Politecnico di Milano
School of Industrial and Information Engineering

Software Engineering 2 Project

e-Mobility for All.
Requirement Analysis and Specification Document

Authors:
Enrico Brunetti
Matteo Gionfriddo

Academic Year 2022/2023

Milano, 23/12/2022

Version 1.1

Contents

List of Figures

List of Tables

1 Introduction

1.1 Scope

eMall is a system meant to provide services for charging electric vehicles and managing charging stations. This is meant both for normal people (i.e. end users) and for CPO workers. The application domain concerns different types of world phenomena, shared phenomena and machine phenomena.

We can identify the followings world phenomena:

- [W1] User connects his electric vehicle to the charge socket.

- [W2] DSOs provide energy to their associated charging stations.

- [W3] Batteries, if available and enabled, store energy from DSOs and provide it to the charging station.

- [W4] Charging sockets are supplied directly by DSO through the network and/or by batteries of the charging station.

the following shared phenomena controlled by the world and observed by the machine:

- [S1] A user books a charge in a charging station.

- [S2] A user enables a charging column.

- [S3] CPOs decides to take energy from DSOs and/or from batteries, if available.

and the following shared phenomena controlled by the machine and observed by the world:

- [S4] Location, external and internal status of the charging station are visible.

- [S5] The system checks the booking ID and enables the charging operation.

- [S6] Vehicle charging is started.

- [S7] Vehicle charging is stopped and a notification is sent.

- [S8] The system elaborates the payment of the charging service.

1.2   Purpose

The system is characterized by the following goals:

- [G1] Users must know position, prices and special offers of charging stations.

- [G2] Users must be able to complete the whole process of charging for their electric vehicles.

- [G3] CPOs workers must be able to manage charging stations associated to their CPO.

- [G4] CPOs must be able to communicate with DSOs in order to provide energy furniture.

In the following table is shown which world and shared phenoma are involved for each goal.

	W1	W2	W3	W4	S1	S2	S3	S4	S5	S6	S7	S8
G1					X			X				
G2	X	X	X	X	X	X			X	X	X	X
G3		X	X	X			X	X				
G4		X					X					

Table 1.1: Correspondence between goals and world and shared phenoma

1.3   Definitions, acronyms and abbreviations

- *eMSP*: e-Mobility Service Provider.

- *CPO*: Charging Point Operator.

- *CPOW*: Charging Point Operator Worker.

- *CPMS*: Charge Point Management System.

- *DSO*: Distribution System Operator.

1.4   Revision history

- 21/12/2022 Version 1.0

- 23/12/2022 Version 1.1:

1.5   Reference documents

- Assignment Document "Assignment RDD AY 2022-2023_v3.pdf".

1.6 Document structure

The document consists of 6 different chapters:

Chapter 1: in this first chapter the application domain is presented starting with the scope, which includes world and shared phenomena that the application can observe and/or control. Furthermore, in the purpose section, the main goals are presented and at the end, some basic information is provided in order to help in the understanding of the whole document.

Chapter 2: the second chapter starts with an overall description of the whole application, made by the definition of the most important possible scenarios which will involve the different types of users and their relationships with the eMall system. Some UML diagrams, i.e. class diagram and state diagrams, are here provided to help to understand which are the different entities of the system and how they should work together in order to provide all required functionalities. The chapter contains also an exhaustive explanation of the main product functions, a description of the different actors, and a list of all needed domain assumptions.

Chapter 3: the third chapter is the heart of the RASD document. First of all external interface and functional requirements are defined and then the second ones are mapped into goals that satisfy and domain assumptions that exploit. Subsequently, the behavior of the system is described by use cases, which are enriched with use cases and sequence UML diagrams. The chapter then ends with performance requirements, design constraints, and software system attributes.

Chapter 4: the fourth is the last effective chapter of the RASD document. It contains a formal analysis of the problem presented using the alloy modeling language. It is especially used to formally verify the consistency of functional requirements.

Chapter 5: in this chapter, the effort spent by each team member is briefly presented in order to provide an overview of how the work has been done.

Chapter 6: the last chapter contains the references used during the developing process of the document.

2 Overall Description

2.1 Product perspective

This section is dedicated to the identification of the most important scenarios and to a first description of the whole system through class and state diagrams.

2.1.1 Scenarios

For all scenarios described below we assume that the system provides a mobile application for end users and a web interface for CPOWs.

Unregistered end-user

A possible end-user has just bought his brand new Tesla Model X and has seen some interesting ads from a famous eMSP about the new application eMall, which will allow him to know where to charge his vehicle and manage the whole process. He decides to download the app and sign-up, filling in all the requested fields and starting using the app. He decides also to insert a picture of his driving license and to register his car inserting dates about the model and the plate. After that, he will include also all information about his debit card so, when he will decide to book a recharge everything will be ready.

Unregistered CPOW

A new CPOW has just been hired and on his first day of work, he needs to register to the eMall web service in order to manage the CPMS of the CPO he is working for. During the sign-up process, he has to insert, in addition to all requested fields, also a special code provided by his employer to register him as a CPOW of that particular CPO.

End-user wants to book a charge

An end-user decides to open eMall app in order to see if there are new offers in the charging stations of his city and discovers that the one of the shopping mall has a discount of 25%. Tomorrow he will finish working at 6:30 pm and he needs to buy also a pair of shoes. So he decides to take advantage also of the special offer and clicks on the book button of the charging station on the map, selects his car and a payment method, and books a fast charging socket between 7 pm and 8 pm tomorrow.

End-user charges his vehicle

An end user arrives at the charging station where he has booked a medium-speed charging socket, presses the unlock button from the app, connects the socket to his car, and starts the charging process. Now he goes to get his hair cut and while he was looking at his new hairstyle in front of the mirror he receives a notification from eMall app which says that the charging process is finished. So he comes back to the charging station, presses the disconnect button on the app, and disconnects the charging socket. At this point, he receives a notification from his bank which says that 20 euros have been charged to his debit card.

CPOW registers a newly opened charging station

A CPOW has just received the assignment to manage through the CPMS a new charging station that will be inaugurated today in the main square of the city, that can be accessed only by electric vehicles. Since all connections between the station and the systems have been already configured by technicians he only has to add it on his web interface. He presses the *configure new station* button and fills all requested fields taking special care about the unique physical id identifier of the station that will allow him to manage it from his CPMS. At this point, the charging station configuration is ended and the management can start to work properly.

CPOW temporarily disable batteries due to maintenance

This night 4 batteries of charging station 1046 have to be removed following the regular maintenance schedule and a CPOW that manages the CPMS related to that charging station has to do some operations before the end of his turn at 6 pm. First of all, he selects that specific charging station on his web interface clicks the battery management button, and selects the 4 interested batteries. At this point, he selects programmed disabling and sets the batteries off from 9 pm of that evening. Since the recharging station has 2 more batteries he sets up to take 25% of energy from batteries and 75% directly from DSO during the charging processes of that night. The day after when the CPOW starts his turn at 9 am he receives a notification that the maintenance has ended and batteries have been correctly reconnected. So he re-enables them and set the charging station to take again 100% of energy from batteries.

CPOW shows to his manager dates about a charging station

Yesterday a new charging station has been opened in the parking of the city shopping center and this morning the manager knocks on the door of the office of the CPOW who manage, trough the CMPS, that specific station, in order to see if there are already some users. The CPOW opens his web interface, selects charging station 1050, and clicks on show statistics. The charging station is composed of 6 charging sockets, 3 at fast-speed and 3 at medium-speed, and 10 batteries. Currently, 2 cars are charging: the first one is a Honda-e, connected to socket 1 (fast-speed charge), that has absorbed 19/35.5kwh of power and will take 12 minutes to complete the charging process. The second one is a Hyundai Kona Electric, which is connected to socket 4 (medium-speed charge), that has absorbed 10/64kwh of power and will take 55 minutes to complete the charging process. The CPOW and the manager see also that 12 other recharges have been booked from now to midnight, 7 on fast-speed sockets and 5 on medium-speed ones. Actually only one of the 10 batteries is at 53% of his charge and the remaining 9 are all full. The manager is pretty satisfied since it is only the second day since the station has opened and he can see encouraging results. So he invites the CPOW for a coffee break and they go to the coffee machine together.

CPOW manually decides where to get energy

A CPO company made a deal with a DSO company and if they will purchase energy only from them for one month they will reduce the price making it the most convenient for the CPO. So all CPOWs of the interested CPO have been instructed to proceed in this way. We took as an example one of them. This specific CPOW decides to have a look at the different offers from all available DSO and he notices that actually the one of the deal isn't the cheaper, so the system won't dynamically decide to acquire energy from that specific company. So, through the web interface of the CMPS, he switches off the auto-deciding acquisition for all the charging stations under his control and set the specific DSO as the only supplier. He also sets all charging stations to use only energy from the batteries until they are empty trying to save some money using before all the already purchased energy that is available.

2.1.2 Class diagram

The UML class diagram which describes the domain application is presented in figure 3.8. The User class is abstract and both EndUser and CPOW classes inherit from it since are the two different types of users of the whole system. Each EndUser is registered to at least one eMSP, owns vehicles and can book recharges which refer to them. Each recharge has a status and happens in a charging socket of a charging station related to one of the eMSPs that the user is registered to. Charging stations are composed of charging columns which are in turn composed of charging sockets and optionally also of batteries. There can be discounts associated to them. Each charging station refers to one or more eMSPs. Each CPO administrates one and only one CPMS that manages one or more charging stations and can use DSOs to purchase energy forniture. There is a provisional class that is EnergyForniture, because at this stage of the project the best way to keep track of energy furniture is to separate them and locate different bills in different objects. This is meant in order to have a detailed report of the precise amount of energy purchased, its price and its destination since each energy forniture can refer to one/more batteries or directly to one/more charging sockets.

2.1.3 State charts

Charging process

In the state chart displayed in figure 2.2 the whole charging process is presented. This chart starts describing the booking process and all the requirements that have to be satisfied in order to make a correct booking of a charge (i.e. vehicle selection). For simplicity, we assume that the user is already logged in to the application and that the *select charging station* operation is meant to include also the type of socket if more options are available (i.e. fast charge socket). After the charge is booked, when the user inserts the plug of his vehicle into the socket the system enters the *Waiting charge completed* state, which is composed of two different states that describe the effective charging process. When the plug is removed the whole process ends. With this last action, we assume also that the payment is done at the moment of the plug removal since the found availability has been already checked during the booking phase.

Charging station energy supply

In the state chart displayed in figure 2.3 is described the state machine related to the energy supply types of a charging station. This representation has to be assumed to be applied independently to all the charging sockets of that specific charging station and the selection between batteries and network can be made manually by CPMS operators or dynamically by the system.
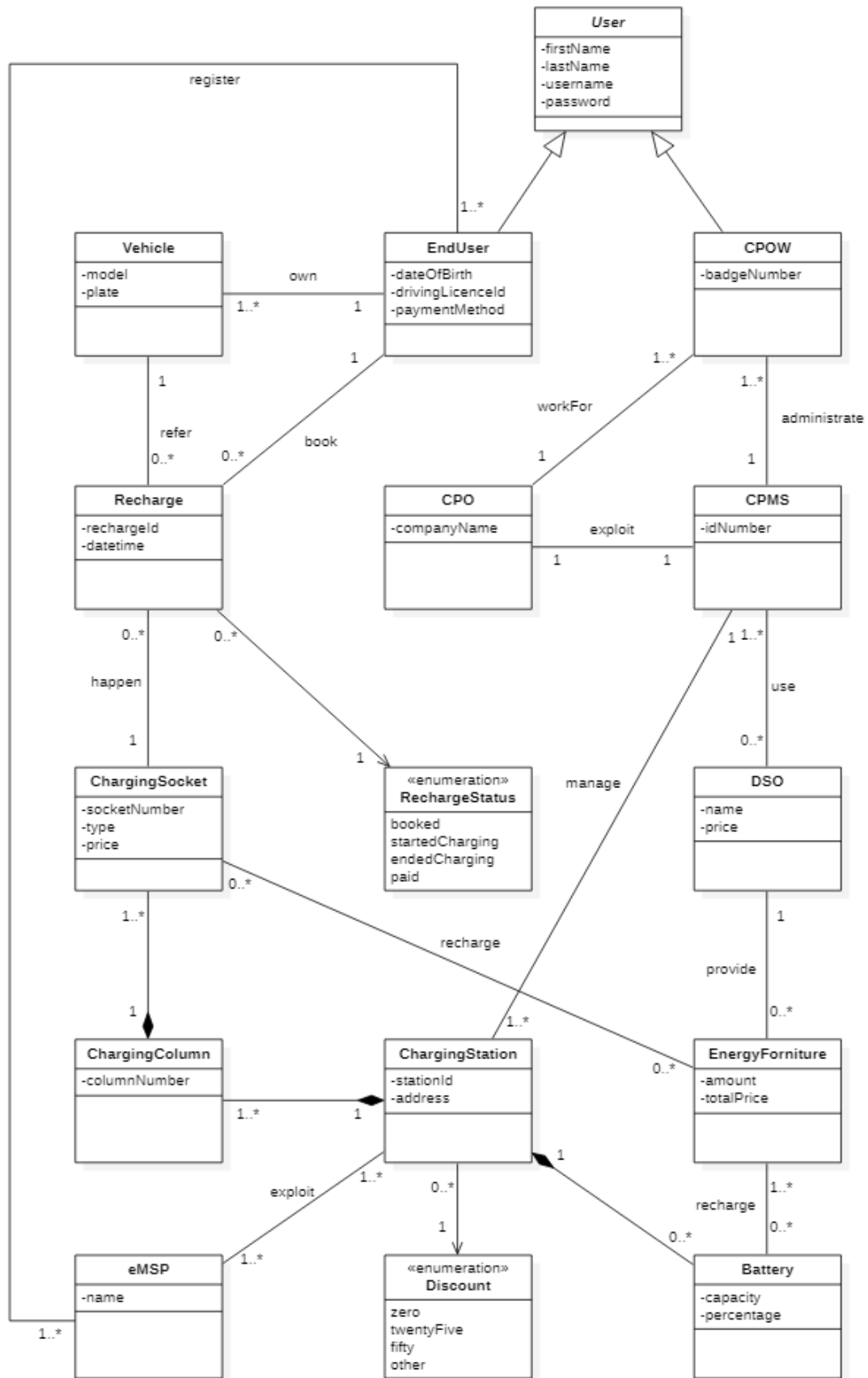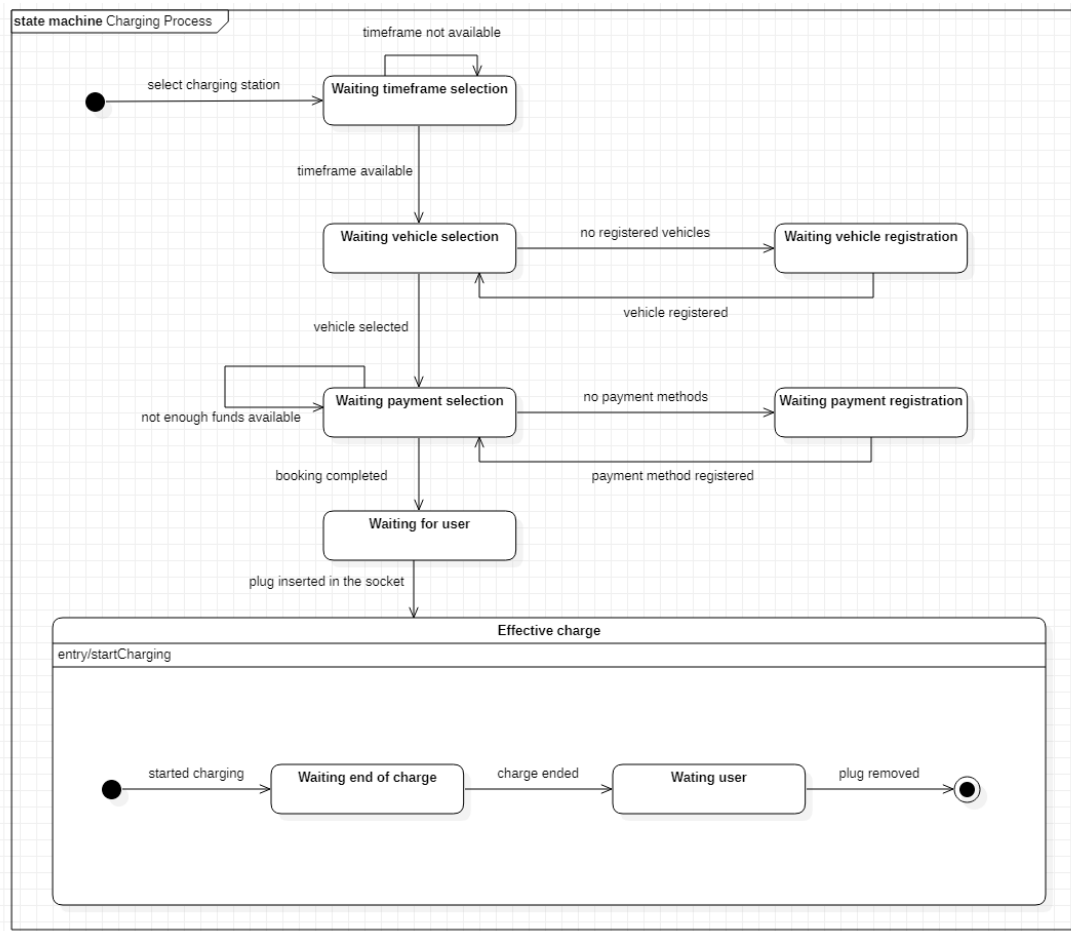
Figure 2.1: UML Class Diagram
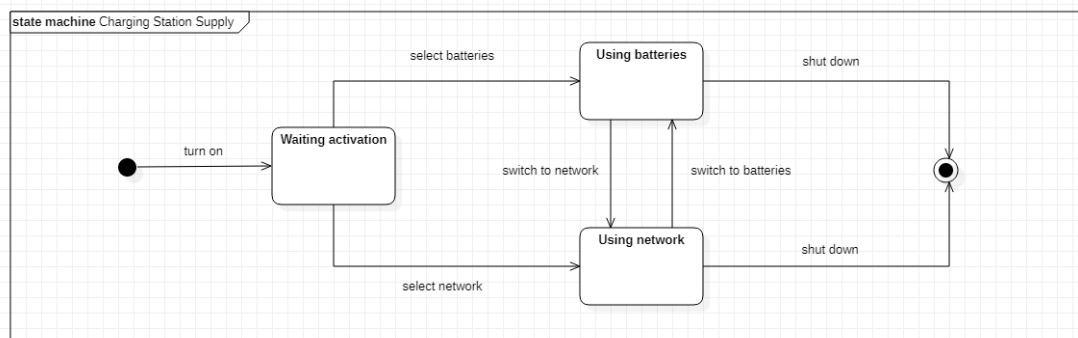
Figure 2.2: UML State Diagram for a whole charging process

Figure 2.3: UML State Diagram for Charging Station energy supply

Other

Other objects are not represented in UML state charts since they don't change state at all during the system activities or the changes are too trivial to be represented in a diagram.

2.2 Product functions

In this section the most important functions that the eMall app will offer are presented divided by the type of user that will exploit them.

2.2.1 Unregistered user

Application exploring

When a user in not logged-in into the application the system will show a map containing the nearest charging stations and the related prices. This information can be accessed also in a list format where also some filters are available, but any operation (e.g. book a recharge) can't be performed until the login/registration.

End user registration

The system allows unregistered users to register and become effective end users of the application. They have to provide their first name, last name, username, password, date of birth and driving license ID. All these fields are mandatory in order to complete successfully the registration process. Optionally a user will be able to register a car and a payment method during the registration process, but this operation can also be performed later.

CPOW registration

The system allows newly hired CPOWs to set up their accounts in order to access the CPMS management web interface. They have to provide their first name, last name, username, password, and most important thing, their badge number. Also in this case, all fields are mandatory in order to complete successfully the registration process, otherwise, it will fail.

2.2.2 Authenticated end user

Here the focus is on the purpose of eMall on the end user side. In particular, we have two main functionalities.

Charge booking

The booking of a charge is the most important function that coincides with the main purpose of the whole system on the end user side. When a user is logged in, the system shows a map with all the nearest charging stations and lets the user navigate through them and sees current prices for different types of sockets of the different stations. Furthermore, the system will inform the user about special discounts and offers.

Once the user has selected a charging station and a date the system will show him all the available time frames and will allow completing the booking only if both a vehicle and a payment method with enough funds have been correctly selected. In all the other cases the operation couldn't be completed correctly.

Charging process

The system must manage also the whole charging process. So, when a user presses the charge button on the app and connects his car to the booked charging socket, the system will start dispensing energy to the car. All the process will be monitored by the system and the app will display to the user the remaining time and will send a notification to him once the process will be ended. At this point, the system will charge the import of the whole recharge on the payment method of the user.

2.2.3 Authenticated CPO worker

Now let's put focus on the CPO side.

Selecting from which DSO acquire energy

In normal execution, DSOs are selected dynamically by the system, trying to exploit the best quality/spent agreement based on prices and dates of different companies.

But can happen that a CPOW needs to manage the selection through his web interface. When he enters the related section, the system will show him a list of all the charging stations and for each one of them a list of DSOs available in that specific geographical area and their prices, with the possibility to turn off the auto-deciding process for each one of the stations and manually select a new DSO.

Charging station energy management

Also charging station use of energy in normal execution is managed dynamically by the system. In this case, the focus is to use batteries in order to exploit price swings in trying to reach the best level of revenue (e.g. when the energy price is low the system will use energy directly from the DSO to charge vehicles and recharge also batteries. On the other hand when the energy price increase over a certain threshold the system will, first of all, consume all energy in batteries bought at lower prices before acquiring new).

Also in this case a CPOW can, always through his web interface, go to the related section. The system will show him a list of all the charging stations and for each one of them, there is the possibility to set the price threshold or to directly manually manage the use of batteries or directly of the network.

Show charging station status

The system will also allow CPOWs to know both the internal and external status of each charging station. When a CPOW access the related section on the web interface a list of all charging stations related to his CPO is shown. By clicking on one of them the station structure is shown (e.g. number of sockets and their types). Also current status is visible, so the list of both currently happening charges and booked charges are shown, with all the informations related to the interested vehicles.

2.3 User characteristics

The system has the following different types of users:

- *Unlogged End User*: an unlogged user of the system can see charging stations near his position on the map and their prices but he can't do any type of operation (i.e. booking a charge). He can change his status by registering to the service or logging in as *End User*.

- *End User*: is the main user of the system that can register vehicles, add payment methods, exploit special offers, and book and complete vehicle charges.

- *Unlogged CPOW*: an unlogged CPOW can only see the registration/log-in interface on the web interface but won't be able to perform any kind of operation. He can change his status by registering or logging in as a CPOW.

- *CPOW*: an operator which is related to a *CPMS* through which he can manage different charging stations (i.e. deciding to supply sockets from batteries or directly from the network), see *DSO* offers and manually decide from which of them purchase energy supplies.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain assumptions

In the following rows domain assumptions are presented:

- [D1] Each CPO is associated with one and only one CPMS.

- [D2] All charging stations have been already correctly added in the right coordinates to third party map used.

- [D3] An End User can't disconnect the plug during the charging process.

- [D4] A unique identifier has been associated physically to every charging station.

- [D5] CPOs must manage their charging station only through his related CPMS.

- [D6] Energy can't be supplied by different DSOs at the same time to the same charging station.

- [D7] Each charging socket can't be supplied by both batteries and the network at the same time.

- [D8] Every end user make charges at the charging station with the specific vehicle he has booked for.

- [D9] The networks that allows the energy furniture from DSOs to charging sockets or batteries must work properly.

- [D10] The system exploits APIs to retrieve information about energy prices from different DSOs.

- [D11] Charging sockets are equipped with an *RFID* receiver in order to be unlocked.

2.4.2 Dependencies and constraints

Some third-party applications and dates from public institutions are necessary to make all functionalities of the system work properly.

- *Vehicle registering check*: check if inserted car model and plate exists.

- *Vehicle model information get*: get dates of all the electric vehicle models in commerce in order to know in advance which is the maximum battery capacity for each booked charge.

- *Payment method check*: check if the payment method added by the user works properly. It will be exploited also before every charge starts in order to ensure that on the payment method there are enough funds to complete a full charge.

- *Map service*: exploit a map service to show a map with all charging stations in the application.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

eMall provides two different types of user interfaces, one for each type of user. In detail, end users exploit provided services through a mobile application, absolutely necessary to unlock charging sockets and start the charging process. On the other hand, CPOWs, which are meant like normal office workers, exploit their managing functionalities through a web application. Both interfaces have to satisfy accessibility requirements and, in particular, the mobile application must be user-friendly. The goal is e-mobility for all, so also older people with low technological skills must be able to perform all required operations to book and complete a charge.

3.1.2 Hardware Interfaces

Since eMall is a software system, no specific hardware interfaces are provided to users. But clearly, end users will interact with charging station hardware (i.e. RFID receivers of charging sockets) that are already provided by domain assumptions. So the communication hardware of the mobile phone on which the mobile up runs is exploited.

3.1.3 Software Interfaces

Since no other applications need to retrieve any kind of data from eMall there are no software interfaces provided. But, like for Hardware Interfaces, the system exploits third-party software as reported in the dependencies section and also other APIs for information retrieval (e.g. DSOs prices) as declared in the domain assumption section.

3.1.4 Communication Interfaces

The whole eMall service is based on communication between different actors. Both the mobile app and CPMS communicate with the central eMall server through uniform internal APIs exploiting HTTPS protocol. In either the two subsystems type of communication is bidirectional, they need to send and receive data from the central server. For example, the mobile app must get updated availabilities of free charging slots but also update the central server when a booking is made. On the other side, also CPMS must always get updated information from the server, but also set, for example, a new DSO for a certain charging station.

Also communication with all the charging stations is necessary. In detail, the main server communicates with charging stations always through internal uniform APIs. Furthermore, RFID technology is exploited to allow communication between the mobile app and the charging socket

(i.e. for unlocking the charging socket). In particular, mobile phones communicate with the RFID receiver of the sockets.

All other types of communication, like the one with all third-party software presented in the dependencies section, happen through API, as the retrieval of DSOs prices.

3.2 Functional Requirements

- [R1] The system must allow unregistered/registered users to see a map of available charging stations and their prices and offers.

- [R2] The system must allow unregistered/registered users to see a list of charging stations and filter on it (e.g. offers, prices and positions).

- [R3] The system must allow unregistered users to register as end user or CPOW (if they have a badge id).

- [R4] The system must verify if the all data inserted by users are correct (e.g. personal data, vehicle and payment method information, charging station ID).

- [R5] The system must allow registered end users or CPOW to log in through their username and password.

- [R6] The system must allow registered end users to associate vehicles to their account and keep track of them.

- [R7] The system must allow registered end users to associate payment methods to their account.

- [R8] The system must allow registered end users to view the available time-slots for a certain type of socket of a certain charging station in a specific day.

- [R9] The system must allow registered end users to book a charge in a specific charging station.

- [R10] The system must allow end users to unlock a charging socket if they have booked it and it's the correct time-slot.

- [R11] The system must let a charge start if a vehicle is correctly connected and the charging socket is unlocked.

- [R12] The system must show to end users the charging status (e.g. remaining time to complete charge).

- [R13] The system must notify the end-user when the charge is complete.

- [R14] When a charging process is correctly finished the system must charge on the end user selected payment system the correct import.

- [R15] The system must allow CPOW to register a new charging station through its physical id.

- [R16] The system must be able to dynamically select if using batteries, directly the network or a mix of the two for each charging station.

- [R17] The system must allow CPOWs to manually select if using batteries, directly the network or a mix of the two for each charging station associated to them.

- [R18] The system must be able to dynamically select which DSO use to provide energy to a specific charging station.

- [R19] The system must allow CPOWs to manually select which DSO use to provide energy to a specific charging station associated to them.

- [R20] The system must be able to dynamically decides if a certain energy furniture is destinated to a battery or directly to sockets of charging stations .

- [R21] The system must allow CPOWs to manually decides if a certain energy furniture is destinated to a battery or directly to sockets of charging stations associated to them.

- [R22] The system must keep track of the current status of each charging process (e.g. booked, startedCharging, etc.).

- [R23] The system must keep track of the structure of each charging station (e.g. number of columns, number and type of sockets).

- [R24] The system must allow CPOWs to view the structure of each charging station associated to them.

- [R25] The system must keep track for each charging station of all bookings related to it.

- [R26] The system must allow CPOWs to view all the bookings of a certain charging station associated to them and their status.

- [R27] The system must allow CPOWs to set prices and special offers for a certain charging station associated to him.

In the following table is shown which Domain Assumptions and Requirements are involved for each goal.

	Domain Assumptions	Requirements
G1	D2	R1 R2 R8
G2	D3 D8 D9 D11	R3 R4 R5 R6 R7 R9 R10 R11 R12 R13 R14
G3	D1 D4 D5 D6 D7 D11	R5 R15 R16 R17 R20 R21 R22 R23 R24 R25 R26 R27
G4	D9 D10	R5 R18 R19

Table 3.1: Correspondence between Domain Assumptions and Requirements and Goals

3.2.1 Use cases

In this section the various use cases present in the use case diagram and derived from the scenarios previously described are illustrated in detail.

1. Registration

Actors: Non registered End-User or CPOW

Entry condition:

- The End-User/CPOW does not have an account and has the application/web interface opened

Flow of events:

- The End-User/CPOW clicks on the "Login or Register" button.

- The End-User/CPOW clicks on the "Register" button.

- The End-User/CPOW provides name, surname, date of birth, email address, all information about his debit card and all information about his electric vehicles that he wants to register/the associated ID of the CMPS where he works and a special code provided by his employer.

- The End-User/CPOW provides username and password.

- The End-User/CPOW clicks on the "confirm" button.

Exit conditions:

- A successful message is sent to the User which is redirected to the main page of the account

Exceptions:

- The personal information are not correct. The system shows an error message and asks the End-User to re-write the incorrect ones.

- Debit Card information are not correct. The system shows an error message and asks the End-User to re-write the incorrect ones.

- Electric vehicle information are not correct. The system shows an error message and asks the End-User to re-write the incorrect ones.

- The CMPS ID is incorrect or doesn't exists. The system shows an error message and asks the CPOW to re-write it.

- The username was already taken by another user. The system shows an error message and asks the End-User to choose another username.

- The End-User interrupts the registration process clicking to "exit" button. He's redirected to the main page

2. Log in

Actors: Registered End-User or CPOW

Entry condition:

- The End-User/CPOW has the application/web service opened

Flow of events:

- The End-User/CPOW clicks on the "Login or Register" button.

- The End-User/CPOW inserts username and password.

- The End-User/CPOW clicks on the "Login" button.

Exit conditions:

- The End-User/CPOW is redirected to the main page of the account

Exceptions:

- Username and/or Password are/is incorrect. The system shows an error message and asks the End-User to re-write it.

3. Non Registered End-User charging stations map view

Actors: Non registered End-User

Entry condition:

- The End-User does not have an account and has the application opened

Flow of events:

- The End-User clicks on the "Browse Map" button.

- A map with all nearby charging station each with a price review is displayed.

- The End-User clicks on a charging station in the map.

- A short description of that charging station is displayed (including price and special offers). account.

Exit conditions:

- A summary of that charging station is displayed (including price and special offers). If the End-User tries to book a charge with the "book a charge" button the system asks him to login or register.

Exceptions:

- No exceptions for this particular use case

4. End-User charge booking

Actors: End-User

Entry condition:

- Registered End-User opened the app and logged in.

Flow of events:

- The End-User clicks on "browse map" button.

- A map with all nearby charging station each with a price review is displayed.

- The End-User clicks on a charging station in the map.

- A short description of that charging station is displayed (including price and special offers).

- The End-User clicks on "book a charge" button.

- The End-User selects a Day and a type of a socket.

- The End-User selects one of the free time-slots available.

- The End-User selects an electric vehicle associated to his account.

- The End-User selects the amount of charge in percentage.

- A summary of the booking is displayed.

- The End-User clicks on the "Confirm" button.

Exit conditions:

- A successful message which presents the summary of the booking with in particular the number of associated charging column, number of associated socket and the booking id is sent to the End-User.

Exceptions:

- There are no socket of that type available for that day. The system shows an error message and asks the End-User to change the date.

- The End-User interrupts the booking process clicking to "exit" button. He's redirected to the charging stations map.

5. End-User benefits from a special offer from the list panel

Actors: End-User

Entry condition:

- Registered End-User opened the app and logged in.

Flow of events:

- The End-User clicks on "browse map" button.

- The End-User clicks on "list charging stations" button.

- The End-User selects a Day, a type of a socket, a price range and an order.

- All charging stations with a socket type of the selected one available in the selected Day and with a price included in the price range selected are displayed in the right order.

- The End-User selects a certain charging station.

- The End-User selects one of the free time-slots available.

- The End-User selects an electric vehicle associated to his account.

- The End-User selects the amount of charge in percentage.

- A summary of the booking is displayed.

- The End-User clicks on the "Confirm" button.

Exit conditions:

- A successful message which presents the summary of the booking with in particular the number of associated charging column, number of associated socket and the booking id is sent to the End-User.

Exceptions:

- There are no charging stations that satisfies the searching conditions. The search result is empty.

- The End-User interrupts the booking process clicking to "exit" button. He's redirected to the main charging stations list page

6. End-User vehicle charging

Actor: End-User

Entry condition:

- The End-User has previously booked a charge and now is at the charging station near the associated charging column with his vehicle. He has opened the application and logged in.

Flow of events:

- The End-User clicks on "my Bookings" button.

- All available charge bookings are displayed.

- The End-User selects one of those

- The End-User clicks on "unlock charging column" button.

- A successful message is displayed and the associated socket of the booked charging column became available.

- The End-User connects his vehicle to the socket.

- The charging process starts automatically and the application displays progressively the amount of charge done.

- When the booked amount of charge is reached the charging process stops and the application displays a message.

- The End-User disconnects the vehicle from the socket.

- The used socket of the charging column is blocked.

- The End-User clicks on "end charge" button.

Exit conditions:

- The system elaborates the payment and displays a successful message to the End-User.

Exceptions:

- The End-User has arrived too late and the booking has been expired. The system shows an error and deletes the booking.

- The End-User has arrived too early respects to the time-slot of the booking and is not able to click on "unlock charging column" button.

- The charging process cannot be doing correctly. The system shows an error message, displays a list of possible solutions to solve the problem.

- The payment has not been successful. The system shows an error message indicating that the balance is still to be paid and displays a list of possible solutions to solve the problem.

7. CPO charging station registration

Actors: Registered CPOW

Entry condition:

- Registered CPOW opened the app and logged in.

Flow of events:

- The CPOW clicks on the "Charging Stations" button.

- The CPOW clicks on the "New Charging Station" button.

- The CPOW provides charging station name, position and physical ID.

- The CPOW clicks on the "confirm" button.

Exit conditions:

- A successful message is sent to the CPOW

Exceptions:

- Charging station information are incorrect or it doesn't exists. The system shows an error message and asks the CPOW to re-write it.

- The CPOW interrupts the registration process clicking to "exit" button. He's redirected to the main page

8. CPO energy acquiring from DSO

Actor: Registered CPOW

Entry condition:

- Registered CPOW opened the app, logged in.

Flow of events:

- The CPOW clicks on "View DSOs Offers" button.

- A list of all DSOs offers is displayed.

- The CPOW clicks on a certain DSO offer.

- A page with all information about that offer is displayed.

- The CPOW clicks on "Acquire energy from this DSO".

- The CPOW clicks on "Confirm" button.

Exit conditions:

- The system displays a successful message to the CPOW.

Exceptions:

- Energy can't be acquired by that DSO for the selected charging station. The system shows an error message to the CPOW and displays a list of possible solutions to solve the problem.

- The CPOW interrupts the process clicking to "exit" button. He's redirected to the main page of the account.

9. CPO batteries management

Actor: Registered CPOW

Entry condition:

- Registered CPOW opened the app, logged in.

Flow of events:

- The CPOW clicks on "Charging Stations" button.

- A list of all associated charging stations is displayed.

- The CPOW clicks on one of those.

- A page with all features of the selected charging station is displayed (e.g "internal" and "external" status, all bookings associated).

- The CPOW clicks on "Batteries Management".

- A list of all batteries associated to that charging station is displayed.

- The CPOW clicks on one of those.

- A page with all features about that battery is displayed.

- The CPOW clicks on "Store energy" button.

Exit conditions:

- The system starts the energy storing process and then shows a successful message to the CPOW.

Exceptions:

- There are no charging stations available. The systems shows a message which says that the list is empty.

- The are no available batteries. The systems shows a message which says that the list is empty.

- The are no available DSO agreements for the selected charging station. The systems shows an error message.

- The energy storing process doesn't work properly. The system stops it, shows an error message to the CPOW and displays a list of possible solutions to solve the problem.

10. CPO energy supply

Actor: Registered CPOW

Entry condition:

- Registered CPOW opened the app, logged in.

Flow of events:

- The CPOW clicks on "Charging Stations" button.

- A list of all associated charging stations is displayed.

- The CPOW clicks on one of those.

- A page with all features of the selected charging station is displayed (e.g "internal" and "external" status, all bookings associated).

- The CPOW clicks on "Energy Furniture" button.

- A list of all DSOs services purchased and of all charged batteries associated is available with a brief summary of their main features for each of them are displayed.

- The CPOW clicks on one of the DSOs services.

- The CPOW inserts the amount of energy in percentage.

- The CPOW clicks on "Confirm" button.

- The CPOW is redirected to the previous list.

- The CPOW clicks on one of the batteries.

- The CPOW inserts the amount of energy in percentage.

- The CPOW clicks on "Confirm" button.

- The CPOW is redirected to the previous list.

- The CPOW clicks on "Continue" button.

- A summary of the total energy supply is displayed.

- The CPOW clicks on "Confirm" button.

Exit conditions:

- The system starts the energy transfer process and then shows a successful message to the CPOW.

Exceptions:

- There are no charging stations available. The systems shows a message which says that the list is empty.

- There are no DSO agreements and/or charged batteries available. The systems shows a message which says that the list is empty..

- The energy transfer process doesn't work properly. The system stops it, shows an error message to the CPOW and displays a list of possible solutions to solve the problem.

- The CPOW interrupts the process clicking to "exit" button. He's redirected to the main page of the account.

11. CPOW set a special offer

Actor: Registered CPOW

Entry condition:

- Registered CPOW opened the app, logged in.

Flow of events:

- The CPOW clicks on "Charging Stations" button.

- A list of all associated charging stations is displayed.

- The CPOW clicks on one of those.

- A page with all features of the selected charging station is displayed (e.g "internal" and "external" status, all bookings associated).

- The CPOW clicks on "Set Special Offer".

- The CPOW selects a type of socket, the amount of discount, a start and an end date.

- The CPOW clicks on "Confirm" button.

Exit conditions:

- The system shows a successful message to the CPOW.

Exceptions:

- There are no charging stations available. The systems shows a message which says that the list is empty.

- The are no available DSO agreements for the selected charging station. The systems shows an error message.

- The CPOW interrupts the process clicking to "exit" button. He's redirected to the main page of the account.

In the following table is shown which goals are involved in every use case.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
G1	X	X	X		X						
G2	X	X		X	X	X					
G3	X	X					X		X	X	X
G4	X	X						X			

Table 3.2: Correspondence between use cases and goals

In the following table is shown which requirements are involved in every use case.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
R1	X	X	X								
R2	X	X			X						
R3	X										
R4	X										
R5	X	X									
R6	X	X									
R7	X	X									
R8	X	X	X		X						
R9	X	X		X	X						
R10	X	X		X	X	X					
R11	X	X		X	X	X					
R12	X	X		X	X	X					
R13	X	X		X	X	X					
R14	X	X		X	X	X					
R15	X	X					X				
R16*											
R17	X	X					X	X	X	X	
R18*											
R19	X	X					X	X		X	
R20*											
R21	X	X					X	X	X	X	
R22*											
R23*											
R24	X	X					X				
R25*											
R26	X	X					X				
R27	X	X					X				X

Table 3.3: Correspondence between use cases and requirements

*These requirements refers to the functions done automatically by the system. For this reason there are no use case related to them.

3.2.2 Use cases diagrams

In figure 3.1, 3.2 and 3.3 are reported the UML Use Case Diagrams of the Visitor, the End-User and the CPOW respectively.

3.2.3 Sequence diagrams

In figure 3.4, 3.5, 3.6, 3.7 and 3.8 are reported the UML Sequence Diagrams for the most interesting Use Cases.
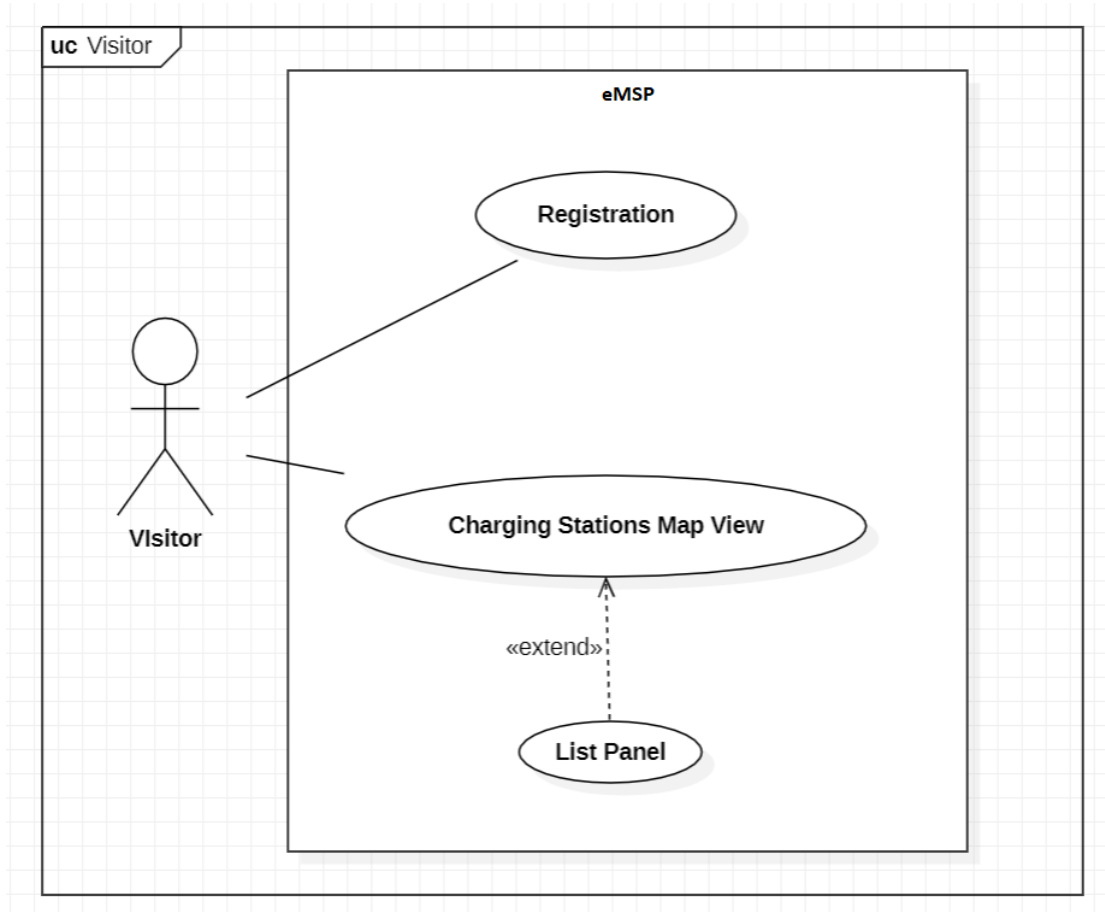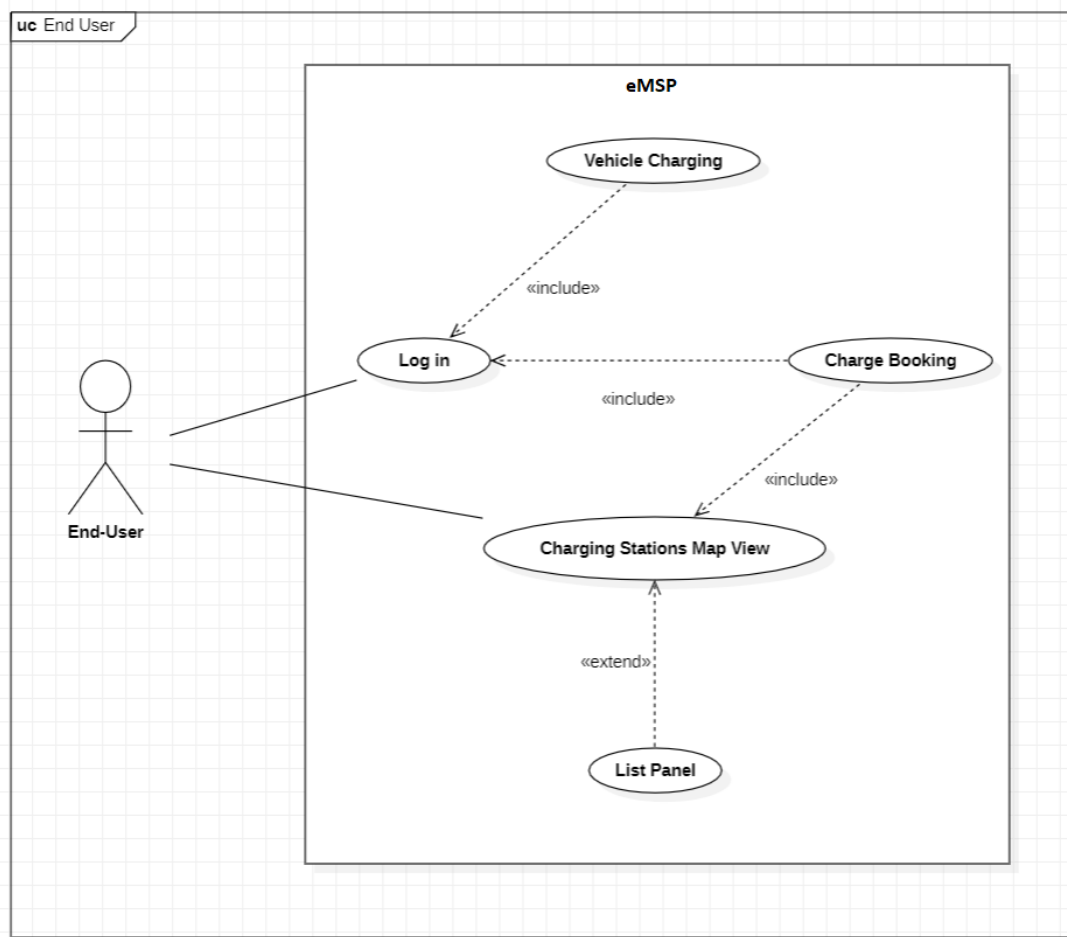
Figure 3.1: Visitor UML Use Case Diagram

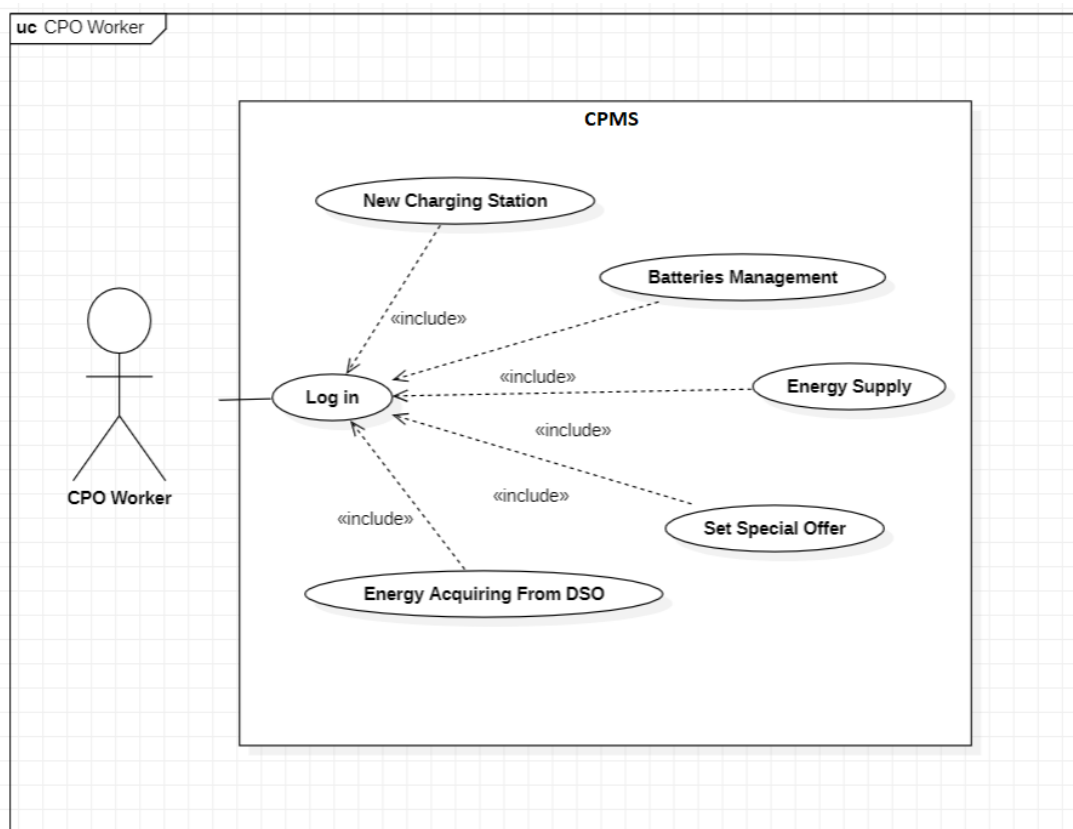Figure 3.2: End-User UML Use Case Diagram

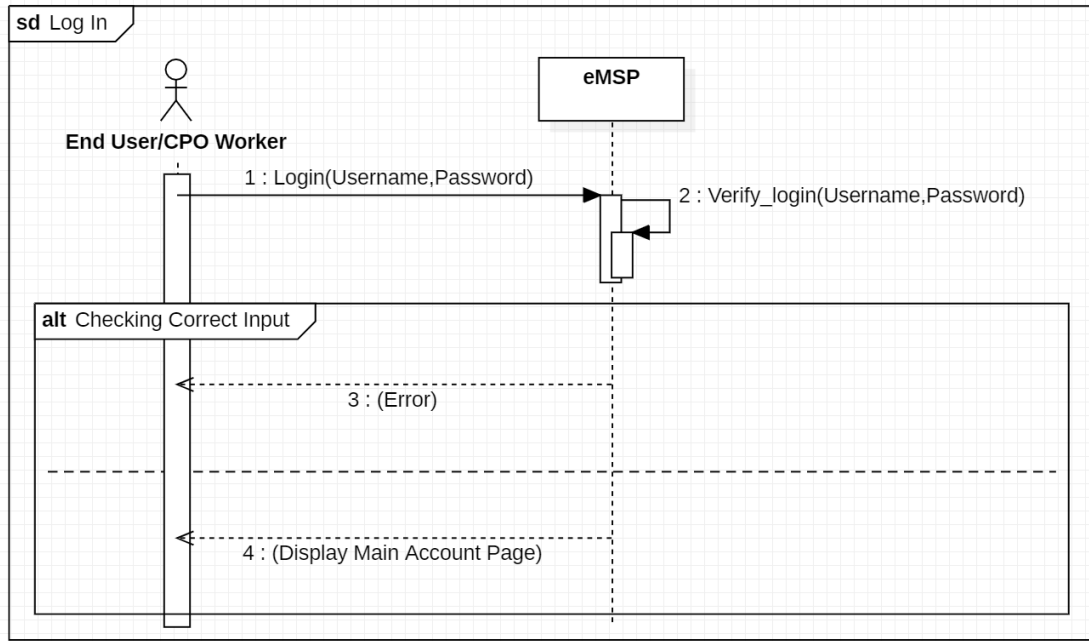Figure 3.3: CPOW UML Use Case Diagram

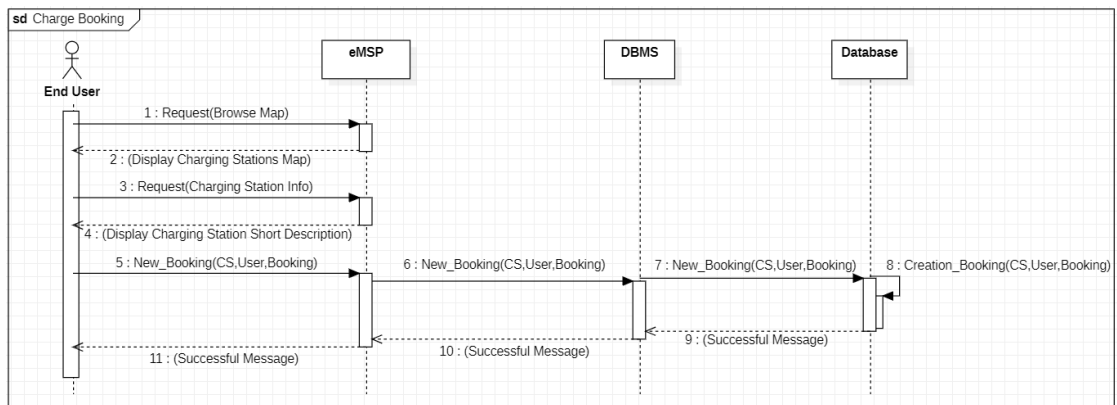Figure 3.4: Sequence Diagram for Use Case n° 2 (Log In)

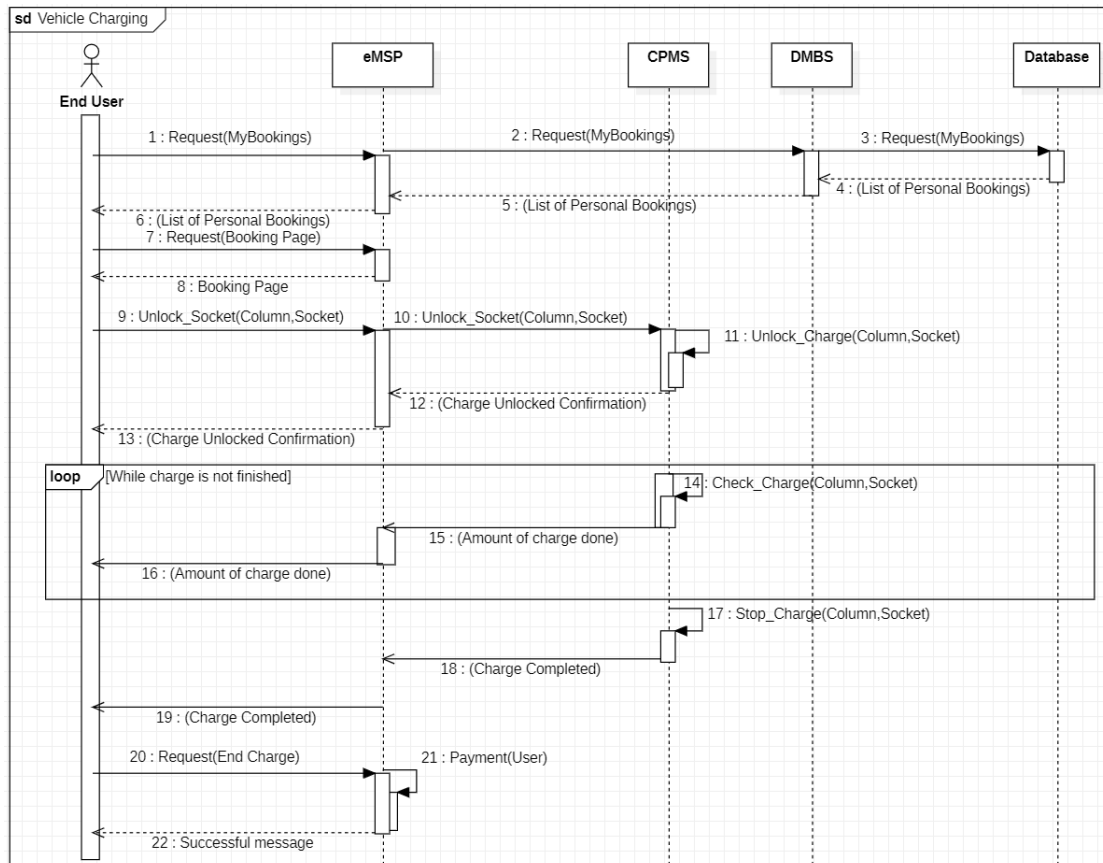Figure 3.5: Sequence Diagram for Use Case n°4 (End-User charge booking)

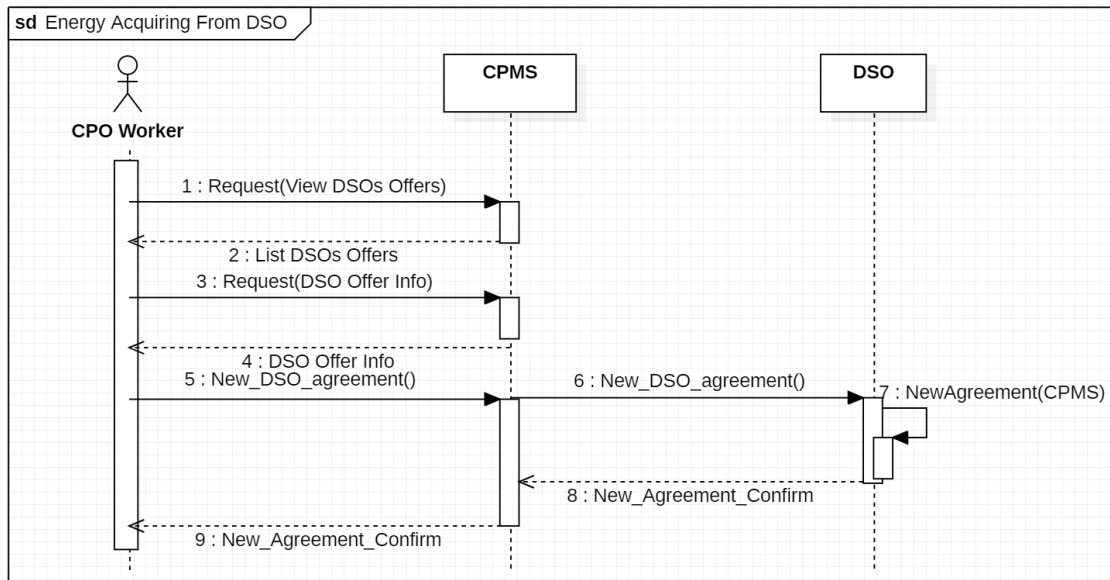Figure 3.6: Sequence Diagram for Use Case n°6 (End-User vehicle charging)

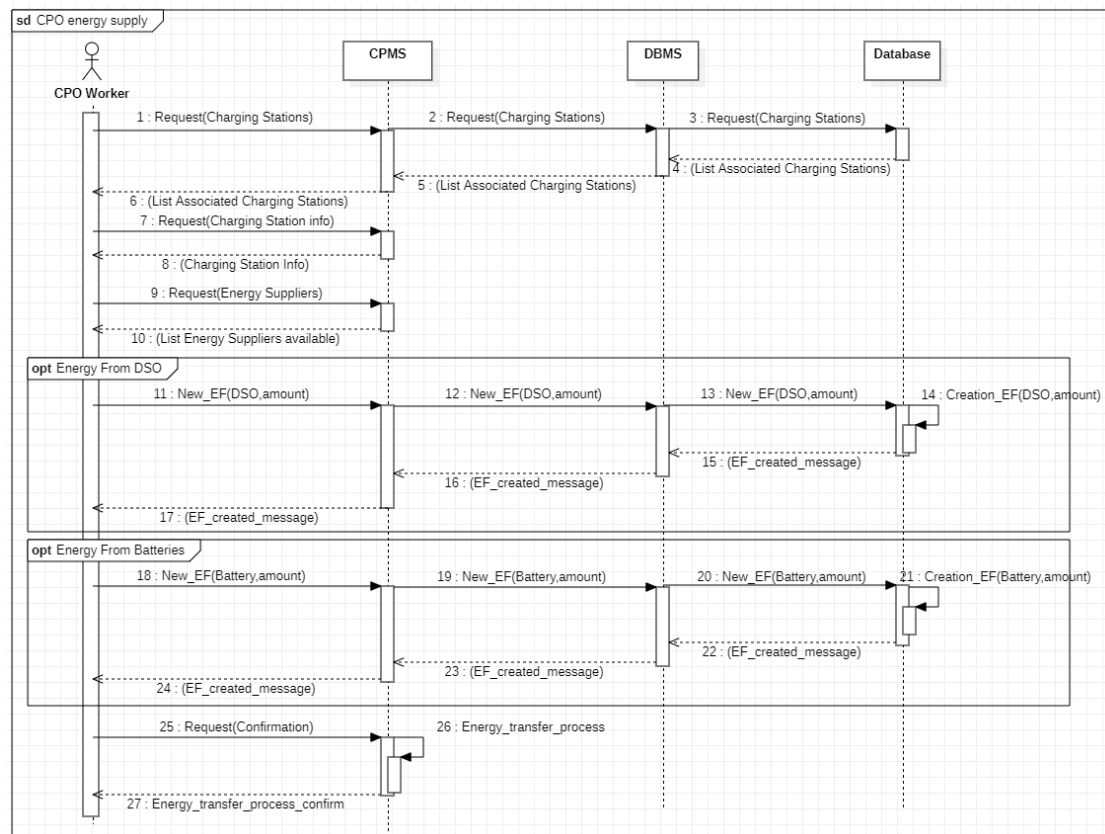Figure 3.7: Sequence Diagram for Use Case n°8 (CPO energy acquiring from DSO)

Figure 3.8: Sequence Diagram for Use Case n°10 (CPO energy supply)

3.3 Performance Requirements

Performances are particularly critical for eMall system. On the end user side, all operations have to be performed as fast as possible and in the correct order, e.g. overbooking of charging sockets must be avoided. Those specifications must fit also end users with low-speed internet connections. Also on the CPMS side all operations, like switching from battery power to network power, must be performed in less than 1 second. The whole computing infrastructure must be thought to be scalable, in order to allow future server capacity to increase due to the increasing number of users.

3.4 Design Constraints

3.4.1 Standards compliance

The system will only store data that are necessarily necessary to ensure the correct execution of all functionalities. The data will only be used for the purpose of eMall system and will be treated confidentially and never shared with the extern, according to European GDPR rules. In particular, the personal data of users and all information about payment methods (e.g. debit card numbers) will never be shown publicly and will be treated with the maximum level of security.

3.4.2 Hardware and software limitations

The mobile application devoted to end users usage needs the following requirements to be satisfied:

- **Hardware**:
  - The mobile phone of end users must support RFID communication protocol in order to be able to unlock the charging sockets.
  - Also a GPS is necessary to show the map of near charging stations.
  - The device must be connected to an internet connection in order to be able to perform any operation.

- **Operating system**: since the main purpose of the system is e-mobility for **all**, the mobile application will be compatible also with older smartphones and minimum requirements are set quite low.
  - Android 5.0 (Lollipop)
  - iOS 9.0

On CPOW side there aren't particular limitations since the web interface can be accessed from every device with an internet connection through every currently used web browser.

3.5 Software System Attributes

3.5.1 Reliability

The system must have the highest reliability possible since all booked recharge must be performed correctly by end users. Furthermore, the updated status of each charging station must be always

known by CPMS. Based on these assumptions multiple instances of data must be available in more than one place in order to satisfy those requirements.

3.5.2   Availability

Also availability is a particularly critical requirement for the system for pretty much the same motivation of reliability. In particular, the system will guarantee 99,999% of up-time. In other words only 5 minutes of downtime per year.

3.5.3   Security

Security is a critical requirement of this system too. The main reasons are the fact that personal data are treated (e.g. identities of users, their cars, and possibly their position if they are going to have a recharge at a specific moment) and also sensible information about payment methods (e.g. bank ID or debit card numbers). All communications between different parts of the system must be properly encrypted and also communication with external APIs must be under strict control. Furthermore, also places where data are stored must be secured and possible theft of personal data must be prevented.

3.5.4   Maintainability

The system will be implemented following coding patterns and specific software engineering practices, all accurately documented, in order to ensure fast and easy maintenance and consequently increment also availability and contribute to downtime reduction as much as possible.

3.5.5   Portability

As already mentioned, the main purpose of the system is e-mobility for **all**, so since compatibility with old devices is already guaranteed on the end user side, here there is the condition that the system is meant to be compatible also with future developed operating systems and devices as long as it will be possible. On the CPOWs side, the system will be compatible with all modern web browsers and the compatibility will be extended also to the one that will be developed in the future.

4 Formal Analysis using Alloy

The purpose of this chapter is to show, through alloy, a formal analysis of the most important aspects which characterize the whole eMall system. The scope is to define and highlight all the constraints that are absolutely necessary to make the system works properly. All the classes of the class diagram have a correspondence with alloy signatures and also the *enumerations*, which represents the state of a charge (e.g. booked, paid, ..) and the type of a discount, have been modeled. Also some simplifications have been done, i.e. for the representation of time instants the *Time* type of alloy has been used, but for sure it hasn't the same power and clearness of *timestamp* or *datetime* types from high-level programming languages. Some facts have been used to ensure that the most important constraints and requirements are satisfied, as can be seen from the diagrams reported below. In the end, some predicates have been exploited to represent the most important functionalities provided by the system and to generate some simple example worlds. In the following section, all the alloy code is shown and after, some worlds are described, in order to allow easy comprehension of how the system will work.

4.1 Alloy Code

```
open util/time

module eMallRASDAlloy

abstract sig User {
        username: one String
}

sig EndUser extends User {
        paymentMethod: set PaymentMethod,
        vehicle: set Vehicle,
        book: set Recharge,
        eMSPs : some EMSP
}

sig PaymentMethod { }

sig Vehicle {
} {
        one this.~vehicle
}
```

```
sig Recharge {
        rechargeId: one Int,
        startTime: one Time,
        endTime: one Time,
        target: one Vehicle,
        payment: one PaymentMethod,
        currentStatus: one RechargeStatus,
        chargingSocket: one ChargingSocket
} {
        gt[endTime, startTime] ∧ rechargeId > 2
}

abstract sig RechargeStatus { }

one sig Booked, StartedCharging, EndedCharging, Paid extends
   ↪ RechargeStatus {}

sig ChargingSocket {
        type: one Int,
        chargingColumn: one ChargingColumn
} {
        //0 slow speed socket, 1 medium speed, 2 fast speed
        type ≥ 0 ∧ type < 3
}

sig ChargingColumn {
        chargingStationC: one ChargingStation
} {
        some this.~chargingColumn
}

sig Battery {
        chargingStationB: one ChargingStation
}

sig ChargingStation {
        discount: one Discount,
        eMSP: some EMSP
} {
        some this.~chargingStationC
}

sig EMSP {}

abstract sig Discount { }

one sig Zero, TwenyFive, Fifty, Other extends Discount {}
```

```alloy
sig CPO { }

sig CPOW extends User {
        badgeNumber: one Int,
        work: one CPO,
        cpms: one CPMS
} {
        badgeNumber > 5
}

sig CPMS {
        cpo: one CPO,
        dso: some DSO,
        manage: some ChargingStation,
        control: some EnergyFurniture
} {
        some this.~cpms
}

sig DSO {
        energyFurniture: set EnergyFurniture
}

sig EnergyFurniture {
        battery: lone Battery,
        chargingSocket: lone ChargingSocket
} {
        one this.~energyFurniture
}

fact usersPartition {EndUser + CPOW = User}

fact avoidDifferentUsersWithSameUsername {
        no disjoint u1, u2: User | u1.username = u2.username
}

fact avoidDifferentCPOWWithSameBadgeNumber {
        no disjoint c1, c2: CPOW | c1.badgeNumber = c2.badgeNumber
}

fact avoidDifferentRechargesWithSameId {
        no disjoint r1, r2: Recharge | r1.rechargeId = r2.
            ↪ rechargeId
}

fact allRechargeHasOneUser {
        all r: Recharge | one r.~book
}
```

```alloy
// Ensure that the EndUser who books a recharge is also the one
    ↪ that owns the car related to that recharge
fact rechargeHasUniqueEndUser {
        all r: Recharge | one v: Vehicle | one u: EndUser | u.book
            ↪ = r ∧ r.target = v ∧ v in u.vehicle
}


// If a user has a recharge he also must have a payment method
fact rechargeHasUniqueEndUser {
        all r: Recharge | one p: PaymentMethod | one u: EndUser |
            ↪ u.book = r ∧ r.payment = p ∧ p in u.paymentMethod
}


fact allCPOHasOneCPMS {
        all c: CPO | one c.~cpo
}


fact allChargingStationsHasOneCPMS {
        all c: ChargingStation | one c.~manage
}


fact allEnergyFurnitureHasOneCPMS {
        all e: EnergyFurniture | one e.~control
}

// Ensure that each CPOW is connected to the CPMS of his CPO
fact cPOWAndCPMSameCPO {
        all w: CPOW | one i: CPMS | one c: CPO | w.work = c ∧ w.
            ↪ cpms = i ∧ i.cpo = c
}

// Ensure that an energy furniture is related to a battery or to a
    ↪  charging station but that can't be related to them both
fact energyFurnitureUniqueness {
        all e: EnergyFurniture | (e.battery ≠ none ∧ e.
            ↪ chargingSocket = none) or (e.battery = none ∧ e.
            ↪ chargingSocket ≠ none)
}

// Ensure that if an energyFurniture arrives to a battery must be
    ↪ controlled from the CPMS that manage the related charging
    ↪ station
fact energyFurnitureCoherenceBattery {
        all c: CPMS | all e: EnergyFurniture | all b: Battery |
            ↪ all s: ChargingStation | (e in c.control ∧ e.battery
            ↪ = b ∧ b.chargingStationB = s)
implies s in c.manage
}
```

```
// Ensure that if an energyFurniture arrives to a chargingSocket
    ↪ must be controlled from the CPMS that manage the related
    ↪ charging station
fact energyFurnitureCoherenceChargingSocket {
        all c: CPMS | all e: EnergyFurniture | all t:
            ↪ ChargingSocket | all l: ChargingColumn
| all s: ChargingStation | (e in c.control ∧ e.chargingSocket = t
    ↪ ∧ t.chargingColumn = l ∧ l.chargingStationC = s) implies s
    ↪ in c.manage
}


// Ensure that if an EnergyFurniture comes from a DSO that DSO
    ↪ must be related with the CPMS that controls the
    ↪ EnergyFurniture
fact dSOCoherence {
        all d: DSO | all e: EnergyFurniture | all c: CPMS | (e in
            ↪ c.control ∧ d in c.dso) implies e in d.
            ↪ energyFurniture
}


// Avoid different charges in the same timeframe in the same
    ↪ socket
fact avoidOverlapping {
        all r1, r2: Recharge | one s: ChargingSocket |
(r1.chargingSocket = s ∧ r2.chargingSocket = s) implies
gt[r2.startTime, r1.endTime] or gt[r1.startTime, r2.endTime]
}


// Ensure recharge state order coherence between different
    ↪ recharges
fact rechargeStatusCoherence {
        all r1, r2: Recharge | (gte[r2.startTime, r1.endTime] ∧ r1
            ↪ .currentStatus ≠ Paid) implies r2.currentStatus =
            ↪ Booked
}


// Ensure that each user is registered to a eMSP that is related
    ↪ to the charging station he wants to exploit
fact eMSPCoherence {
        all u:EndUser | all r:Recharge | all e:EMSP | (e in u.
            ↪ eMSPs) ∧ (r in u.book) implies (r.chargingSocket.
            ↪ chargingColumn.chargingStationC.eMSP = e)
}


fact stringPool {
        none ≠ "a" + "b" + "c" + "d" + "e"
}


pred userAddVehicle [u, u': EndUser, v: Vehicle] {
```

```
            u'.vehicle = u.vehicle + v
}

pred userBookCharge [u, u':EndUser, r: Recharge, v: Vehicle, t1:
    ↪ Time, t2: Time, c: ChargingSocket, p: PaymentMethod, s:
    ↪ Booked] {
        r.target = v
        r.startTime = t1
        r.endTime = t2
        r.chargingSocket = c
        r.currentStatus = s
        u'.book = u.book + r
}

pred startCharging [r, r': Recharge, s: StartedCharging] {
        r'.currentStatus = s
}

pred provideEnergyFurnitureToBattery [c, c': CPMS, d: DSO, e:
    ↪ EnergyFurniture, b: Battery] {
        c'.dso = c.dso + d
        d.energyFurniture = e
        e.battery = b
}

//run provideEnergyFurnitureToBattery

pred world1 {
        #EndUser = 2
        #PaymentMethod = 2
        #Vehicle = 3
        #Recharge > 1
        #ChargingSocket > 1
        #ChargingColumn > 1
        #EMSP < 2
}

pred world2 {
        #CPOW > 1
        #CPMS = 2
        #DSO > 1
        #EnergyFurniture = 5
        #ChargingSocket > 2
        #Battery = 3
        #ChargingStation = 3
        some e: EnergyFurniture | e.chargingSocket ≠ none
        some e: EnergyFurniture | e.battery ≠ none
}
```

```
run world1 for 5
```

```
//run world2 for 5
```

4.2 Worlds

We decided to represent three different worlds to give a presentation on the most important aspect of eMall system.

4.2.1 First world

In the first world, the focus is on the end-user and on the recharge process, as can be seen from figure 4.1. To provide a better clean report we decided to hide the CPO subsystem in order to focus only on the end-user. As can be observed from the picture two end-users that book two different recharge are the main actors. There are two different charging sockets of two different charging columns belonging to the same charging station. The two recharges are on the same charging socket in two different timeframes and users already paid for both. The most important checked constraints are:

- Only end users with a registered vehicle can be associated with a recharge.

- Only end users with a registered payment method can be associated with a recharge. In that specific case, we see that the different users used the same payment method. This is a kind of situation that can happen. E.g. the two users are a wife and a husband and decided to pay with the same debit card. This assumption can be supported by the fact that the two recharges are consecutive (one starts when the other end) so they might be together with their two electric cars.

- The target vehicle of each recharge must be associated with the end user who books the recharge.

- More recharges can't happen in the same socket at the same time.

- Status must be coherent (e.g. a recharge scheduled after another can't be paid if the other one is still booked).

- Charging sockets belongs to only one charging column that belongs to only one charging station.

4.2.2 Second world

In the second world, we move our focus on the CPO subsystem, as can be seen in figure 4.2. Also in this case, to provide a cleaner report users and recharges have been hidden. Now the main actors are two CPOWs that manage two different CPMSs and so work for two different CPOs. Both the CPMS exploit a single DSO to provide energy furnitures. The first CPMS manages a charging station and the second two. The first provides from the DSO energy furniture to a battery of his charging station and the second one provides two energy furniture to a battery and the other two to two different charging sockets of the two different charging stations he manages. The most important checked constraints are:

- Each CPOW works for one and only one CPO and can access one and only one CPMS.

- The CPMS that each CPOW access must be the one of the CPO he is working for.

- Each energy furniture must be controlled by a CPMS that exploits the DSO from which it comes from.

- Each energy furniture must recharge a battery or a charging socket of a charging station that is managed by the CPMS which controls that specific energy furniture.

- In order to be active, each charging socket must be associated with an energy furniture, or at least the charging station it belongs to must have at least a battery that is associated with an energy furniture.

- Different CPOs can work with different eMSPs.

- Different CPOs can work with the same DSO.

- Charging sockets belongs to only one charging column that belongs to only one charging station.

4.2.3 Dynamic world

Some dynamic modeling has been done in alloy. The most important functions have been represented as predicates. Most of them are trivial and for completeness, we report in figure 4.3 the *provideEnergyFurnitureToBatteryOne*. The main purpose of this predicate is to exploit a new DSO to provide an energy furniture to a battery of a charging station managed by the interested CPMS.
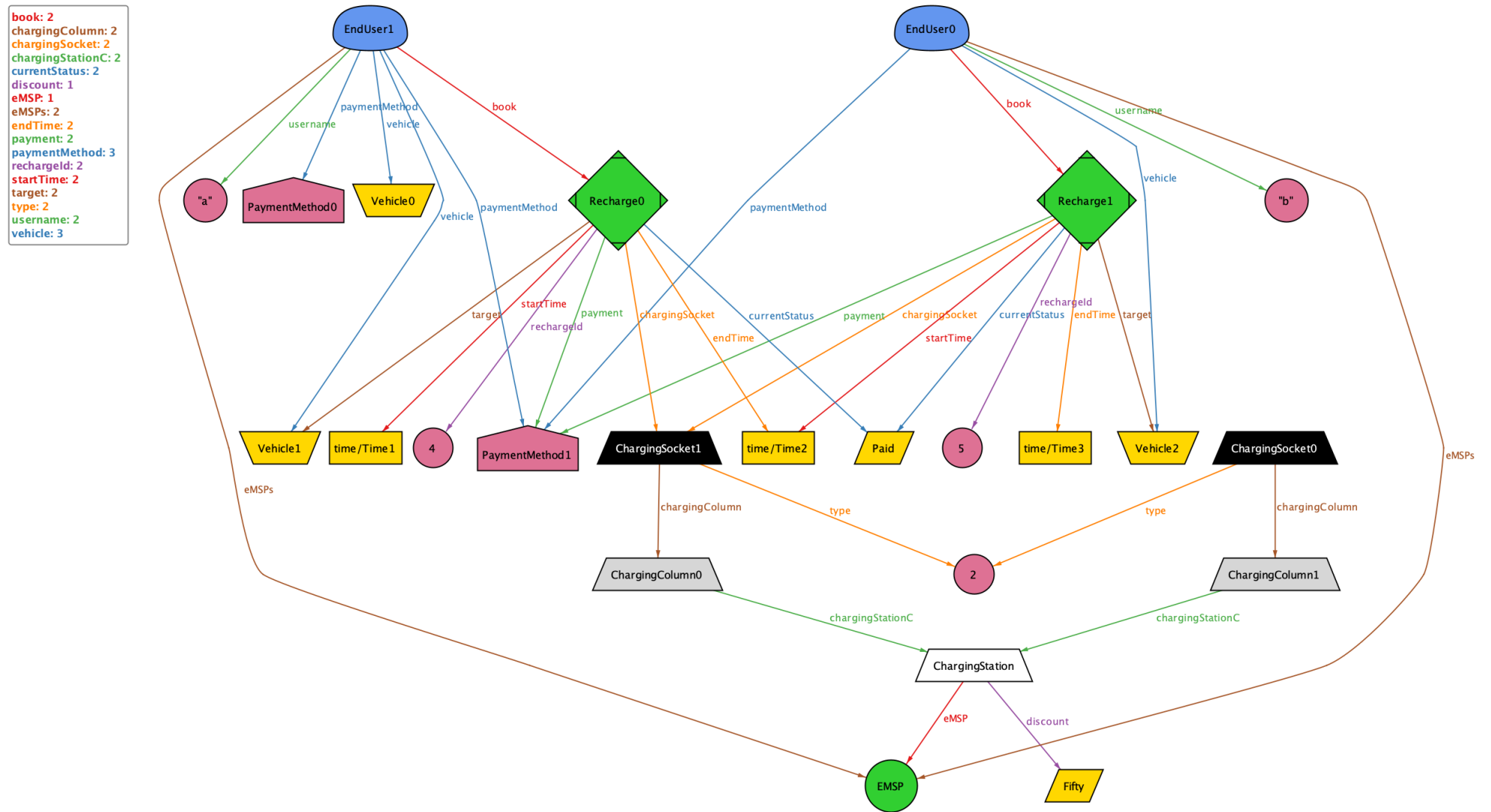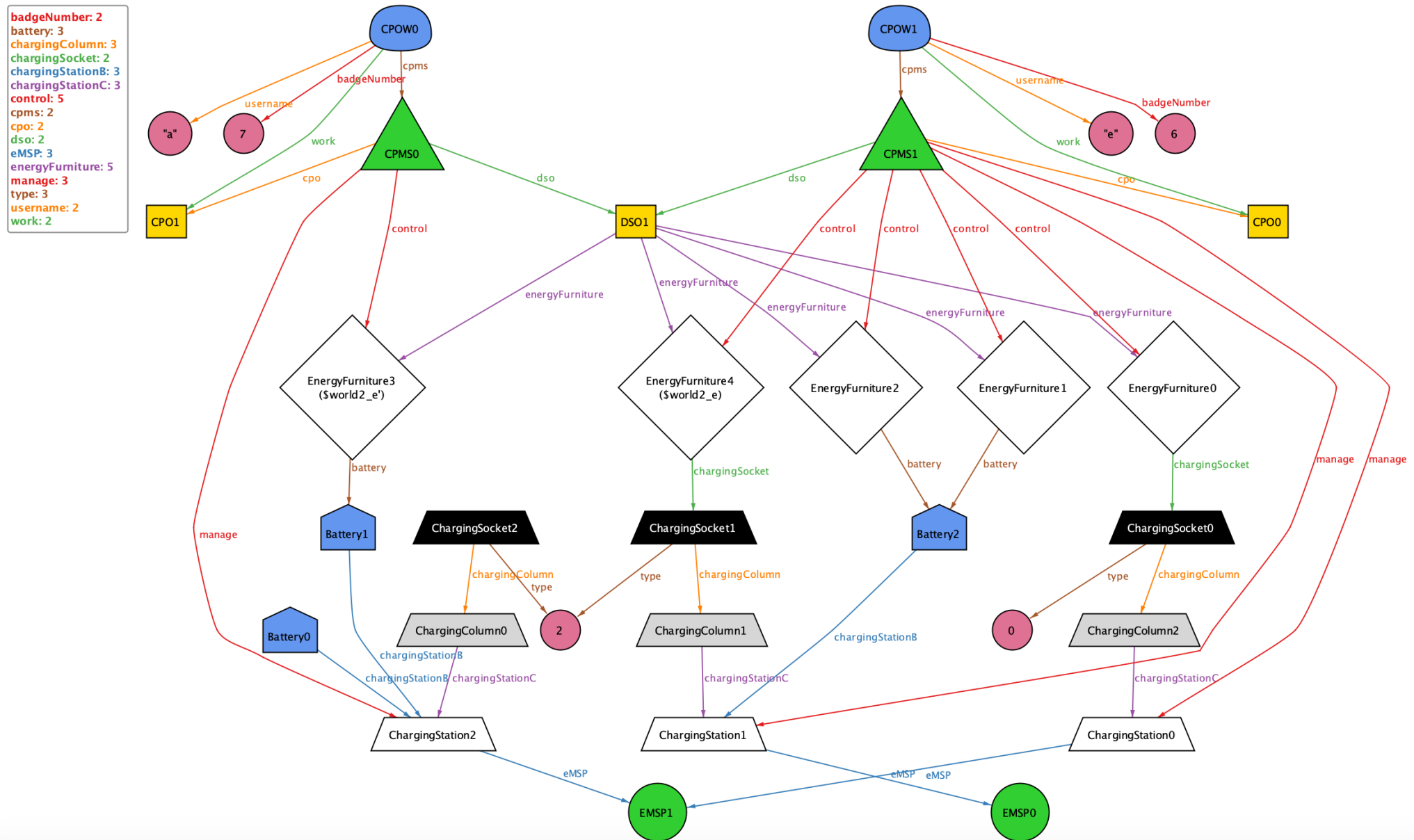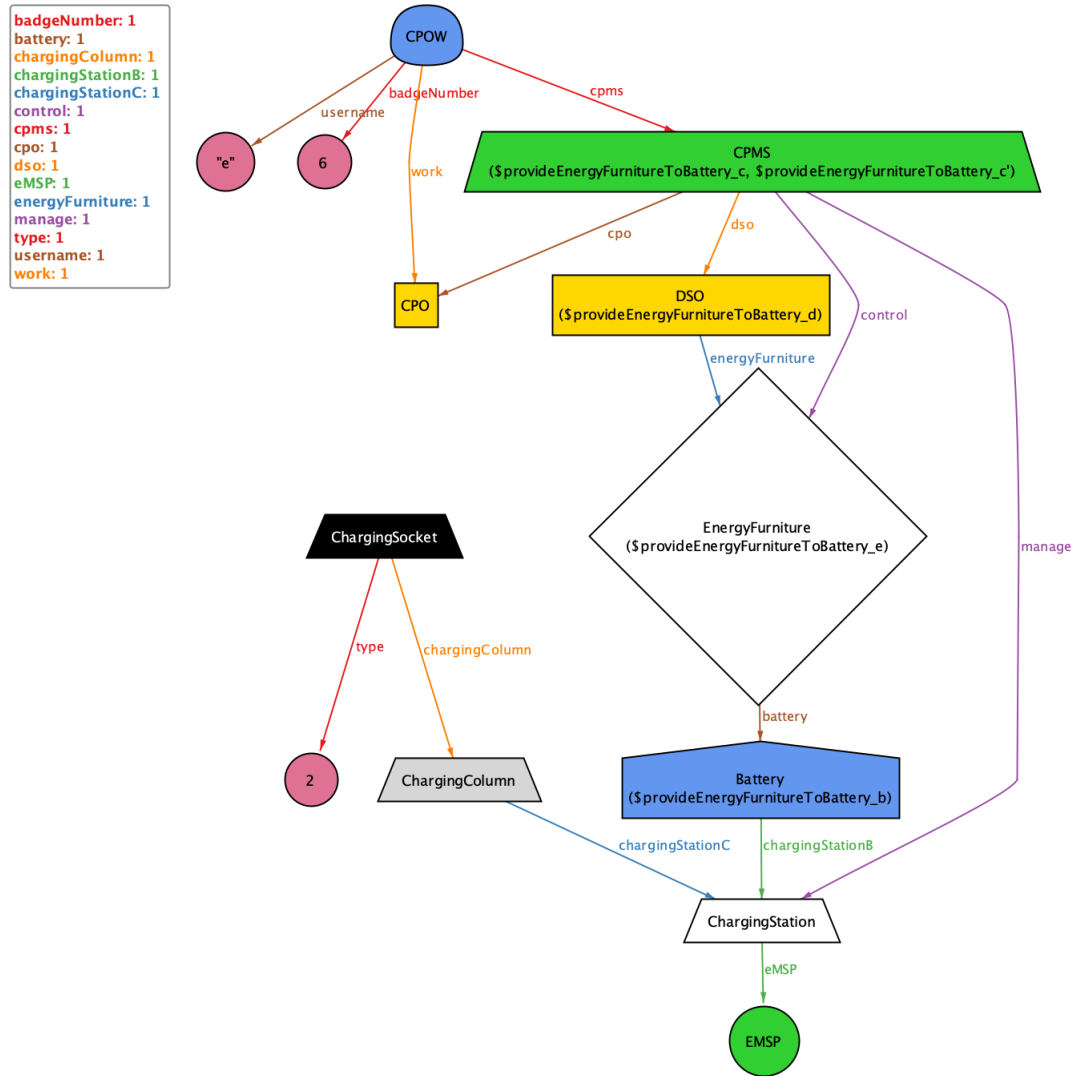
Figure 4.1: Alloy World n.1

Figure 4.2: Alloy World n.2

Figure 4.3: Alloy predicate *provideEnergyFurnitureToBattery*

5 Effort spent

	Enrico Brunetti	
Date	*Hour*	*Section*
09-11-2022	1.5 h*	Introduction
16-11-2022	4 h	Overall Description
17-11-2022	1 h	Overall Description
18-11-2022	2 h	Overall Description
27-11-2022	2 h	Overall Description
30-11-2022	1 h*	Overall Description
04-12-2022	3 h*	Specific Requirements
05-12-2022	1.5 h	Overall Description
6-12-2022	1 h	Specific Requirements
8-12-2022	4 h	Formal Analysis Using Alloy
9-12-2022	4 h	Formal Analysis Using Alloy
20-12-2022	0.5 h	Overall Description
20-12-2022	2 h	Specific Requirements
21-12-2022	2.5 h*	Final Revision
22-12-2022	1 h	Formal Analysis Using Alloy

Table 5.1

	Matteo Gionfriddo	
Date	*Hour*	*Section*
09-11-2022	1.5 h*	Introduction
15-11-2022	2.5 h	Overall Description
19-11-2022	1.5 h	Overall Description
22-11-2022	2 h	Overall Description
27-11-2022	1 h	Overall Description
29-11-2022	2 h	Overall Description
30-11-2022	1 h*	Overall Description
04-12-2022	3 h*	Specific Requirements
07-12-2022	1.5 h	Specific Requirements
11-12-2022	3 h	Specific Requirements
13-12-2022	1.5 h	Specific Requirements
15-12-2022	1 h	Overall Description
17-12-2022	1.5 h	Overall Description
19-12-2022	2 h	Overall Description
20-12-2022	1.5 h	Specific Requirements
21-12-2022	2.5 h*	Final Revision

Table 5.2

* Group work

6   References

- Specification Document: "Assignment RDD AY 2022-2023_v3.pdf"

- Alloy Reference Document Examples: "http://tmancini.di.uniroma1.it/teaching/courses/2007-2008/mfis/materiale/progetti/Pagliaro

- IEEE Std 830-1993 - IEEE Guide to Software Requirements Specifications.

- IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.