RASD

Software Engineering Project
A.Y. 2022-2023

Marco Ronzani, Alessandro Sassi

November 2022

Contents

1 Introduction

1.1 Purpose

Electric vehicles are starting to grow in number, and their takeover of combustion engines is bound to happen, consequently to support such a thriving trend adequate easy access to charging stations is of utmost importance. In this landscape the goal of eMall is to allow owners of electric vehicles to easily know where charging stations are and carefully plan their charging process according to their schedules at any such station.

This document will discuss goals and requirements that regard the system necessary to make eMall a reality, with the purpose of guiding the development process.

1.1.1 Goals

Goal	Description
G1	Allow Users to see the list of available Charging Stations
G2	Allow Users to see the cost of a recharge and special offers for a Charging Station
G3	Allow Users to book charging time slots
G4	Allow Users to monitor, manage and pay for their booked charging sessions
G5	Allow Charge Point Operators to manage the energy sources each Charging Station uses
G6	Allow Charge Point Operators to manage the price of electricity for each Charging Station

1.2 Definitions, Acronyms and Abbreviations

1.2.1 Definitions

Term	Definition
Charging Station	Device with a connection to the electric grid which brakes out power to one or more socket(s) for vehicles. Monitoring of each socket's status.
Socket	One of the charging outlets available at a CS where a vehicle connects, its type determines the vehicles that can connect.
Charging Session	The process in which a User performs a recharge of their vehicle at a specific socket.
Energy source of a Charging Station	Batteries or the DSO currently assigned to the CS, whichever the CS is currently drawing its power from.
Charging Station External Status	Number of charging sockets available, their type such as slow/fast/rapid, their cost, and, if all sockets of a certain type are occupied, the estimated amount of time until the first socket of that type is freed.
Charging Station Internal Status	Amount of energy available in the batteries, if any, number of vehicles being charged and, for each charging vehicle, amount of power absorbed and time left to the end of the recharge.
Energy Source	Method of energy production that results in a known fraction of the energy supplied to an endpoint in the electric grip.
User-price	Cost of a recharge that is shown to the User when they inspect a CS and is what they are charged for. It is set by the CPO/CPMS on a per-CS basis.
Nominal-price	Cost of a recharge at a CS without any discount or offer applied, it is set by the CPO/CPMS on a per-CS basis. A user-price < nominal-price implies an ongoing special offer.
Energy source management policy OR Battery usage policy	A per-CS policy given them by the CPMS to allow them to dynamically decide whether to acquire energy from their assigned DSO or from their batteries and when to charge their batteries with energy from their DSO.
Charge Point Management System's policy for "Automatic Mode"	The global policy used by the CPMS to operate autonomously, its mainly built around thresholds and weights to allow the CPMS to decide prices and battery usage policies for its CSs.
Offer reset date	The date, set on a per-CS basis, at which a CS will reset its user-price to the value of its nominal-price, removing any present offer.

1.2.2 Acronyms

Acronym	Full Name
eMall	Electric Mobility for All
eMSP	Electric Mobility Service Provider
CPO	Charging Point Operator
CPMS	Charge Point Management System
CS	Charging Station
DSO	Distribution System Operator
STB	System-To-Be
OS	Operating system

1.3 Scope

This RASD document takes into consideration the requirements and specifications of the eMSP platform "eMall", together with its interaction with one or more CPMSs. The stakeholders considered are the End Users who interact with the "eMall" platform, CPOs owning the respective CSs and CPMSs, and DSOs offering their services to the aforementioned parties.

1.3.1 World Phenomena

Phenomena	Description
WP1	Electric vehicles connect to a CS socket
WP2	CPOs add/remove available CSs
WP3	CPOs add/remove batteries from existing CSs
WP4	CPOs decide the policy with which to assign DSOs to CSs
WP5	DSOs set the price and/or the mix of sources for the electricity they provide
WP6	CPOs pay DSOs for the consumed electricity

1.3.2 Shared Phenomena

In order to represent more clearly whether a phenomenon is world or machine controlled or observed, we define a list of acronyms used only for the scope of the Shared Phenomena definition. These acronyms will also be reported in the Shared Phenomena table, so that for each entry the controller/observer party can easily be identified.

Type	Explanation
MO	Machine Observed Phenomenon, the STB takes notice of the event
MC	Machine Controlled Phenomenon, the STB causes the event
WO	World Controlled Phenomenon, an/some external agent(s) notices the event
WC	World Controlled Phenomenon, an/some external agent(s) causes the event

Phenomena	Type	Description
SP1	WC MO	The User views the available charging stations and their information
SP2	WC MO	Users book a charging session to a CS
SP3	WC MO	Users start their vehicle's charging session on a CS
SP4	MC WO	The eMSP notifies users of the completion of their vehicle's charging session
SP5	WC MO	Users prematurely terminate their ongoing charging session
SP6	WC MO	Users are charged for their finished charging sessions
SP7	MC WO	CPMSs publish the location and external status information regarding their managed CSs
SP8	WC MO	CPOs update the DSO associated to a CS (this determines how much a CPO is charged for the electricity a CS provides)
SP9	WC MO	CPOs update the user-price of energy and set special offers for a CS
SP10	WC MO	CPOs update batteries information and batteries usage policies for their CSs
SP11	WC MO	CPOs update the policy used by their CPMSs to operate autonomously
SP12	MC WO	The eMSP transfers the amount due to CPOs for performed recharges

1.4 Revision History

v0.1 First draft of the document.

1.5 Reference Documents

1. The provided document describing the project: *Assignment RDD AY 2022-2023-v3*.
2. The Software Engineering 2 course held by Prof.s Camilli Matteo, Di Nitto Elisabetta and Rossi Matteo Giovanni.
3. *ISO/IEC/IEEE 29148:2011(E)* standard for Requirement Engineering.
4. Project of last year provided as an assignment.

1.6 Document Structure

This document is based on the *ISO/IEC/IEEE 29148:2011(E)* document and adheres to the requirements analysis procedures of *ISO/IEC 12207*. The document is divided into the following six sections.

The **introduction** informally introduces the scope and purpose of the STB this document analyses, following up such description with a formal elicitation of goals and phenomena that together provide a characterization of the world around the STB and what will the STB bring to such environment. In this section useful information to read the document is provided as well, such as acronyms, the document's revision history and other referenced documents.

The **overall description** follows with detailed modelling of the world around the STB and the scenarion in which the system will be interacting with all the different external users. This section identifies and describes the uses of the system and the functions the STB will offer, as well as posing the needed domain assumptions to supplement the current model.

The **specific requirements** section focuses on functional and non functional requirements, starting from the interfaces of the system towards external entities and other system elements, following with a formal elicitation of functional requirements and their contextualization with goals and domain assumptions. Finally design constraints and software system attributes are described in this section to complement requirements.

Follows the **formal analysis using Alloy** which presents the Alloy code related to the described model and the results provided by its usage.

The **effort spent** section presents data regarding the amount of time each team member invested in the creation of this document.

Finally **references** are reported.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

1. New user registration

Actors: Unregistered User

A person, Bob, who discovered eMall has decided to register into the platform to use its recharge booking and recharge management system. To this end Bob reaches the eMall services and chooses to "sign up", after that Bob fills in the relevant information and his eMall account is created. Now Bob can proceed to login.

2. User login

Actors: Registered User

An already registered user, Bob, reaches the eMall services and intends to either book a charging spot or to manage a currently ongoing recharge booked to his name, hence Bob chooses to "login" and enters its credentials. Assuming Bob inserted valid credentials he now has a valid session within the eMall services. If Bob failed in setting valid credentials he can try again.

3. User looking for charging stations

Actors: Registered User

A user already registered on eMall, Bob, needs to know the location and/or pricing of CSs in a certain are of his interest. To obtain such data, Bob has reached the eMall services, has logged in and has selected the charging stations "search" options. Now Bob can search for CSs and is presented with multiple filters to control:

- His area of interest
- Recharge price ranges
- Connector types available
- Charging speeds available

By manipulating the filters Bob can restrict or widen the scope of his search. Results of the search are displayed with respect of the filters and Bob can see the details of every result such as:

- Location
- User-Price (with offers applied)
- Nominal-Price (base price without offers)
- Expiration date of any ongoing offer
- Booking schedule
- Available connectors
- Available charging speeds

4. User booking a recharge

Actors: Logged-in User

A user Bob has already performed the login and now intends to book a recharge. Bob first performs a search for the CS to reserve for his recharge (see "user looking for charging stations" scenario) and once he finds a suitable and available one chooses "book recharge" for that station. Bob is then presented with the choice for the time slot for his recharge and once he chooses his reservation is completed.

5. **User deleting one of his bookings**
   Actors: Logged-in User
   A user Bob has already performed the login and recently made a few bookings on eMall. However on second thought Bob realized he could not make it in time for one of his bookings and now intends to delete it. To this end Bob selects the option to view his meetings' list on eMall and selects the meeting he desires to remove, he chooses the delete option the meeting list is updated with the deleted meeting now missing, confirming that eMall discarded it.

6. **User performing a charging process (start, monitor and pay)**
   Actors: Logged-in User
   The user Bob has previously booked a recharge at a certain CS for a certain time slot. Bob shows up at the CS during his reserved time slot and proceeds to connect his vehicle properly. Once the vehicle is connected Bob reaches the eMall services to start the charging process. After the process is started Bob has the option to interrupt it at any time, otherwise the charging procedure automatically halts as the CS noticed the vehicle's battery being full or the booked time slot expires. Bob is notified by eMall as soon as the charging procedure ends. Immediately after completing a recharge, Bob is then presented by eMall with the total cost of his recharge and is automatically charged for the service on the payment method he inserted while registering.

7. **User does not show up for a booked recharge**
   Actors: Registered User
   The user Bob has previously booked a recharge at a certain CS for a certain time slot. However Bob forgot about the booking and did not show up to the booked CS during the time slot for the recharge, hence he is notified by eMall after the time slot expires and is charged a fee from eMall. Obviously had Bob deleted the meeting beforehand he would have avoided the fee.

8. **CPO wants information on DSOs' energy prices and mix of sources**
   Actors: CPO
   The CPO Xcorp, or better its employee Mary which authorizes herself with Xcorp's CPMS, wants all information regarding the energy market to evaluate their choice of a DSO. To do this Xcorp uses its CPMS's functionality to gather such information from all the DSOs known to it, hence the CPMS proceeds to recover this information from the various DSOs and once it has finished it presents the result to Xcorp.

9. **CPO chooses prices, energy sources and battery usage policies for a CS**
   Actors: CPO
   The CPO Xcorp might want to change a few parameters for one of its CSs, for instance which DSO supplies it, how the CS manages its batteries (if any are present) or the prices it offers to users. This might be done for a variety of reasons, for example in order to adjust prices in response to the market, to take advantage of better energy mixes provided by other DSO or to improve the CS's batteries lifespan. To this end Xcorp instructs one of its employees, Mary, to authenticates with Xcorp's CPMS. Afterwards Mary selects the CS which will be subject to changes and can choose to assign that CS a new nominal-price, a new user-price, a new DSO among those the CPMS is aware of or to change the battery usage policy for that CS. If Mary were to assign a user-price lower than the nominal-price, that would be the same as her setting up a special offer at that CS, and when doing so Mary can choose to set an expiration date for the offer as well, after which the CS will resets automatically its user-price to its nominal-price.

10. CPO toggles the CPMS automatic mode

Actors: CPO

A CPO employee, Mary, in charge of administering the CPMS of its company, Xcorp, has been tasked to change its operating mode from "Manual" to "Automatic". To do this, She logs in to the CPMS and toggles "Automatic Mode" on. Once the system has finished changing the operating mode, if its now in "Automatic Mode" She loses control over manual overrides, and is prompted to enable "Manual Mode" again in order to change those settings. After completing the action, Mary successfully logs out of the system.

11. CPO changes the CPMS policy

Actors: CPO

Mary, an employee at the CPO Xcorp, has been requested to alter the way the company's CPMS operates automatically, such behaviour is determined by a series of thresholds and weights that Mary has to overwrite. To do so, Mary first logs in to the CPMS and chooses to see its current policy, there She can type in new values in place of the current ones and commits its changes once She is done, only then the CPMS will store the new policy and start using it for its "Automatic Mode". After committing the changes Mary logs out of the CPMS.

2.1.2 Class Diagram

UML class diagram showing the entities of interest for the system and their relationships.

A notably missing relationship is the one between *Vehicle* and *User*, which was deemed not necessary since an instance of the latter should as well be able to book recharges for vehicles they don't own (Ex: rented electric cars), hence binding the two might have been counterproductive.

The *Policy* entity models the criteria the CPMS uses once set to "Automatic Mode" to decide DSOs, energy source and prices for CS.

Meanwhile *Booking* presents the *isActive* flag, which is true only when the charging process associated to it is going on.

The flag *chargingFromBatteries* on *CS* indicates whether the current source of energy the CS is using to charge attached vehicles are its batteries or the associated DSO.

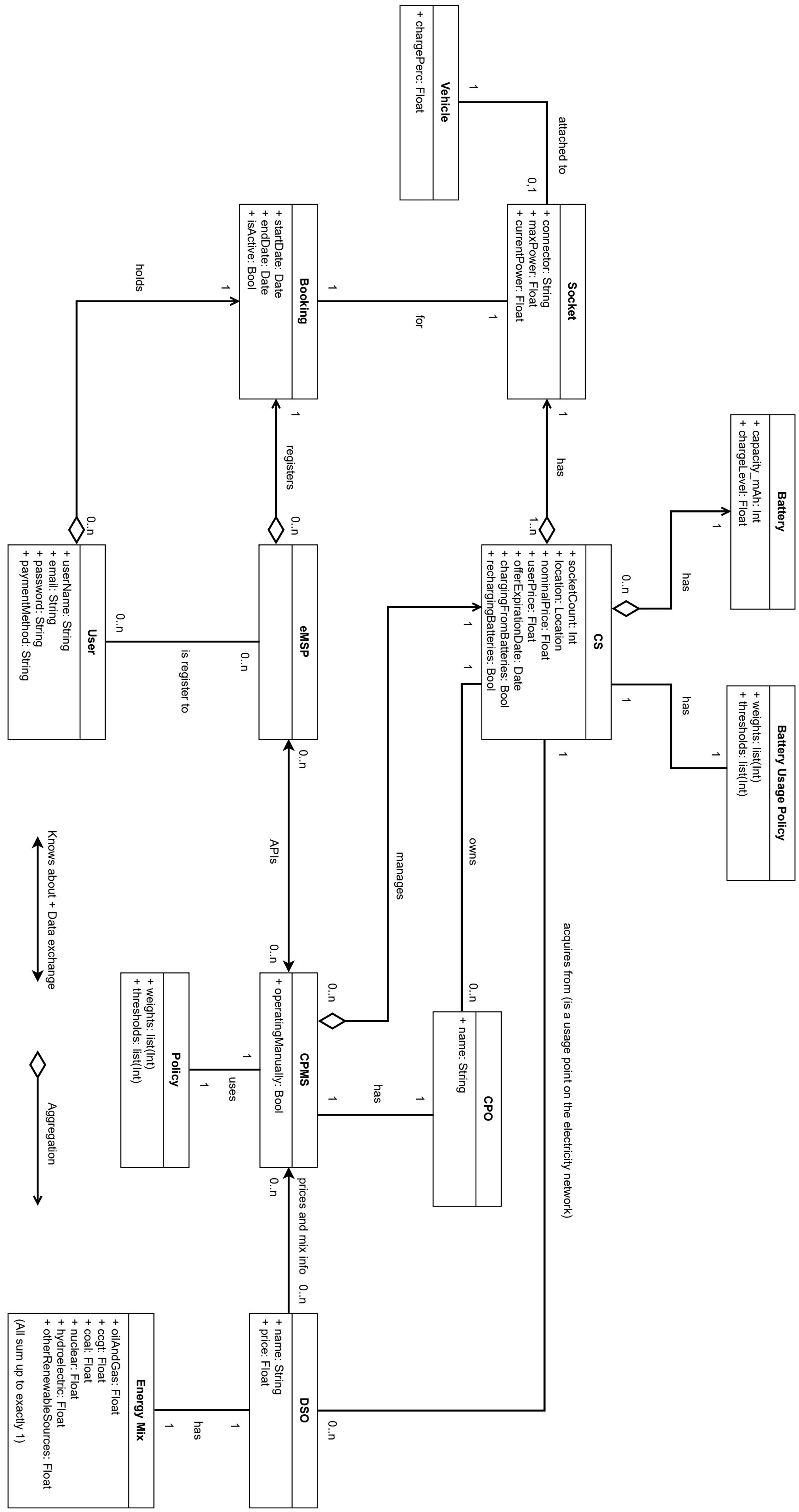

Figure 1: Class diagram

2.1.3 State Diagrams

Those following are representations of the most important interactions that involve the STB, they mostly correspond to the different scenarios.

1. **User registration and/or login**

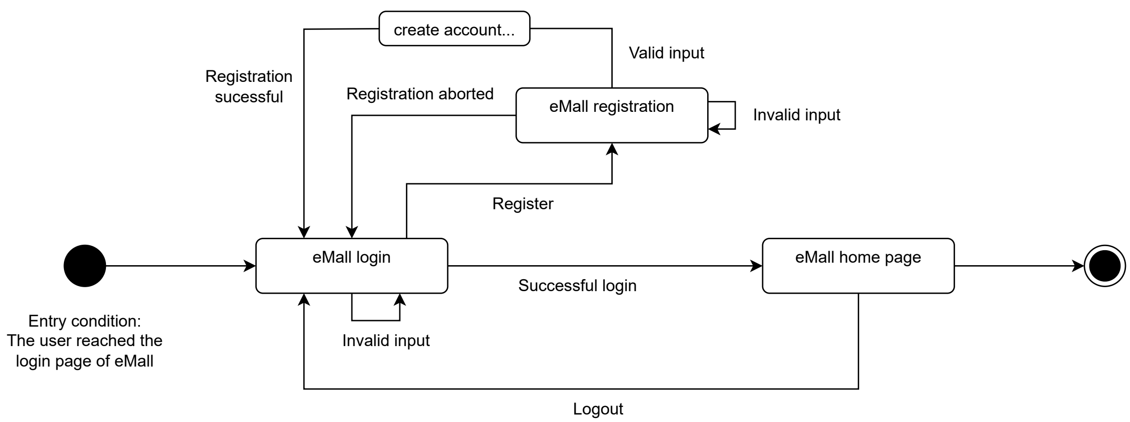

Figure 2: User registration and/or login

2. **User searching CS and booking a recharge**

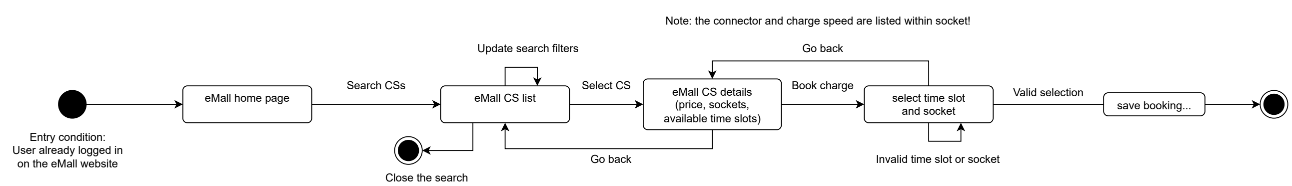

Figure 3: User searching CS and booking a recharge

3. **User deleting a booking**

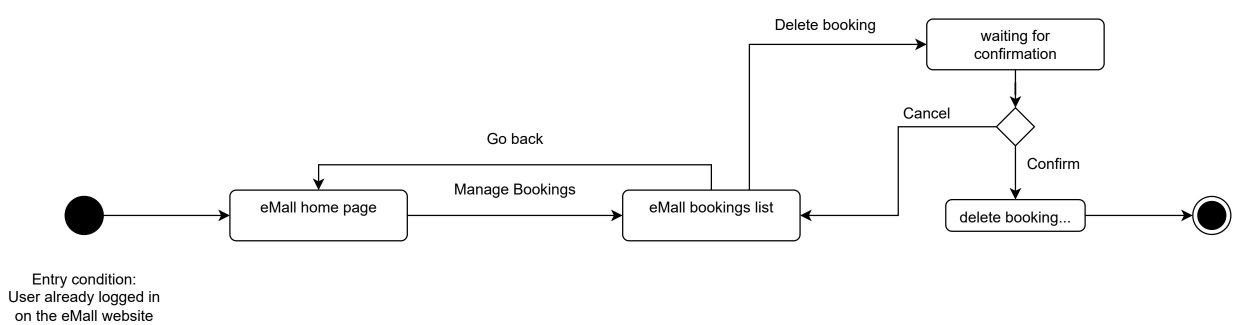

Figure 4: User deleting a booking

4. User performing a charging session

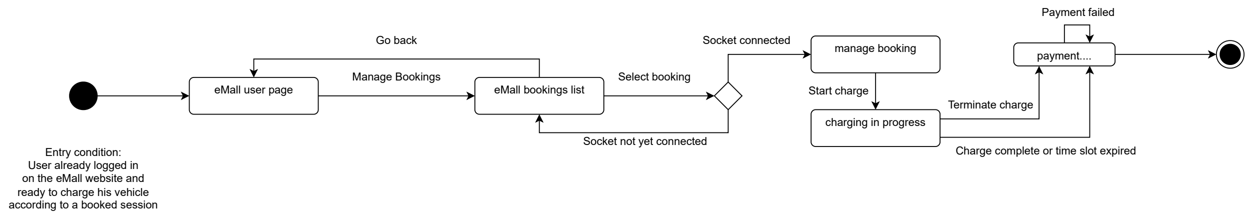

Figure 5: User performing a charging session

5. CPO acquiring information on DSOs' energy prices and mixes

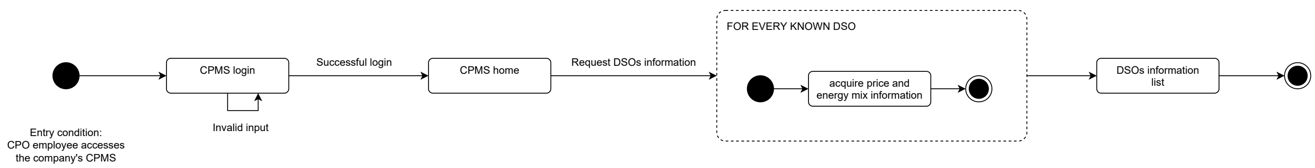

Figure 6: CPO acquiring information on DSOs' energy prices and mixes

6. CPO chooses prices, energy sources and battery usage policies for a CS

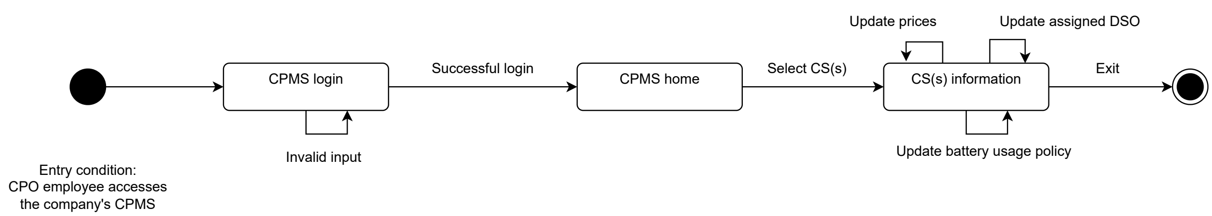

Figure 7: CPO chooses prices, energy sources and battery usage policies for a CS

7. CPO toggles the CPMS automatic mode

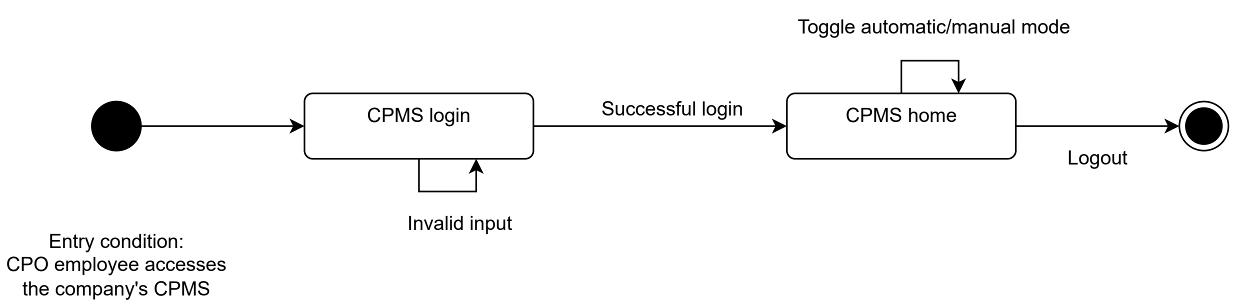

Figure 8: CPO toggles the CPMS automatic mode

8. CPO changes the CPMS policy

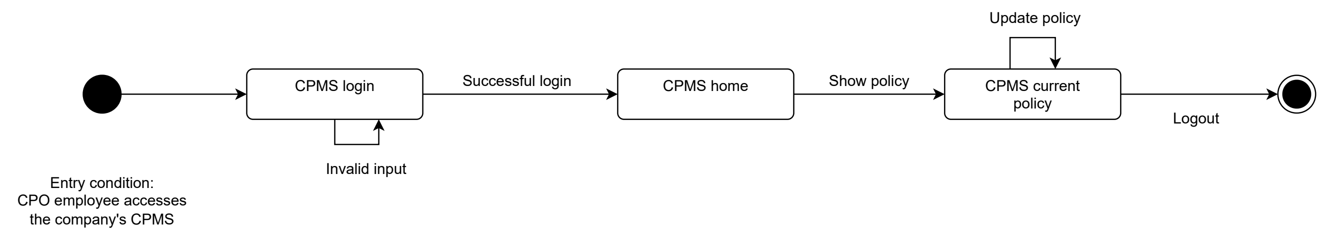

Figure 9: CPO changes the CPMS policy

2.2 Product Functions

2.2.1 User registration

eMall will allow Users to register an account. A User will register by entering:

- A desired UserName
- Their email address
- A password
- A payment method

Given those data the account will be created and will be available to the User. This functionality will allow only one account to be bound to a specific email address.

2.2.2 Search for CSs

Users with an account will be able to search for CSs, multiple selectors will be available to narrow or widen the scope of the search, hence allowing Users to find the best result for their needs. Available filters will be:

- Geographical area of interest
- Reharge price ranges
- Connector types available
- Charging speeds available

Users will be able to select a specific CS, consequently seeing the already booked time slots for it and further details as:

- Location
- Price
- Offers
- Booking schedule
- Available connectors
- Available charging speeds

2.2.3 Book a recharge at a CS

From the list of results for the **search for a CS** a User will be able to book a recharge at a CS, they will be able to select a specific socket and time slot among the available ones for the CS and reserve it for themselves. During their booked time slot they will then have to show up at the CS with their vehicle to perform the recharge.

2.2.4 Perform a recharge at a CS

When the User show up at a CS during the time slot the booked it for and they connect to the booked socket they will be able to log into eMall, reach their bookings and start the charging process for the one they are currently connected for. Once the charging process is going, the User will be able to interrupt it through eMall at any time, otherwise the recharge will terminate after the vehicle is fully charged or the time slot expires, notifying the User. After the completion of a recharge the User will be charged by eMall for the service.

2.2.5 Acquire price and energy mix information from DSOs

A CPO will be able to ask their CPMS to present them with the information regarding energy prices and mix of energy sources used by the different DSOs known to the system. The CPMS will recover the information from DSOs and return it to the CPO.

2.2.6 Choose prices, energy sources and battery usage policies for a CS

The CPMS will provide its CPO with a list of all CSs handled by the system, from that list the CPO will be able to modify the configuration of each CS independently, specifically they will be able to alter the nominal-price and user-price, the date after which the user-price will be reset to the nominal-price hence removing any active offer, the DSO that is currently supplying that CS and the policy with which that CS used its batteries.

2.2.7 Allow the CPMS to operate automatically

The CPO will as well be able to let the CPMS take decisions regarding the prices, DSO and energy source of a CS automatically on their behalf by toggling its “Automatic Mode” at will.

2.2.8 Allow the CPO to change the “Automatic Mode” policy used by the CPMS

The CPMS will allow the CPO to configure its policy which gets used when operating automatically, altering the way the CPMS will take decisions on its own.

2.3 User Characteristics

1. Unregistered User

A User which just accesses the list of available CS without being able to book recharges, he has the option to register and become a *Logged-In User* at any time.

2. Registered User

A User with a registered account within eMall which has yet to login-in with their credentials.

2. Logged-In User

A User with a registered account within eMall which has logged in via their credentials, can still access the CSs list and can also book a recharge. With their bookings he can then manage their charging sessions one he arrives at the CS.

3. **CPO**
   By CPO at large we mean a company whose employees have access to the company's CPMS, hence the actual human beings interacting with the STB will be CPO employees with credentials that allow them to authenticate with the CPMS. The tasks an employee might perform are acquiring DSOs' information, assigning a DSO to a CS owned by their company, changing the battery management policy for any CS controlled by the CPMS or adding/updating CS managed by the CPMS.
   The reason for CPOs being identified as a users instead of their employees is that the decisions justifying the operations performed by the employees cannot be taken by the employees, they are just the executors, decisions are taken by their superiors, hence we can view of the entire company as a user to avoid being too specific about who takes which decision and who interacts with the STB.
   Furthermore a CPO is likely to use its information system to interface with the CPMS, hence identifying the company as a user also prevents distinctions between direct CPMS access or access via the information system.

2.4 Assumptions, Dependencies and Constraints

2.4.1 Domain Assumption

Assumption	Description
D1	CSs never disconnect from their CPMSs nor malfunction in any way as long as they are booked or in use
D2	The only way to charge at a CS is by booking it via the eMSP
D3	CSs are aware of whether a vehicle is connected or not, and if it is connected they know its charging state
D4	CSs have the information required to reach out to their CPMS and connect to it by themselves, placing themselves under its control
D5	No vehicle is connected to a socket of a CS at any given time unless it's booked for that time slot
D6	The only user using a socket of a CS during a given time slot is the one who booked said socket
D7	No vehicle is ever disconnected from a socket unless no charging process is in progress
D8	A DSO's energy price is constant for all the locations where said DSO serves electricity
D9	A DSO's energy mix is constant for all the locations where said DSO serves electricity
D10	The user-price for a recharge at a CS is kept constant during the whole charging process
D11	CPOs pay DSOs for the used electricity via methods outside the scope of this document (Ex: direct bank transfer)
D12	eMSPs know in advance which CPMSs they are going to interact with
D13	Login credentials to grant access to CPMSs are manually created and inserted by CPOs

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The interface eMall will use is a web app, through which its Users will have available all product functions. Such web app will need to be usable both on desktop and mobile devices, since a mobile device is what users will most likely have available to them while at the CS.
In order to allow the User to select an area of interest while searching for CSs, the user interface will make use of a map.

What follows are some mockups for the eMSP's user interface design.

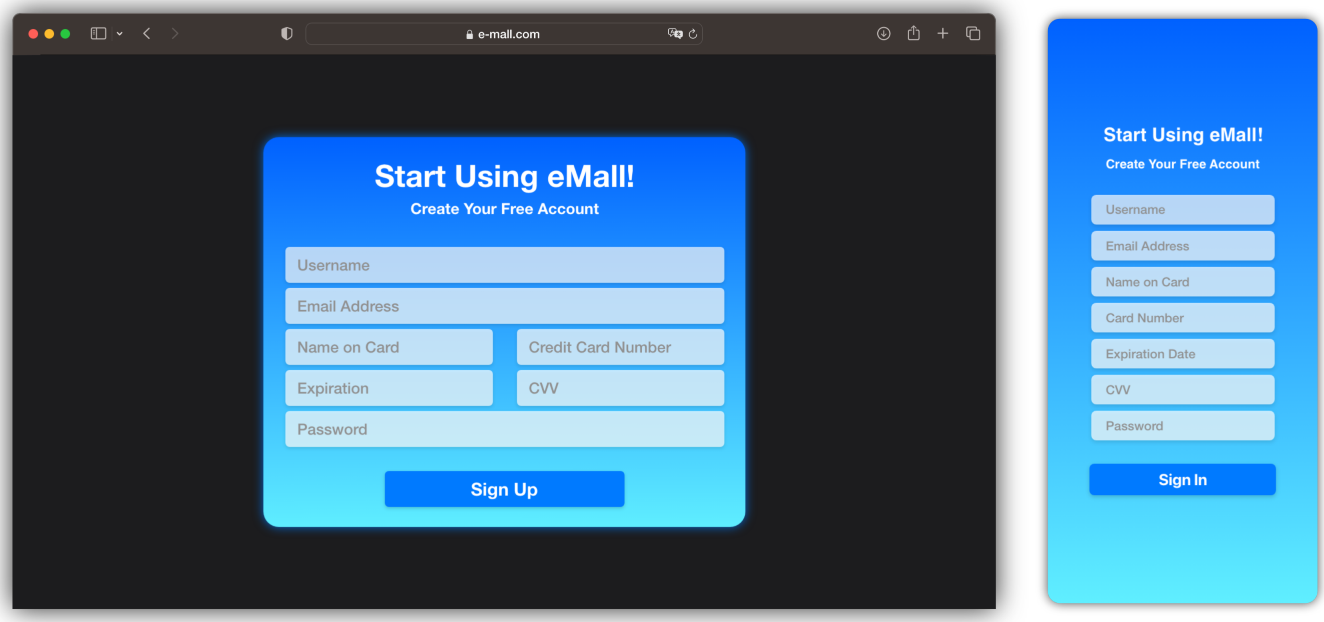

Figure 10: Signup page mockup

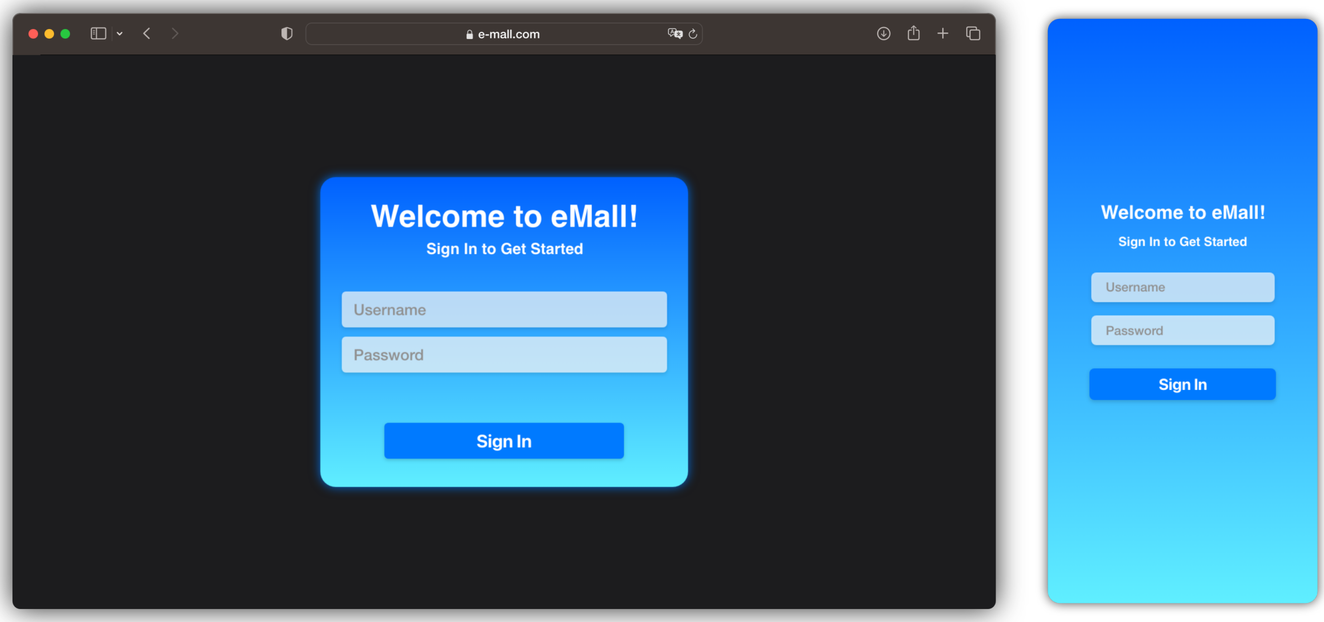

Figure 11: Login page mockup

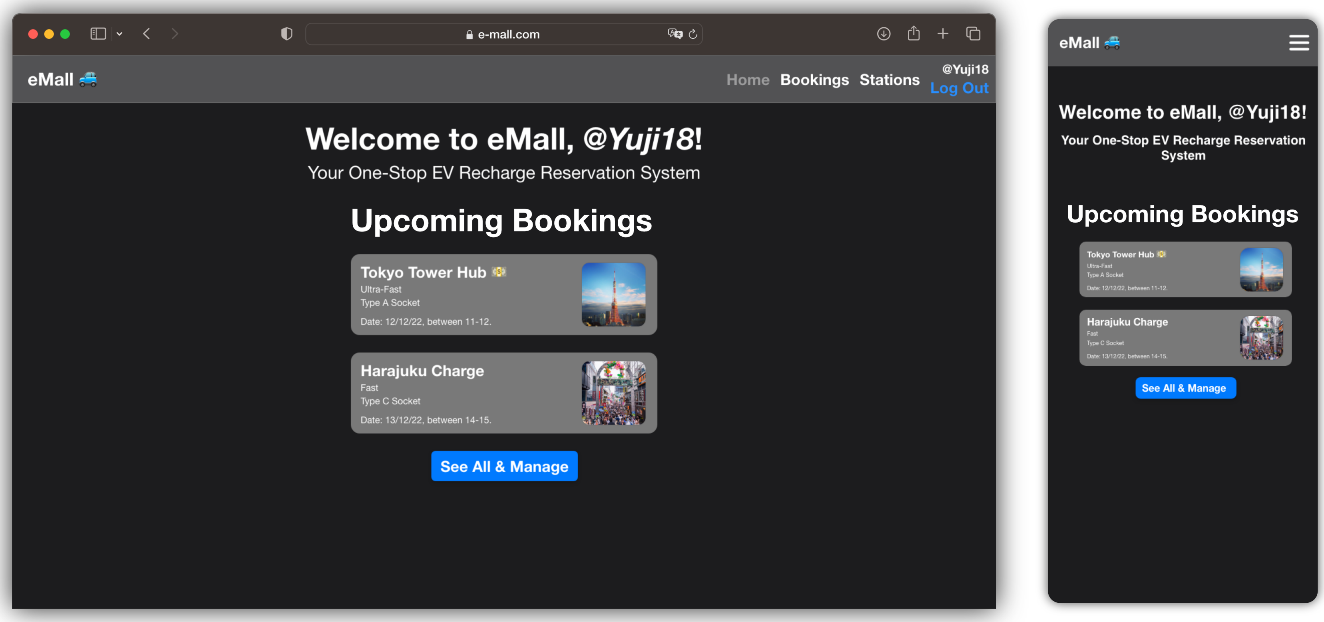

Figure 12: Home page mockup

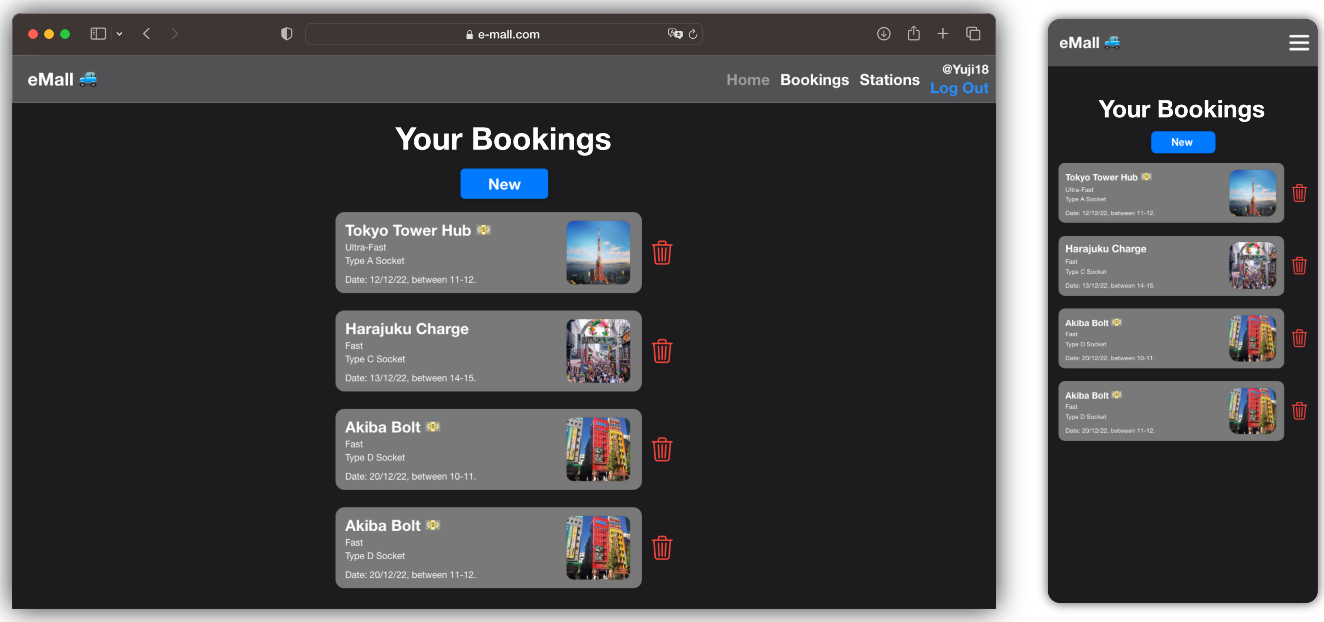

Figure 13: Bookings page mockup

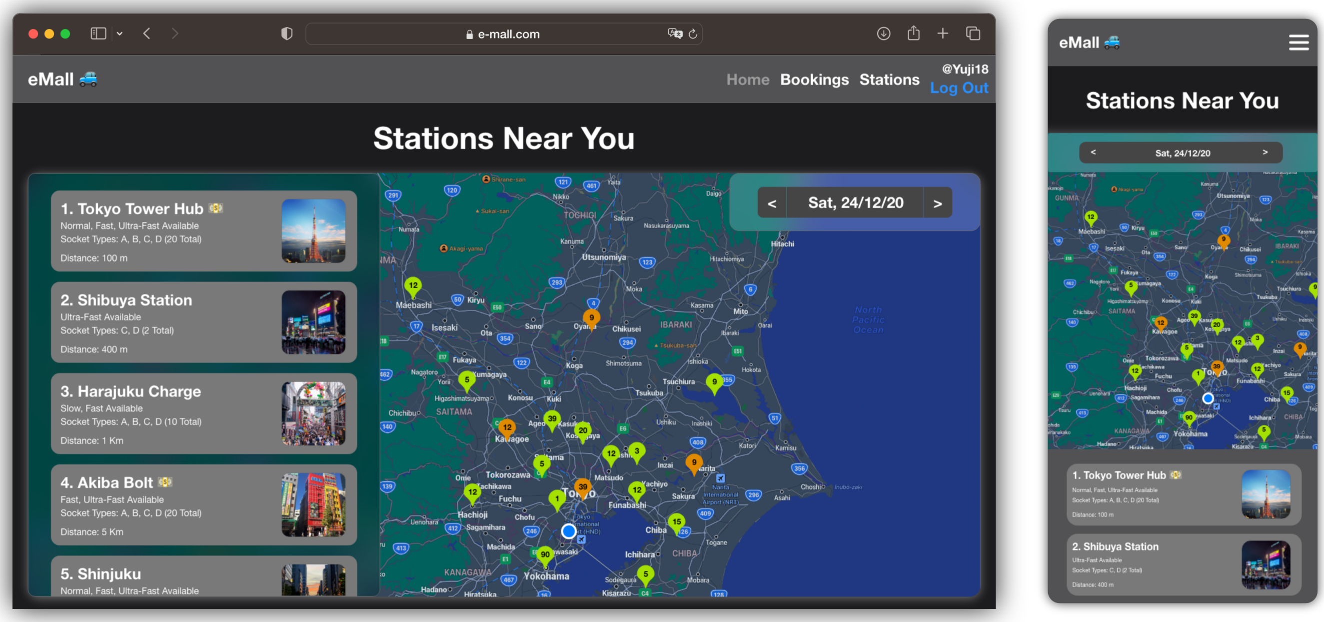

Figure 14: Stations page mockup

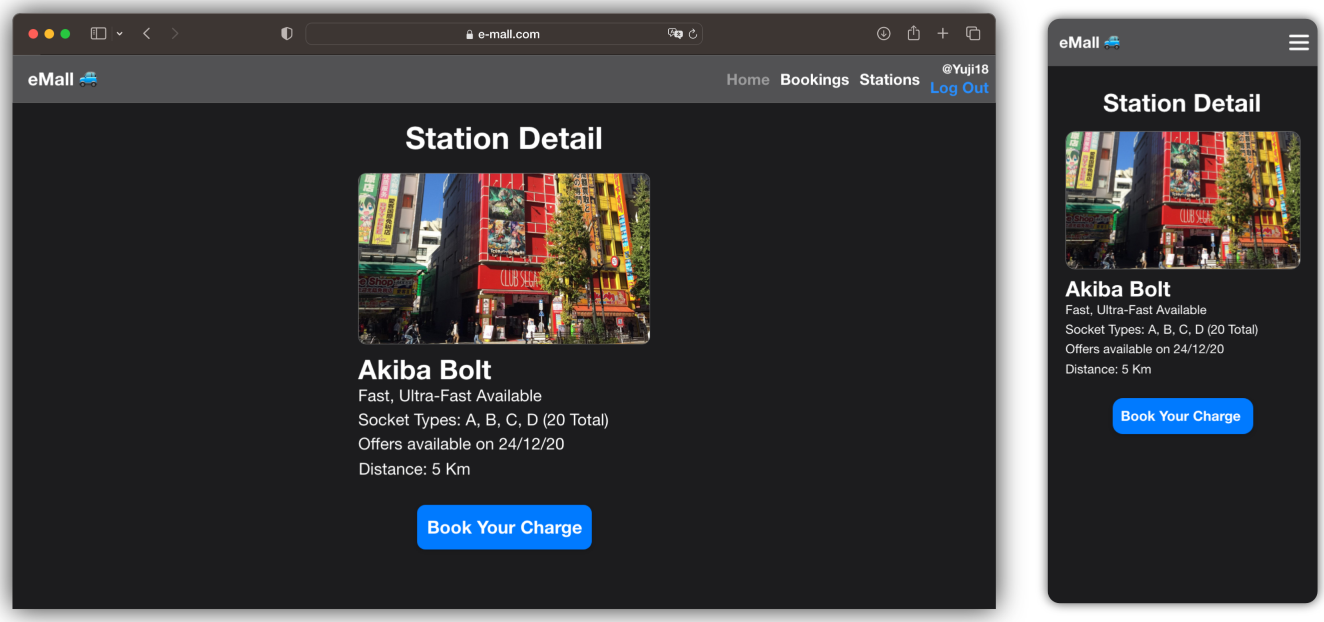

Figure 15: Station Details page mockup

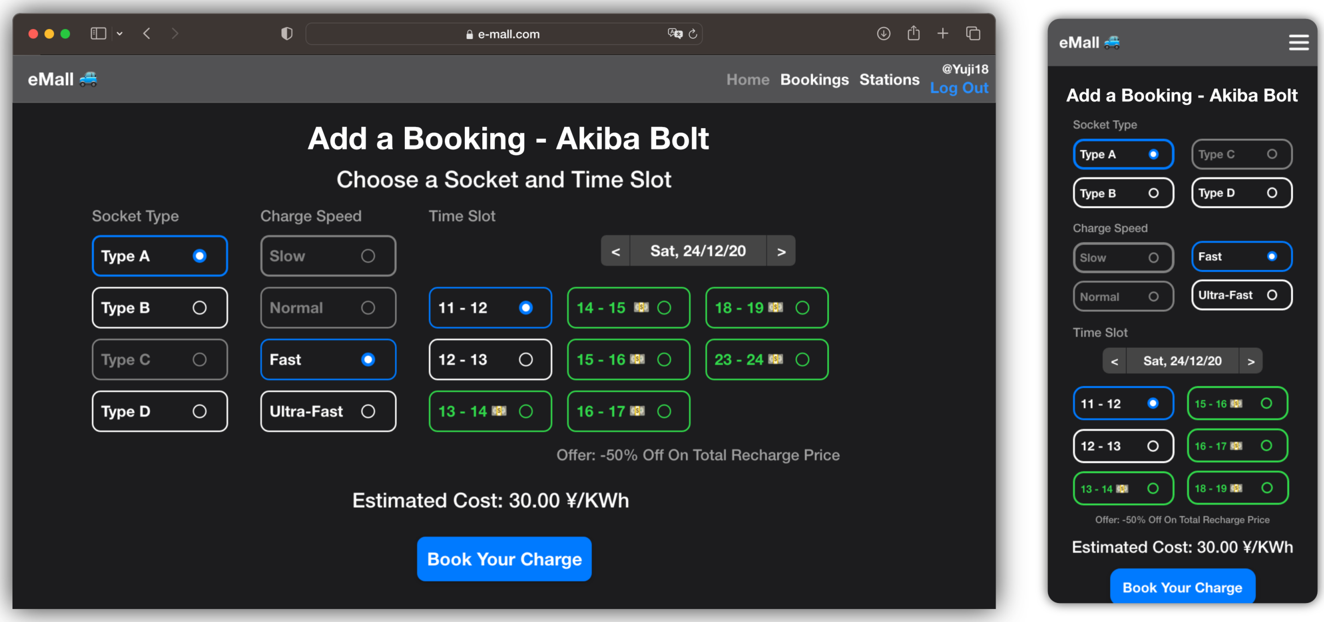

Figure 16: BookCharge page mockup

Similarly, CPMSs will offer their functionality to CPOs via a web portal, here there is no need to support anything other than a desktop device, since CPMSs are intended to be managed by employees in a company environment.

3.1.2 Hardware Interfaces

Users will require a device equipped with a web browser and an internet connection to access the eMall website, the hardware architecture of the specific device is irrelevant as long as it runs an OS equipped with a web browser.

CPOs will as well only require a device capable of web browsing, although in this case such device is intended to a be a desktop machine since its users will be the employees working at the CPOs from their offices.

CSs will need to be equipped with adequate hardware to monitor each socket and control circuitry for their batteries, if there are any connected to them. Furthermore CSs need to connect to and communicate with their respective CPMSs, hence a small computer within each CS capable of collecting sockets and battery(ies) data as well as interface with the internet (via 4G or a wired connection) is necessary.
Furthermore this document will assume that a CS is billed for its used electricity via a in loco electricity meter independently monitored by the DSO currently assigned to the CS by the CPMS.

3.1.3 Software Interfaces

Devices in the end of eMSP users will require an OS equipped with a network stack within which to run a web browser. The same is true on the CPOs side, on their machines there need to be an OS with network capabilities suitable for a web browser.

The computer installed on CS will need to have an OS with networking capabilities too, while also having adequate software to collect and process the information coming from the CS's sockets and eventually connected vehicles, as well as to manage eventual batteries connected to the CS.

3.1.4 Communication Interfaces

The main functionalities of the STB require cooperation of multiple components sometimes owned by different parties and not deployed on machines in the same local network, hence their interactions occur via adequate web APIs. The interactions requiring such interfaces are the following:

1. **eMSP acquiring CS information from CPMSs**
   Whenever a search for CSs, a booking or a recharge at a CS is performed via an eMSP, the ladder needs to fetch information regarding one or more CSs from CPMSs first, hence CPMSs need to expose a web API reachable by the eMSP to communicate the details regarding CSs.

2. **eMSP managing a recharge through a CPMS**
   When a charging process for a previous booking needs to be starter or terminated by a user via the eMSP, the eMSP has no control over the CS, hence needs to interface with the CPSM to request the latter to start/stop the charging process. Hence the CPMS needs to expose in it web API method for eMSP to start/stop charging procedures that were booked through them.

3. **eMSP website acquiring map data from external map provider**
   To allow the User to select its area of interest while looking for CS, a map will be used, and such map will need to be loaded dynamically from the eMSP website thanks to a map provider (Ex: Google Maps).

4. **eMSP requesting a payment via an external provider**
   After the completion of a charging process or when a fee is charged to a user, the eMSP needs to use the payment method a user has bound to its account to charge them for the service, hence it

needs to interface with an API which will allow the eMSP to have the amount due accredited to himself.

5. **CPMS monitoring and controlling its CSs**
   CSs need to know how to reach the CPMS responsible for them over internet, this way whenever a CS is deployed it will automatically find its CPMS, authenticate with it and setup itself under its control. To this end CPMSs need to expose a web API allowing for new CSs to authenticate with them, and CSs need to offer to the CPMS they are connected to an API to allow the CPMS to manage them.

6. **CPMS acquiring DSOs price and energy mix information**
   To allow CPMSs to know their energy prices and mixes, DSO must have an API available via web so that CPMSs can recover such information.

3.2 Functional Requirements

3.2.1 Use cases

1. New user registration

Actor	Unregistered User
Entry condition	The User does not have an eMall account yet and reached eMall's login page
Event flow	1. The User selects the option to register on eMall 2. The User enters his desired UserName for the platform, his email address, a password and a payment method for his new account 3. The User submits the inserted data to eMall 4. eMall registers the account and the User can now log in with it
Exit condition	The User's account has been created
Exceptions	• The User's inserted data was not complete or did not comply with the requested formats • The User's email is already associated to an existing account In all those situation an error is presented to the User.

2. User login

Actor	Registered User
Entry condition	The User already has an eMall account, is not currently logged-in, and reached eMall's login page
Event flow	1. The User selects the option to log in on eMall 2. The User enters his account's email and password 3. The User submits the inserted data to eMall 4. eMall verifies the credential's validity and presents the user with the home page
Exit condition	The User logged in on eMall
Exceptions	• The User's inserted data was not complete or did not comply with the requested formats • The User's credentials were not valid for any account In all those situation an error is presented to the User.

3. User searching and booking a CS

Actor	Logged-in User
Entry condition	The User has already logged in and is on eMall's home page
Event flow	1. The User selects the "search" option for CSs 2. The User is presented with an initial list of CSs (could be empty) and the filters to update the search results **(filters list here)** 3. The User can repeat the search to his hearth's contempt by updating the filters 4. The User selects a CS to be shown its details **(details list here)** 5. The User chooses to book a recharge at the selected CS 6. The User inserts the desired socket and time slot for his booking 7. eMall registers the User's booking and confirms it to the User
Exit condition	The User booked a recharge at a CS
Exceptions	• The User's inserted values in filters did not comply with accepted formats • The User's inserted data was not complete or did not comply with the requested formats • The User's selected socket is already occupied for the chosen time slot • The User already has another booking that presents an overlapping time slot with the currently selected one In all those situation an error is presented to the User.

4. User deleting one of his bookings

Actor	Logged-in User
Entry condition	The User has already logged in and is on eMall's home page
Event flow	1. The User selects the option to see his bookings' list 2. The User selects the booking they want to delete from the list 3. The User chooses the delete option on the selected meeting 4. eMall deletes the selected booking and confirms it to the User
Exit condition	The User logged in on eMall
Exceptions	No exception can happen during within user case.

5. User starting a charging process

Actor	Logged-in User
Entry condition	The User has already logged in and is on eMall's home page, has arrived at a CS during their booked time slot and has connected their vehicle to the booked socket
Event flow	1. The User selects the option to see his bookings' list 2. The User selects the booking made for the current time slot (eMall might highlight it ahead of time) 3. The User is presented with the booking's status and the option to start the charging process 4. The User chooses to start the charging process 5. eMall, via the CPMS, makes the CS's socket start to recharge the connected vehicle
Exit condition	The CS's socket started charging the connected vehicle
Exceptions	No exception can happen during within user case.

6. User interrupting a charging process

Actor	Logged-in User
Entry condition	The User has already logged in and is on eMall's home page while his current booking has an ongoing charging process
Event flow	1. The User selects the option to see his bookings' list 2. The User selects the booking made for the current time slot with the active charging process 3. The User is presented with the charging process's status and the option to interrupt it 4. The User chooses to interrupt the charging process 5. eMall, via the CPMS, makes the CS's socket stop the charging of the connected vehicle 6. eMall charges the User for the received service through adequate external services, using the User's registered payment method
Exit condition	The booking's charging procedure has ended and the payment has succeeded
Exceptions	• The payment is rejected In the above situation eMall retries the payment a given number of times, if it never succeeds eMall notifies the User to change their payment method.

7. Charging procedure self-terminates and User receives a notification

Actor	Logged-in User
Entry condition	The User has already logged in and is on eMall's home page while his current booking has just reached one of its termination conditions: • The booked time slot has expired • The connected vehicle is completely charged
Event flow	1. eMall, via the CPMS, makes the CS's socket stop the charging of the connected vehicle if it hasn't already halted by itself 2. eMall charges the User for the received service through adequate external services, using the User's registered payment method 3. The User receives a notification from eMall telling them that their charging procedure just ended
Exit condition	The booking's charging procedure has ended, the payment has succeeded and the User has been notified
Exceptions	• The payment is rejected In this situation eMall retries the payment a given number of times, if it never succeeds eMall notifies the User to change their payment method. • The User did not receive the notification In this situation eMall send the notification again every predetermined amount of time for a configured number of attempts.

8. User does not show up for a booked recharge

Actor	Registered User
Entry condition	The Registered User had a booking with eMall whose time slot just expired without the User ever starting the charging process
Event flow	1. eMall detects the the expiration of the breached booking and deletes it 2. eMall charges the User a fee for the reservation
Exit condition	The payment of the fee succeeds
Exceptions	• The payment is rejected In the above situation eMall retries the payment a given number of times, if it never succeeds eMall notifies the User to change their payment method.

9. CPO requests information on DSOs' energy prices and mix of sources

Actor	CPO (employee)
Entry condition	A CPO employee has authorized access to the company's CPMS and performing this task is part of his job, hence reached the CPMS login page
Event flow	1. The CPO employee authenticates with the CPMS and reaches its home page 2. The CPO employee queries the CPMS for a list of the energy prices and mixes of DSOs know to the system 3. The CPMS contacts the all APIs of known DSOs to get their energy price and energy mix information 4. The CPMS shows the fetched information to the CPO employee
Exit condition	The CPO employee has the information regarding DSOs energy prices and mixes
Exceptions	• A DSO API does not answer the CPMS's request In the above situation the CPMS retries until a timeout is met, then ignores that DSO.

10. CPO assigns energy sources and battery usage policies for a CS

Actor	CPO (employee)
Entry condition	A CPO employee has authorized access to the company's CPMS and performing this task is part of his job, hence reached the CPMS login page. The CPMS must be in "Manual Mode".
Event flow	1. The CPO employee authenticates with the CPMS and reaches its home page 2. The CPO employee requests a list of the CPMS's CS 3. The CPO employee selects a CS and is presented with its details 4. The CPO employee chooses to alter the CS's currently assigned DSO and/or its battery usage policy 5. The CPO employee assigns the CS's DSO and/or its battery usage policy 6. The CPMS communicates the updated configuration to the CS
Exit condition	The CS's DSO and/or battery usage policy has been updated
Exceptions	• The DSO and/or battery usage policy inserted by the CPO employee are invalid In the above situation the CPMS does not updated the CS and presents an error to its user.

11. CPO assigns nominal-price and user-price to a CS

Actor	CPO (employee)
Entry condition	A CPO employee has authorized access to the company's CPMS and performing this task is part of his job, hence reached the CPMS login page. The CPMS must be in "Manual Mode".
Event flow	1. The CPO employee authenticates with the CPMS and reaches its home page 2. The CPO employee requests a list of the CPMS's CS 3. The CPO employee selects a CS and is presented with its details 4. The CPO employee chooses to alter the CS's nominal-price and user-price 5. The CPO employee assigns new prices and an offer reset date to the CS 6. The CPMS communicates the updated prices to the CS
Exit condition	The CS's prices have been updated
Exceptions	• The prices inserted by the CPO employee are invalid In the above situation the CPMS does not updated the CS and presents an error to its user.

12. CPO toggles the CPMS operating mode

Actor	CPO (employee)
Entry condition	A CPO employee has authorized access to the company's CPMS and performing this task is part of his job, hence reached the CPMS login page.
Event flow	1. The CPO employee authenticates with the CPMS and reaches its home page 2. The CPO employee toggles the CPMS operating mode from "Manual Mode" to "Automatic Mode" or vice-versa 3. The CPMS applies the change
Exit condition	The CPMS's operating mode has been toggled w.r.t. its state before the employee's actions
Exceptions	No exception can happen during within user case.

13. CPO changes the CPMS's "Automatic Mode" policy

Actor	CPO (employee)
Entry condition	A CPO employee has authorized access to the company's CPMS and performing this task is part of his job, hence reached the CPMS login page.
Event flow	1. The CPO employee authenticates with the CPMS and reaches its home page 2. The CPO employee reaches the page with the current policy in use by the CPMS 3. The CPO employee updates the policy's weights and thresholds accordingly to its task 4. The CPMS saves and starts using the new policy
Exit condition	The CPMS's policy has been successfully updated
Exceptions	• The weights and thresholds inserted by the CPO employee are invalid In the above situation the CPMS does not updated the policy and presents an error to its user.

3.2.2 Use case diagrams

1. Unregistered User

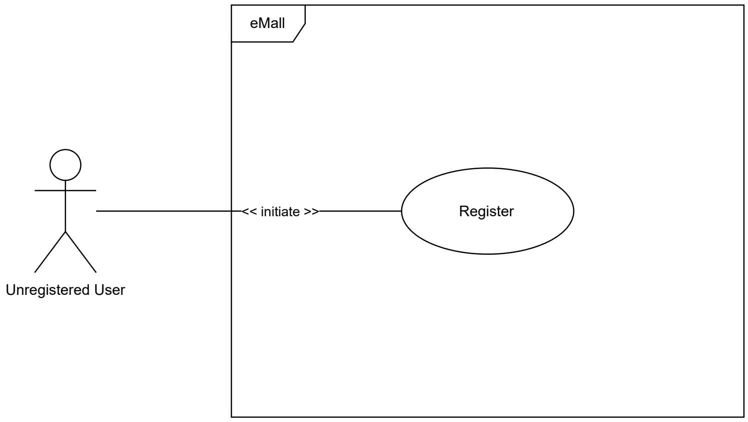

Figure 17: Unregistered User

2. Registered User

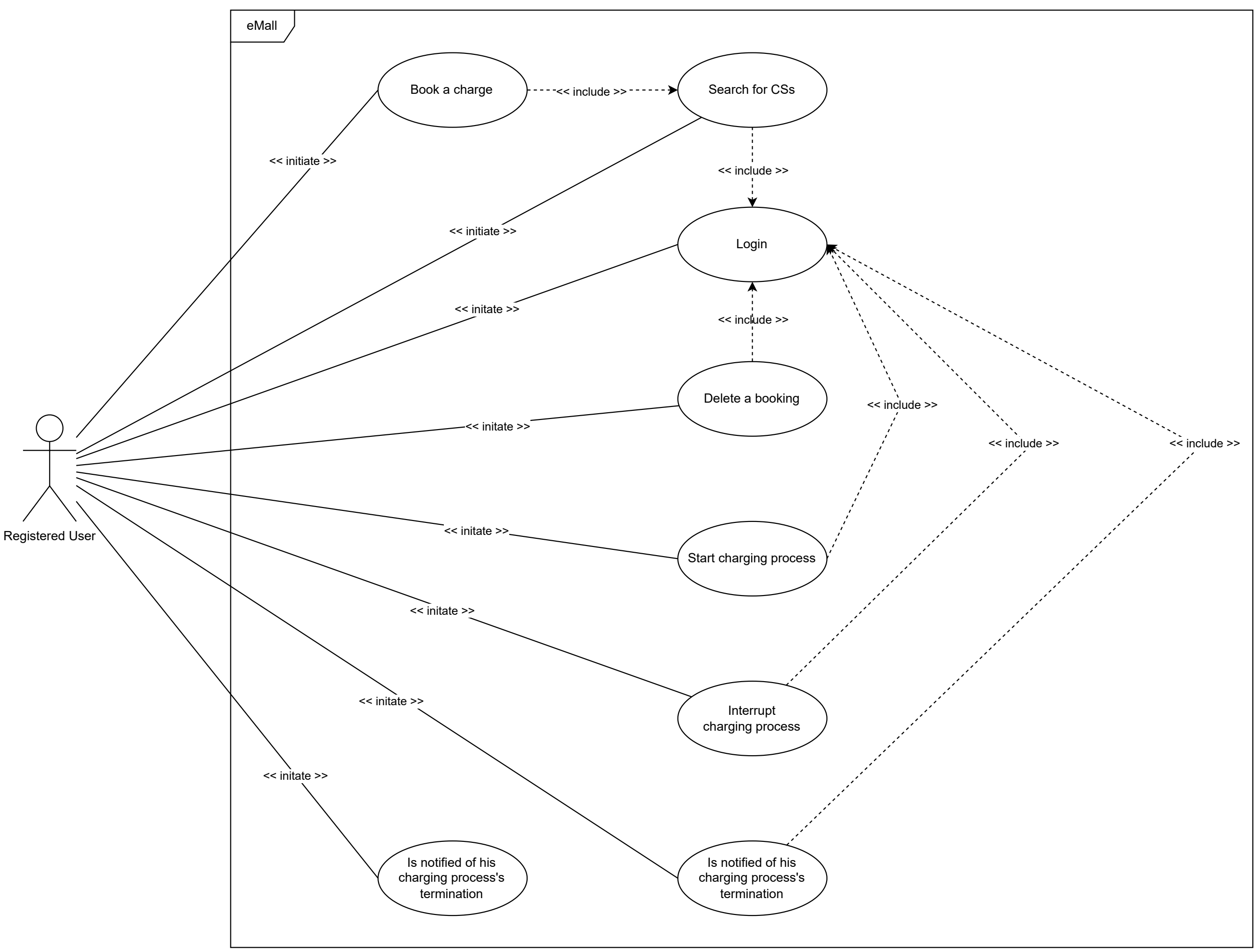

Figure 18: Registered User

3. CPO

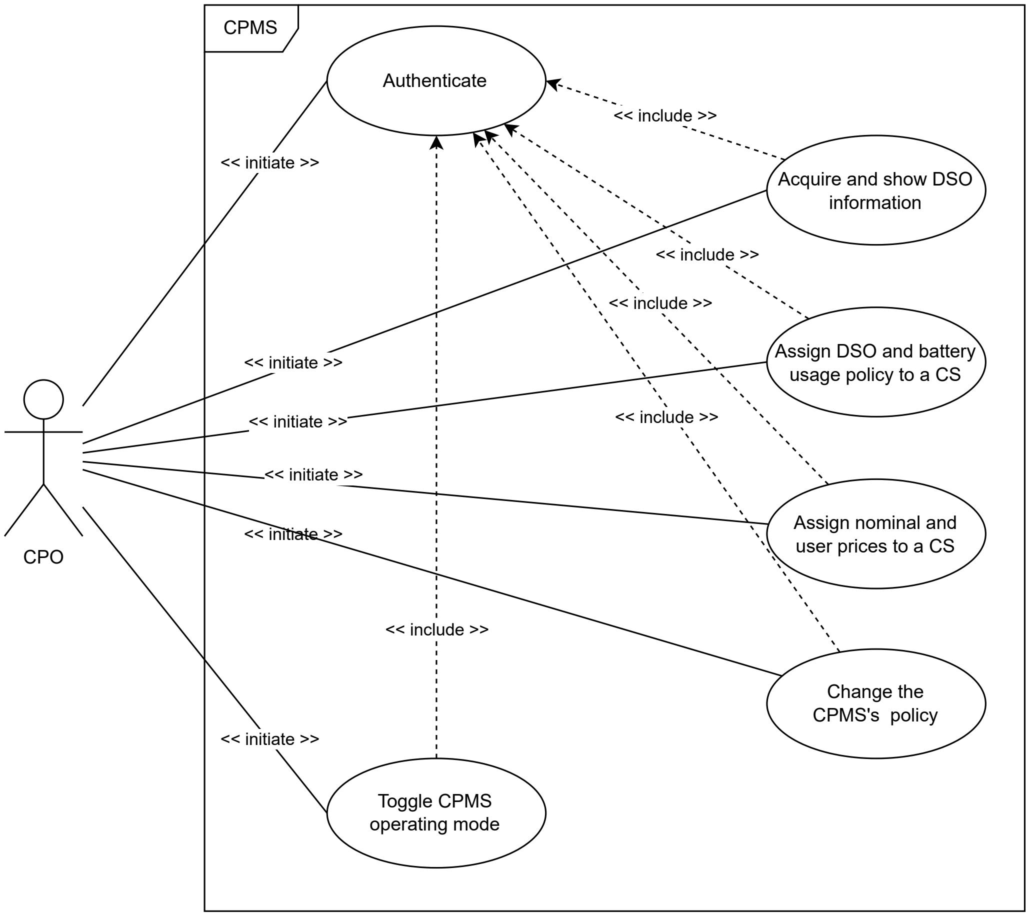

Figure 19: CPO

3.2.3 Sequence diagrams

Here are the sequence diagrams corresponding one-to-one with the use cases in the above section, the entry conditins hold for the diagrams as well, hence things like "login" are not repeated in every diagram.

1. **New user registration**

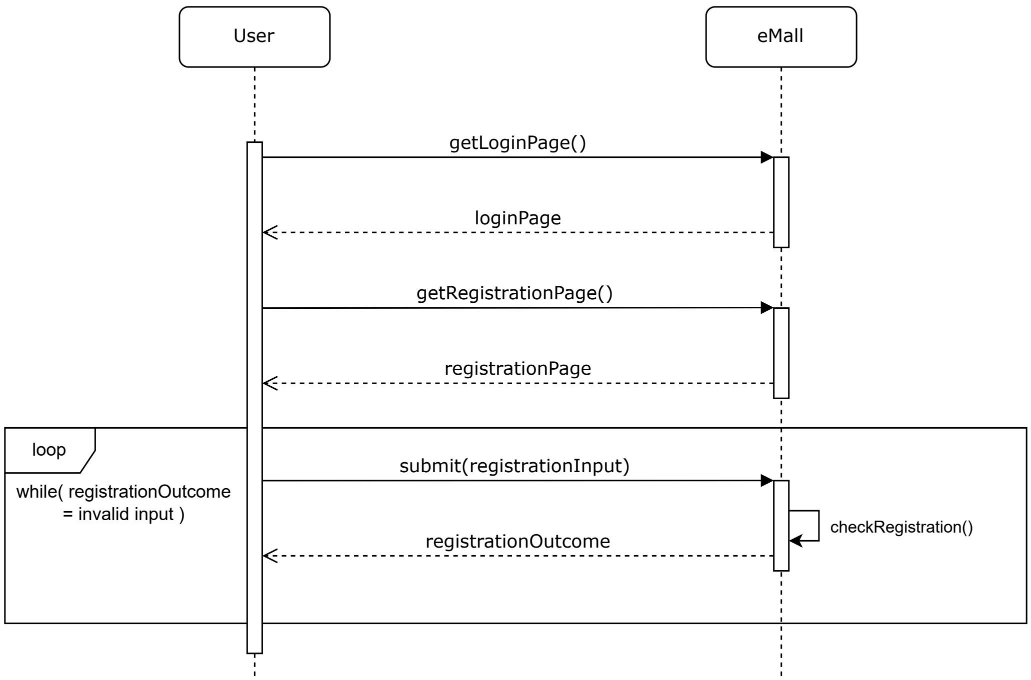

Figure 20: New user registration

2. **User login**

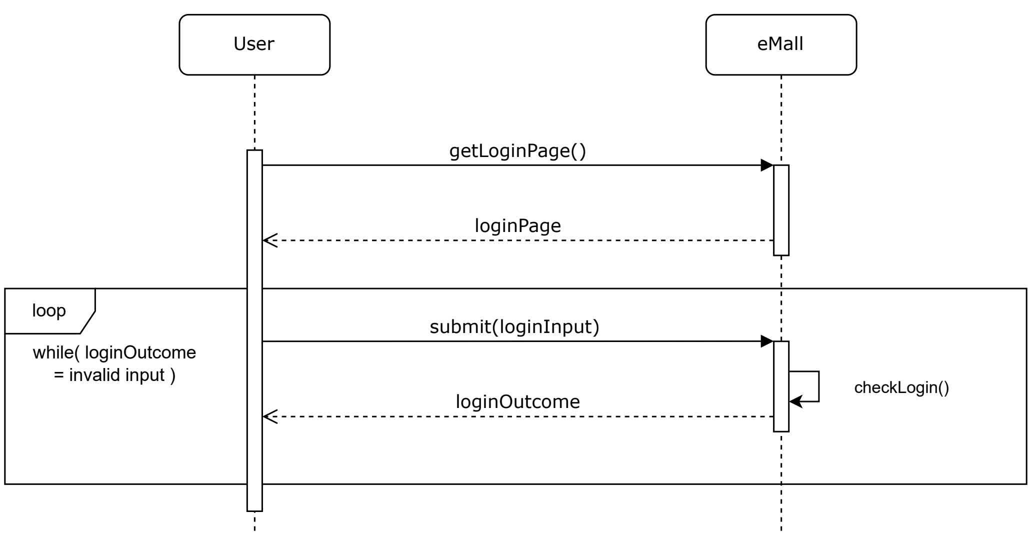

Figure 21: User login

3. User searching and booking a CS

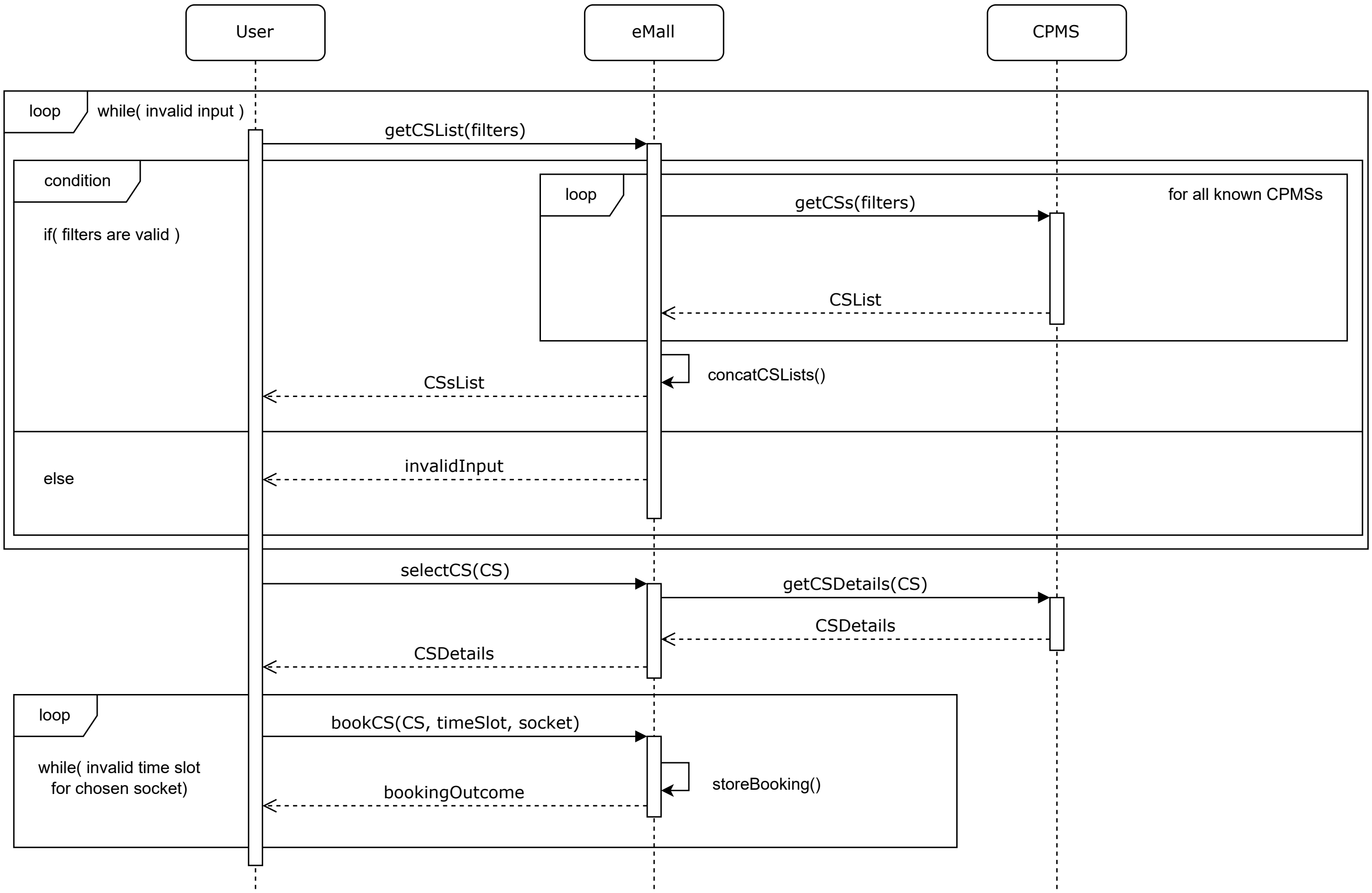

Figure 22: User searching and booking a CS

4. User deleting one of his bookings

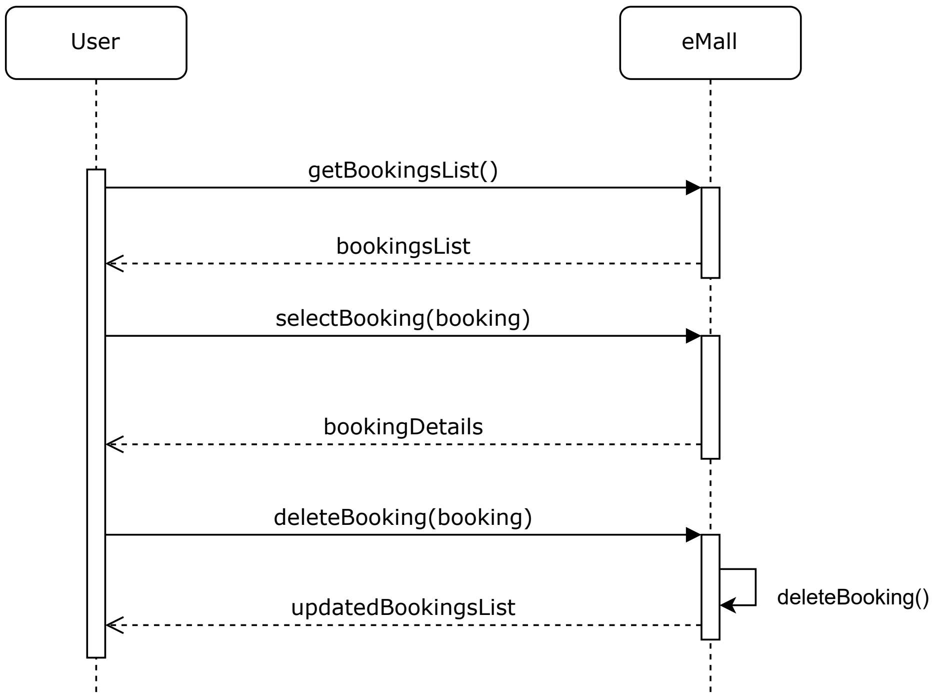

Figure 23: User deleting one of his bookings

5. User starting a charging process

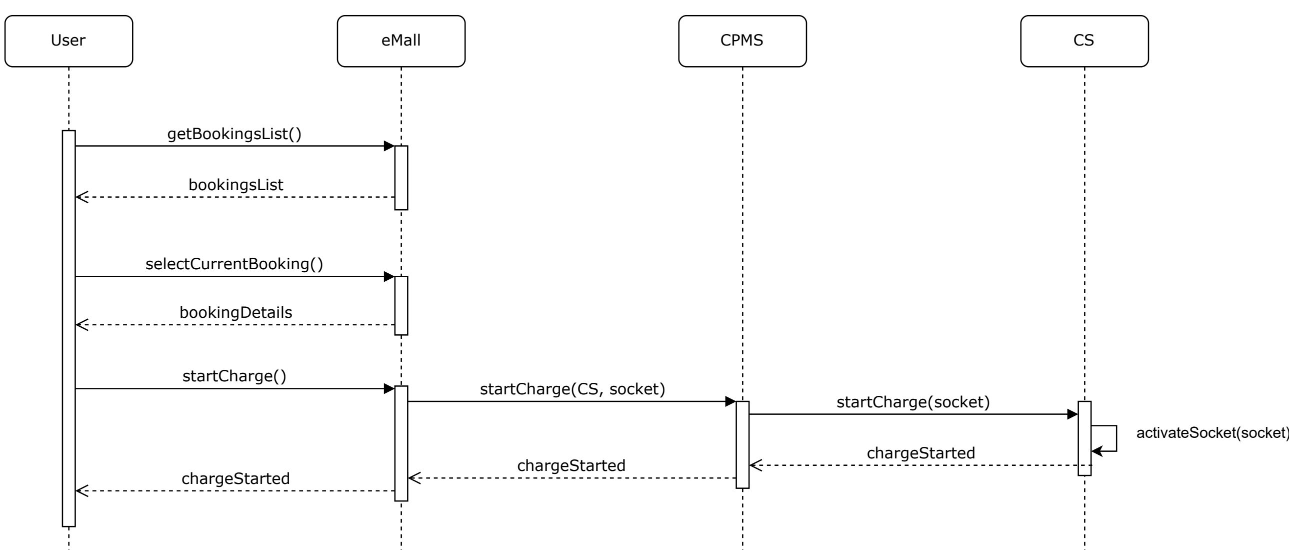

Figure 24: User starting a charging process

6. User interrupting a charging process

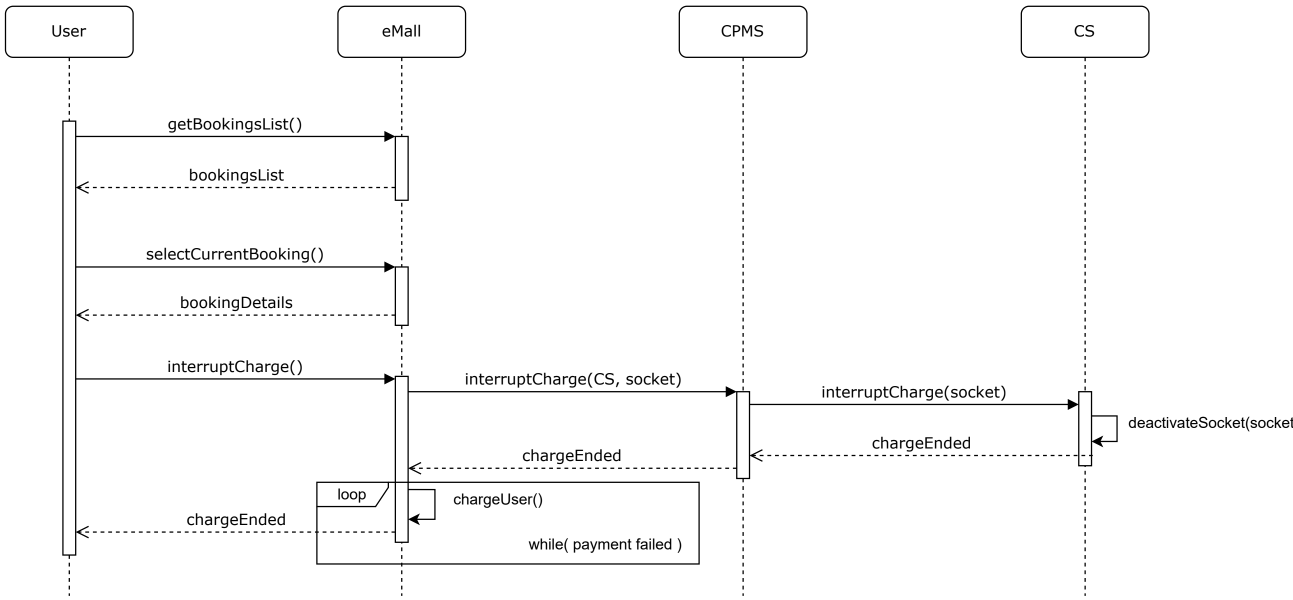

Figure 25: User interrupting a charging process

7. Charging procedure self-terminates and User receives a notification

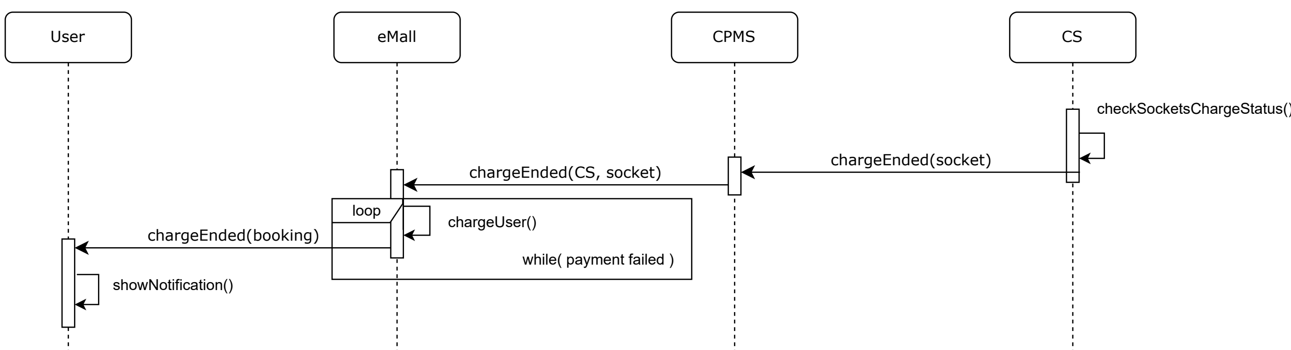

Figure 26: Charging procedure self-terminates and User receives a notification

8. User does not show up for a booked recharge

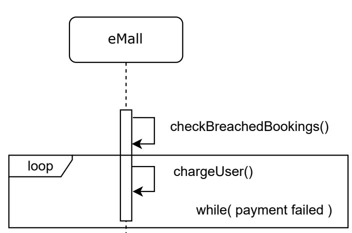

Figure 27: User does not show up for a booked recharge

9. CPO requests information on DSOs' energy prices and mix of sources

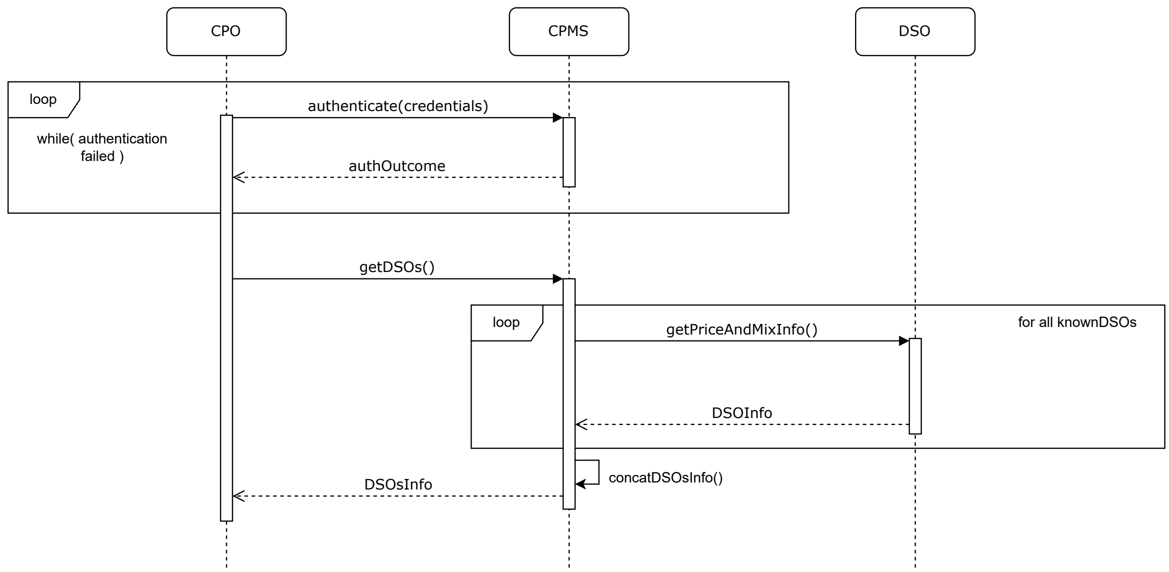

Figure 28: CPO requests information on DSOs' energy prices and mix of sources

10. CPO assigns nominal-price, user-price, energy sources and battery usage policies for a CS

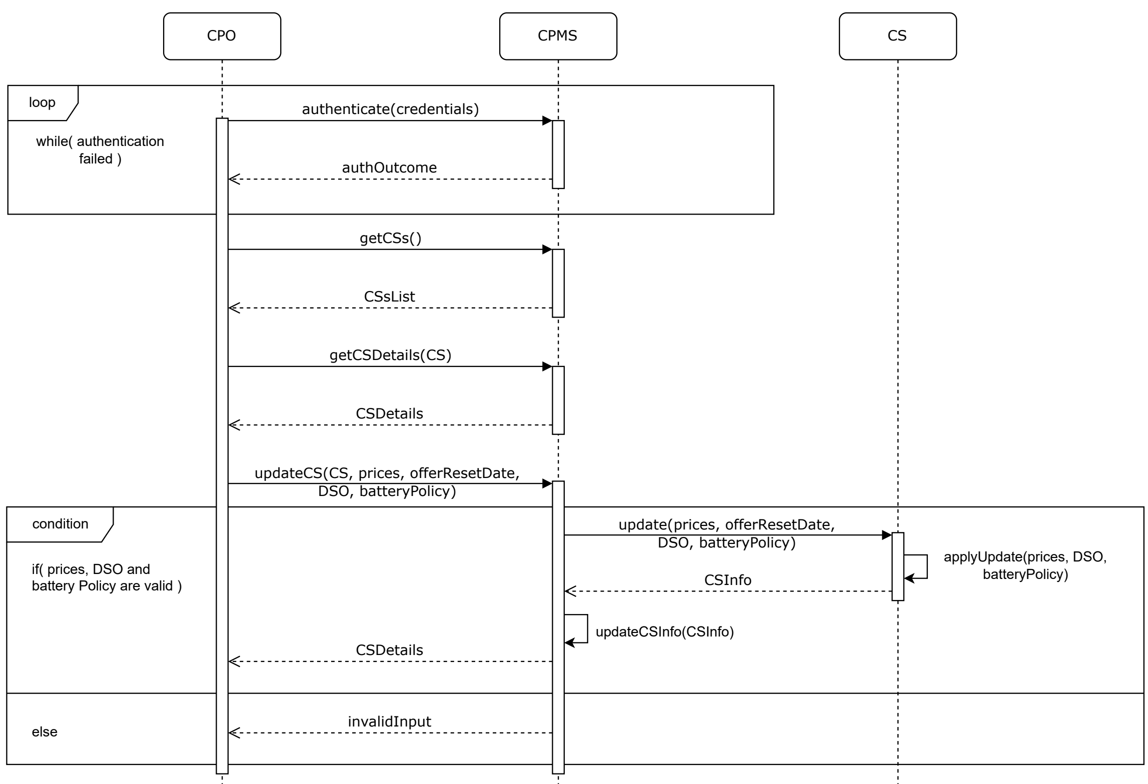

Figure 29: CPO assigns nominal-price, user-price, energy sources and battery usage policies for a CS

11. CPO toggles the CPMS operating mode

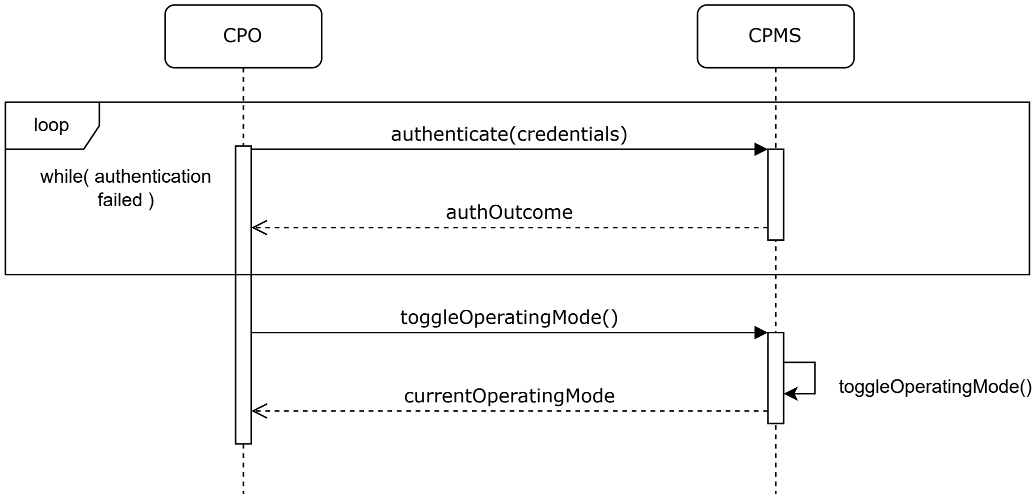

Figure 30: CPO toggles the CPMS operating mode

12. CPO changes the CPMS's "Automatic Mode" policy

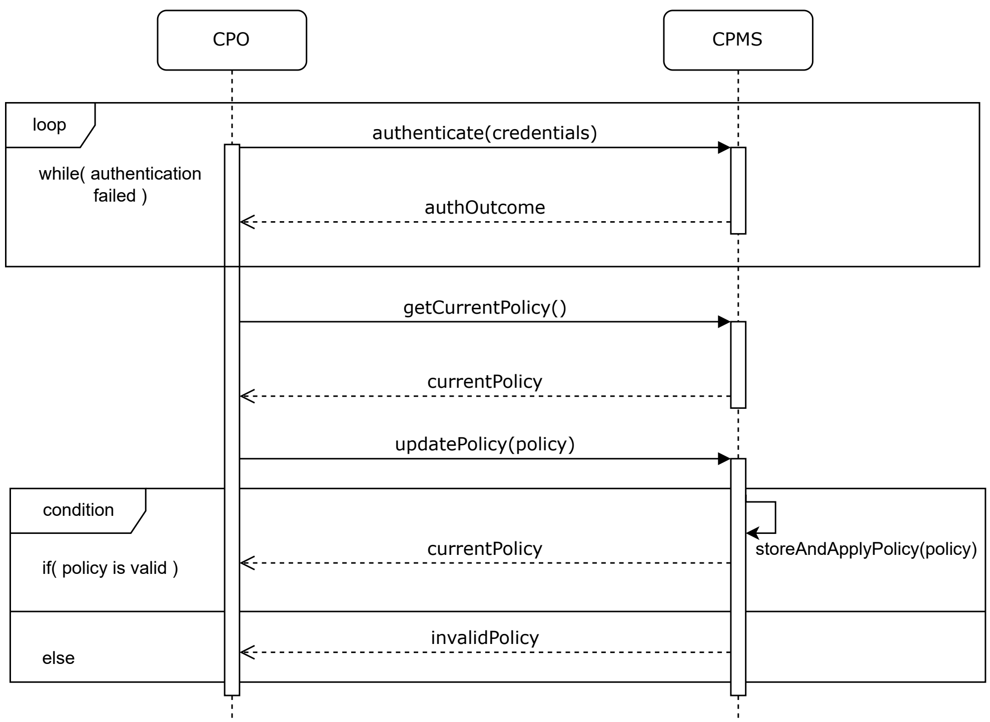

Figure 31: CPO changes the CPMS's "Automatic Mode" policy

3.2.4 Requirements

Id	Requirement
R1	The eMSP shall allow a registered user to login
R2	The eMSP shall allow an unregistered user to register on the platform
R3	The eMSP should report to the user when they sent invalid input data through a form
R4	The eMSP should report to the user when they attempted to perform an unauthorized action
R5	The eMSP should report to the user when the platform encountered an error while processing the action
R6	The eMSP should report to the user any successful action performed
R7	The eMSP shall allow a logged-in user to view charging stations nearby
R8	The eMSP shall allow a logged-in user to know the cost of a recharge at a CS
R9	The eMSP shall allow a logged-in user to know about ongoing special offers present at a CS
R10	The eMSP shall allow a logged-in user to book a recharge at a free socket of a CS for a given time slot
R11	The eMSP shall allow a logged-in user to cancel any of its booked recharges before their scheduled time
R12	The eMSP shall not allow a user to book a recharge at a socket of a CS which has already been booked for the chosen time slot
R13	The eMSP shall not allow a user to book multiple recharges with overlapping time slots
R14	The eMSP shall allow a logged-in user to see their bookings
R15	The eMSP shall allow a logged-in user to start the charging process for one of their booked recharges
R16	The eMSP shall not allow a user without a booking to start the charging process
R17	The eMSP shall not allow a user to start the charging process as long as the socket is vacant
R18	The eMSP should notify a logged-in user when their ongoing charging processes is complete
R19	The eMSP should notify a logged-in user when their ongoing charging process is terminated due to the expiration of its booked time slot
R20	The eMSP shall allow a logged-in user to terminate their charging process earlier
R21	The eMSP should charge a user for everyone of his charging processes as soon as they finish
R22	The eMSP shall charge a user for a fee if they do not start a recharge during their booking's time slot
R23	The eMSP shall be able to offer its functionalities to multiple users at once
R24	The eMSP shall be able to pay CPOs for the amount due for recharges performed through them since the last payment
R25	The CPMS should be able to acquire the location from CSs

Id	Requirement
R26	The CPMS should be able to acquire the internal status from CSs
R27	The CPMS should be able to acquire the external status from CSs
R28	The CPMS should be able to start charging a vehicle connected to a CS socket
R29	The CPMS should be able to monitor the charging process of a vehicle
R30	The CPMS should be able to acquire from DSOs their current energy prices
R31	The CPMS should be able to acquire from DSOs their energy mix
R32	The CPMS shall allow its CPO to authenticate with it
R33	The CPMS shall allow its CPO to assign a DSO to a CS as its energy provider when operating in "Manual Mode"
R34	The CPMS should be able to automatically assign a DSO to a CS as its energy provider when operating in "Automatic Mode"
R35	The CPMS shall allow the CPO to decide the energy source management policies when operating in "Manual Mode"
R36	The CPMS should be able to automatically choose the current energy source for a CS according to the given policy when operating in "Automatic Mode"
R37	The CPMS shall allow its CPO to assign a nominal-price, a user-price and an offer reset date to a CS when operating in "Manual Mode"
R38	The CPMS should be able to automatically assign a nominal-price, a user-price and an offer reset to a CS when operating in "Automatic Mode"
R39	The CPMS shall allow the CPO to choose its policy for operating in "Automatic Mode"
R40	The CPMS shall allow the CPO to choose whether it acts automatically or manually
R41	The CPMS shall never take a decision automatically when it is set in "Manual Mode"
R42	The CPMS should report to the user when they sent invalid input data
R43	The CPMS should report to the user when they attempted to perform an unauthorized action
R44	The CPMS should report to the user when the platform encountered an error while processing the action
R45	The CPMS should be able to decide whether to start charging the internal batteries of a CS (if present)

3.2.5 Mapping on goals

Goal	Domain assumption	Requirement
G1	D4, D12	R1, R2, R3, R4, R5, R6, R7, R23, R25, R27
G2	D4, D8, D9, D12	R1, R2, R3, R4, R5, R6, R8, R9, R23, R27
G3	D1, D2, D4, D12	R1, R2, R3, R4, R5, R6, R7, R10, R11, R12, R13, R22, R23, R25, R27
G4	D1, D2, D3, D4, D5, D6, D7, D10, D11, D12	R1, R2, R3, R4, R5, R6, R14, R15, R16, R17, R18, R19, R20, R21, R24, R26, R27, R28, R29, R30
G5	D1, D4, D8, D9, D11, D13	R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, R41, R42, R43, R44, R45
G6	D1, D4, D10, D13	R30, R32, R37, R38, R40, R41, R42, R43, R44, R45

3.2.6 Mapping on requirements

Use case	Requirements
1. New user registration	R2, R3, R4, R5, R6, R23
2. User login	R1, R3, R4, R5, R6, R23
3. User searching and booking a CS	R1, R3, R4, R5, R6, R7, R8, R9, R10, R12, R13, R23, R25, R27
4. User deleting one of his bookings	R1, R3, R4, R5, R6, R11, R14
5. User starting a charging process	R1, R3, R4, R5, R6, R14, R15, R16, R17, R18, R19, R21, R23, R24, R26, R28, R29
6. User interrupting a charging process	R1, R3, R4, R5, R6, R14, R20, R21, R23, R24, R26, R29
7. Charging procedure self terminates and User receives a notification	R1, R3, R4, R5, R18, R19, R23, R24, R26, R29
8. User does not show up for a booked recharge	R22, R26
9. CPO requests information on DSOs' energy prices and mix of sources	R29, R30, R32, R42, R43, R44
10. CPO assigns energy sources and battery usage policies for a CS	R25, R26, R27, R30, R31, R33, R34, R35, R36, R39, R41, R42, R43, R44, R45
11. CPO assigns nominal-price and user-price to a CS	R25, R26, R27, R30, R37, R38, R39, R41, R42, R43, R44
12. CPO toggles the CPMS operating mode	R39, R42, R43, R44

3.3 Performance Requirements

The system should be able to handle concurrent users, who may want to book or start a recharge at the same time in different geographical locations. As such, the systems should be capable of handling a reasonably high number of concurrent users, which we hypothesize to be in the order of 10.000.
Secondly, the system should operate with reasonable response times (of max 5 seconds), necessary to provide timely feedback to users who are booking or paying for a recharge, as well as users who want to start their recharge quickly.
Finally, the system should work properly even when accessed from a low-speed connectivity, such as 3G networks, since it is expected that users who might want to start a recharge through the system are probably going to access it through their mobile phone. As such, the overall amount of data exchanged with the user should be kept at a minimum.

3.4 Design Constraints

3.4.1 Standards compliance

The system should adhere to the standards and regulations imposed by the GDPR privacy law, since it is expected to be deployed (entirely or partially) in European countries.
Secondly, the APIs used for communications between the eMSP and CPMSs, as well as APIs used to share data between other parties in the STB should follow the current industry standard for API design which, at the time of writing, is the REST architecture.
Finally, the STB should function as expected on all modern web browser, without loss of functionalities according to the browser chosen by the user.

3.4.2 Hardware limitations

Given that the system will be accessed from the Web and will require a connection to the platform to carry out its functions, the hardware must support some form of Internet connectivity (such as Wi-Fi, Ethernet or Cellular) and must have at least one modern Browser installed.
Furthermore, given that it is expected that the STB is mostly going to be accessed on mobile devices, it should not require a high processing power in order to perform its front-end functions.

3.5 Software System Attributes

3.5.1 Reliability

The system should be highly reliable and be up and running most of the times, since without it the user would not be able to start a recharge previously booked by the system. Hence regular maintenance is expected, together with some precautions that help reduce the risk of total outages in case of failures (e.g. geographical redundancy).

3.5.2 Availability

High availability is crucial for this system for 2 main reasons:

- Its entire business model is based off of transactions made through the system (users booking and, later, paying for recharges)
- Users need the system to be up and running in order to start a recharge, even if it was already booked

Taking this into consideration, it is deemed appropriate to have an availability of 99.99%, which would result in a maximum of approximately 52 minutes of downtime in an entire year. To achieve such a high figure, it may be essential to run components in parallel, making use of redundant machines and databases.
Furthermore, since the system also relies on external components (including, but not limited to the systems owned by DSOs), it should be noted that these systems may have lower availability constraints, and as such our STB should not fail entirely due to the failure of such dependencies.

3.5.3 Security

Since the STB has to deal with highly confidential user data (including, but not limited to billing information), the latest security standards should be applied. In practice, this boils down to the adoption of secure protocols for transferring information over the Internet (e.g. HTTPS), actively checking the authorization status of a user when they attempt to access a resource or a specific endpoint, as well as proper encryption of passwords when stored into the system. Prevent an asshole to create 1 trillion accounts and occupy all of a city worth of CS, hence require credit card info on registration!

3.5.4 Maintainability

The system should be easily maintainable, to allow for future expansion of its feature set without requiring to build it from scratch. As such, standard software patterns should be adopted, and the whole architecture and underlying code should be properly and extensively documented, to allow potential future developers to start working on it without reading the entire code base.

3.5.5 Portability

As mentioned in previous sections, the system should be accessible both on a Desktop browser and on Mobile ones, therefore its interface should be fully responsive to adapt to different screen sizes. Futhermore, the system should be able to run on a wide variety of modern browsers, without feature restrictions for some of them.

4 Formal Analysis Using Alloy

Follows a formal analysis of the system discussed in this document using Alloy, a formal language used to specify software models. This aims to prove that the model's constraints are satisfiable by instances that allow for the goals of the system to be archived.

All classes modelled in the **class diagram** are present in this model as *signatures*, together with those attributes that were deemed better modelled by dedicated *signatures* rather than basic types. Note that the representation of time used within *Date* is the same as the Unix Time, and the *Now* signature is randomly chosen during the evaluation of the model to represent the current time.

The present *facts* model the impact of requirements on the model, while *predicates* allow for the **product functions** previously described.

```
//All the numeric types are Integers since the precision of Floats is not required.
//For prices, to have 2 decimals of precision, we consider the prices directly in cents.
//NOTE: Integers appear to be 4-bits, meaning that we can no longer reach 100, hence percentages cap at 10

sig Date {
        unixTime: one Int
} {
        unixTime >= 0
}
//Radomly chosen by Alloy
one sig Now extends Date {}

sig Location {}
sig Email {}
sig UserName {}
sig Password {}
sig CompanyName {}
sig PaymentMethod {}

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

sig User {
        userName: one UserName,
        email: one Email,
        password: one Password,
        paymentMethod: one PaymentMethod,
        bookings: set Booking
} {
        //No user can have two overlapping bookings
        all b1, b2: Booking | (b1 in bookings and b2 in bookings)
        implies
```

```
        (b1 = b2 or (b1.startDate.unixTime < b2.startDate.unixTime and b1.endDate.unixTime <= b2.startDate.unixTime) or
        (b2.startDate.unixTime < b1.startDate.unixTime and b2.endDate.unixTime <= b1.startDate.unixTime))
}

sig Booking {
        startDate: one Date,
        endDate: one Date,
        isActive: one Bool,
        socket: one Socket
} {
        //No negative duration bookings
        startDate.unixTime < endDate.unixTime
        //isActive only when actually within its duration and a vehicle is connected
        and (isActive = True implies startDate.unixTime <= Now.unixTime and Now.unixTime <= endDate.unixTime)
        and (isActive = True implies socket.connectedVehicle != none)
}

one sig eMSP {
        users: some User,
        knownCPMSs: some CPMS,
        bookings: some Booking
} {
        //No user outside of the eMSP
        all u : User | u in users
}

//The CS->CPO one-to-one relation can be inferred by going through here
sig CPMS {
        operatingManually: one Bool,
        CSs: some CS,
        knownDSOs: some DSO,
        owner: one CPO,
        policy: one Policy
}

sig Policy {
        weights: set Int,
        thresholds: set Int
} {
        //Weights and thresholds must be positive integers
        all w: Int | w in weights implies w >= 0
        and all t: Int | t in thresholds implies t >= 0
}

sig BatteryUsagePolicy {
        weights: set Int,
        thresholds: set Int
```

```
} {
        //Weights and thresholds must be positive integers
        all w: Int | w in weights implies w >= 0
        and all t: Int | t in thresholds implies t >= 0
}

sig CS {
        socketCount: one Int,
        location: one Location,
        nominalPrice: one Int,
        userPrice: one Int,
        offerExpirationDate: one Date,
        chargingFromBatteries: one Bool,
        rechargingBatteries: one Bool,
        sockets: some Socket,
        batteries: set Battery,
        currentDSO: one DSO,
        batteryUsagePolicy: one BatteryUsagePolicy
} {
        socketCount > 0
        and nominalPrice > 0
        and userPrice > 0
        //Lets avoid givin electricity away for free...
        and userPrice <= nominalPrice
        and nominalPrice >= currentDSO.price
        //Force the system to never charge the batteries and discharge them at the same time
        and (not ((chargingFromBatteries = True) and (rechargingBatteries = True)))
}

sig Connector {}

sig Socket {
        connector: one Connector,
        maxPower: one Int,
        currentPower: one Int,
        connectedVehicle: lone Vehicle
} {
        maxPower > 0
        and currentPower >= 0
        and currentPower <= maxPower
}

sig Vehicle {
        chargePerc: one Int
} {
        //As a percentage it must be between 1 and 100 (-> 10 for integer limitations in alloy)
        chargePerc >= 0 and chargePerc <= 10
}
```

```
sig CPO {
	name: one CompanyName
}

sig DSO {
	name: one CompanyName,
	price: one Int,
	energyMix: one EnergyMix
} {
	price > 0
}

sig EnergyMix {
	oilAndGas: Int,
	ccgt: Int,
	coal: Int,
	nuclear: Int,
	hydroelectric: Int,
	otherRenewableSources: Int
} {
	//The sum will need to reach 100 in reality, but here we are working with
	↪ 4-bit integers
	oilAndGas + ccgt + coal + nuclear + hydroelectric + otherRenewableSources =
	↪ 10
	and oilAndGas >= 0
	and ccgt >= 0
	and coal >= 0
	and nuclear >= 0
	and hydroelectric >= 0
	and otherRenewableSources >= 0
}

sig Battery {
	capacitymAp: one Int,
	chargeLevel: one Int
} {
	capacitymAp > 0 and
	chargeLevel >= 0 and chargeLevel <= 10
}

//No two users with the same email or userName
//Fine for payment methods being used for more than one User
fact noUsersWithSameEmail {
	no u1, u2: User | u1 != u2 and (u1.email = u2.email or u1.userName =
	↪ u2.userName)
}

//No two CPOs can share the same name
fact noCPOsWithSameName {
```

```
        no c1, c2: CPO | c1 != c2 and c1.name = c2.name
}

//Company names 1:1 with companies
fact companyNamesToCompanies {
        all n : CompanyName | ((one c : CPO | c.name = n) or (one d : DSO | d.name = n)) and !((one c : CPO | c.name = n) and (one d : DSO | d.name = n))
}

//No unused Passwords
fact noUnusedPasswords {
        all p : Password | (some u : User | u.password = p)
}

//No unused Payment Methods
fact noUnusedPaymentMethods {
        all p : PaymentMethod | (some u : User | u.paymentMethod = p)
}

//No two DSOs can share the same name
fact noDSOsWIthSameName {
        no d1, d2: DSO | d1 != d2 and d1.name = d2.name
}

//No CPO and DSO can share the same name
fact noCPODSOWithSameName {
        no c: CPO, d: DSO | c.name = d.name
}

//Each CPO must be 1:1 with its CPMS
fact CPOOneToOneCPMS {
        no cp1, cp2: CPMS | cp1 != cp2 and cp1.owner = cp2.owner
}

//No booking can be without a User associated to it
fact noBookingsWithoutUsers {
        all b: Booking | (one u: User | b in u.bookings)
}

//No expired booking must be in the system
fact noExpiredBookings {
        all b: Booking | b.endDate.unixTime >= Now.unixTime
}

//No two bookings with overlapping time frames must be for the same socket
fact noOverlappingSocketBookings {
        all b1, b2: Booking | (b1 = b2 or b1.socket != b2.socket or (b1.startDate.unixTime < b2.startDate.unixTime and b1.endDate.unixTime <= b2.startDate.unixTime)
```

```
        or (b2.startDate.unixTime < b1.startDate.unixTime and b2.endDate.unixTime <= b1.startDate.unixTime))
}

//Every socket must be free if it is not booked
fact socketFreeIfNotBooked {
        all s: Socket | ((no b: Booking | b.socket = s)
        implies
        no s.connectedVehicle)
}

//No socket should be free if there is an active booking on it
fact socketNotFreeIfActiveBooking {
        all s: Socket | ((some b: Booking | b.socket = s and b.isActive = True)
        implies
        s.connectedVehicle != none)
}

//All CSs must be associated to one and only one CPMS
fact allCSHaveACPMS {
        all cs: CS | (one cpm: CPMS | cs in cpm.CSs)
}

//Each Socket must be related to one and only one CS
fact uniqueSocketsForCS {
        no cs1, cs2: CS | cs1 != cs2 and (some s: Socket | s in cs1.sockets and s in cs2.sockets)
}

//No CS of an unknown CPMS should be booked by an eMSP User
fact noBookingsOverUnknownCPMSs {
        no b : Booking | (some cpms : CPMS | b.socket in cpms.CSs.sockets and (no e : eMSP | cpms in e.knownCPMSs))
}

//Add User booking
pred addUserBooking[u0, u1: User, b: Booking] {
        u1.bookings = u0.bookings + b
}

//Booking deletion
pred delUserBooking[u0, u1: User, b: Booking] {
        u1.bookings = u0.bookings - b
}

//Start charging process
pred startUserCharge[b0, b1: Booking, v: Vehicle] {
        b1.socket.currentPower = b0.socket.maxPower
        b1.socket.connectedVehicle = v
        b1.isActive = True
```

```
}

//End charge
pred endUserCharge[b0, b1: Booking, v: Vehicle] {
        b1.socket.currentPower = 0
        b1.socket.connectedVehicle = none
        b1.isActive = False
}

//Assign energy source and policy
pred changeEnergySource[d1: DSO, es: EnergyMix] {
        d1.energyMix = es
}

//Change CPMS policy
pred changeCPMSPolicy[c: CPMS, p: Policy] {
        c.policy = p
}

//Change nominal and user price
pred changePrices[c0, c1: CS, np: Int, up: Int] {
        c1.nominalPrice = np
        c1.userPrice = up
}

//Check that we dont't have overlapping bookings on the same Socket of the same CS
assert noOverlapForSocket {
        no disj b1, b2: Booking | (b1.socket = b2.socket and (b1.startDate.unixTime >= b2.startDate.unixTime and b1.startDate.unixTime < b2.startDate.unixTime))
}

check noOverlapForSocket
```

4.1 Results

```
run {} for 12 but 6 Int, exactly 3 User, exactly 3 Password, exactly 3 PaymentMethod,
↪ exactly 3 CS, exactly 2 CPMS, exactly 3 Booking
```

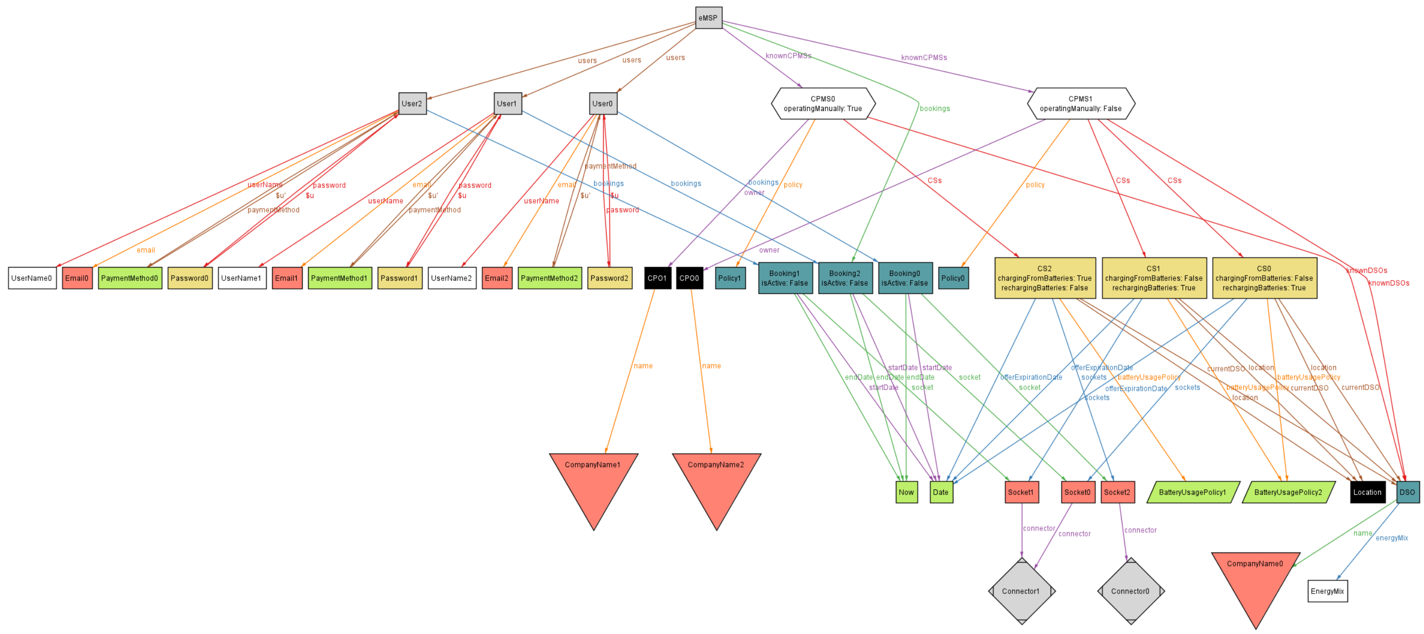

Figure 32: First run

```
run {} for 12 but 6 Int, exactly 2 User, exactly 1 CS, exactly 1 CPMS, exactly 3
↪ Booking, exactly 1 DSO
```

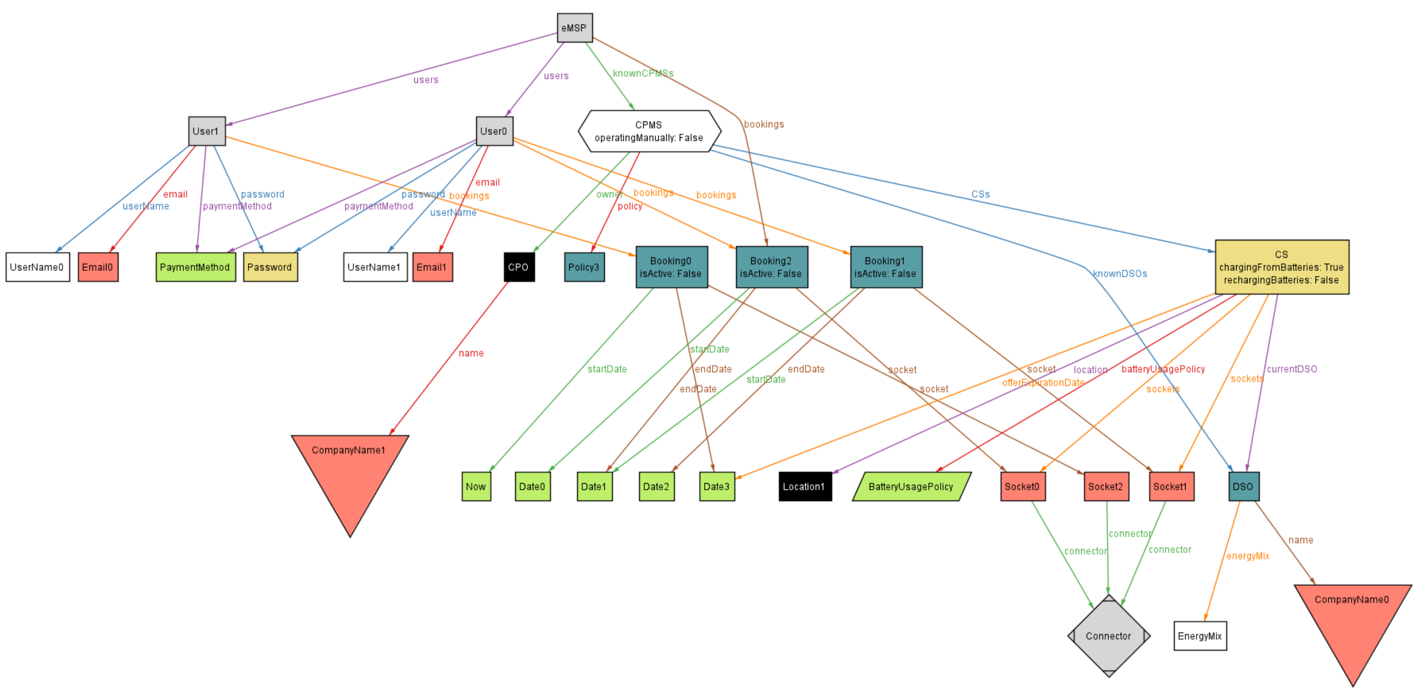

Figure 33: Second run

5 Effort Spent

5.1 Ronzani Marco - mat: 224578

Task	**Time spent**
Introduction	17 h
Overall Description	20 h
Specific Requirements	15 h
Formal Analysis	8 h
Total	60h

5.2 Sassi Alessandro - mat: 220837

Task	**Time spent**
Introduction	11 h
Overall Description	16 h
Specific Requirements	15 h
Formal Analysis	8 h
UI Mockups	6 h
Total	56 h

6 References

1. ChargeLab - operating system for EV charges
2. Platform for Electromobility. EV Charging: How to tap in the grid smartly?.
3. F. Campos, L. Marques, and K. Kotsalos, Electric Vehicle CPMS and Secondary Substation Management.
4. Shu Su, Hui Yan, and Ning Ding. 2018. Machine Learning-Based Charging Network Operation Service Platform Reservation Charging Service System. In Proceedings of the 2018 International Conference on Signal Processing and Machine Learning (SPML '18). Association for Computing Machinery, New York, NY, USA, 1–5.
5. Jan Mrkos, Antonín Komenda, and Michal Jakob. 2018. Revenue Maximization for Electric Vehicle Charging Service Providers Using Sequential Dynamic Pricing. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 832–840.