions, Dependencies and Constraints

Identifier	Descriptions
D1	The voltage of charging is compatible with the user's car model.
D2	EV driver inserts the car specifications correctly
D3	The charging stations always have either batteries or are supported by DSOs
D4	The locations and information of each station that is inserted into eMSP by CPOs, are observed and correct *****
D5	Each user(CPMS/CPO) of eMSP has a unique ID which occurs for uniform APIs
D6	Energy supply from Energy Company(DSO) is continuous and without problems

D7	The tokens(according to the amount of energy consumption) are transferred to the user's virtual account after each charging***
	The Current price of energy provided by CPMS is correct and updated instantly****
	After the charging process is finished, the user pays the bill first, and then plugs out his/her car
	The eMSP notifies the user exactly when the charging process is finished*****
	The external and internal situations of each station are updated after finishing each charging process at each station****
	The user is able to attend the station in 15 min after booking a charging space
	The user interface and facilities available for CPMSs and users are totally different from each other
	The comments and rates which are given by every EV-driver for each station are correct
	Each EV-driver uses each station's facilities correctly and after finishing the charging process, the EV-driver put the hose back in its space correctly

### 3. Specific Requirements

#### 3.1. External Interface Requirements

##### 3.1.1. User Interfaces

The following mockups show the different main parts of both eMSP and CPMS systems interacting with EV-drivers and CPOs respectively. Complete sections of the systems will be mentioned and discussed in Design Document.

### **3.1.2. Hardware Interfaces**

The hardware that both systems eMSP and CPMS need are:

- EV-drivers have to use a smartphone or web browser where they can search for charging stations and book a charge and make payments.
- We also need some scanners at the charging stations so that the charging process start after the device scans the user booking confirmation.

### **3.1.3. Software Interfaces**

The application uses an external interface:

- MAP API for user and CPO to view the locations
- API for connecting banks' systems

### **3.1.4. Communication Interfaces**

- Communication with DSOs:

CPMSs must be able to communicate with DSOs to get the current price of energy in order to calculate the amount that the EV-driver should pay for the obtained service.

- Communication with CPMSs:

eMSP must be able to communicate with CPMSs to get any required information such as the location of charging stations, the charging process, the amount that must be paid by the EV-driver and etc.

### 3.2. Functional Requirements

Requirement	Descriptions
R1	The eMSP must allow an unregistered user to register
	The eMSP must allow the user to contact the supporting team in case of having a problem with registering or logging the process
	The eMSP must send an error if the user inserts an incorrect form of needed blanks or if he/she is already registered
R2	The eMSP must send an email to the user's email address in order to verify him/her
R3	The eMSP must allow a logged-out user to log in again
	The eMSP must allow users to add a virtual wallet into his/her personal portal in order to enable them to easily pay via that in the next payments
	The eMSP shall allow users to have authority so that they can pay either with their virtual wallet or with another card
R4	The eMSP shall allow a registered user to enter the specifications of their cars
R5	The eMSP must allow the user to search for the nearest charging stations
R6	The eMSP must show the nearest charging stations in their near specific area
	The eMSP must show the related stations to the user according to the user's car specifications
	The eMSP must allow the user to view other driver's comments on one station and its rate
	The eMSP must show the types of each space of the station
	The eMSP must show the free spaces of each station to the user
R7	The eMSP must allow users to sort the shown charging stations by different categories and according to the user's car's specification



	The eMSP must show the 15 min countdown after the booking process
	The eMSP must send a receipt containing a QR code to the user after he/she books a charge space
	The eMSP must provide the direction to the specific charge station via the map
	When the user plugs in his/her car, The CPMS must scan the QR code provided by the user to verify him/her
	The CPMS must start the charging process after the verification
	The eMSP must alert the user that the charging process is started by sending a notification
	The user must be able to see the remaining time of his/her car to be fully charged processed by CPMS through eMSP
	The user must be able to see the current price of energy provided by CPMS
	The eMSP must allow the user to know that his/her car is being charged by the internal batteries of the station or from DSO
	The CPMS must end the charging process when the car is fully charged
	The eMSP must send a notification to the user that the charging process is finished
	The eMSP must allow users to plug out his/her car whenever they wanted
	The CPMS must calculate the price of charging and show it to the user via eMSP
	The eMSP must allow the user to use his/her tokens which were taken before, for payment
	The eMSP must allow the user to use different bank payment gateways to pay the bill
	The CPMS must calculate the amount of token acquired by the user after paying the bill according to the amount of the bill and transfer it to his/her personal account
	The eMSP must send a notification to the user that the payment process was successful

0	The eMSP must allow the user to download or print the payment receipt
	After finishing payment, the eMSP must allow the user to leave a comment about the station and rate it
	The eMSP must allow the user to edit, remove his/her written comments and ratings
	The eMSP must save the user's comment in his/her personal portal
	The eMSP must store the history of previous stations used by the user and the bills
	The CPMS must calculate the value of available tokens and shows the facility which can be used by those tokens

### 3.2.1. Use Case Diagrams

### 3.2.2. Use Cases

Name	Register
Actor	EV-driver
Entry Conditions	EV-driver opens the eMSP application.
Events Flow	<ul style="list-style-type: none"><li>1- EV-driver selects the register button to enter the registration page.</li><li>2- EV-driver fills the required blank fields.</li><li>3- EV-driver selects the “Register” button to submit his/her information.</li><li>4- eMSP sends an email to EV-driver for account confirmation.</li><li>5- EV-driver confirms his/her email.</li><li>6- eMSP allows EV-driver to sign in to the eMSP system.</li></ul>
Exit Conditions	EV-driver is registered in the eMSP system.
Exceptions	<ul style="list-style-type: none"><li>1- EV-driver does not fill all the needed blanks. In this case, eMSP warns him/her to fill all the blanks.</li><li>2- EV-driver does not confirm his/her email.*****</li><li>3- The inserted email address/phone number already exists in the application. In this case, eMSP warns him/her to log in instead of registering.</li><li>4- Inserted password is too weak. In this case, eMSP warns him/her to choose a stronger password.</li></ul>

<b>Name</b>	<b>Login</b>
<b>Actor</b>	EV-driver
<b>Entry Conditions</b>	1- EV-driver opens the eMSP application. 2- EV-driver has already registered in the eMSP application.
<b>Events Flow</b>	1- Ev-driver fills the email address/phone number and password. 2- EV-driver clicks on the “Login” button. 3- EV-driver enters his/her personal account.
<b>Exit Conditions</b>	- eMSP allows EV-driver to log in.
<b>Exceptions</b>	1 EV-driver enters the wrong email or password. In this case, eMSP warns him/her to enter the correct data.

<b>Name</b>	<b>Search for charging stations</b>
<b>Actor</b>	EV-driver
<b>Entry Conditions</b>	1- EV-driver already logged in on eMSP 2- EV-driver is on the main view of his/her personal portal on eMSP
<b>Events Flow</b>	1- Ev-driver clicks on the search button to view the list of charging stations 2- The eMSP system shows the lists of charging stations 3- The user can filter the provided stations based on different factors 4- Ev-driver chooses one station 5- USER views the available types of the sockets in the station and their prices 6- The user books the charge space
<b>Exit Conditions</b>	- eMSP shows the requested information by Ev-driver
<b>Exceptions</b>	- losing internet connection ***** - eMSP Fails to Connect to CPMS

Name	Book a charge in a charging station
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- EV-driver is on the main view of his/her personal portal on eMSP
Events Flow	1- Ev-driver clicks on the search button to view the list of charging stations 2- The eMSP system shows the lists of charging stations 3- User can filter the provided stations based on different factors 4- Ev-driver chooses one station 5- The user views the available types of sockets in the station and their prices 6- The user books the charge space
Exit Conditions	- The eMSP reserve a place for the user in the specific CPO
Exceptions	1- eMSP Fails to Connect to CPMS 2- eMSP fails to save the booking process 3- The user loses his/her internet connection while the booking process 4- The user chooses the wrong type of socket

Name	Start charging
Actor	EV-driver
Entry Conditions	1- EV-driver already logged in on eMSP 2- Ev-driver has booked a charging place 3- Ev-driver got the receipt containing the QR code
Events Flow	1-
Exit Conditions	
Exceptions	1- The user plugs in his/her car in the wrong socket 2- Ev-driver plugs his/her car 15 min after booking 3- The user shows a different QR code 4- CPMS fails to verify the user through the API sent/to eMSP*****

<b>Name</b>	<b>Pay for obtained service</b>
<b>Actor</b>	EV-driver
<b>Entry Conditions</b>	1- EV-driver already logged in on eMSP 2- The charging process is finished
<b>Events Flow</b>	1- The eMSP sends a notification to the user to inform him/her that the charging process is finished 2- eMSP shows the price which is calculated by CPMS 3- EV-driver clicks on the pay button 4- EV-driver can choose between paying via his/her virtual wallet or another wallet 5- If the user has an offer code, he/she can insert it into offer code button 6- CPMS calculates the new price according to inserted offer code 7- EV-driver pay the bill 8- eMSP completes the payment process 9- eMSP sends the payment receipt to the user
<b>Exit Conditions</b>	- eMSP does the payment process
<b>Exceptions</b>	1- The user loses internet connection after clicking pay button 2- EV-driver closes the application

<b>Name</b>	<b>Rate a charging station</b>
<b>Actor</b>	EV-driver
<b>Entry Conditions</b>	1- EV-driver already logged in on eMSP 2- The payment process is finished
<b>Events Flow</b>	1- The eMSP sends a message to the user to invite him/her to rate the station which was used by them 2- EV-driver clicks on the rating button 3- EV-driver writes a comment about the service he/she has gotten and rates the station 4- EV-driver clicks on the submit button 5- eMSP sends a notification to the user that his/her comment is submitted
<b>Exit Conditions</b>	- The rating process is finished

<b>Exceptions</b>	<ul style="list-style-type: none"> <li>1- The user closes the application</li> <li>2- The eMSP is unable to submit the rating due to weak internet connection or losing connection*****</li> <li>3- The user leaves the blanks and try to submit, in this case, the eMSP warns him/her to fill all blanks</li> </ul>
-------------------	--

<b>Name</b>	<b>Add money in wallet</b>
<b>Actor</b>	EV-driver
<b>Entry Conditions</b>	1- EV-driver already logged in on eMSP
<b>Events Flow</b>	<ul style="list-style-type: none"> <li>1- The user clicks on the (add money) button.</li> <li>2- The user writes the amount money he/she wants to add to her/his wallet</li> <li>3- The user chooses the cart that he/she wants to use to get money from</li> <li>4- The user clicks on the (next) button</li> <li>5- The eMSP transfers the user to another page to enter the information of the selected cart</li> <li>6- The user enters the cart's information</li> <li>7- The user clicks on the (pay) button</li> <li>8- The eMSP makes an API to CPMS in order to complete the transfer process</li> <li>9- The eMSP gives the payment receipt to the user</li> </ul>
<b>Exit Conditions</b>	- CPMS does the adding money process
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>1- The user has another bank account which is not available in the list of banks</li> <li>2- The user does not enter money amount or does not select the bank, in this case, the eMSP warn him/her to fill the needed blanks</li> </ul>

<b>Name</b>	<b>Login</b>
<b>Actor</b>	CPO
<b>Entry Conditions</b>	1- CPO manager opens the eMSP application. 2- CPO manager has already registered in the eMSP application.
<b>Events Flow</b>	1- CPO manager fills the email address/phone number and password. 2- CPO manager clicks on the “Login” button. 3- CPO manager enters his/her personal account.
<b>Exit Conditions</b>	- eMSP allows the CPO manager to log in.
<b>Exceptions</b>	1- CPO manager enters the wrong email or password. In this case, eMSP warns him/her to enter the correct data.

<b>Name</b>	<b>View charging stations’ locations</b>
<b>Actor</b>	CPO
<b>Entry Conditions</b>	1- CPO manager opens the eMSP application. 2- CPO manager has already logged in the eMSP application.
<b>Events Flow</b>	1-
<b>Exit Conditions</b>	
<b>Exceptions</b>	1-



<b>Name</b>	<b>View Internal Status of a charging station</b>
<b>Actor</b>	CPO
<b>Entry Conditions</b>	1- CPO manager has already logged in to the eMSP application.
<b>Events Flow</b>	1- CPO manager clicks on the (internal status) button 2- In the (internal status) page, the CPO manager clicks on the batteries energies button to view the status of energies available in the batteries 3- In the (internal status) page, he/she is able to see the number of vehicles that are being charged at that moment 4- CPO manager clicks on one of the car's status 5- The eMSP transfer CPO manager to the page in which he/she can visit the amount of power absorbed and time left to the end of the charge through it's CPMS
<b>Exit Conditions</b>	- The eMSP shows the requested information to the CPO manager
<b>Exceptions</b>	1- The eMSP fails to make an API to CPMS in order to get the required information

<b>Name</b>	<b>View External Status of a charging station</b>
<b>Actor</b>	CPO
<b>Entry Conditions</b>	- CPO manager has already logged in the eMSP application
<b>Events Flow</b>	1- CPO manager clicks on the (external status) button 2- On the (external page), he/she can view the number of charging sockets available and their types(slow, fast, rapid) 3- CPO manager clicks on one of the sockets icons 4- eMSP transfers the CPO manager to the specific socket's page 5- In the specific page of the socket, CPO manager is able to view the cost of that socket and in case of being full, the estimated time to be freed which is calculated by CPMS
<b>Exit Conditions</b>	- The eMSP shows the requested information to the CPO manager
<b>Exceptions</b>	1- The eMSP fails to make an API to CPMS in order to get required information

### **3.2.3. Sequence Diagrams**

### **3.2.4. Mapping on Requirements**

## **3.3. Performance Requirements**

- The eMSP system must be able to serve a great number of users simultaneously.
- The eMSP system must guarantee correct responses.
- The eMSP system must be able to send a response to a query less than 5 seconds since it has been received.
- The eMSP system must be available 99% of the time.
- The CPMS system must be able to serve a great number of users simultaneously.
- The CPMS system must guarantee correct responses.
- The CPMS system must be able to send a response to a query less than 5 seconds since it has been received.
- The CPMS system must be available 99% of the time.

## **3.4. Design Constraints**

### **3.4.1. Standard Constraints**

- The eMSP system requires EV-drivers' permissions to retrieve their position.
- Both systems must maintain the data retrieved from the EV-drivers with respect to the privacy laws.

### **3.4.2. Hardware Constraints**

- The web browser or the smartphone which user is using must have the ability of connecting to the internet and using GPS services.

### **3.4.3. Any other Constraints**

- The estimations that the CPMS system makes about the remaining time of full charge must be accurate 99% of the time.

## **3.5. Software System Attributes**

### **3.5.1. Reliability**

Both eMSP and CPMS systems must be able to run continuously without any interrupts. The reliability of the systems depends on the services of the systems and should be up 99% of the time. This means the MTTR or downtime should be 3.65 days per year. In order to guarantee this time of downtime, the systems must have an appropriate infrastructure with a full backup system located in a different office that replicates the core services for covering the general failure of the main systems.

### **3.5.2. Availability**

The two systems do not relate to the emergency; thus, we don't need high availability. Moreover, the service is not fully automated because has to rely on the policymakers that follow office hours.

### **3.5.3. Security**

For security issues and to avoid any problems, we need to encrypt the stored data and passwords of each actor of the two systems, EV-drivers and CPOs. Also, if CPMS wants to give some suggestions to EV-drivers based on the information and data it receives through APIs from the navigation system of the car, this information must be encrypted and then stored in the DBMS. Therefore, a layer of protection must be implemented to keep users' personal data safe.

### **3.5.4. Maintainability**

The software must be written in Python and codes must be written with good standards and in OO (object-oriented) structure so that in case of any occurrence of problem or bug, it can be fixed easily and without requiring to change a big portion of the code. Moreover, any bug in one component must have not interrupted with the functionality of other components.

### **3.5.5. Portability**

The software must be designed simply and implemented on different platforms. The software Run in different platforms must support Android and iOS operating systems for mobile devices, as well as a Web application for use simply in other platforms.

## **4. Formal Analysis Using Alloy**

### **4.1. Code**

### **4.2. Results**

### **4.3. Generated**

## **5. Effort Spent**

## **6. Refernces**

- Specification Document: “Assignment RDD AY 2022-2023\_v2.pdf”
- Course slides