RASD e-Mall

1 Introduction

1.1 Purpose

The main challenge of a massive deployment of electric mobility is the reduction of transportation's impact on climate by limiting the carbon footprint caused by our urban and sub-urban mobility everyday needs.
With this in mind, our aim is to develop and implement a new system called eMall having the purpose to expand the electric charging infrastructure with the means of realizing a world-wide network which fullly connects all the actors involved in the charging processes. This is achieved by making all the back-end operations and transactions transparent to the everyday user which is fully supported in the charging process.
eMall will act as a platform deployed as a mobile application and implemented through several servers (eMSPs) located throughout the globe allowing to monitor electric mobility by communicating with various CPOs, each of them administrated through a CPMS.
Thereby the platform with expose several services such as:
- knowing where to charge the e-vehicle (locate charging stations owned and managed by CPOs)
- planning charging processes in a way to limit constraints on our daily schedule giving the possibility to the system to access our personal calendar
- choose from various charging possibilities based on special offers set by CPOs which can dynamically decide from where to acquire energy to be distributed (from which DSO)

In order to guide the development step by step, the document will focus on the goals and requirements of the system to achieve.

1.1.1 Goals

Sub-system = eMSP (e-Mobility Service Provider)
Sub-system = CPMS (Charge Point Management System)

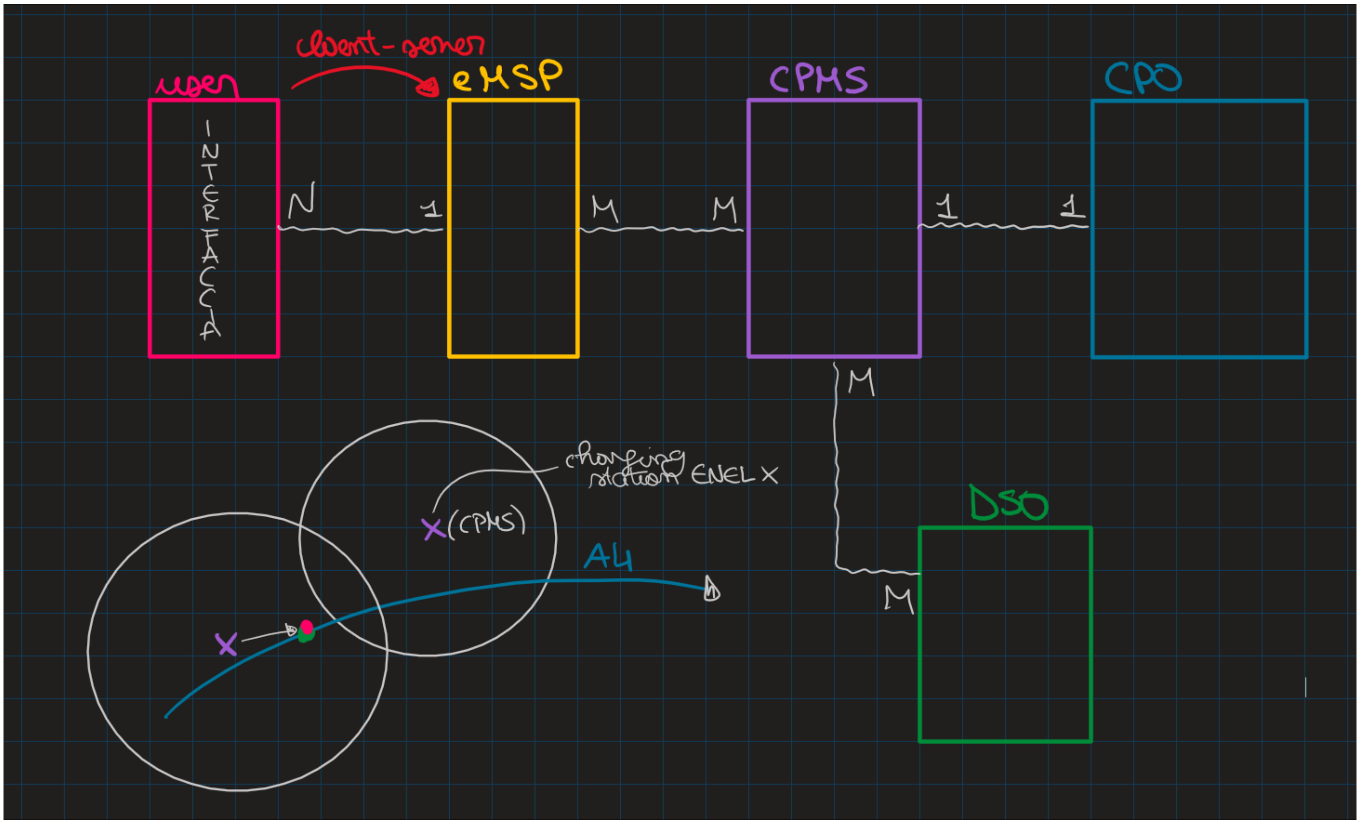
- G1: The eMSP should allow the user to obtain information about the charging stations.
- G2: The eMSP should allow the user to book a charge in a certain charging station before the vehicle runs out of energy.
- G3: The eMSP should allow the user to manage the charging process (from its start to the payment) once he/she reaches the charging point.
- G4: The eMSP should give the possibility to the user to receive updates on the charging process.
- G5: The eMSP should show the user the optimized path to the destination based on the residual energy of the vehicle.
- G6: The CPMS should be able to handle the acquisition of energy from external third party providers (DSO).
- G7: The CPMS should be able to gather information about the DSOs' current price of energy.
- G8: The CPMS should store informations about the location, external and internal status of a charing stations. ("external/internal").
- G9: The CPMS should provide energy according to the type of socket chosen by the user.
- G10: The CPMS should decide where to get energy from (station batteries, DOS, mix).

1.2 Scope

This document focuses on the interaction between the most important stakeholders, even though several ones might be left undefined. In particular we consider the users, which might want to use the services provided by the system, the CPOs (e.g. ENEL X, IONIX or others) physically deployed through charging stations and administrated by ad hoc pre-implemented CPMS sub-systems, and DSOs, external third party energy providers. The system we are focusing on is the e-MSP, a remote sub-system (passive server) which receives requests from the several users and uses uniform APIs to interact with other external sources such as CPMSs and other systems.

/**

- Registered USER
- e-MSP (e-Mobility Service Provider) is the remote sub-system (passive server) called by the user which gathers informations by contacting CMPSs through uniform APIs.
  -which uses the exposed interfaces by CMPSs and used by users on their own local app instances.
- CPO (e.g. ENEL X/IONIX) (Charging Point Operator) administrated through CPMS and deployed through:
  - Charging stations
- CPMS (Charge Point Management System) administrates CPO's IT infrastructure by managing the relationship between the energy acquired by a DSO and the charging vehicle connected to the charging station's sockets.
- DSO (Distribution System Operators): external third party energy providers.
  /**



1.2.1 World Phenomena

- WP1: User's e-vehicle is running out of energy.
- WP2: User physically looks for an available charging station.
- WP3: Charging station's socket doesn't work properly.
- WP4: Catastrofic environmental disasters (earthquakes, avalanches, hurricanes, floods, heavy storms).
- WP5: Accidental damages occured on the infrastructures.
- WP6: User prepares the vehicle (parks, open the charging socket) for the charing process.
- WP7: There is a fully geographical network coverage and connectivity to the system is ensured.

1.2.2 Shared Phenomena

#todo TODO: system = sub-system + sub-system or not
#todo Insert new Controller/Observer columns
SP1: The charging station doesn't provide the guaranteed services presented by the eMSP. (Discrepancies between the real-world situations and the services

CPO intends to guarantee)
SP2: The user looks up for a nearby charging station.
SP3: The user chooses the best charging station from the available ones based on its preferences.
SP4: The system shows the user informations about the charging process.
SP5: The system shows the user several payment methods.
SP6: The system automatically decides from which DSO energy should be acquired.

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

#todo

1.3.2 Acronyms and abbreviations

eMall - Electric Mobility for All
RASD - Requirements Analysis and Specification Document

WPx - World Phenomena
SPx - Shared Phenomena
Gx - Goal number X
Dx - Domain assumption number X
Rx - Requirement number X
e-MSP - e-Mobility Service Provider
CPO - Charging Point Operator
CPMS - Charge Point Management System
DSO - Distribution System Operator
API - Application Programming Interface
VIN - Vehicle Identification Number
COTS - Component Off The Shelf
RAFT - Algorithm regarding replication and consistency for distributed systems.
MTTR - Mean Time To Repair

EV - Electric Vehicle
DMV - Department of Motor Vehicles

1.4 Reference Documents

The specification document: "Assignment RDD AY 2022-2023_v3.pdf"

1.5 Document Structure

#todo ..

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

1. Unregistered user wants to use e-Mall
    1. Registering by adding:
    1. Personal information
    2. Associating Vehicle : e-Mall System with VIN (assumption: integration with the Department of Motor Vehicle databases for person:vehicle association AND integration with Car Manufacturer's database for vehicle's real time data and details)
    3. Access its profile and consult next charges

    Description:

    The user Alice has just bought an electric car and wants to use e-Mall. She can only do so if she is registered, so through the index page of the corresponding site she fills the registration module by entering her personal information and vehicle details. e-Mall is integrated with the DMV's database to confirm the association between vehicle and owner, and with vehicle manufacturers' database to associate the vehicle owner with the vehicle details. After registering, Alice can finally use all the services offered by e-Mall: searching for charging stations, booking charging stations, scheduling trips and managing the charging process. Every time she uses the app data will be updated so that she is in control of any information, directly from the app. Now Alice can get in her car and make her first recharge thanks to e-Mall.

2. **User looking up for charging stations (+filters)**

    1. User wants to know the position of charging stations so selects SEARCH section of APP
        1. based on current location
        2. at a certain position
    2. Filtering and Sorting (ASC, DESC)
        1. range of distance from current or remote position
        2. range of price
        3. preferred CPOs
        4. preferred payment methods
        5. availability of sockets
        6. type of sockets
        7. special offers (range of sales percentage)

        Description:

        The user Cassandra is at home which is 10km away from the main town. She wants to charge her electric vehicle so she accesses e-Mall app from her smartphone by using her fingerprint. From the Home Page she selects the Searching Page to which she's being redirected. She can now choose to either let the system find the most convenient charging station based on the information about the vehicle's status (park location, battery level) acquired by e-Mall or to insert preferences (range of distance from current or remote position, range of price, preferred CPOs, preferred payment methods, availability of sockets, type of sockets, special offers) about the searching operation on her own. Cassandra wants to go get some groceries from the town centre so she inserts the destination of the grocery shop and wants to retrieve the most economic (range of price) nearby charging stations within

200m from it (distance range filter). The system provides a list of three charging stations and Cassandra decides to consult them from the map point of view. Two of them are availble but characterized by a slightly high cost, meanwhile the last one is occupied but the system estimates that the charging process is about to end within the next 15 minutes, exactly the time Cassandra needs to reach the location. Furthermore, this last charging station is managed by a CPO providing a special offer lasting for other few hours. Cassandra reasons about the next step.

3. **User booking a charge**

1. Selecting a certain charging station from the list provided
2. Reads infos about the charging station
2. Selects a timestamp for the booking and make a reservation
3. Requires booking
4. Receiving ack for the booking or notifying a failure (overlapping of the timestamp with other events in the calendar) and requiring other timestamps.

Description:

The user Mario notices that the car battery is not fully charged, so he decides to charge it. After searching for the charging stations nearest to him based on the search criteria (filters) entered, he chooses one, thus obtaining more information regarding the charging station (available sockets, price, special offer,...). If the information convinces Mario, he makes the reservation by inserting the initial and end time of the charging process. He clicks on the BOOK button, and if he receives a notification of successful booking, this means that he can reach the charging station and the book is added to the user's calendar, otherwise, Mario will receive a notification with the error that occurred (eg. at that time on his calendar he has an appointment so you have an overlap), in which case Mario will have to choose a different timestamps.

4. **User scheduling a travel**

1. Select TRAVEL section of the APP
2. Inserts destination and time of travelling.

3. System calculates the best path using certain algos (e.g. Dijkstra, Bellman-Ford) and possibly based on battery level if requested by the user and shows it to the user
4. User confirms or declines the proposal.

Description:

The user Vittorino has to travel from Milan to Budapest for a business trip the next Monday. He is registered to the e-Mall app which has updated information about Europe's most recent electric mobility infrastructures. From Home Page he clicks on the Travel Button which immediately redirects him to the Travel Page. Vittorino inserts the destination he wants to reach, the day and time by which he needs to arrive and suggests the app to assume the vehicle will be fully charged by the time he leaves. The app calculates two convienent routes (using algorithms like Dijkstra or Bellman-Ford) Vittorino could choose from based on the time he wants do start the trip but also detects an overlap between the trip time and a certain event inserted onto the user's calendar. A notification is showed to Vittorino which clearly forgot to reschedule the event on the calendar and decides to continue to choose from the routes proposed anyway, ignoring the notification. Later on he will reschedule his event.

5. **User's vehicle is running out of battery**

1. System detects that battery is running out of ernergy
2. System notifies the user in order to go an charge the vehicle suggesting the most convenient one.

Description:

Mrs. Braz is on her way to Portugal to visit her family. After a few hours of driving, Mrs. Braz is focused on the road and does not notice that her car's battery is getting low, so having arrived at the remaining 10% the system recognizes that the battery is getting low and approaching zero, so it searches for the most convenient charging stations close to the car's current location, and sends a notification to Mrs. Braz advising her to recharge the vehicle at the recommended charging station as soon as possible.

6. **Charging process executing (failure managements)**

1. User reaches the charging station
2. User takes the proper socket and connects it to the vehicle
3. User selects the percentage to achieve
4. User selects Start on APP to start the charging process and selects target battery level or duration of the charge.
5. Different sources (battery, direct (DSO deciding), mix (deciding), deciding DSO)

Description:

The user Nicola has booked a charge at a charging station through e-Mall. He reaches the charging stations, parks his car, and connects the socket. Once he connects the socket he receives a notification with the details of his reservation and with the option to start the process. He chooses to start the process and in real time, both from the app and on the display of the charging point, he receives details about the status of the charging that is taking place according to the strategy defined by the CPMS (take advantage of the battery inside the charging point, or use an external DSO). As soon as it reaches 100%, charging is complete and Nicola can disconnect the socket and leave the area, after having payed for the service. However, if some unforeseen event happens during the charging process (the socket is accidentally disconnected, the vehicle's motor is turned on, or the charging point has problems), the charging is suspended and details of the problem appear on his own device with options to fix or report what happened to the CPO, or even stop the process and pay only for how much energy was dispensed. Nicola is very pleased with such a smart process and is looking forward to charging his vehicle again by using e-Mall application.

7. **Payment**

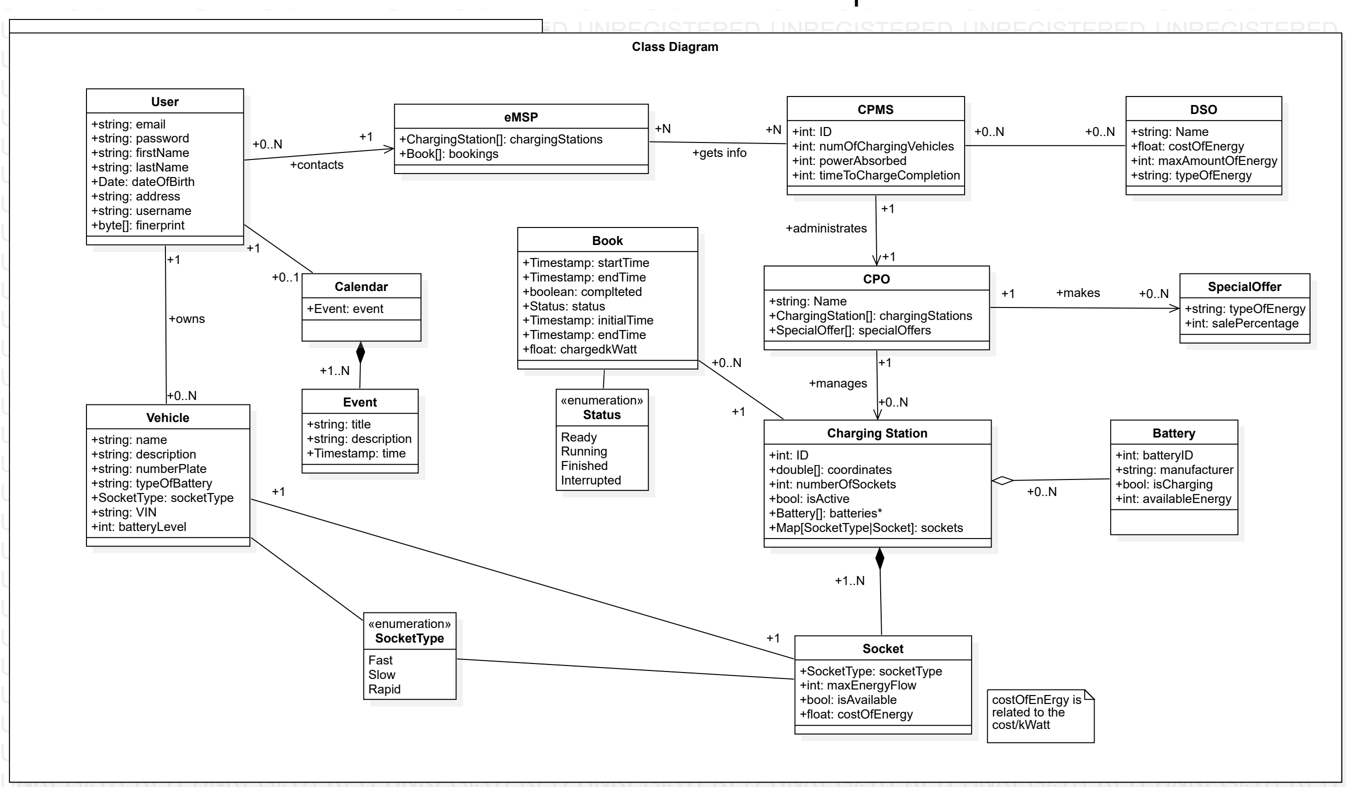
1. Several methods (+ Apple/Google/Samsung payments)
2. 3DS secured methods

Description:

The user Stefan is taking a walk while he is waiting for his vehicle to be fully charged at the charging station. Suddenly his phone rings and he notices a notification from e-Mall app has been received. Stefan accesses the app by using his fingerprint. He then realizes his vehicle is up about to reach the desired battery level decided at the booking phase and decides to reach the charging station. Finally continues using the app and selects one from the payment methods provided, all supported by 3DS secure methods. He waits for the transaction to be completed but a failure occurs and the app suggests to choose another payment method. He then realized his card just expired two days ago and decides to use Paypal in order to complete the payment. Once the transaction has finally completed the app suggests Stefan to disconnect the socket and leave as soon as possible as another charging process is about to get started within the next 10 minutes at that location.

2.1.2 Class Diagram

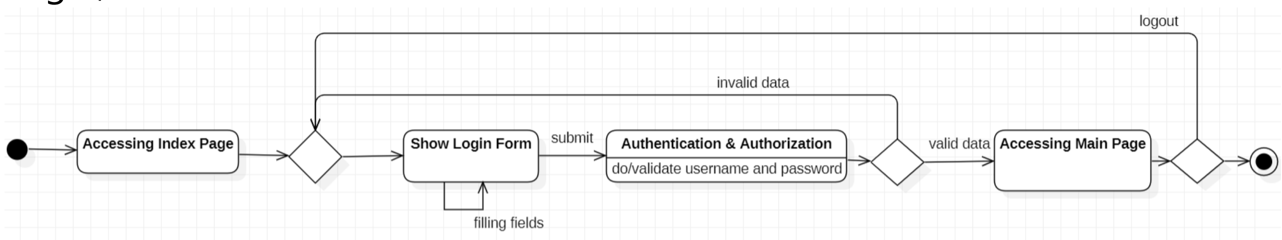
The below class diagram shows the system's main objects class diagram.
Note: The attributes of the objects are set as public even if some of them should be assigned as private. Setters and Getters implementation must be provided in order to set attributes' values and retrieve them as pleased.



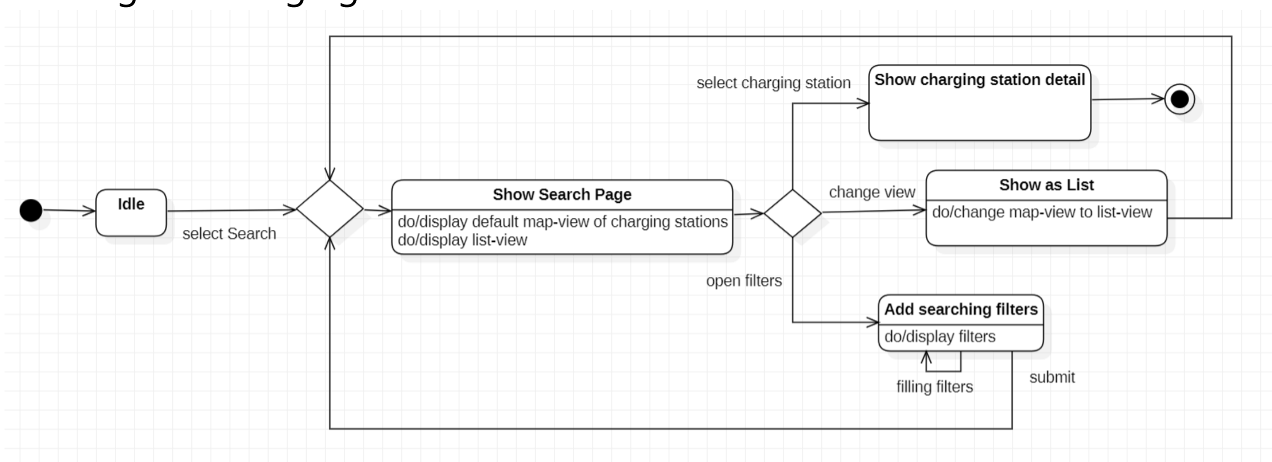
2.1.3 State Diagrams

This section is focused on the statecharts of some aspects of eMall system

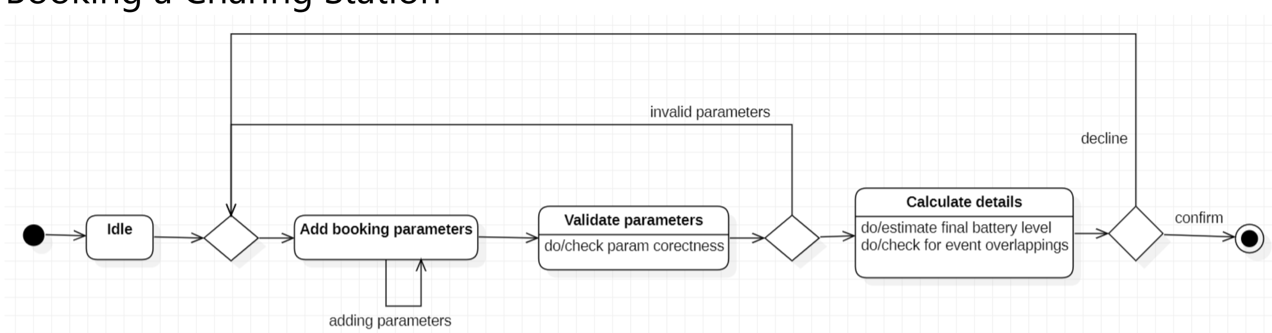
1. Login, Authentication and Authorization



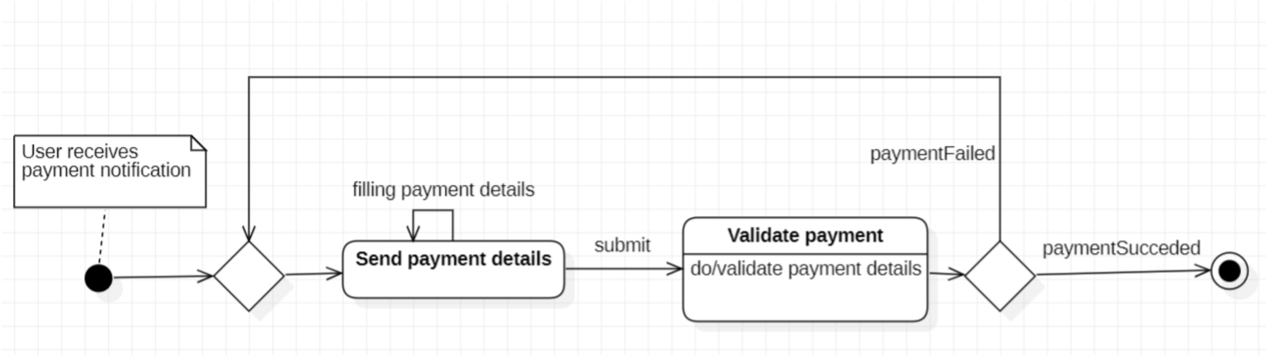
2. Looking for Charging Stations



3. Booking a Charing Station



4. Payment



2.2 Product functions

In this section we present the main functionalities introduced by e-Mall.

2.2.1 Identification of the nearby charging points

One of the most important aspects of the system is the identification of charging stations.
The search for charging stations can be carried out in different ways, quickly, by using the GPS, which returns a list of the nearest charging stations to the vehicle, by applying filters to the search, such as the maximum distance the charging station must be from the vehicle, or by choosing a charging station anywhere on the map.
Once the user has received a list of the nearest charging stations, he can see details of each charging station such as the price of charging, the types of sockets that are available, and some other detailed information about the charging station.
If the information on the charging station convinces the user then he can carry out the next booking operation, otherwise he can return to the list of charging stations to look in detail at other charging stations and choose the one he prefers.

2.2.2 Efficient and optimized charging station's booking system

The booking process is another key aspect of the system; after looking at the charging stations, there is the booking phase.
The system allows the user to choose a start time and an end time for recharging, also estimating the battery level that will be reached at the end of recharging based on the types of sockets of the station, so that the user can choose to book for a longer recharging time if he/she needs it.
Once the time has been chosen, the system checks the user's calendar, to which it has access thanks to the smart function implemented on the system, looking for overlapping events. In the event of overlapping the system notifies the user of the presence of simultaneous events and allows the user to choose another time slot.
If there is no overlapping, the user can continue by finishing the booking process, and can reach the booked charging station to start recharging.

2.2.3 Interoperability and connectivity ensured for all the Infotainment systems (cars manufacturers)

The eMall system is fully connected with the infotainment systems of various EVs, due to the integration with various car manufacturers and full support for Android Auto or Apple Car. Connection to the system is ensured through the mobile network connection, to be assumed reliable, of the smartphone associated to the car and the GPS antenna.

2.2.4 Fault tolerance towards operations

Since some operations may occur under precarious network and geolocation conditions, the system implements a mechanism to handle client-side faults: from re-transmitting data, to waiting for a more stable connection. On the server side, infrastructure redundancy and geographic distribution of machines allows for greater up-time and response to requests from outside, as well as disaster recovery.

2.2.5 Sharing knowledge through the network among participants

The system will provide the possibility to share your charging experience at a certain charging station by providing a rating along with a comment once the charging process has finished or interrupted by the user. This will enhance or decrease the charging station visibility on the map based on the level of agreement between the users who took advantage of its services. The overall rating will be periodically notified to the specific CPO which will analyse the data and decide how to further proceed. The rating is totally subjective and could be influenced by the environment, the access to the station or the nearby infrastructure such as restaurants, shops or other activities. All this data will be used to enforce the the system and upgrade it based on the users requests.

2.2.6 Collect and share information for Data Analytics purposes

The system collects data about users as long as it can track their movements and their decisions. The eMall back end will have a hidden-to-the-user component which will be given the function to analyze data coming from different users, to profile them and to compute meta-data and statistics about system's efficiency and find ways to improve the effect on the everyday user's routine.

2.3 User characteristics

1. Unregistered user
   A certain user needs to register to the system in order to use the functionalities provided.
2. Registered user
   A registered user who is able to improve its electric mobility routine by using the functionalities provided by the system such as looking for a charging station from anywhere and at any time.

2.4 Assumptions, dependencies, constraints

2.4.1 Domain assumptions

- D1: There exist uniform APIs allowing the user to interact through the eMSP with one or multiple CPOs.
- D2: There exist uniform APIs allowing the CPOs to interact with one or multiple DSOs through the CPMS.
- D3: The user inserts valid data when performing the registering phase.
- D4: The user approaches the charging station once he/she has book a charge within a certain time.
- D5: The user interacts with the charging station by following the instructions attached to the charging station.
- D6: The user interacts with eMSP by following the guideline provided by the eMSP.
- D7: Absence of inconsistencies between the a-priori defined energy flow offered by the DSO and the actual one used to charge the vehicle.
- D8: The user pays the booked service after the charging process has finished.
- D9: The user schedules a travel to a timestamp greater than the current one.
- D10: The system works properly in absence of inconvenient and unexpected events during travel.
- D11: The user frees the charging station once the charging process has finished.
  #todo

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

The eMall user interface is an application used by:

- a non-registered user, who can register by providing the information requested during the registration process
- a registered user, allowing him to log in by entering the credentials used during registration, or by using the fingerprint

- a logged-in user, allowing him to use the various functionalities provided by the system, such as searching and booking a charging station
  The system must be designed to run both on the smart-phone and on the vehicle's infotainment system.
  It should therefore be simple and user-friendly, so that it can also be used while driving and by any type of person.

3.1.2 Hardware Interface

From a hardware point of view, the system must be able to receive input data, to perform login, registration, etc., and display information on the screen such as the nearest charging station and battery charging status.

The system is interconnected with the various charging stations and so therefore knows their location by the means of GPS technology. This requires Internet connection in order to locate the vehicle worldwide and suggest the nearest charging station to reach.

The system must be connected to the EV's battery as it must display the battery charge status, during charging, and in the event of a low battery, to send a notification to the user in order to remind him/her to reach a charging station as soon as possible. The system-battery connection could be performed by using a black-box hardware device directly connected to the battery which main function is to acquire real-time data and send it to the system.

3.1.3 Software Interface

#todo

3.1.4 Communication Interface

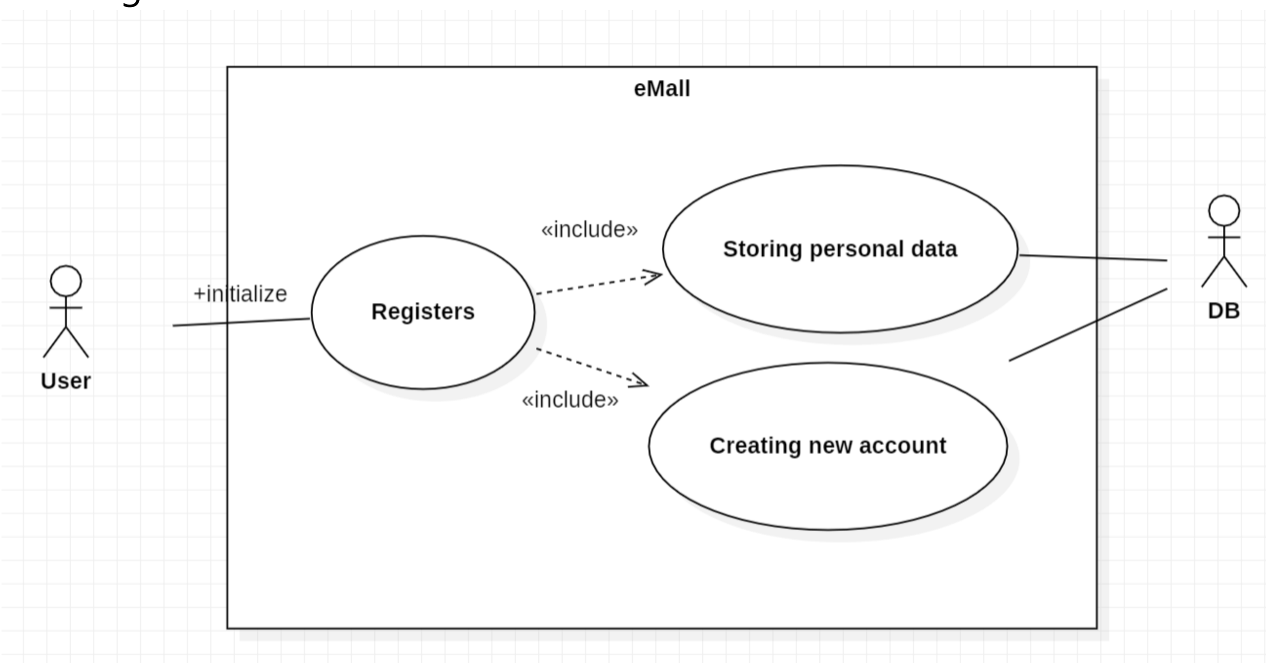
Most of the functionalities provided by the system return information, and only the registration, login and filter search operations allow the user to enter data. There are 4 different interfaces that eMall utilizes:

- retrieval of nearest charging stations (either based on proximity, or based on filters provided)
- receipt of notification of successful or error booking
- receipt of notification when the charging process is complete
- at the payment stage (where you will be directed to a payment provider)

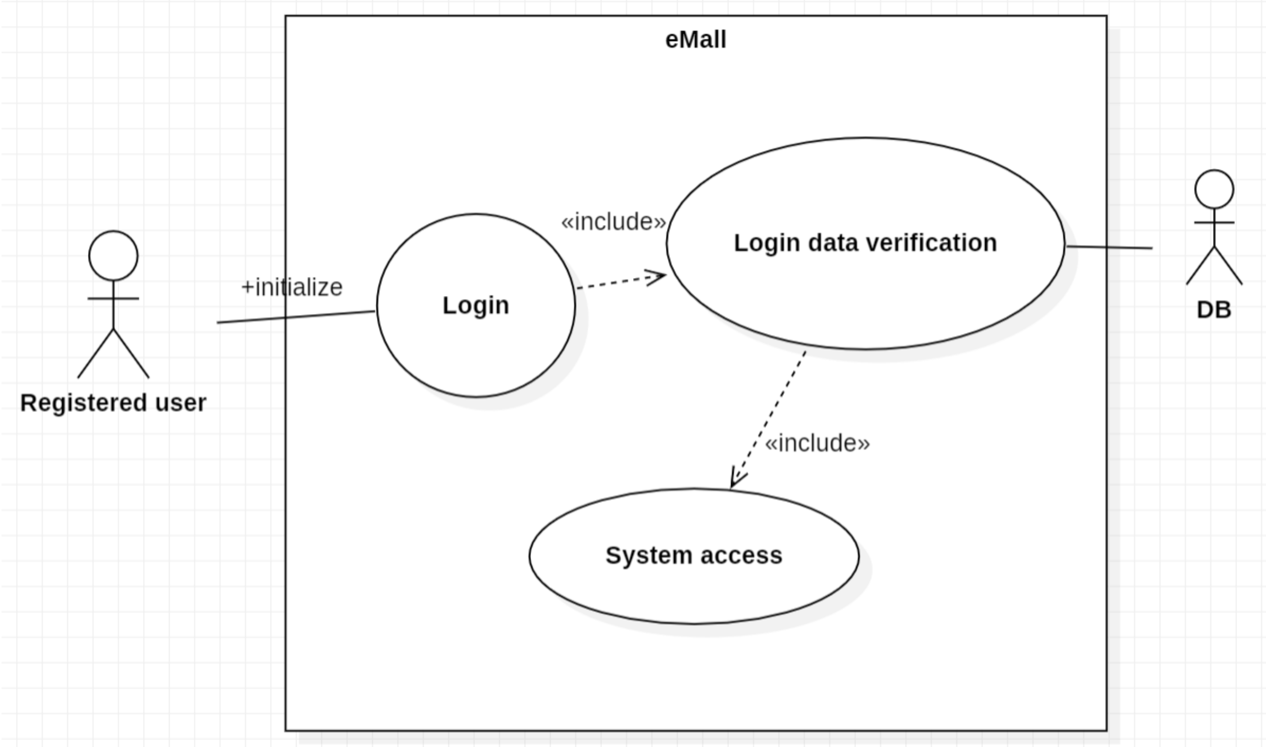
3.2 Functional Requirements

3.2.1 Use case diagrams

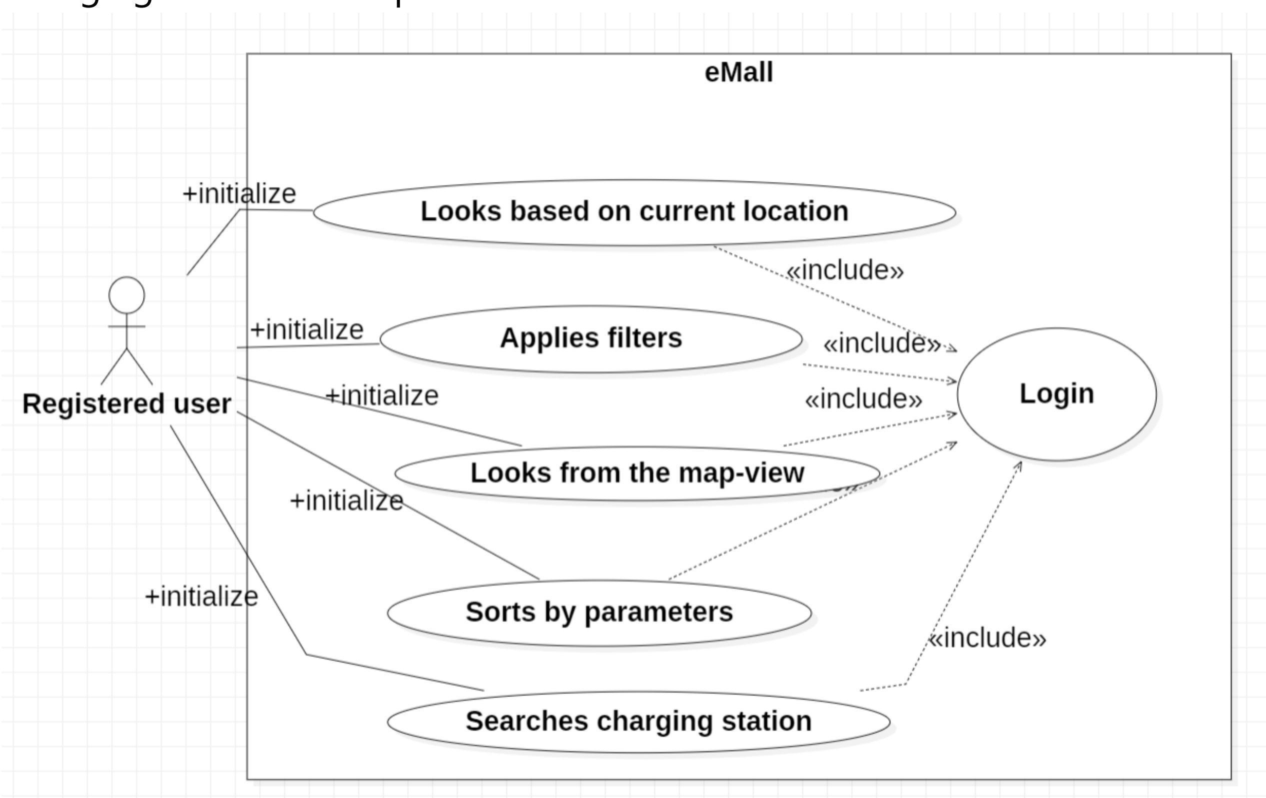
1. User registration



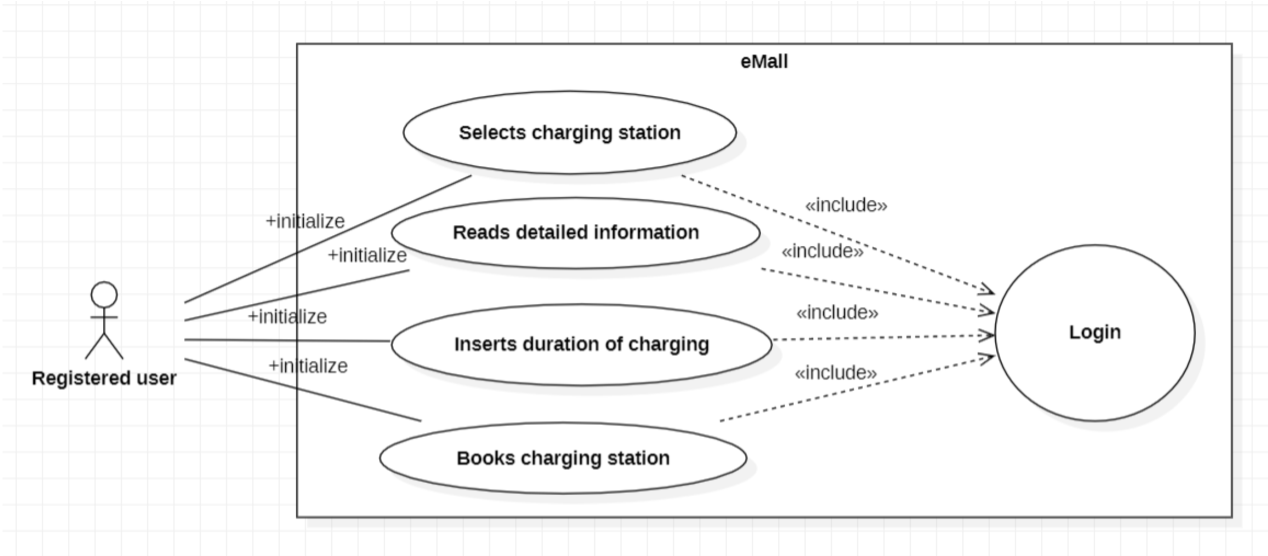
2. User login



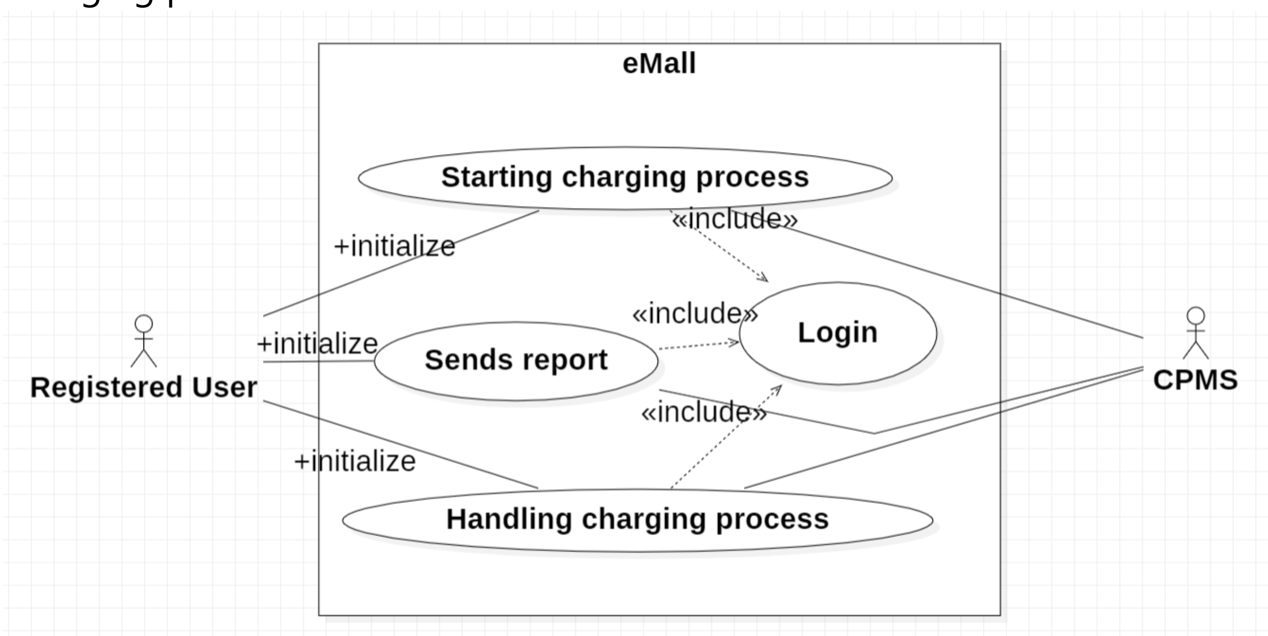
3. Charging station look up



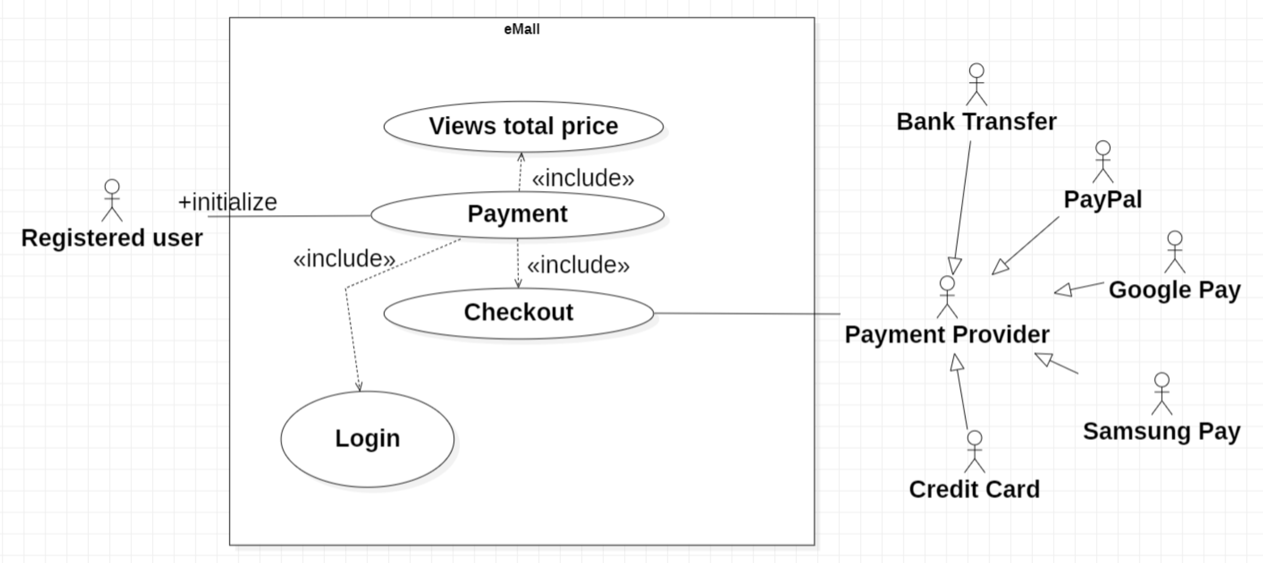
4. Charging station book



5. Charging process



6. Payment



3.2.2 Use cases

1. User registration

    1. Actor: Unregistered user
    2. Entry condition:
        1. The user does not have an account and is on the index page of the application.
    3. Event flow:
        1. The user clicks on "Create Account"
        2. The user inserts his/her personal data: first and last name, date of birth, place of residence, eMail address, username, password and telephone number (#todo update the class diagram).
        3. The user can also enter biometric data such as fingerprint.
        4. The user enters an identification document to confirm his/her identity (ID card or driver's license) or uses a digital identity provider such as SPID or CIE.
        5. The user enters the VIN of the vehicle(s) he wants to associate with his profile. By querying the DMV (Deparment of Motor Vehicle) dataset the system will confirm whether that vehicle is owned by the user or not.

    6. If confirmation occurs, the vehicle manufacturer's dataset is also queried to associate the vehicle's details with the user, which will be updated in real time (or at least periodically).
    7. e-Mall processes the information, verifies it, and returns a successful transaction message to the user and redirects it to the Home Page.
    8. The user can remember the credentials for future access.
4. Exit condition:
    1. The account is created.
5. Exceptions:
    1. The user did not enter all the required data.
    2. The user provided invalid data.
    3. The query of the database of the civil DMV and/or the vehicle manufacturer was unsuccessful.
    In all cases the user receives a notification.

2. User login

1. Actor: Registered User
2. Entry condition: User is registered, not logged in and on the e-Mall's index page.
3. Event flow:
    1. User presses login button.
    2. User insert either email address and password or uses fingerprint to access the application.
    3. User submits.
    4. e-Mall processes the information and displays a success message after having checked .
4. Exit condition:
    1. User is logged in to e-Mall.
5. Exceptions:
    1. User does not enter password or email before submitting.
    2. User insert invalid email/password combination.

    3. User does not have fingerprint saved.
    4. User uses invalid fingerprint.

3. Charging station look up

1. Actor: Registered user
2. Entry conditions: The user is registered, logged in to the e-Mall app and on the Home Page.
3. Event flow:
    1. The user presses the Search button which redirects the user to the Searching Page.
    2. The user can either directly look up for the nearest charing stations using default preferences imposed by the system itself based on the information acquired by the calendar and the vehicle's Infotainment system by selecting Fast Search (e.g: battery status) or insert his own current preferences to filter the search. (further detailed infos in the corrisponding scenario).
    3. The system looks up for the charging stations based on the searching preferences and returns a list of them. The list shows an overview station by station and can contain available charging stations which are waiting to be booked or even occupied charging station characterized by imminent ending charing process. Both of the types can be booked. Charging stations can also be visualized from a map view.
    4. Based on the user's choice, he/she can select a certain charging station from the provided list in order to view futher detailed information about it.
4. Exit condition:
   1. The user selects Exit/Home Page from the Searching Page.
   2. Search operation succedes and an the page is updated with requested data.
   Any of the above operations leads to the exit condition.

5. Exceptions:
   1. The user doesn't insert all mandatory data for the searching filters.
   2. The user selects an invalid data for certain mandatory fields.
   If any of the above occurs, e-Mall shows the user a notification error message and reccomends to retry.

4. Charging station book

1. Actor: Registered User
2. Entry condition: A registered user, logged on to e-Mall, and about to look for a charging station.
3. Event Flow:
   1. The user selects a certain charging station from the list provided or chooses it from the map view.
   2. Reads informations about the charging station selected.
   3. Selects the initial timestamp and the end timestamp to make a reservation for the charging station.
   4. Press button BOOK.
   5. Receives ack for the booking (with the estimation of the battery level at the end of the charging process) or nack in case of failures (eg. overlapping of the timestamp with other events in the calendar) and the system requires other timestamps to be inserted.
4. Exit condition:
   1. An ack is received and a book is scheduled.
5. Exceptions:
   1. User unauthorized to book.
   2. Overlapping with other events from the user's calendar.
   3. Concurrent book operations between several different users on the same charging station within the same range of time.

5. Charging process

1. Actor: Registered user

2. Entry condition:
    1. The user has arrived at the charging station, inserts the socket, and receives a notification on the e-Mall app in order to consent the start of the charging process.
3. Event flow:
    1. The user receives the notification with a reminder to their reservation details.
    2. The user starts charging by clicking on the "Start Charging" button on the app.
    3. Energy is delivered according to the strategy dynamically defined by the CPMS (the energy source can be a DSO, or the internal battery if present, or both) which will impact on the duration of the recharge.
    4. Charging stops when the end time of the booking is reached.
    5. The user can monitor the charging status in real time either from the app, the vehicle's Infotainment system or the display located at the charging station.
    6. The user receives a notification that the recharge is completed.
4. Exit condition:
    1. The end time of the charging process has been reached .
5. Exceptions:
    1. The socket is suddenly plugged out from the vehicle.
    2. The user turns on the vehicle's motor.
    3. User suddenly stops the charging process from the app before reaching the end time.
       In all cases, charging is suspended and the user is notified with options to fix the problem (if within his or her scope), send a report to the CPO, or stop charging definitely.

6. Payment

1. Actor: Registered user
2. Entry conditions:

    1. The user is registered, logged on to e-Mall, the charging process has finished and the system sent him/her a notification.
3. Event flow:
    1. The user receives a notification about the completion of the charging process of the vehicle and is invited to:
        - reach the vehicle
        - pay for the service
          The above activities can be executed on the prefered sequence.
        - disconnect the socket from the vehicle and reconnect it to the charging station
    2. The user confirms that the notification has been received and the system provides the user with different payment methods:
        1. Paypal
        2. Credit/debit card
        3. Bank transfer
        4. Several payment apps: e.g Google Pay, Samsung Pay, etc
    3. The user provides payment based on the chosen payment method.
4. Exit condition:
    1. Payment transaction committment is received by the system and notified to the user.
5. Exceptions:
    1. The user reached the card's daily limit.
    2. Fake fraudulent purchase detection.
    3. Card/Paypal account/Bank coordinates are invalid.
    4. Payment transaction rolled back.
       If any of the above operations occurs the payment operation starts from scratch again.

3.2.3 Requirements #todo system should be able to notify user

- R1: The eMSP should allow an unregistered user to register an account.
- R2: The eMSP should allow a logged user to look for a charging station.
- R3: The eMSP should be able to access the registered user's details (vehicle's Infotainment's system, user's calendar).
- R4: The eMSP should be able to retrieve a list of the nearby charging stations based on the current position of the user's vehicle.
- R5: The eMSP should be able to show the logged user the list of the nearby charging stations.
- R6: The eMSP should allow the logged user to select a charging station from the list of nearby charging stations.
- R7: The eMSP should be able to detect if the battery status is going below a fixed threshold.
- R8: The eMSP should automatically show the logged user a list of nearby charging stations when the battery threshold is overpassed.
- R9: The eMSP should allow the logged user to book a charge at the selected charging station.
- R10: The eMSP should have access to the logged user's infotainment system.
- R11: The eMSP should allow the logged user to insert filters on the looking up operation of the charging stations.
- R12: The eMSP should allow the logged user to get information about the charging station selected (sockets available, type of sockets, costs, special prices).
- R13: The CPMS should allow the CPO to choose the DSOs from which to acquire energy.
- R14: The CPMS should allow the CPO to choose either to directly distribute the energy acquired or to store it into batteries collocated at charging stations.
- R15: The CPMS should allow the user to monitor the information about the charging process (price, energy flux rate, estimated remaining time, battery's charging status).

- R16: The CPMS should manage payment processes with different payment methods, responsible for the transactions.
- R17: The eMSP should allow the logged user to decide the duration of the charging process.
- R18: The CPMS should be able to dynamically calculate the residual time for the charging process.

HINTS:

- User can insert preferencies on the range of the charging stations's look up but the system can scale it based to the vehicle's remaining battery level.
- User decide where to go by inserting a destination and the system proactively calculates the most convenient path by individualizing the possible charging stations on it.

3.2.4 Mapping on Goals #todo to justify

G1 - D1, D3, D6 - R1, R2, R4, R5, R11, R12, R18
G2 - D1, D3, D4, D6, D9 - R1, R2, R3, R4, R6, R9, R11, R12, R17, R18
G3 - D1, D2, D5, D7, D8, D10, D11 - R13, R14, R15, R16
G4 - D1, D3, D5, D6, D7, D10 - R1, R15, R16, R18
G5 - D1, D3, D10 - R1, R2, R4, R5, R6, R7, R8, R9, R10, R11, R12
G6 - D2, D7, D10 - R13, R14
G7 - D2 - R12, R13
G8 - D1, D4 - R12, R15
G9 - D2, D7 - R13
G10 - D2, D7 - R13, R14

3.2.5 Mapping on Requirements #todo to justify

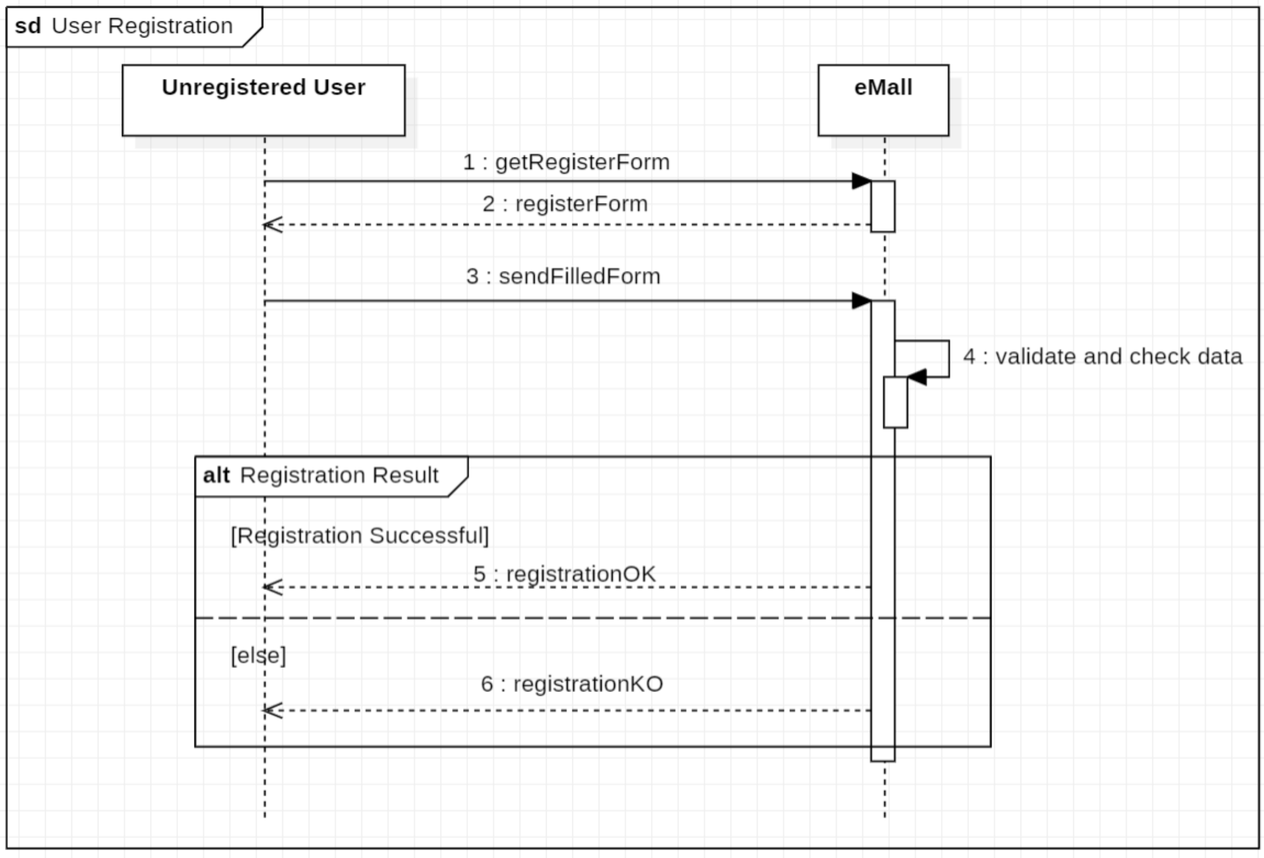
User regisration - R1
User login - #todo R10, receiving notifications from the system
Charging station look up - R2, R3, R4, R5, R6, R7, R8, R11, R12
Charging station book - R9, R10, R17

Charging process - R13, R14, R15, R16, R18

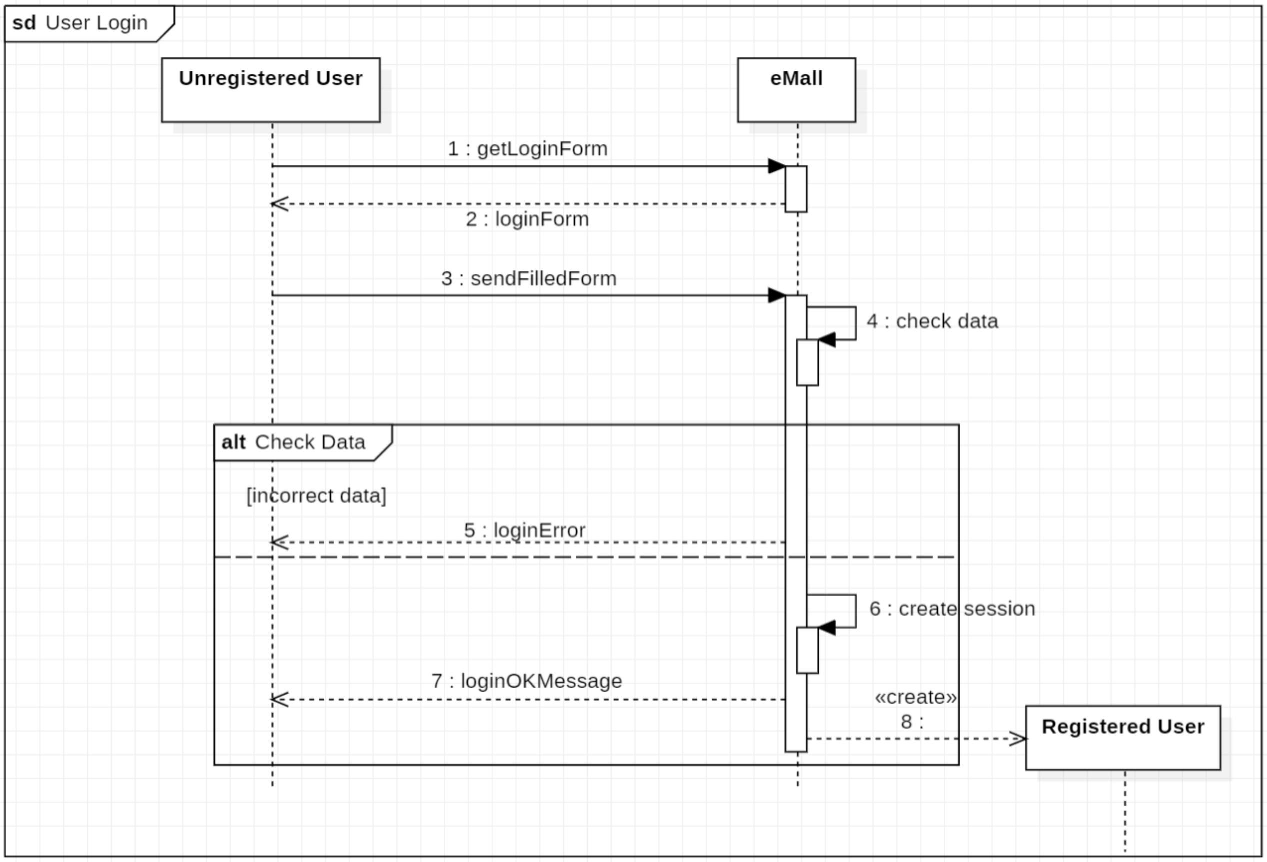
Payment - R16

3.2.6 Sequence Diagrams

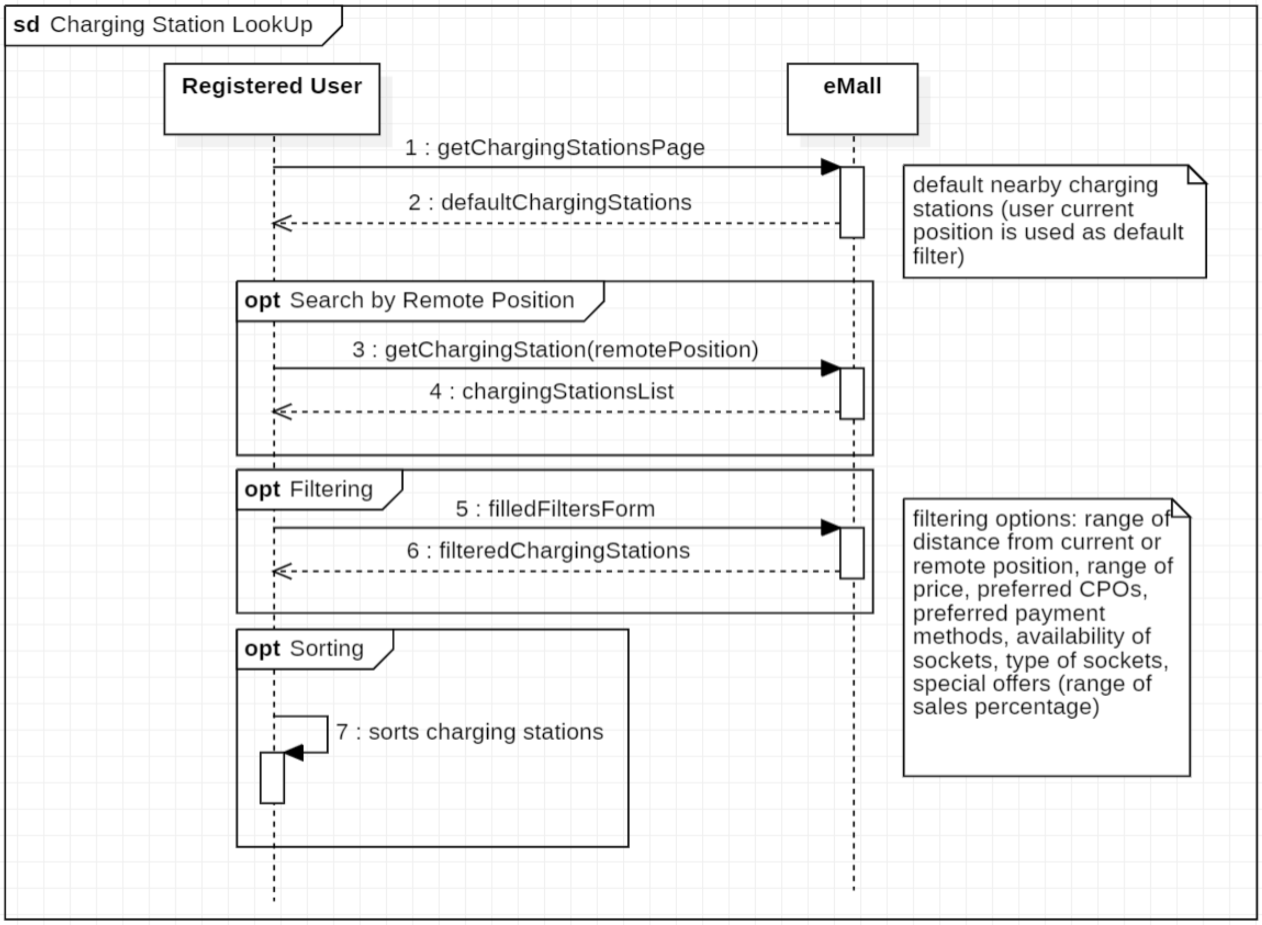
1. Registration



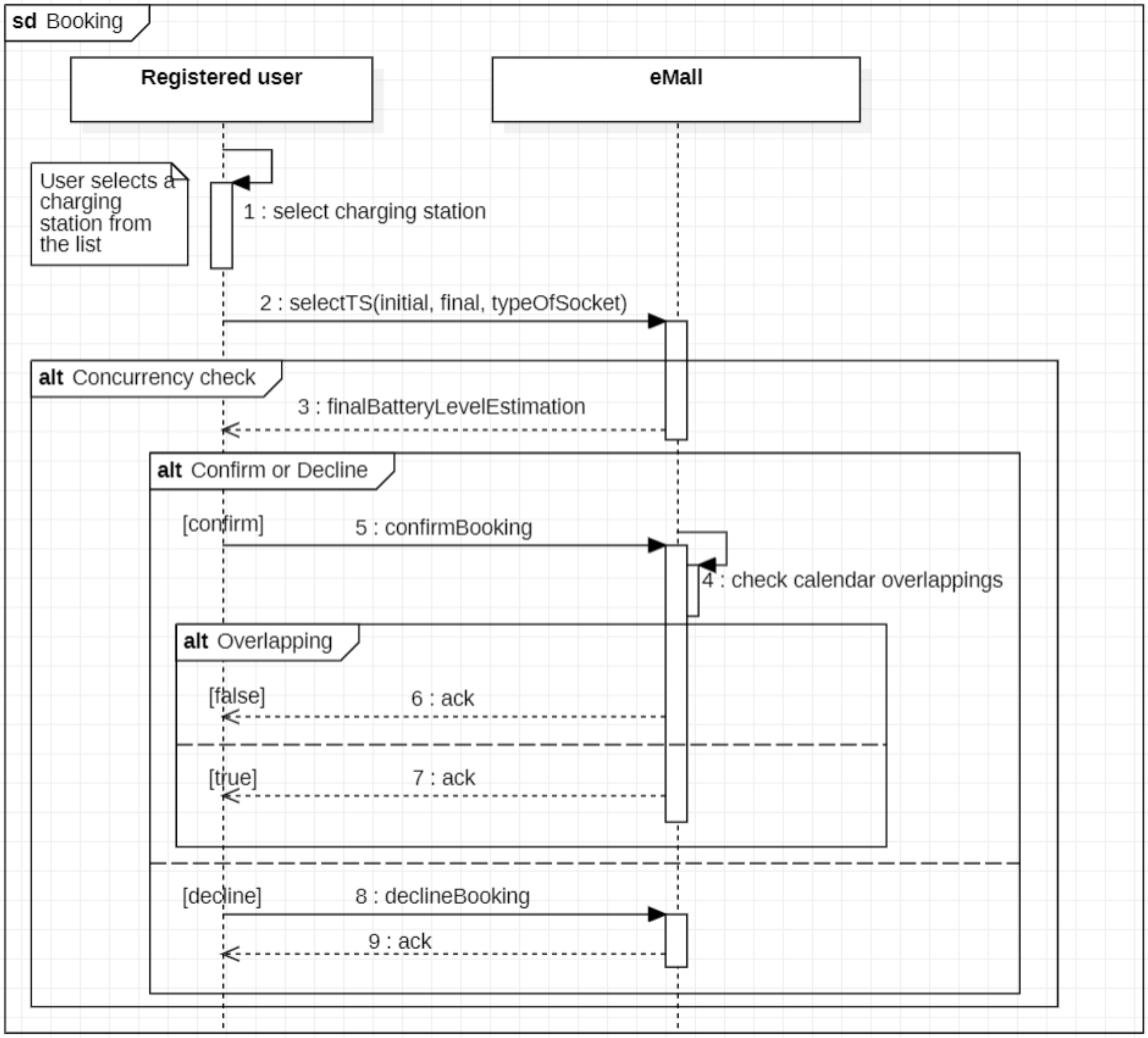
2. Login


sd User Login
Unregistered User
eMall
1 : getLoginForm
2 : loginForm
3 : sendFilledForm
4 : check data
alt Check Data
[incorrect data]
5 : loginError
6 : create session
7 : loginOKMessage
«create»
8 :
Registered User

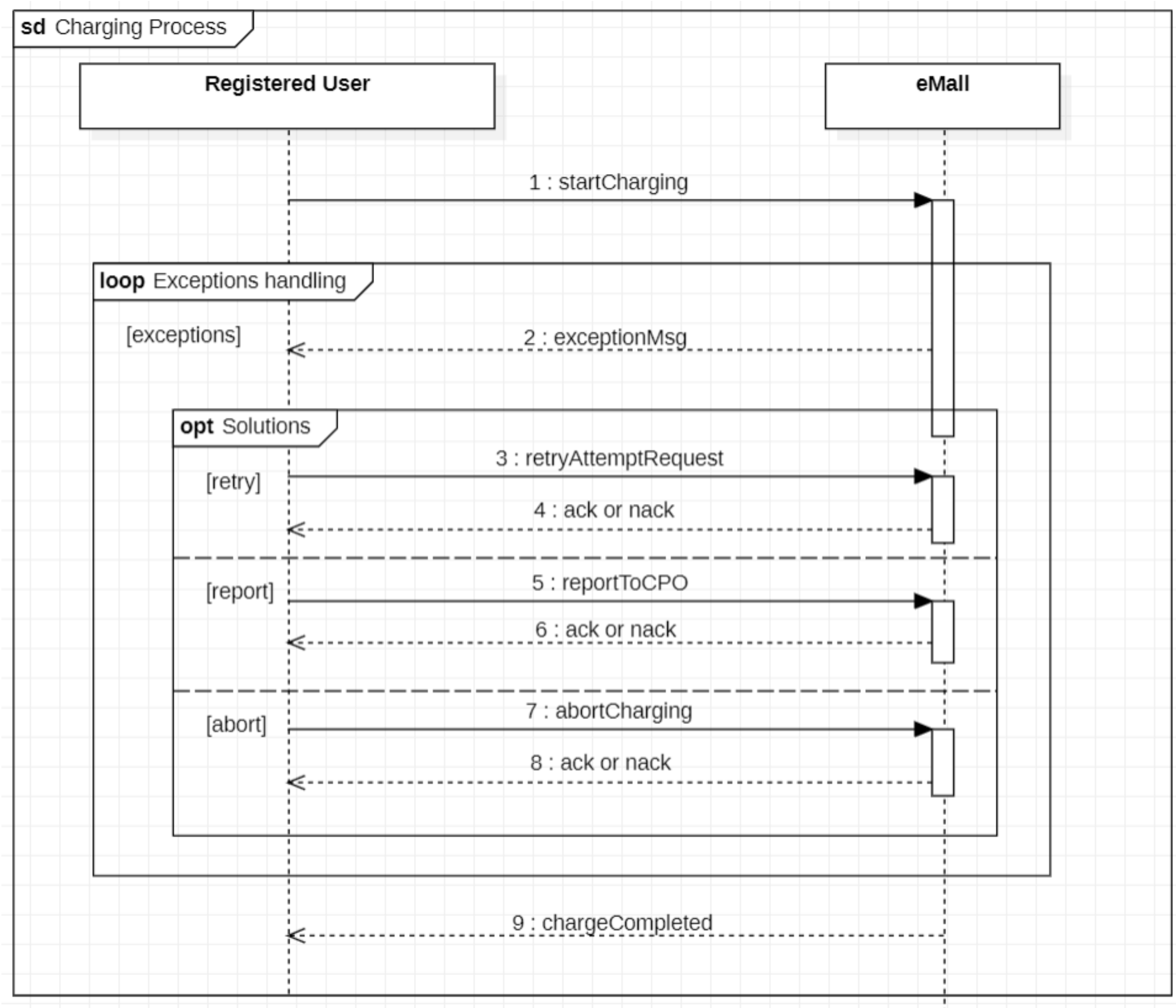
3. Charging station looking up



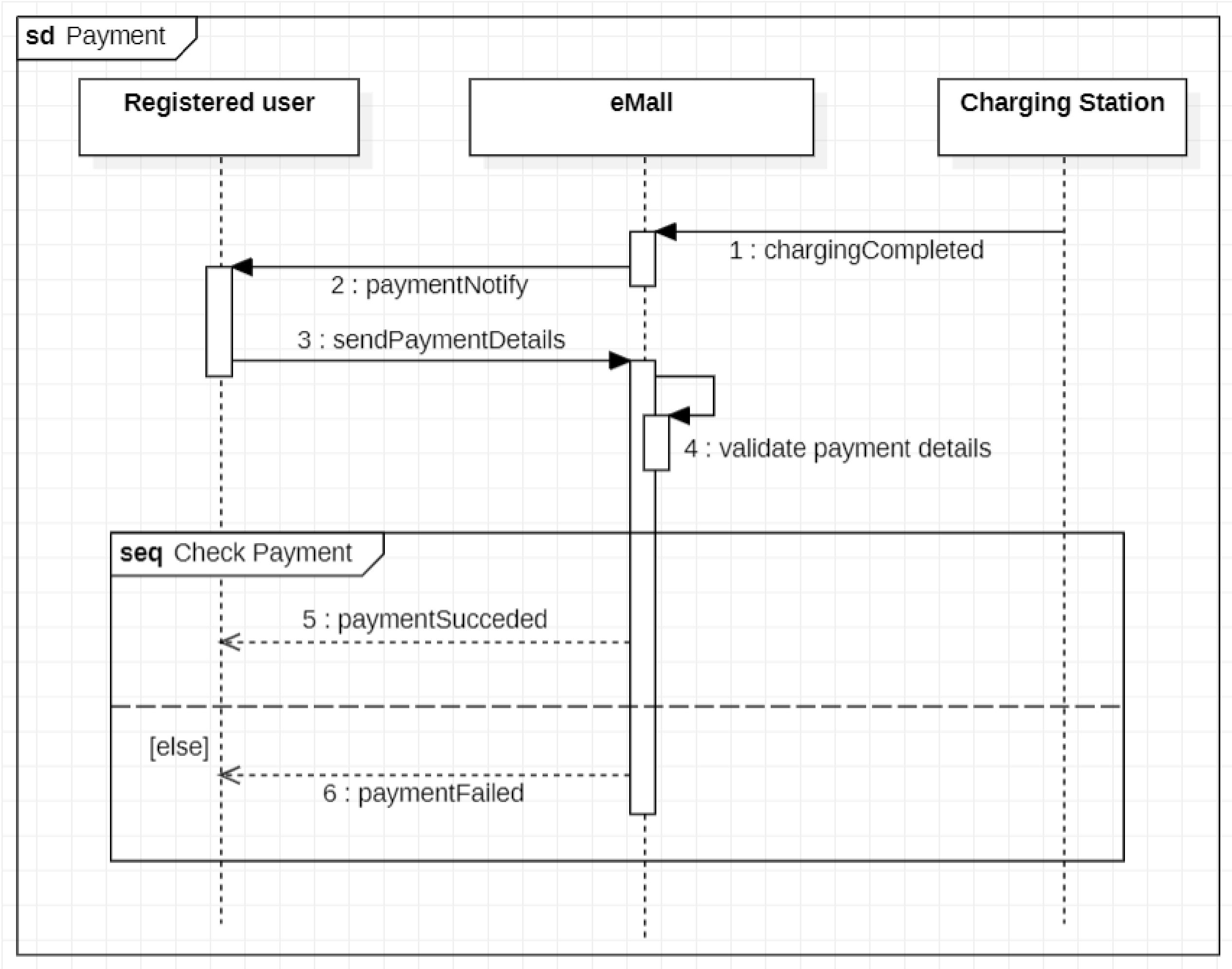
4. Charging station booking



5. Charging process



6. Payment SD



3.3 Performance Requirements

The system should always be available (24/7) and able to meet a demand of several simultaneously connected users (as we estimate to be around 1 million at most out of all 5.6 million EVs around the globe). It is possible to achieve this through redundant infrastructure and load balancing mechanisms that ensure server-side stability and scalability.
Since the user is connected to the system via mobile internet connection, communication problems may arise as a result of the user's mobility and low bandwidth (think of a user traveling to an area where there is little cellular coverage).
In order to make it possible to operate under precarious bandwidth and connection conditions, only essential information should be exchanged between client and server, and processing should be delegated to the business logic

present on the client as much as possible. In more stable connection situations (home network or 5G), all nonessential synchronization operations can be performed.

3.4 Design constraints

3.4.1 Standard compliance

eMall must have compliance with current regulations in terms of data management and protection (see the European GDPR and Law 196/2003 regarding privacy). Reference [3]
Next, it must comply with all good practices for responsive and accessible design, with full support of the devices used by most users.
Thorough documentation should be provided for each component of the application explaining how that component can be extended and implemented. Obviously, this is extended to APIs meant to allow dialogue with CPMS and external providers such as authentication via public digital identity system (SPID or CIE), the DMV or the car manufacturer.

3.4.2 Hardware limitations

It is possible to use the application from different types of devices: car infotainment systems, smartphone. The basic requirement is that there is a (hopefully stable) connection to the Internet, a GPS antenna to take advantage of the geolocation services, and that the device is working properly. An optional requirement is a fingerprint reader, which would make login and confirmation easier.

3.5 Software System Attributes

3.5.1 Reliability

The eMall system should be able to run continously for a long time. If a system crashes frequently, even for a very short amount of time (e.g. milliseconds), it can be regarded as a very unreliable system and this is to be avoided by implementing fault tollerance, meaning that the system would provide its services even in presence of faults. The key technique to mask a failure to the user is to apply redundancy, in particular, physical redundancy. In this case, a good choice is to adopt a COTS solution such as RAFT Algorithm [1] within a redundant server processes group, characterized by a leader process with the ability to distribute the workload through the different replicated ones.

3.5.2 Availability

The eMall system has to be ready for use as soon as a user needs it to make a request. The goal is to have the highest availability possible. Statistics [2] state that there are over 5.6 million electric vehicles worldwide at the moment with a 64% increase, seeing total number of EVs rise from 3.4 million to 5.6 million in the last few years, number to be mapped to one possible eMall user and probably doomed to increase. In order to provide system availability to so many users, the MTTR has to be automated through sophisticated recovery techniques. We deem that an availability of 99.9% (3-nines notation), equivalent to 8.76 hours/year downtime, should be enough to cover user's needs. All the reasoning is assuming that communication links are reliable and remain so.

3.5.3 Security

The system is storing confidential information about user's movements and charging routines which could be exploited by external agents such as hackers or robbers. This means that security has to be ensured by using passwords, biometric authentication, authorization methods, firewalls and encryption. In fact, the system must be closed, meaning that nobody can access it without passing the pre-defined access rules, managed by several deployed firewalls throughout the system.

3.5.4 Maintainability

The eMall system has to be open to various updates during its life-cycle as new features may be implemented or pre-existing ones fixed. Software architectural patterns have to be chosen wisely to ensure decoupling between components in order to facilitate maintainability in accordance with prescribed procedures. This would smooth the path to restore a failed component within a specified period, fix a bug spotted within a certain component without affecting others, or enhance the software to provide new requested features.

3.5.5 Portability

The system should give the possibility to use the deployed platform in different environments and to different users. All heterogeneities must be covered as for Dev-ops, portability comes with stability. The platform should not have different behaviours on different devices and should ensure consistency. For this case, the implementation could be done using Java Enterprise Edition (Java EE) in order to take advantage of all the functionalities it provides, along with high maintainability standards.

4 Formal analysis

4.1 Alloy code

```
//Signatures
sig Email{}
sig Byte{}
sig Date{
year: one Int,
month: one Int,
day: one Int,
hour: one Int,
minute: one Int,
second: one Int
}

sig User{
userID: one Int,
email: one Email,
dateOfBirth: one Date,
fingerprint:one Byte,
```

```
type: one Type
}

//---------------------------------------------------
abstract sig Type{}
one sig UNREGISTERED extends Type{}
one sig REGISTERED extends Type{}
one sig LOGGED extends Type{}
//---------------------------------------------------
abstract sig Bool{}
one sig TRUE extends Bool{}
one sig FALSE extends Bool{}
//---------------------------------------------------
sig VIN{}
sig Vehicle{
vin: one VIN,
user: one User,
socketType: some SocketType,
batteryLevel: one Int,
typeOfBattery: one TypeOfBattery
}

abstract sig SocketType{}
one sig FAST extends SocketType{}
one sig RAPID extends SocketType{}
one sig SLOW extends SocketType{}

sig TypeOfBattery{}
//---------------------------------------------------
//todo
//sig Calendar{}
//sig Event{}

//-------------------------------------------------
```

```
sig CPO{
cpoID: one Int,
specialOffer: some SpecialOffer,
dso: some DSO
}

sig SpecialOffer{
salePercentage: one Int
}

//--------------------------------------------------
sig DSO{
dsoID: one Int,
costOfEnery: one Int,
maxAmountOfEnergy: one Int,
cpo: some CPO
}

//-------------------------------------------------
sig ChargingStation{
location: one Location,
numberOfAvailableSockets: one Int,
isActive: one Bool,
cpo: one CPO,
sockets: set Socket
}

sig Location{}
sig Battery{
batteryID: one Int,
isCharging: one Bool,
availableEnergy: one Int,
chargingstation: one ChargingStation
}
```

```
//--------------------------------------------------
sig Socket{
socketType: one SocketType,
maxEnergyFlow: one Int,
isAvailable: one Bool,
costOfEnergy: one Int,
vehicle: lone Vehicle,
chargingstation: one ChargingStation
}

//--------------------------------------------------
sig Book{
startTime: Date,
endTime: Date,
completed: Bool,
bookStatus: BookStatus
}
sig BookStatus{}

//fact
//FATTO: ogni colonnina ha un CPO solo
//FATTO: un CPO può avere più charging station
//FATTO: ogni veicolo ha un tipo di batteria e può collegarsi con un solo socket
//FATTO: una chargingstation può avere più batterie
//FATTO: associazione un DSO tanti CPO e viceversa
//FATTO: && batteryLevel>=0
//FATTO: specialprice
//FATTO: i vehicle possono essere associati solo a socket che sono available
//FATTO: chargingstationa available se c'è almeno 1 socket available
//FATTO: socket a una sola charging station
//FATTO (è una conseguenza di quello precedente): le socket non possono
esistere se non ci sono charging station

fact allUserHasVehicle{
all u:User |
```

```
( some v:Vehicle | u.type = LOGGED && u in v.user or u.type = REGISTERED &&
u in v.user)
}

fact distinctEmail{
no disjoint u1, u2: User | u1.email = u2.email
}

fact distinctByte{
no disjoint u1, u2: User | u1.fingerprint = u2.fingerprint
}

fact distinctUserID{
no disjoint u1, u2: User | u1.userID = u2.userID
}

fact distinctLocationChargingStation{
no disjoint c1, c2 : ChargingStation | c1.location = c2.location
}

fact distinctVIN{
no disjoint v1, v2: Vehicle | v1.vin = v2.vin
}

fact PositiveuserID{
all u:User | u.userID >= 0
}

fact PositivedsoID{
all d:DSO | d.dsoID >= 0
}

fact PositivecpoID{
all c:CPO | c.cpoID >= 0
}
```

```
fact PositivebatteryID{
all b:Battery | b.batteryID >= 0
}

fact PositiveBatteryLevel{
all v:Vehicle | v.batteryLevel >= 0
}

fact PositiveSpecialPrice{
all s:SpecialOffer | s.salePercentage>0
}

fact PositiveAvailableEnergy{
all b:Battery | b.availableEnergy >= 0
}

fact PositiveCostOfEnergy{
all s:Socket | s.costOfEnergy > 0
}

fact AvailableChargingStation{
all c:ChargingStation | c.isActive = TRUE iff (some s:Socket | s.isAvailable = TRUE
and s in c.sockets)
}

fact VehicleLinkedToSocket{
all s:Socket, v:Vehicle | s.vehicle = v iff (s.isAvailable = TRUE and v in s.vehicle)
}

fact SocketIsContainedOnlyInOneChargingStation{
all s:Socket, c:ChargingStation | s in c.sockets iff c in s.chargingstation
}

fact OneSocketWithOneVehicle{
no disjoint s1, s2 : Socket | s1.vehicle = s2.vehicle
}
```

```
fact DSOCPOrelation{
all d:DSO, c:CPO | d in c.dso iff c in d.cpo
}

fact ConsistentDate{
all b:Book | b.endTime.day > b.startTime.day and
b.endTime.month > b.startTime.month and
b.endTime.year > b.startTime.year and
b.endTime.hour > b.startTime.hour and
b.endTime.minute > b.startTime.minute and
b.endTime.second > b.startTime.second and
b.endTime.day >=0 and
b.endTime.month >=0 and
b.endTime.year >=0 and
b.endTime.hour >=0 and
b.endTime.minute >=0 and
b.endTime.second >=0 and
b.startTime.day >=0 and
b.startTime.month >=0 and
b.startTime.year >=0 and
b.startTime.hour >=0 and
b.startTime.minute >=0 and
b.startTime.second >=0

}

//assert
assert EachChargingStationHasOnlyOneCPO{
all c:ChargingStation| one p:CPO | p in c.cpo
}
check EachChargingStationHasOnlyOneCPO

assert UserWithSomeVehicle{
all u:User | some v:Vehicle | u in v.user
```

}

check UserWithSomeVehicle

//pred

pred show1{
#User >=2
#Vehicle = 5
#Book >= 2
#ChargingStation > 1
#Socket >= 3
}

pred show2{
#CPO = 2
#DSO = 3
}

run show1 for 5

run show2 for 3

REFERENCES

1. In Search of an Understandable Consensus Algorithm (Extended Version) Diego Ongaro and John Ousterhout Stanford University
2. Electric Car Statistics In The US And Abroad by Alex Kopestinsky
3. The Italian Data Protection Authority

#todo

1.5 (at the end)

2.2

3.1.3
3.2.3
3.2.4
3.2.5