RASD

1 Introduction

1.1 Purpose

Electric mobility (e-Mobility) is a way to limit the carbon footprint caused by our urban and sub-urban mobility needs. When using an electric vehicle, knowing where to charge the vehicle and carefully planning the charging process in such a way that it introduces minimal interference and constraints on our daily schedule is of paramount importance.

There are four main entities that need to interact in order to provide the mentioned service:

1. eMSP (e-Mobility Service Providers): an application that links together the final users (owners of electric vehicles) and the charging stations
2. CPOs (Charging Point Operators): they own and manage the charging station
3. DSOs (Distribution System Operators): energy providers

The purpose of this project is to develop e-Mall (e-Mobility For All), a set of applications that:

- will grant the user the possibility to book charges for its vehicles and pay for it, monitoring costs and special offers;
- allows CPOs to handle their own charging areas through CPMS (Charge Point Management System) that manages the charging columns and the energy acquisition for the single charging stations, automatically or manually by employees.

This document later will further expand on the goals and requirements put on the system to be with the purpose of guiding the development.

1.1.1 Goals

Goals	**Description**
G1	Allow user to pay for a charge
G2	Allow user to book a charge
G3	Allow user to register to the eMSP system
G4	Allow user to know about charging station prices and special offers
G5	Allow user to perform a charging process
G6	Allow user to make reservations
G7	Allow CPMS to manage manually or automatically the power input/output of the charging station

1.2 Scope

Our system focuses on the eMSP and CPMS subsystems with all the features listed in the specification document without making the eMSP smarter than it needs to for the end user.

1.2.1 World phenomena

Identifier	Description
WP1	Energy cost shifts
WP2	Car is low on battery
WP3	User gets to the charging station

1.2.2 Shared phenomena

Identifier	Word controlled
SP1	User makes a reservation
SP2	User plugs the car to the charging station
SP3	User registers an account
SP4	User validates it’s reservation through NFC on the charging column
SP5	User unplugs the car from the charging station

Identifier	Word controlled
SP6	User searches for a specific charging station
SP7	User is late to the reservation [MAYBE WORLD PHENOMENA]
SP8	Charging Station operator decides on which energy provider to use
SP9	Charging Station operator decides on the price/offer for the energy
SP10	Charging Station operator decides whether to store energy in batteries or to use stored energy

Identifier	Machine controlled
SP11	A payment is charged to the user
SP12	CPMS allows charging column to charge vehicle
SP13	System shows information about charging stations
SP14	System shown reservation list
SP15	System sends “Charging done” notification
SP16	System shows charging stations map
SP17	CPMS asks for energy prices to the DSOs
SP18	CPMS decides which energy source to use

1.3 Definitions, Acronyms, Abbreviations

Definition	Description
Charging column	A device with one ore more standard charging sockets equipped with a NFC tag
Charging station	A group of charging columns displaced in a nearby area owned by a CPO and managed through the CPMS
User	Person interested in using the system
Operator	Instructed personnel that manages a charging station

Acronyms	Description
eMSP	e-Mobility Service Providers application that links together the final users and the charging stations
CPOs	Charging Point Operators owners and managers of the charging station
DSOs	Distribution System Operators energy providers
CPMS	Charge Point Management System manages reservations and energy for charging stations
eMall	Electric Mobility for All
IoT	Internet of Things
NFC	Near Field Communication
CS	Charging station

Abbrevations	Descrpition
RASD	Requirements Analysis and Specification Document

1.4 Revision history

1.5 Reference Documents

The specification document “Assignment RDD AY 2022-2023_v3.pdf”

1.6 Document structure

This document is composed of 5 sections, detailed below.

In the first section the purpose of the project is introduced and the scope is defined. Lastly, definitions and abbreviations are reported.

In the second section an overall description of the system is reported, including its main functions. Moreover, class diagram, scenarios, state-charts and domain assumption are described.

In the third section the focus is on the requirements of the system and the mapping between project goals and them. This includes functional, non functional and external requirements.

In the fourth section the formal analysis is presented with the help of alloy.

The fifth section reports project team member total efforts.

2 Overall description

2.1 Product perspective

2.1.1 Scenarios

S1. User sign up

Elon want's to sign up to the eMSP system because he just bought a Tesla Model S. He opens the previously downloaded app and selects the option to sign up. He inserts his data and the payment methods which will first need to be verified to be effectively registered to the service.

S2. Making a new reservation for the next week

Jeff will be on travel next week so he wants to make a reservation so that he will be able to have lunch while the car is charging. He opens the app and through the search bar he searches for the location he will be having lunch in. The map moves and shows the nearby charging station marked with different colors accordingly to their max charging rate. Jeff selects the nearest CS with ultra fast charging speed, views it's cost and selects "Book a charge". He now inputs the date and, from the available time-frames list with relative charging rates, he selects the most suitable for his time of arrival and expected permanence. A booking fee proportional to the booked time-frame is charged on Jeff's payment method which will be refunded after the charging process.

S3. Making a reservation for the immediate future

Bill's car is signaling that is low on charge so he pulls over and opens the app. He selects "Charge NOW" and the map moves to his current location. Different SC with available slots for the current time are shown marked with colors accordingly to their max charging rate and with an exclamation mark in case there's a ongoing offer. Bill notices that a nearby charging station is promoting a good offer so he clicks on it and the current price is shown. He selects the required duration and then "Go NOW" to confirm his booking and arrival. A booking fee proportional to the booked time-frame is charged on Bill's payment method which will be refunded after the charging process.

S4. Charging process

Mark arrived at the charging station right on time for his quick charge reservation. He identifies the right charging column for the requested charging speed and parks near it. After getting out of the car and opening the app he places his NFC enabled phone on the tag marked on the charging column. The CPMS validates the reservation and unlocks the charging socket to which Mark plugs the car to before leaving for the bar. When the booked time-frame is up the eMSP app notifies Mark that his reservation ended and to go retrieve his car to which he goes. He unplugs it and get's in to drive home; on his way the payment is processed and receives a notification with the total charging cost minus the refund for the reservation fee.

S5. Missed reservation

Warren had a reservation for a charge at 11:00 am but he forgot about it. After 15 minutes of delay he receives a notification on his eMSP app that his reservation time is up and that his reservation fee is not refundable.

S6. CPO employee manually overrides the CPMS

Silvio, the charging station operator, has received directives to start buying greener energy. He overrides the automatic decision of the CMPS from his terminal, he chooses from a list of DSOs the most convenient one that utilizes green energy and, since the batteries of the charging station are almost empty, to also fully charge them.

2.1.2 Class diagram

Here presented is the class diagram of the entire system composed of both the eMSP and CPMS

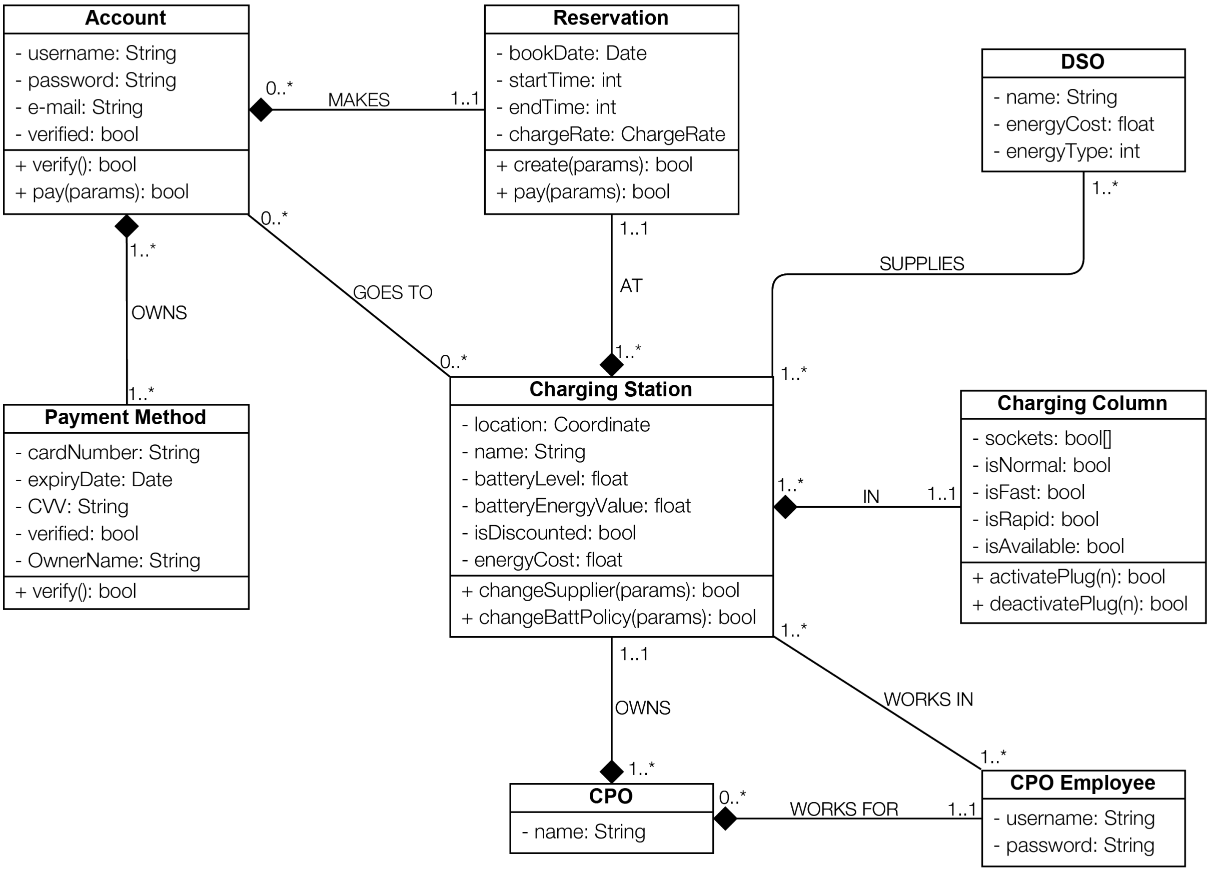
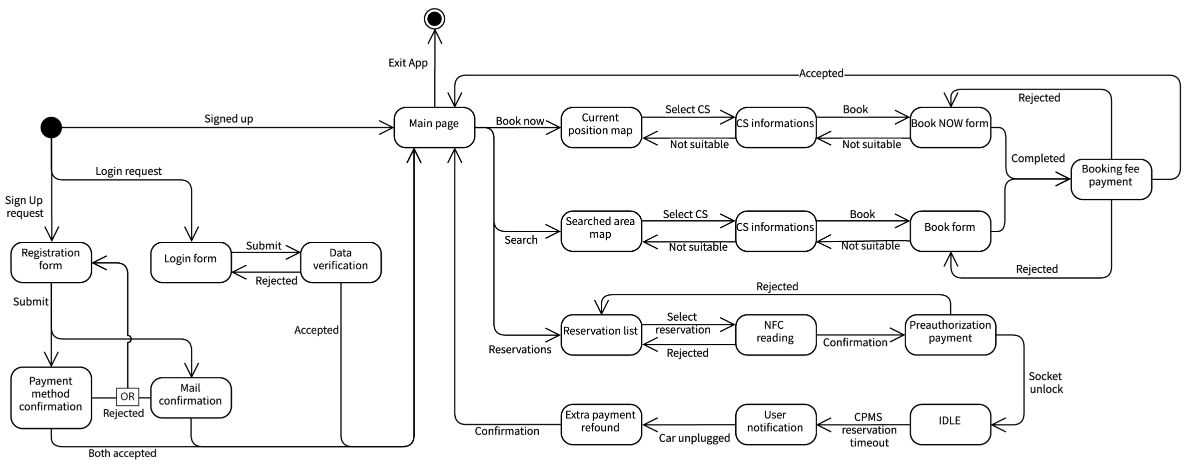
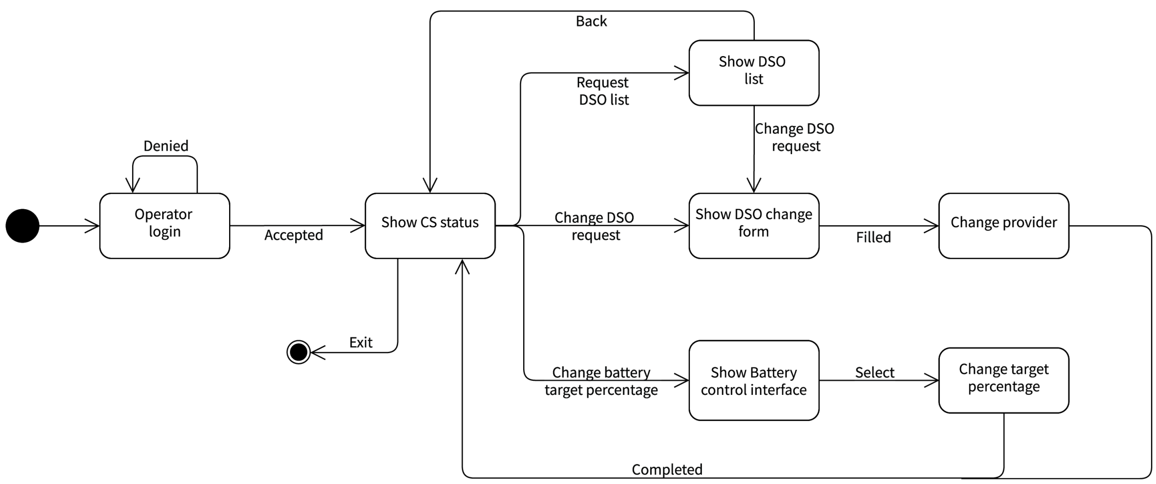


Figure 1. Class Diagram of the eMall system

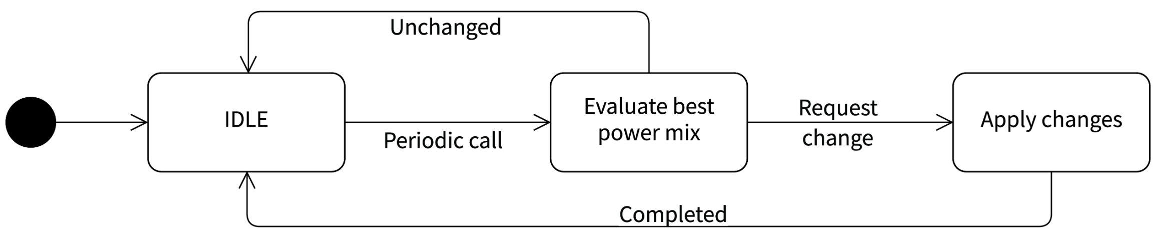
2.1.3 State charts



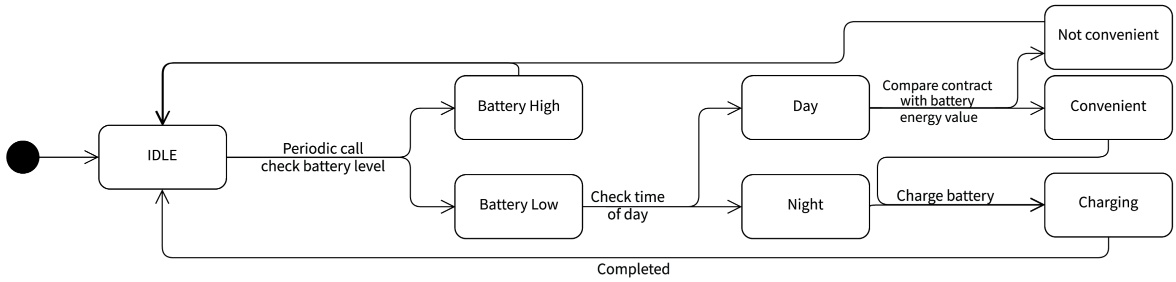
eMSP: Full app functionality



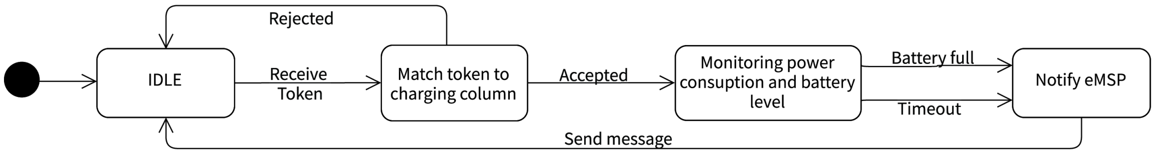
CPMS: operator login state diagram



CPMS: decision of vehicles charging policy



CPMS: decision of internal battery pack policy



CPMS: charging vehicle process

Here we include the most important features of both eMSPs and CPMSs.

Booking of a time-frame

A booking can be performed either for the immediate future or later. The first one is based upon the fact that a user might need to charge his vehicle immediately and thus is searching for a place nearby to book a time-frame. The software shows a map centered in his current position with each charging station signaled by a different color with respect to the maximum charging rate available at the current time and a special mark for those who are providing a limited time special offer. The user is able to look for and select the one that most suits his needs and book a charge by selecting the arrival, residence time and charging speed. In the case that he wants to book a charge for the future, the map also has the possibility to search for specific locations or charging stations. When the research string is submitted the map moves to the new location and shows the charging stations nearby with the same iconography as stated before. The user is now able to select arrival date and time, charging duration and charging rate.

In both cases after the chosen time-frame is submitted and confirmed by the CPMS a fee proportional to the booked time is required. When the payment is finalized the reservation is effectively confirmed.

Charging a vehicle

After the user has reached the charging station and identified a suitable charging column for his booked charging speed, he approaches it and through the app confirms his arrival by reading a NFC token or by inserting a one time key displayed on the column directly on his phone. After confirmation, based on the time-frame duration and charging rate the CPMS calculates the maximum energy that the charge should consume and processes the payment through eMSP. If the payment succeeds the socket is freed and the user has 1 minute to plug the car in. He then can leave and the CPMS takes control of the charging process by monitoring power consumption and vehicle's battery level. After the timeframe is up or the vehicle is fully charged the system notifies the user to unplug the car and leave the charging column free. The charging bill of the actual consumed energy is then processed by the eMSP and the difference, if any, is refunded to the user which then gets notified of the final cost. The user is also able to leave early by simply unplugging the vehicle. When the car is unplugged the user must leave in five minutes and the remaining time of the booked timeframe is rendered available for other users to reserve.

Power management

One of the most critical requirements of the CMPS is to manage the power input and output. For this it is capable to monitor the battery packs level and decide smartly from where to acquire power or whether to use the stored energy either to provide competitive costs or to increase incomes, despite an increase in general DSO costs. In order to do this the CPMS periodically (hourly) checks the DSO prices and if a better than the current provider is found a new contract is stipulated. The batteries level are also checked and if the battery level is low and it's currently daytime, only if the current DSO price is lower or equal than the energy cost inside the battery, than they are charged. Otherwise if it's night, so, the price of energy is lower than the daytime one, batteries should be charged to full.

When a new vehicle is plugged the system checks battery levels If battery are not low and the price of energy stored in battery is lower than the actual DSO price the vehicle will be charged using battery energy otherwise it is charged through the DSO.

2.3 User characteristics

The following three actors are considered in the eMSP and CPMS systems:

- Unregistered customer
  A user that owns an electric vehicle and wants to start using the eMSP system services
- CPO employee
  An employee with given credentials to access the CPMS system which is able to take decisions about the charging stations
- Customer
  A registered user which necessitates to use the app to charge his vehicle

2.4 Assumptions, dependencies and constraints

2.4.1 Domain assumptions

Assumption	Description
D1	During the night hours energy is less expensive
D2	Inside charging areas there is free Wi-Fi that users can access in order to avoid low signal problems during the charge
D3	After a vehicle reserved time-frame finishes the charging socket is freed in 5 minutes
D4	The device used by the CPO employee has already the software installed and ready for use
D5	There exists an API that the CPMS can use to communicate with multiple DSOs
D6	An API that allows to manage battery pack and retrieve information about its status exists
D7	An API that allows to manage charging columns and retrieve information about its status exists
D8	An API that allows to manage power switcher and retrieve information about its status exists
D9	An API that allows eMPS to obtain a list of CS owned by multiple CPOs exists.
D10	There exist an API that manages payments on behalf of the eMSP
D11	Each charging column has a unique identifier which contains informations about iteself and the CS it is contained in

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

The user interface of eMSP is a phone app available on both Android and iOS which uses a simple and intuitive graphical interface. It should be as easy to use as possible, regardless of the technical background of the user.

The user interface for CPO employees should be available only for laptops and desktops since it would be used only by trained personnel in the work environment. It should be made professional but intuitive so that training would be easy to perform.

3.1.2 Hardware interfaces

For the eMSP the only hardware necessary should be a phone with either an Android or iOS operating system and, for added functionality, the presence of NFC and GPS.

The CPMS system has:

- on the charging columns a display for the one time keys, and the different sockets with respective power output sensors
- on the battery packs, network interface used to communicate power levels measured in order to allow CPMS to manage charging
- inside an employee reserved area a space with a computer capable of running the CPMS software for CPO employees

3.1.3 Communication interfaces

Many of the functionalities offered by the system relies on communication with other services. The communication is either of the type where the CPMS or eMSP communicates with external services, or where CPMS and eMSP exchange information/requests between each other. Specifically, there are seven different interfaces that CPMS and eMSP utilize, possibly through Web APIs. Below we list the required function offered by each interface. Note that the description of each interface are assumptions made in this project.

- Retrieval of data from battery pack
  Each battery pack inside the charging area is equipped with an IoT device that is capable to provide information about the battery level and input/output power delivery;
- Retrieval of data from charging column
  Each charging column in the charging area is able to provide information about the number of unplugged sockets and the power consumption of each one, the charging percentage of the vehicle attached to the socket. It also has a NFC module and a display that respectively sends and expose the one-time-use code supplied by the CPMS;
- Information exchange with power switcher
  The power switcher of the charging area is able to receive directives on how to deliver power to the different sockets and provide information about which kind of energy source each one is utilizing;
- Booking information exchange between CPMS and eMSP
  For the eMSP app to be able to provide a booking service it’s necessary that the CPMS communicates which time-frames of a specific dates are free and the eMSP communicates which time-frames must be reserved for a user.
- Positional information of charging stations
  The CPO exposes and API that is able to answer with the geolocation of all it’s charging stations to the eMSP so that it can display to the user the correct map with all neighboring charging areas of the different CPOs
- Retrieval of data from DSO
  The CPMS makes use of the DSOs interface to retrieve data about price and availability so that it can decide on which one to use
- Exchange of information with payment service
  Both eMSP and CPMS must check and ask for payments to be performed with the different payment services so that users get correctly billed for the obtained service

3.2 Functional requirements

Requirements	Description
R1	eMSP shall allow users to sign up to the eMPS service
R2	eMSP shall allow registered users to sign in
R3	eMSP shall allow users to link a payment method to his account
R4	eMSP must verify users payment methods
R5	eMSP must verify users email upon registration
R6	eMSP shall allow registered users to make a reservation to a specific CS at a specific date
R7	eMSP must show registered users correct positional data and nearby charging stations
R8	eMSP shall allow registered users to select an area in which to search for charging stations
R9	eMSP must be able to order payments from given payment methods
R10	eMSP must notify registered users of missed reservations
R11	eMSP must notify registered users of the final billing for the received service
R12	eMSP must notify registered users when the booked time-frame is up
R13	eMSP must notify the user when the CPMS sends a charge complete response
R14	eMSP must notify CPMS of code read through NFC
R15	eMSP must notify CPMS of the one-time-token inserted through the app
R16	CPMS must be able to allow CPO employees to manually decide how to manage battery and net power
R17	CPMS must be able to allow CPO employees to manually decide which DSO to buy energy from
R18	CPMS must be able to automatically decide how to manage its own battery and net power
R19	CPMS must be able to automatically decide which DSO buy energy from
R20	CPMS must be able to answer eMSP requests for free timeframes and relative prices
R21	CPMS must allow a user to charge it's vehicle at a specific time-frame if and only if that specific time-frame has been reserved by that user
R22	CPMS must be able to manage reservations through a DBMS
R23	CPMS must be able to authorize users with reserved time-frames
R24	CPMS must be able to interact with all APIs that manage charging stations components
R25	CPMS must notify eMSP when the vehicle of a user is fully charged
R26	CPMS must be able to notify user of exceptions
R27	CPMS must be able to notify user of successful actions
R28	eMSP must be able to notify user of exceptions
R29	eMSP must be able to notify user of successful actions
R30	CPMS bust be able to notify eMSP of exceptions
R31	CPMS bust be able to notify eMSP of successful actions
R32	eMSP must be able to retrieve charging stations locations from a CPO list
R33	CPMS must notify eMSP of final power bill
R34	CPMS shall allow CPO employee with corporate credential to sign in
R35	CPMS must be able to show CPO employees a filtered and sortable list of DSOs

3.2.1 Mapping on Goals

Goal	Domain assumption	Requirement
G1	D10, D2	R2, R3, R4, R9, R11, R28, R29, R30, R31, R33
G2	D2, D3, D9	R2, R3, R4, R6, R7, R8, R9, R20, R22, R28, R29, R30, R31, R32
G3	-	R1, R3, R4, R5, R28, R29

G4	-	R2, R7, R8, R16, R17, R18, R19, R20, R24, R28, R29, R32
G5	D2, D3, D6, D7,D8, D11	R2, R3, R4, R6, R12, R13, R14, R15, R16, R18, R21, R23, R24, R25, R28, R29, R30, R31
G6	D3, D7, D9, D10	R2, R3, R4, R6, R9, R10, R20, R22, R28, R29, R30, R31
G7	D1, D4, D5, D6, D7, D8	R16, R17, R18, R19, R24, R26, R27, R34

3.2.2 Use cases

eMSP sign up

Actor	User
Entry conditions	The user doesn't have an account and he is on the initial view of the system
Event flow	1. User presses the "Register account" button 2. The user fills the registration form with username, mail, password, and payment method 3. User receives an e-mail on the given one containing a link that must be clicked to confirm the address 4. The system verifies the payment method though the relative API 5. The eMSP system processes the information and displays a success message
Exit condition	The account is created
Exceptions	In all cases eMSP will notify the user 1. The user doesn't insert the mandatory data 2. The mail is not verified 3. The payment method is rejected

eMSP sign in

Actor	User
Entry conditions	The user is registered, not logged in and in the initial view of the system
Event flow	1. User presses the "Login" button 2. User inserts mail or username and password in the form 3. User submits 4. The eMSP processes the information and displays a success message
Exit condition	User is logged in the system
Exceptions	In all cases eMSP will notify the user+ 1. User inserts partial or incorrect data

Making reservation for the immediate future

Actor	User
Entry conditions	The user is logged in and in the main page of the system
Event flow	1. User presses the "Book NOW" button and a map with his current location is shown 2. The user selects a charging station and is shown free timeframes with relative charging speeds and cost 3. The user inserts the arrival and duration of permanence 4. The eMSP sends the reservation information to the CPMS and waits for confirmation 5. The system proceeds to request the reservation fee (proportional to the booked timeframe) with one of the inserted payment methods and displays a success message
Exit condition	User has made a reservation
Exceptions	In all cases eMSP will notify the user 1. User inserts partial or incorrect data 2. The payment gets refused (also the CPMS gets notified of the canceled reservation) 3. The CPMS returns an error

Making reservation for the future

Actor	User
Entry conditions	The user is logged in and in the main page of the system
Event flow	1. User searches with the search bar a location and the map moves to that position showing the charging stations nearby 2. The user selects a charging station and is shown free timeframes with relative charging speeds and cost 3. The user selects "Book for a charge" and gets prompted to a form 4. The user inserts the arrival date and time and duration of permanence 5. The eMSP sends the reservation information to the CPMS and waits for confirmation 6. The system proceeds to request the reservation fee (proportional to the booked timeframe) with one of the inserted payment methods and displays a success message
Exit condition	User has made a reservation

Exceptions	In all cases eMSP will notify the user 1. User inserts partial or incorrect data 2. The payment gets refused (also the CPMS gets notified of the canceled reservation) 3. The CPMS returns an error

Charging process with NFC

Actor	User
Entry conditions	User who previously made a charge reservation arrives at the charging station and parks near a charging column enabled to the booked charging rate. The user has a mobile phone with NFC module enabled
Event flow	1. The user opens the eMSP app and select the desired reservation from a list 2. The user puts the mobile phone on the NFC tag of the charging column if his phone is anable to use of NFC technology otherwise The user insert the code displayed by the charging column into the eMSP app 3. The eMSP submits the couple (reservation, charging column code) to the CPMS 4. The CPMS authorizes the user 5. The user plugs the car 6. The charging column starts to supply energy 7. The CPMS notifies the user through its eMSP app if the battery fills before the end of the booked time-frame 8. The eMSP app notifies the user 10 minute before the ending of the booked time-frame 9. The user unplugs the car 10. The CPMS sends the final bill to the user through the eMSP 11. The eMSP pay the bill trough the user’s register payment method 12. If the charging spot has been freed before the end of the booked time-frame, the CPMS makes the remaining time available to other users.
Exit condition	The charging process is successfully completed, the bill is paid and the charging spot is available to other users.
Exceptions	In all cases CPMS notifies user trough eMSP 1. The charging columns NFC does not accept the user token 2. The payment does not successfully complete.

Missed reservation

Actor	User
Entry conditions	The user didn’t show up to the reservation at the start time
Event flow	1. The CPMS notifies the eMSP that the user is late and the reservation has been canceled 2. The eMSP notifies the user that the reservation has been missed and the fee will be held 3. The CPMS frees the remaining time of the missed reservation
Exit condition	The lost reservation time is freed
Exceptions	No exceptions should occur

CPO employee sign in

Actor	CPO employee
Entry conditions	The CPO employee has corporate credentials, not logged in and in the initial view of the CPMS system
Event flow	1. The CPO employee presses the “Login” button 2. The CPO employee inserts its own corporate employee code and password 3. User submits 4. The CPMS processes the information and displays a success message
Exit condition	The CPO employee is logged in the system
Exceptions	In all cases CPMS will notify the CPO employee: 1. User inserts partial or incorrect data

Change DSO manually

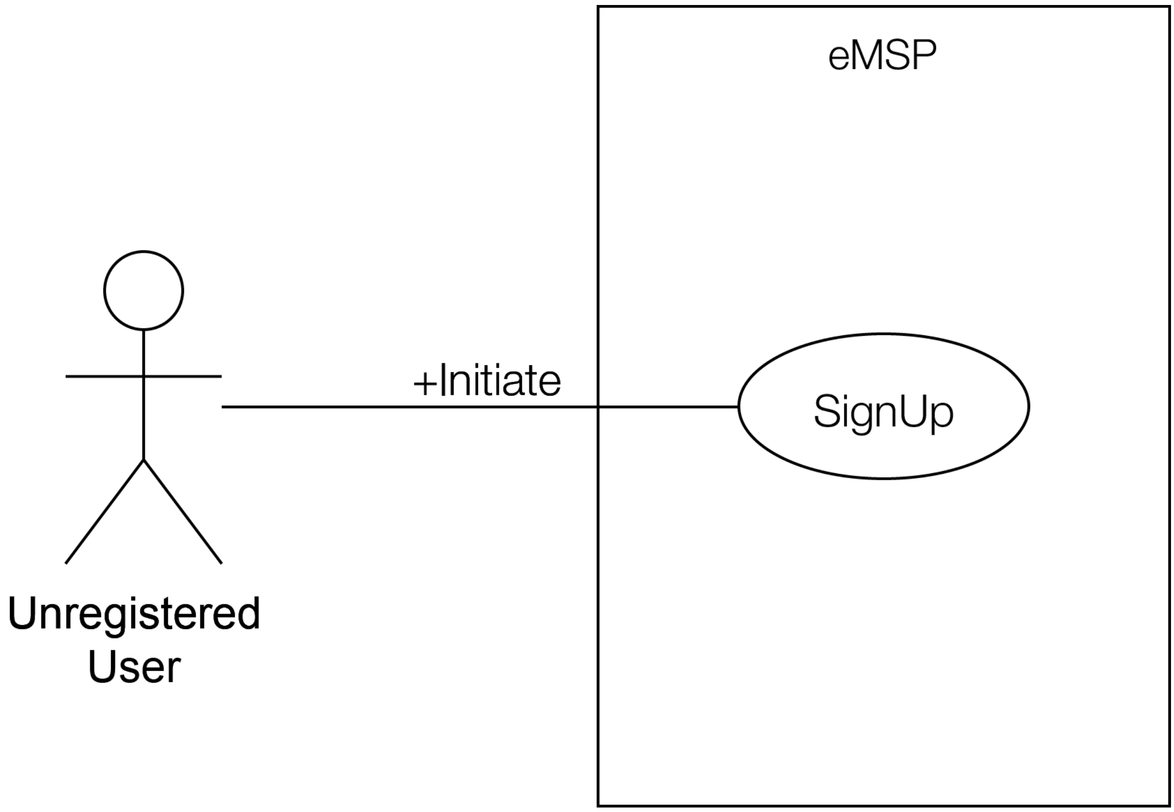
Actor	CPO employee
Entry conditions	CPO employee is logged in the CPMS management software trough its corporate credentials
Event flow	1. The CPO employee select option “manually select DSO” 2. The CPO employee checks some flags about requisites that the DSOs which will be select must complies. 3. The CPO employee submit 4. The CPMS shows the list of available DSOs that complies the above selected requisites, sortable by each of them. 5. The CPO employee select the DSO that better match the requisites and submits his decision. 6. CPMS stipulate a new energy acquisition contract with the chosen DSO

Exit condition	The CPMS energy supplier is been correctly changed.
Exceptions	In all cases CPMS will notify CPO employee 1. No DSOs satisfy the constraints imposed trough the checked flags 2. The chosen DSO refuses the new contract

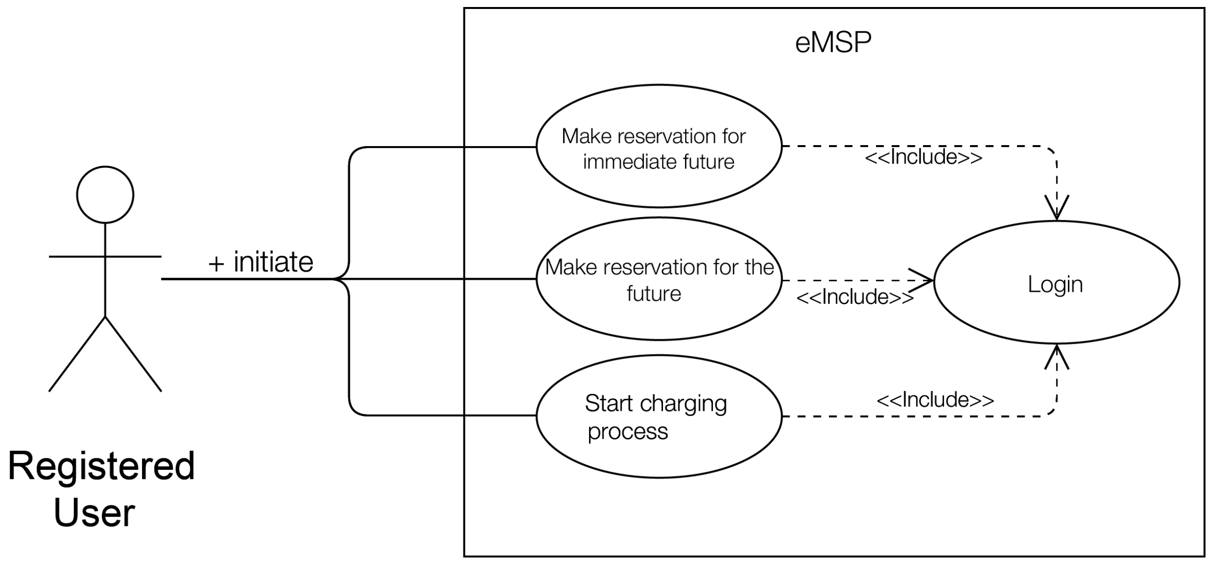
Manually overriding CPMS battery policy

Actor	CPO employee
Entry conditions	CPO employee is logged in the CPMS management software trough its corporate credentials
Event flow	1. The CPO employee select “manually battery management” 2. The CMPS shows the current level and status of the battery pack 3. The CPO employee insert an objective charge level to achieve through charge/discharge 4. The CPO employee submit
Exit condition	The CPMS make suitable decision in order to achieve the target
Exceptions	In all cases CPMS will notify CPO employee 1. Battery pack API report an error

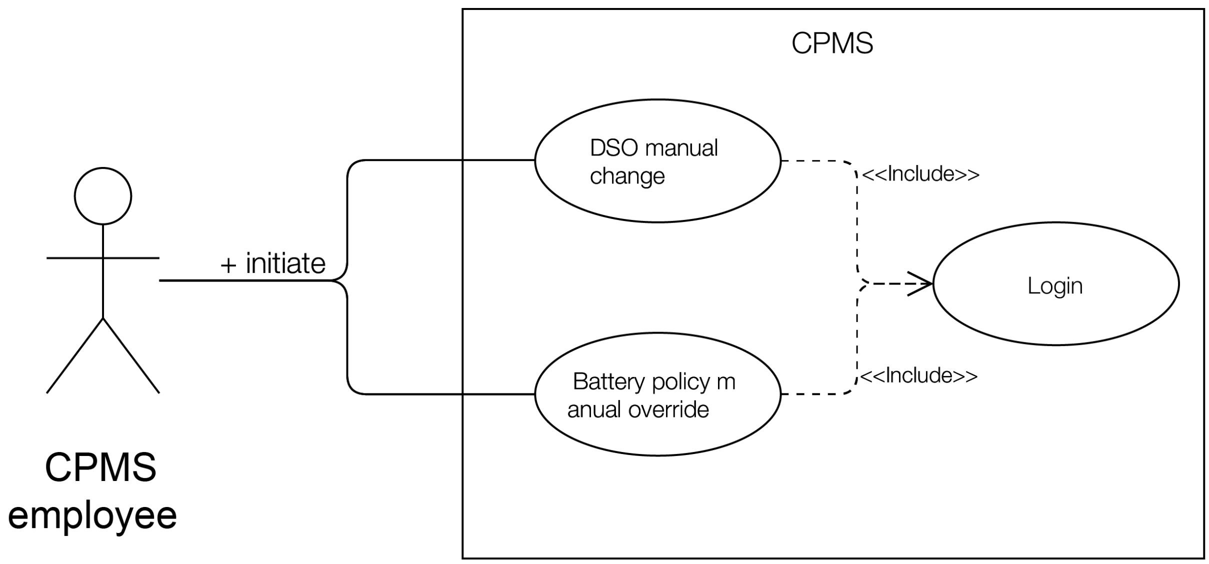
3.2.3 Use cases diagram



Use case diagram for eMSP unregistered user

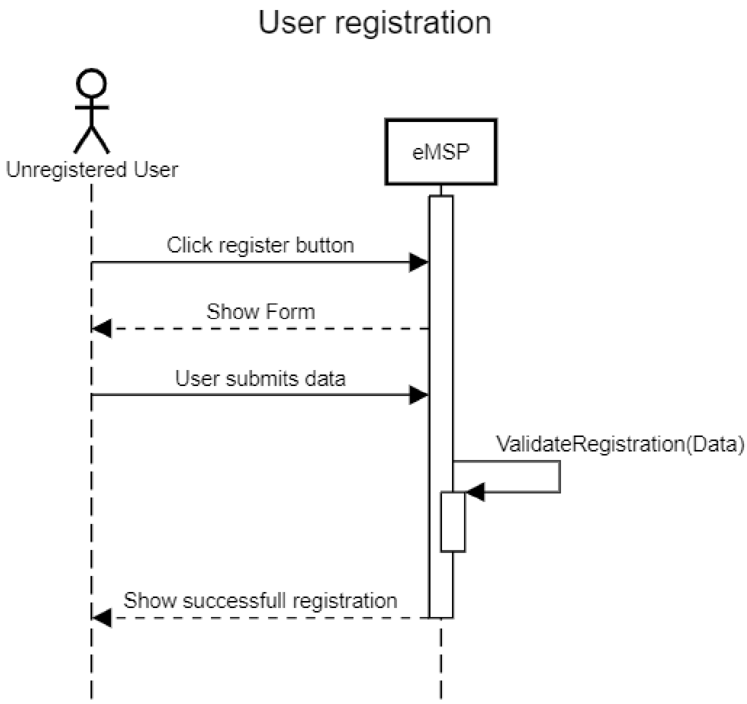


Use case diagram for eMSP register user

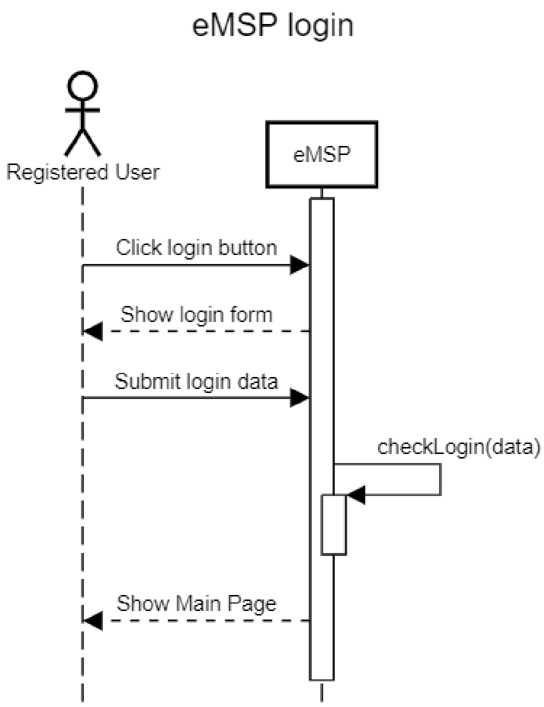


Use case diagram for CPMS employee

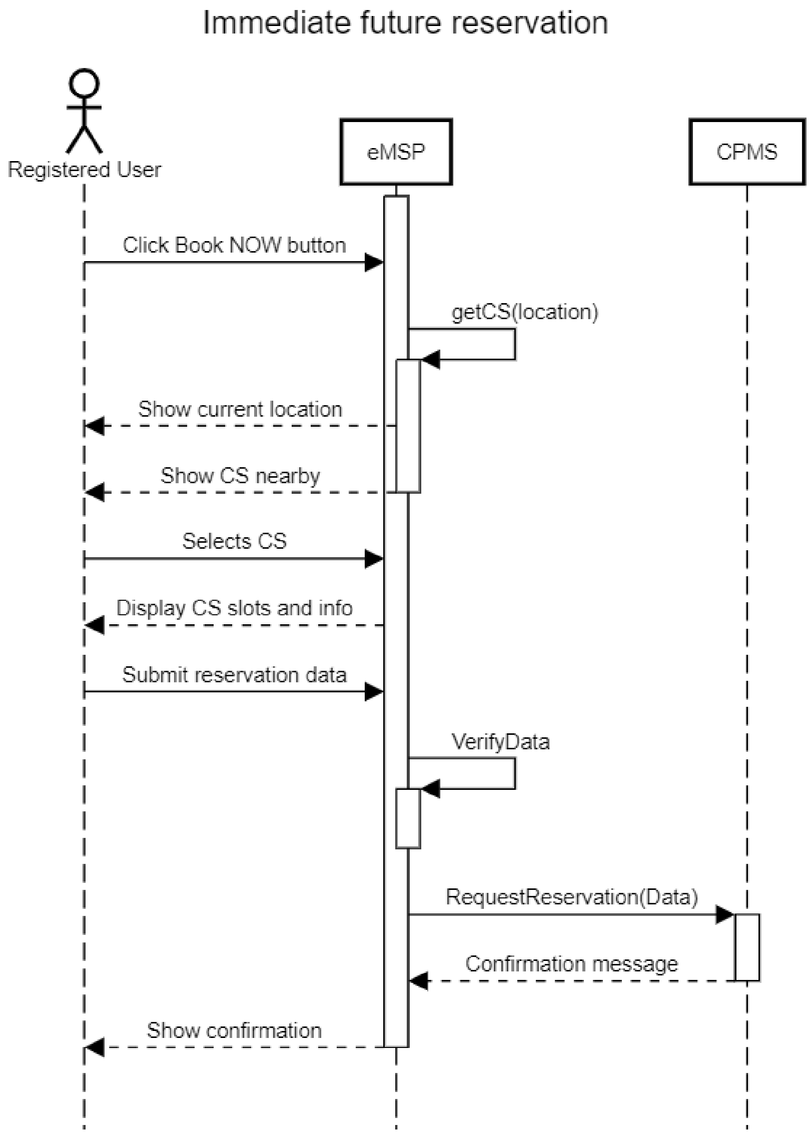
3.2.4 Sequence diagram



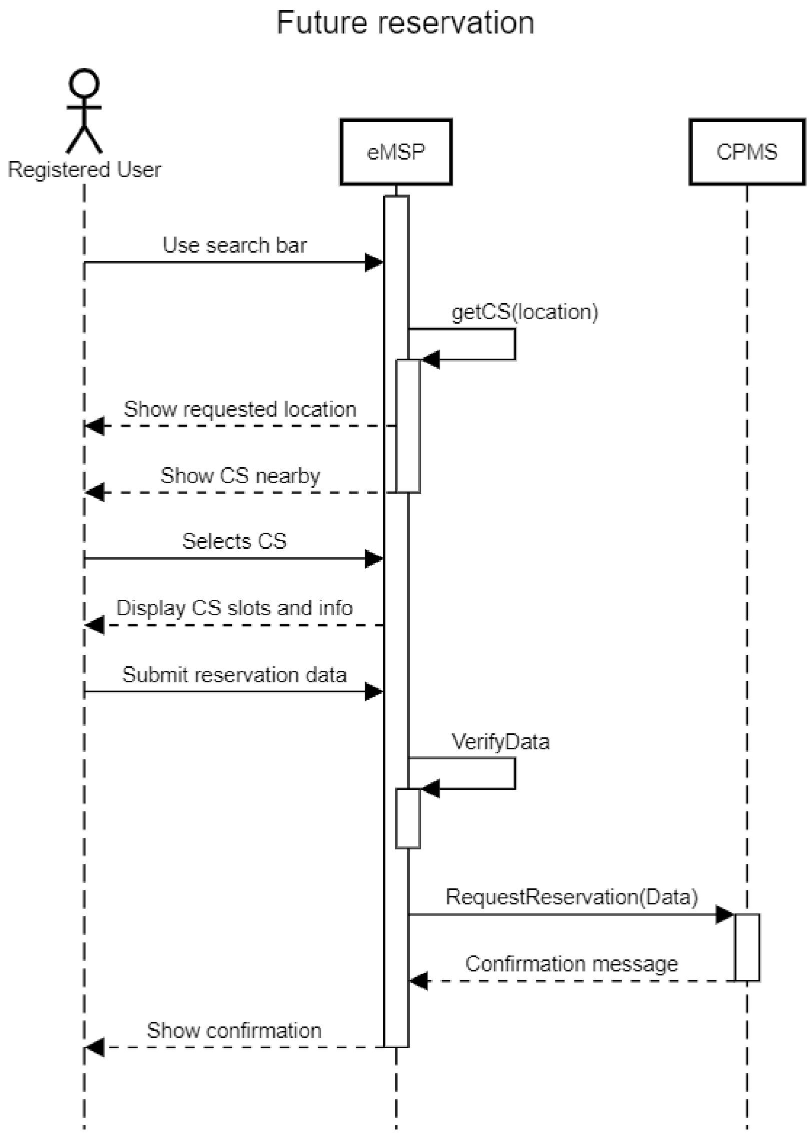
Sequence diagram for user registration



Sequence diagram for eMSP login

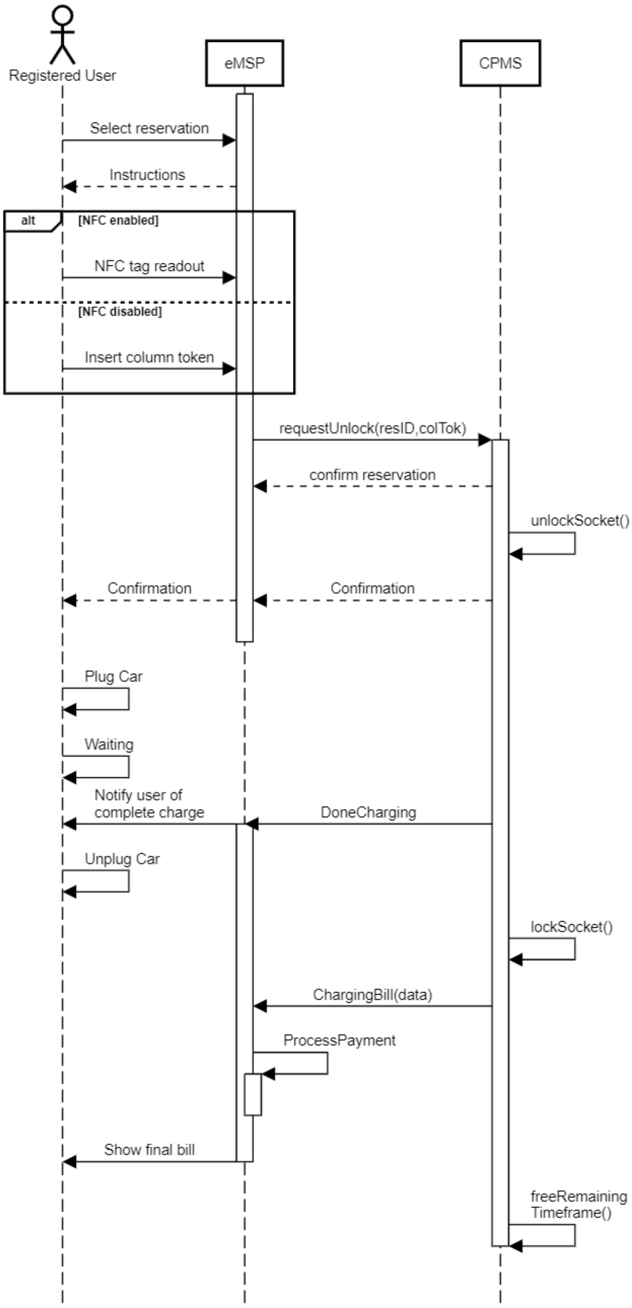


Sequence diagram for immediate future reservation

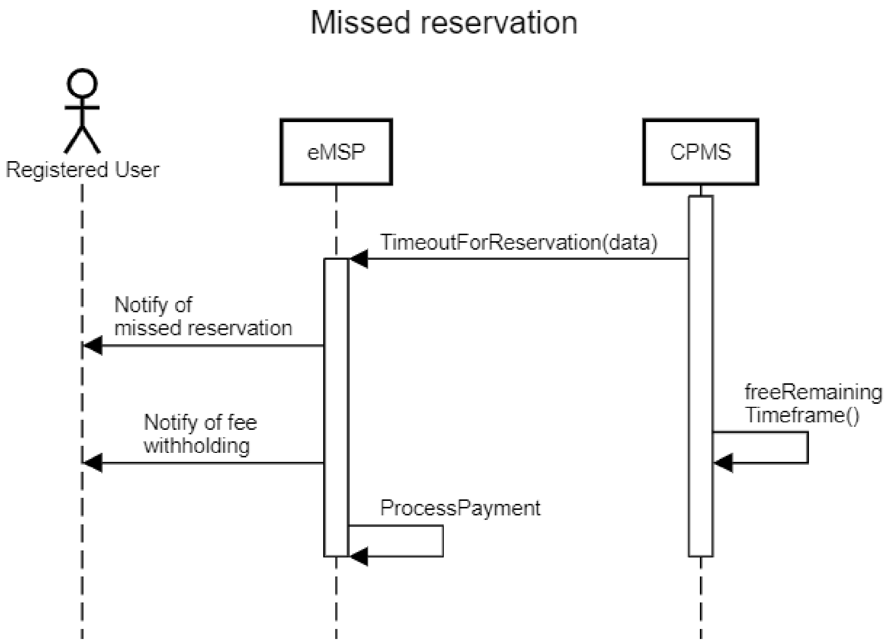


Sequence diagram for future reservation

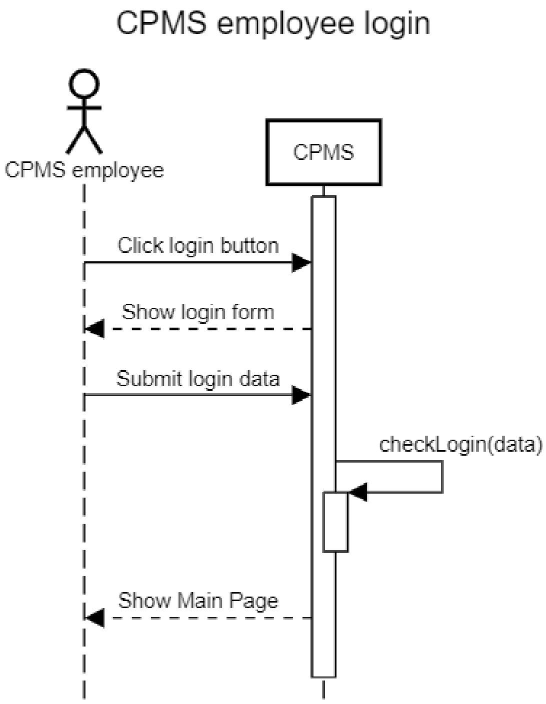
Charging process



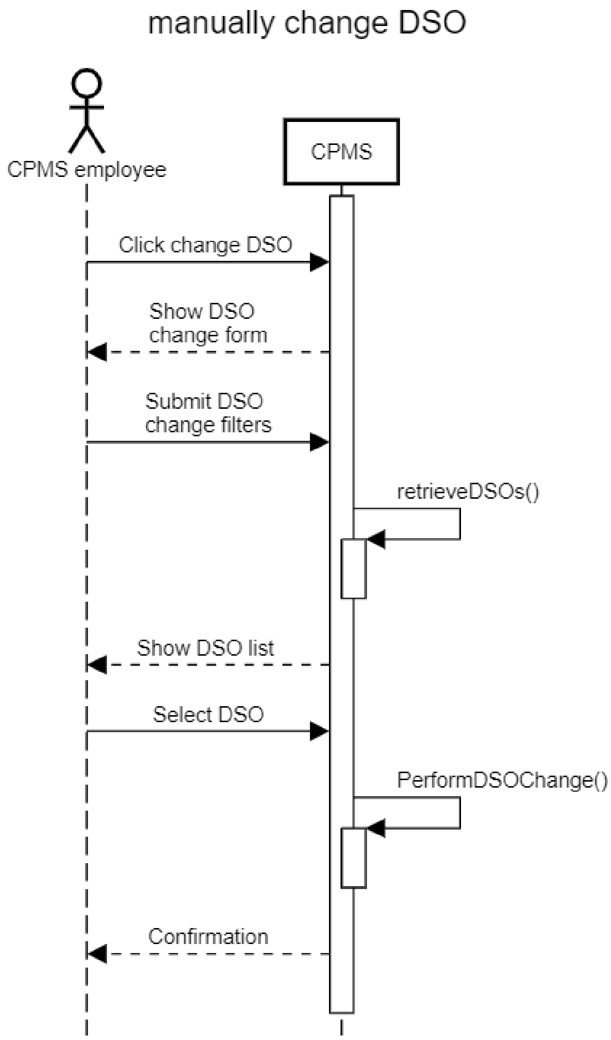
Sequence diagram for the charging process



Sequence diagram for a missed reservation

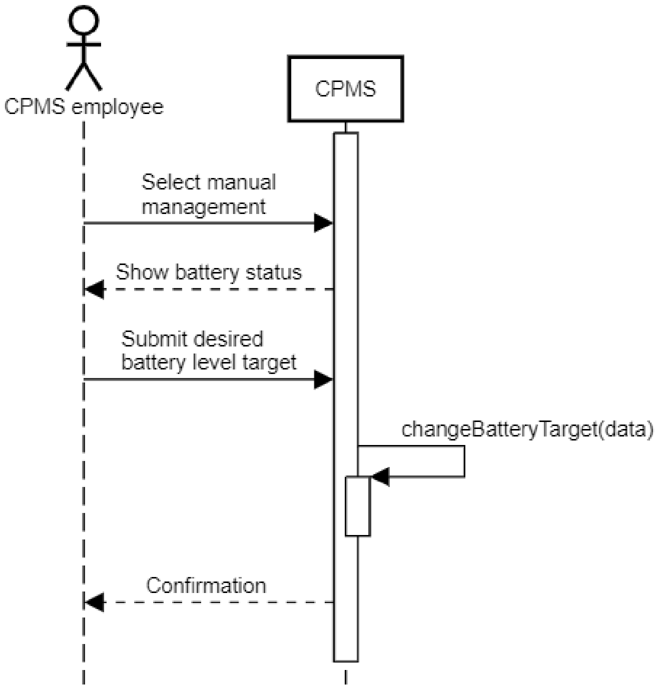


Sequence diagram for the CMPS employee login



Sequence diagram for manually change DSO

Modify CPMS battery policy



Sequence diagram for modify CPMS battery policy

3.2.5 Mapping on requirements

Use Cases	Requirements
eMSP sign up	R1, R3, R4, R5, R28, R29
eMSP sign in	R2, R28, R29
Making reservation for the immediate future	R6, R7, R20, R22, R28, R29, R30, R31, R32
Making reservation for the future	R6, R7, R8, R20, R22, R28, R29, R30, R31
Charging process with NFC	R9, R11, R12, R13, R14, R23, R31, R24, R25, R28, R29, R30, R31, R33
Charging process without NFC	R9, R11, R12, R13, R15, R23, R31, R24, R25, R28, R29, R30, R31, R33
Missed reservation	R10, R22, R28 R29, R30, R31
CPMS sign in	R26, R27, R34
Change DSO manually	R17, R26, R27, R35
Manually override CPMS battery policy	R16, R24, R26, R27

3.3 Performance requirements

With respect to the eMSP system, it should be available for both android and iOS at least, since the majority of mobile phones run one of this two operating system.
The eMSP app should be not particularly graphic intensive, since the information that needs to be displayed are just a few. In order to keep the information exchange between eMSP app and server low, since is not assumed that all users have access to high speed internet connection, most of the computation should be handled directly by the app.

The CPMS consists in two main subsystems: one that communicates with eMSPs and manages the reservations and one that manages the internal components. The first one should be able to manage an unbounded number of incoming request in a parallel way, so to make the responsiveness of the eMSP higher, since the payload of the system it relative to the geographic area in which it is deployed. The second one should be a lightweight system that can operate with high priority on the same hardware while not affecting capabilities of the concurrent processes.

3.4 Design constraints

3.4.1 Standard compliance

For one, all user data shall be treated in compliance with GDPR, or the equivalent privacy laws with jurisdiction in the state in which the user is located.

Secondly, the eMSP application should function fully on all widely used mobile phones operating systems.
Thirdly, the regulation and guidelines of external APIs must be followed

3.4.2 Hardware limitations

For the eMSP app the only limitation is that it works just for mobile phones able to access internet. For what concerns the CPMS the hardware must be able to satisfy the performance requirements assessed in the Performance requirement section.

3.5 Software System Attributes

3.5.1 Reliability

The eMSP app should have high reliability in order to ensure a optimal user experience. The eMSP server should also have high reliability and be up and running most of the time. In order to achieve this preventive regular maintenance should be performed and a duplication of the server could be run in parallel such that upon failure of one the other can be put in place.

The CPMS system should also have high reliability so to ensure safety in the charging stations. In order to achieve this a physical copy of the system should be available in case of failure of the main one. Also procedures to ensure correct management of emergencies should be put in place.

3.5.2 Availability

The eMSP app availability depends directly on the eMSP server and phone software. The eMSP server should be as high as 99.99% to ensure high usability of the system from a user perspective since this correspond to about 50 minutes of downtime yearly .

In similar way, the CPMS system should have an availability of 99.99%, since the high dependency with the eMSP for booking has a direct impact on income and the downtime should be minimal and used mainly for maintenance purposes.

In both cases, this availability level can be achieved through a duplication of the main servers and possibly of the databases.

It should also be noted that some of the functionality of the systems relies on external APIs out of our control, though the system should not completely fail because of failure in one of those.

3.5.3 Security

Since both eMSP and CPMS systems manage sensitive data as payment and location information, the security of the system is of paramount importance. In practice, all information exchange between systems and endpoints must be encrypted as well as stored data about customers.

3.5.4 Maintainability

Both systems should be easy to maintain also to achieve the above-mentioned level of availability. Moreover, both systems should grant the possibility of feature extensions. In order to achieve these goals, both system should be well documented and code should be as modular as possible, with comment to enable better feature extendibility.

3.5.5 Portability

With respect to the eMSP app, the software should be executable on more than 90% of devices. In order to do so, the software should be compatible with Android versions, going back to v8, and iOS versions, going back to v14.8.

Since the CPMS software executes on server-grade machines, it can be developed specifically for the required architecture and version.

4 Formal analysis

4.1 Alloy code

The formalization through Alloy code is here presented to highlight some of the constraints that should be imposed to achieve a satisfactory representation of the modeled world. With this model, is formally shown that the system is sound and consistent and the relations between classes and constraints results in a realistic representation.

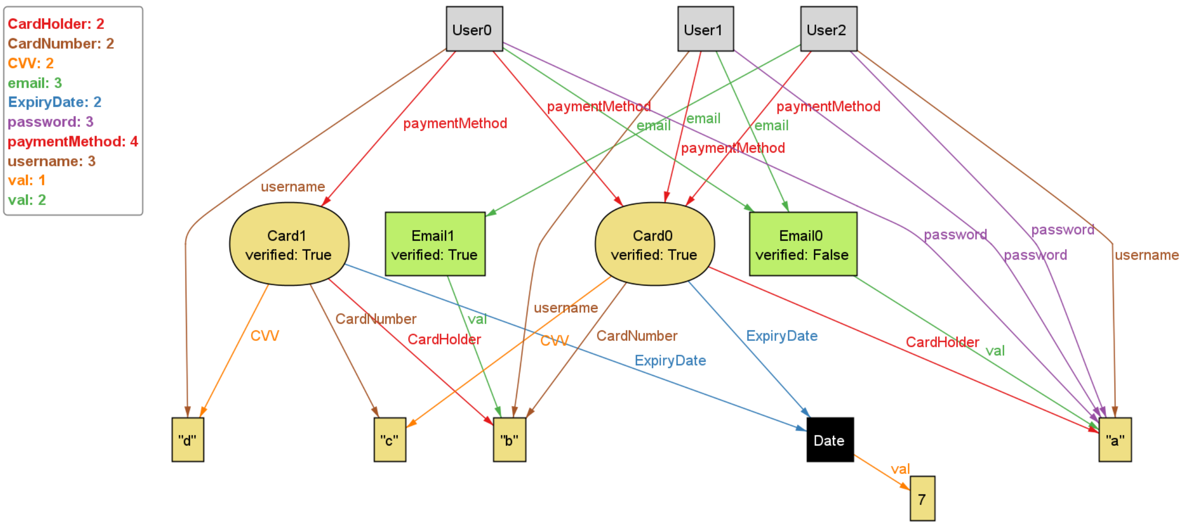
```
//--------Signatures---------

abstract sig Bool{}
one sig True extends Bool{}
one sig False extends Bool{}

abstract sig ChargeRate{}
one sig Slow extends ChargeRate{}
one sig Fast extends ChargeRate{}
one sig Rapid extends ChargeRate{}

abstract sig EnergyType{}
one sig Green extends EnergyType{}
one sig Carbon extends EnergyType{}
one sig Nuclear extends EnergyType{}

//simplified date signature
sig Date{
    val : one Int
}

sig Coordinate{
    lat : one Int,
    lon : one Int
}

sig Email{
    val: one String,
    verified : one Bool
}

sig Card{
    CardNumber: one String,
    CardHolder: one String,
    ExpiryDate: one Date,
    CVV: one String,
    verified : one Bool
}

sig User{
    username: one String,
    password: one String,
    email: one Email,
    paymentMethod: some Card
}

sig Reservation{
    time: one Date,
    startTime: one Int,
    endTime: one Int,
    chargeRate: one ChargeRate,
    user: one User,
```

```
    chargingStation: one ChargingStation
}

sig ChargingStation{
    location: one Coordinate,
    name: one String,
    isDiscounted: one Bool,
    energyCost: one Int,
    batteryLevel: one Int,
    batteryEnergyValue: one Int,
    chargingColumns: some ChargingColumn,
    supplier: one DSO
}

sig DSO{
    name: one String,
    energyCost: one Int,
    energyType: one EnergyType
}

sig ChargingColumn{
    type: one ChargeRate,
    isFull: one Bool,
    sockets: some Socket,
}

sig CPO{
    name: one String,
    employees: some CPOEmployee,
    chargingStations: some ChargingStation
}

sig CPOEmployee{
    username: one String,
    password: one String,
    workPlace: one ChargingStation
}

sig Socket{
    occupied: one Bool,
}

//----------Constraints----------

//all users must have verified mail and payment method to make a reservation
fact {
    all r:Reservation | (r.user.email.verified in True and r.user.paymentMethod.verified in True)
}

//no two charging stations can be at the same location and all coordinates must be in at least one of the charging stations
fact{
    all disj c1,c2:ChargingStation | c1.location != c2.location
    all c:Coordinate | c in ChargingStation.location
}

//all mails must be in at least one user
fact{
    all e:Email | e in User.email
}

//all Cards must be in at least one user
fact{
    all c:Card | c in User.paymentMethod
}

//all Dates must be in at least one reservation or card
fact{
    all d:Date | d in Reservation.time or d in Card.ExpiryDate
}

//no date can exist without a reservation or a DebitCard
fact{
    all d:Date | (some r:Reservation | (r.time = d)) or (some r:Card | (r.ExpiryDate = d))
}

//each chargingColumn is always in one and only one chargingStation
fact{
    all c:ChargingColumn | no disj c1,c2:ChargingStation | c in c1.chargingColumns and c in c2.chargingColumns
    all c:ChargingColumn | c in ChargingStation.chargingColumns
}
```

```
//no two CPOEmployees can have the same username
fact{
    all disj e1,e2:CPOEmployee | e1.username != e2.username
}

//each employee must work for one and only one CPO
fact{
    all e:CPOEmployee | no disj c1,c2:CPO | e in c1.employees and e in c2.employees
    all e:CPOEmployee | e in CPO.employees
}

//if all charging stations sockets are occupied, the charging column is full
fact{
    all c:ChargingColumn |  (no s:Socket | s in c.sockets and s.occupied in False) iff c.isFull = True
}


//all sockets must belong to one and only one charging column
fact{
    all s:Socket | no disj c1,c2:ChargingColumn | s in c1.sockets and s in c2.sockets
    all s:Socket | s in ChargingColumn.sockets
}

//all startTime must be greater than endTime
fact{
    all r:Reservation | r.startTime < r.endTime
}

//a reservation can be made only if the charging station has a charging column of the required type
fact{
    all r:Reservation | r.chargeRate in r.chargingStation.chargingColumns.type
}

//users mail and username must be unique
fact{
    all disj u1,u2:User | u1.email != u2.email or u1.username != u2.username
}

//Strings exist
fact{
    all s:String | s in "a"+"b"+"c"+"d"
}

//all CPOEmployees must work in a charging station that belongs to the CPO they work for
fact{
    all c:CPO | c.employees.workPlace in c.chargingStations
}

//all charging station must belong to one and only one CPO
fact{
    all c:ChargingStation | no disj c1,c2:CPO | c in c1.chargingStations and c in c2.chargingStations
    all c:ChargingStation | c in CPO.chargingStations
}

//Assertions to be verified

assert Check_CC_Status{
    all c:ChargingColumn | c.isFull = False
    all s:Socket | s.occupied = True
    #sockets > 0
}

//------------Predicates----------

//dynamic predicates

//add a new card to user
pred addCardToUser[u,u':User, c:Card]{
    u'.paymentMethod = u.paymentMethod + c
}

//add a new reservation
pred addReservation[r,r':Reservation, cr:ChargeRate, cs:ChargingStation, u:User,]{
    r'.chargeRate = cr
    r'.chargingStation = cs
    r != r'
}

//change DSO for a charging station
```

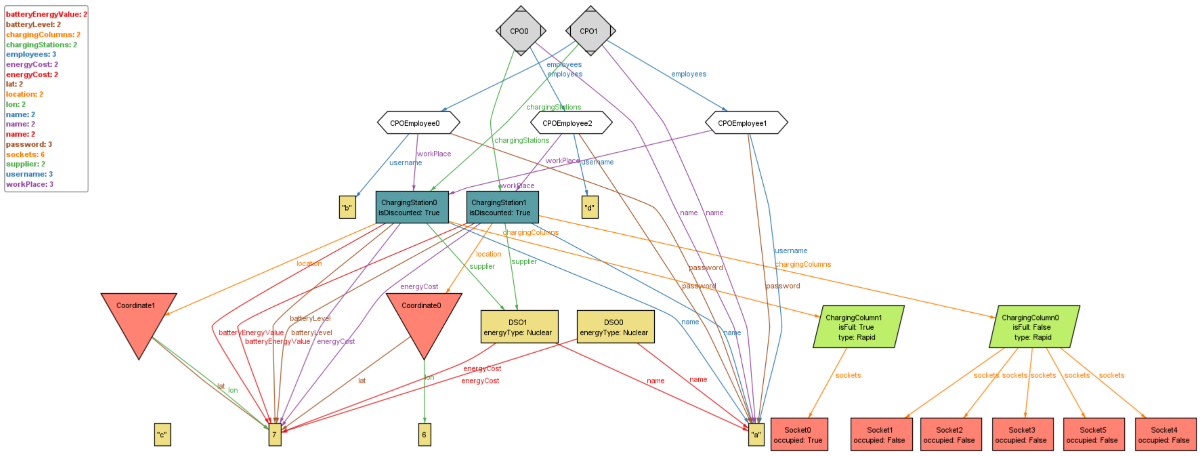
```
pred changeDSO[c:ChargingStation, d,d':DSO]{
    d != d' and
    d.energyCost >= d'.energyCost and
    c.supplier = d'
}

//Charging station changes price
pred changePrice[c:ChargingStation, p,p':Int]{
    c.energyCost = p and
    p != p'
}

//----------Worlds-------------
pred eMSPWorld{
    #ChargingStation = 0
    #DSO = 0
    #Reservation = 0
    #User = 2
    #Card = 3
    #Email = 1
}

pred CPMSWorld{
    #ChargingStation = 2
    #DSO = 2
    #Reservation = 0
    #CPO = 2
    #CPOEmployee = 3
    #Socket = 6
}

pred World{
    #Reservation = 2
    #User = 2
    #Card = 1
    #Email = 2
    #ChargingStation = 2
    #ChargingColumn = 2
    #Socket = 3
    #DSO = 1
}

//------------run------------

run eMSPWorld for 20
run CPMSWorld for 20
run World
```

4.2 eMSP world



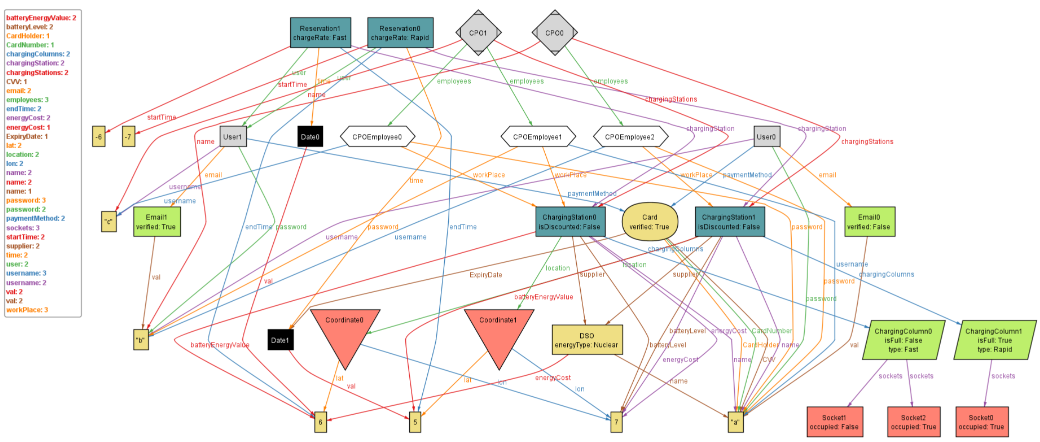
In this representation we can see how the eMSP world is composed of just users and their respective credentials

4.3 CPMS world



In this representation we can see that the EMSP world is composed of CPOs with their relative employees and charging stations which in turn have their own charging columns, DSOs and personell

4.4 Complete world



In the representation of the full world the eMSP and CPMS world are joined by the reservations witch connect users and charging stations each with their relative attributes and connected entities

5 Effort

5.1 Riccardo Bravin

Task	Time spent
Introduction	7h 30 min
Overall description	10h
Specific requirements	16h 30 min
Formal analysis	7h
Reasoning and writing	6h 30 min
Total	**47h 30 min**

5.2 Elia Feltrin

Task	Time spent
Introduction	7h 30 min
Overall description	10h
Specific requirements	16h 30 min
Review	3h
Reasoning and writing	7h
Total	**44h**