Software Engineering 2 - Prof. Di Nitto Elisabetta
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

eMall - e-Mobility for All

RASD
Requirement Analysis and Specification Document

December 23, 2022

Giovanni De Lucia (10700658)
Lorenzo Battiston (10618906)
Matteo Currò (10940719)

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose

With the higher focus on the impact of our urban and suburban travel on the environment and the higher accessibility of electric mobility, an increase of circulating electric vehicles can be observed. [1] [2] [3] This increase concerns both private vehicles and goods transporting ones. As a result of restrictions on fuel vehicle production and sell that will concern a large part of the world's population[4], the number of electric vehicle is still set to increase. For these reasons, the main vehicle manufacturers have started making huge investments in electric mobility[5], which will lead to greater accessibility to the market by drivers.

A main problem of electric vehicles is that a full charge requires much more time than a fuel vehicle refuel. [6] Thus, a single charge can have a huge impact on our daily schedule, and it is necessary to plan wisely when and where to charge. Furthermore, some electric vehicle owners don't have the proper equipment to recharge at home, or their vehicle discharges in the middle of the road and the driver doesn't have the possibility to go home to recharge.
To solve these problems is one of the main objective of the eMall - e-Mobility for All system.
This system aims to develop an efficient planning of the charging process of electric vehicles that limits the carbon footprint caused by people mobility needs.

The following document is the RASD for the eMall - e-Mobility for All system. It provides a description of the system focusing on the requirements and specifications, developing scenarios and use cases to specify what the system must do, how it will interact with the stakeholders and the constraints it is subject to.

1.1.1 Goal

G1	Allow EV - Electric Vehicle driver to plan efficiently their charging process
G2	Allow EV - Electric Vehicle driver to have a single application for all the processes involving the charge with a personalized experience based on the car and the user commitments
G3	Allow CPOs - Charging Point Operators to be reached by EV drivers looking for charging points
G4	Provide smart managing of charging stations, including the register of charging reservations
G5	Allow CPOs - Charging Point Operators to choose between contracts of energy providers and to determine the energy source mix

1.2 Scope

The main actors in this system are the drivers and the CPOs - Charging Point Operators, who manage their charging columns, along with the DSOs - Distribution System Operator, in charge of distributing the energy. The digital system eMall should provide three main features:

- **Booking** allows EV owners to book a charge. The remote booking avoids interference in the daily schedule of the owners, and it includes a notification system that alerts owners when their reservation is going to start.

[1] https://www.eea.europa.eu/ims/new-registrations-of-electric-vehicles
[2] https://www.statista.com/statistics/1101415/number-of-electric-vehicles-by-type/
[3] European Investment Bank Climate Survey
[4] Places with planned fossil-fuel vehicle restrictions
[5] EV plans from major manufacturers
[6] Why consumers don't buy electric vehicles

- **Charging** allows EV owners to charge an EV, remotely monitor their charging process and be notified at the end of the charge. Thanks to these features, owners have not anymore the need to physically go to the CP when they want to retrieve details of their charge.
- **Managing an EVCP** allows CPOs to get statistics on live and historical details about their EVCP - Electric Vehicle Charging Pool, to acquire information about the current energy price by DSOs and to decide in an automated way where to get energy for charging.

1.2.1 World phenomena

WP1	An EV driver arrives at a charging station
WP2	An EV driver wants to charge the car
WP3	An EV driver wants to plan a charge
WP4	A charging station is connected to the electrical grid
WP5	Some charging station has solar panels
WP6	Some charging station has a storage battery
WP7	An EV battery discharges

1.2.2 Shared phenomena

ID		Controlled by
SP1	An EV driver books a charge at a certain charging station	world
SP2	An EV driver search for a specific charging station	world
SP3	The system suggest to charge based on daily schedule, special offers and availability	machine
SP4	An EV driver starts the charging process	world
SP5	An EV driver receives a notification when the charging process is completed	machine
SP6	An EV driver pays for the charge	world
SP7	The system shows to CPO the status of its charging station as amount of energy in batteries, number of vehicle being charged and for each the time left of the charge	machine
SP8	The system shows to CPO information about the DSOs	machine
SP9	A CPO decide to acquire energy from a certain DSO	world
SP10	The system notifies an EV driver that the charging shift will begin shortly	machine
SP11	An EV driver monitors the charging status	machine
SP12	An EV driver deletes a reservation	world
SP13	A CPO decide to retrieve the historical reservations on its CPs	world
SP14	An EV driver retrieves the historical reservations	world

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- EV Driver - Electric Vehicle Driver, people or entities who own an EV car and want to use the system for their charging needs
- EVCP - Electric Vehicle Charging Pool, is a station with multiple CPs
- CP - a synonym of EVSE - is a single charging column with multiple connectors
- Connectors - charging sockets which can be of different types (e.g. CCS2, Type2)
- OCPP - Open Charge Point Protocol [7] - is a protocol that dictates the communication between CPMS and a controlled CP to achieve smart charging functionalities
- OCPI - Open Charge Point Interface [8] - is a protocol that dictates the communication between

[7] OCPP Protocol
[8] OCPI Protocol

CPMSs and eMsps to let the CPMSs to be accessible by multiple eMsps achieving roaming functionalities

- Rate -

1.3.2 Acronyms

eMall	e-Mobility for All
RASD	Requirement Analysis and Specification Document
eMSP	Electric Mobility Service Provider
EV	Electric Vehicle
CPO	Charging Point Operator
DSO	Distribution System Operator
CPMS	Charging Point Management System
EVSE	Electric Vehicle Supply Equipment
CP	Charging Point
EVCP	Electric Vehicle Charging Pool
GPS	Global Positioning System
API	Application Programming Interface
OCPP	Open Charge Point Protocol
OCPI	Open Charge Point Interface
OS	Operative System
VAT	Value-Added Tax (number)
IBAN	International Bank Account Number
HTTPS	HyperText Transfer Protocol Secure
TLS	Transport Layer Security

1.3.3 Abbreviation

$\mathbf{WP}_x$	x-World Phenomena
$\mathbf{SP}_x$	x-Shared Phenomena
$\mathbf{G}_x$	x-Goal
$\mathbf{D}_x$	x-Domain Assumption
$\mathbf{Dep}_x$	x-Dependency
$\mathbf{R}_x$	x-Functional Requirement
$\mathbf{U}_x$	x-Use Case
$\mathbf{NFR}_x$	x-Non Functional Requirement

1.4 Revision history

1.5 Reference Documents

Assignment document A.Y. 2022/2023 ("Requirement Engineering and Design Project: goal, schedule and rules")

1.6 Document Structure

This document is composed of six sections:

1. we introduce the problem and the goals of the system to be. In the scope subsection we provide a description of the various world and shared phenomena occurring. Lastly, we provide useful information to read this document properly, such as definitions and abbreviations
2. we provide an overall description of the system, along with a description of the users and their main functionalities. Moreover domain diagrams are presented and several scenarios are described, lastly we provide the domain assumption of the system to be
3. we specify the requirements of the system to be. This includes functional and non-functional requirements. Furthermore use cases diagrams are presented, along with a description of each use cases and a related sequence diagram. Lastly, we provide a mapping of the requirements on both goals and use cases
4. we provide a formal analysis of the system to be with Alloy
5. we provide an estimate of the effort spent by each group member
6. we list the used references

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

A. Registration
Einar is a driver of an electric vehicle that uses every day to go to his office. He decided to download the eMall app because he heard by a friend of him that he can discover all the charging points in the entire world, booking one, starting a charge and pay for the charge, entirely through the app. After having downloaded it, launches the app for the first time and select sign in button to register into the system. He provides all the personal data required to access in the system and accept to personalize his experience by selecting his car from a provided list of all the EVs. He submits his data and the system asks him to verify his account through email or phone number.

B. Book a charge
Edvar has the necessity to go shopping in the next days at the blue and yellow furniture retailer of the city. Knowing that shopping will take some time he wants to find and book a EVCP nearby the shopping center to charge his EV that uses every day. In the home of the app he filters the results on the location of the mall and the day he wants to go. After submitting the form, the home page change according to his information and displays a map of the selected zone with the available charging stations to book. He discovers that there is one available really close from the shopping center. He selects the marker of the EVCP and are displayed the information about the CPO, the types of CP, the availability at the actual moment of the research, the availability of them for the filtered date, the charging power at which the connector operates and the cost for recharging 1 kWh. To book the charge he selects one connector that is available and indicate when he wants to start the charge and when to finish. He then confirms the booking and pays the charge, then see the reservation on the reservations' tab.

C. Charging process
Anne plans a long trip from Oslo to Stockholm to do with her brand-new EV. Considering the suggested time by the app to do a full charge on her EV, she books a three-hour charge through the eMall app at her trusted charging station with high power connectors. When she arrives at the CP station she parks in a free slot with the booked connector. Through the app she selects the reservation in the reservations' tab and starts the charge inside the app. The connector socket is unlocked, and the charge can start by plugging the connector on the EV. Meanwhile, waiting for the complete charge, Anne goes for a walk because she feels relaxed to control at any time the status and the remaining time of the charge with the app. When the charge is completed, as expected before the three hours, Anne is already back from the walk and a notification about the end of the charge appears on Anne's phone, she disconnects the connector and gets back home to prepare the luggage for the trip. She doesn't worry about the payment because it's executed in background with the payment method that she indicates during the booking operation.

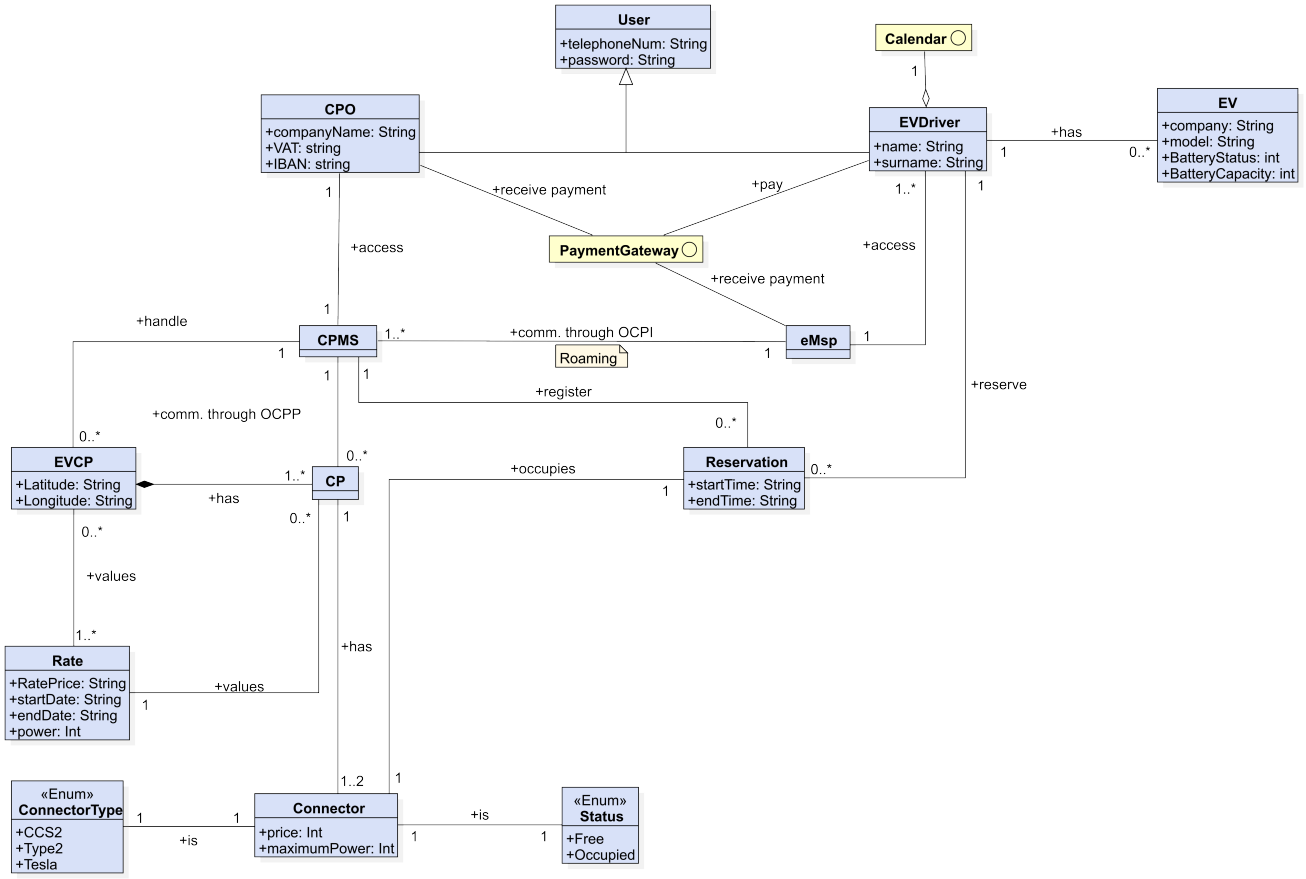
D. Charging point status
Erling is the owner of a restaurant and installed two charging columns in the parking slot in front of the restaurant because he wants to acquire good clients that can stop for charging the EV and have a meal at the restaurant. To make the CP accessible to the largest possible public he subscribes to the eMall-business for CPOs because he is interested in a service that permits to add and manage CPs and make the CPs visible by EV drivers in the eMall app. After submitting the registration by providing essential information about the company, including VAT number of the restaurant and IBAN bank account to get payments from the driver he waits for the approval to be inserted into the app. When the approval arrives Erling inserts the charging point of the restaurant by specifying the number of sockets by type, the amount of power supplied by each socket and the API to connect the charging columns to the dashboard. With the dashboard he can visualize how many vehicles are charging in real time and for each charging vehicle the amount of power absorbed, and the time left to the end of the charge. He can visualize the import that gets from each charge, decide the price for a charge and add special promotions to the charge to win the loyalty of the existing clients or acquire new clients.

E. Charging point management
Grethe, admin of a CPO using the CPMS function of eMall from the early days, receives a mail from the CPMS about a malfunction involving one of her CPs. She enters the CPMS and directly

from the dashboard she modifies the CP status to "in maintenance". She is sure that the CP under maintenance now results as unavailable for every user trying to book it until she modifies back the status to "available". Then verifies the status of the others CP controlling that they work as intended.

2.1.2 Class diagram

The UML class diagram below represents a conceptual, high-level model of the software to be. Given its nature, it may model entities that will not be represented in the actual system that will be developed. At this level, it not include any references to methods and other low-level details, those will be detailed during the design phase.



The main entity in the diagram are:

- **User**
  A user can be a CPO or an EV Driver and their experience strictly depends on their role. An EV Driver has access to the eMsp entry point, while the CPOs access their CPMS. The EV Driver can pay for a charge through a Payment Gateway and the associated CPO receives the payment. Furthermore, EV Drivers can add their EVs to the system to receive a personalized experience and give the system access to their Calendar to receive suggestion based on their daily schedule
- **CPMS**
  The CPMS is a subsystem of eMall that allows CPO to smartly handle their EVCPs. It provides automatically management of CPs but lets also the possibility to manually set options (the energy source mix, the DSO to acquire energy from ecc.). CPMSs communicate with eMsp following the OCPI protocol, this lets them being accessible from external eMsp too
- **eMsp**
  The eMsp is a subsystem of eMall, the EV Drivers entry point. It allows them to search for CPs,

to book a connector for a given timeframe, pay and start a charge along with other functionalities described in the next chapters. The eMsp communicate with CPMSs following the OCPI protocol, this lets the EV drivers accessing CPs owned by CPO that aren't subscribed to our system and then leads to a larger set of possibilities

- **EVCP, CP, Connector**
  EVCP is a station with multiple CPs. Each CP has usually 1 or 2 connectors that has a status and type. A CP communicates with CPMS following the OCPP protocol

2.1.3 State diagrams

In the paragraphs below, a representation of the behavior of the main conceptual components of the system. The focus is on how these components respond to external influences and modify accordingly their states. For this purpose, UML State Diagrams are proposed.
The following state diagram represents the possible status of a charging point. When is added to the system it results "available" until the system requests to pass in "reserved" mode. When it is in "reserved" status, only the user with the matching reservation can start a charge through the eMall APP. When the user starts a charge the system communicates the charging point to change status in "charging". In this status the charging point performs the charge as expected and when the charge is ended, the system notify to modify the status in "available" back again. It is possible that for some sort of malfunction or manual intervention by the operators through the system forces the charging point to change the status in "out of order", and this can happen from all the state. The charging point will return "available" only when the operators decide to.

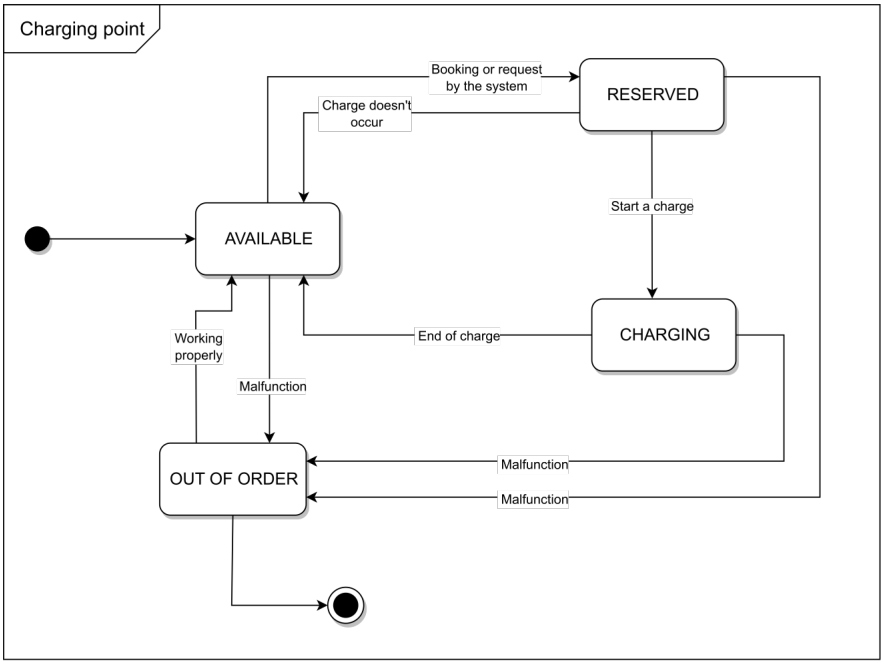


Figure 1: Charging Point state chart

The following diagram describes the possible status for a reservation. When a reservation is created through the APP it is in "not started" state. In the exact moment in which the time of the reservation starts then it passes in the "pending" status. Will remain in pending status until the charge is started or until the time for the reservation runs out. When the charge is started it is in "running" mode and when finishes is "completed" and archived in the list of past transaction.

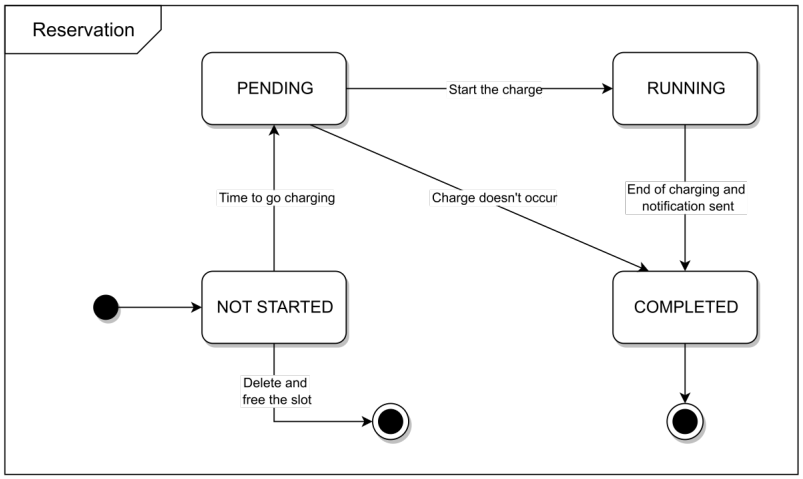


Figure 2: Reservation state chart

2.2 Product functions

The major functions offered by our system are organized according to the stakeholder that is being addressed

- **EV driver's functions**
  The person that owns the EV to be smart charged. The function provided to the EV drivers aim to reduce the impact of charging processes on their daily schedule
  - **Search in Map**
    Users can retrieve charging points based on some filtering, such as location, connectors available and booking date
  - **Receive smart charging suggestion**
    Users receive suggestions on when and where to charge based on their location, the battery status of their EV, their daily schedule and special offer by CPOs
  - **Book a Charge**
    Users can book a charge in a given charging point for a specified time frame and a chosen date. They can also pay through the system for the service
- **CPO's functions**
  The operator, person or entity, that manage the charging point at which the driver recharge their EV
  - **Monitor Status of CPs**
    CPOs can retrieve the status of their CPs. The status contains the details about current charging processes, aggregate or detailed views of energy consumption, profit by each CP, and any problem reported by customers
  - **Manage DSOs contract**
    CPOs can retrieve a list of available DSOs in their operating area, along with their energy prices. They can also create contract with DSOs and start getting energy from them
  - **Manually change status of a CP**
    CPOs can manually reserve CPs or decide to set up a maintenance task, e.g. following a malfunction
  - **View reservations on their CPs**
    CPOs can see a list of historical and live reservations on their CPs. The system provides them

an aggregate or detailed view of reservations

- **Managing CP**
  CPOs can add, delete and modify CP. The system gives the possibility to make special offer, change prices and choose an energy sources mix

2.3 User characteristics

It is possible to distinguish two different types of actors who use the system:

- **EV driver**
  People who want to book a charge remotely avoiding interference in their daily schedule, be notified when their reservations is going to start and end, monitor charging processes. Since there are not many statistical analyses on tech-friendliness among EV drivers, the system must aim to be usable by as wide an audience as possible providing easy to use interfaces
- **CPO**
  People or entities who want to manage efficiently their EVCP, make statistics on live and historical details on the EVCP, to acquire information on the current price of energy offer by DSOs and to decide in an automated way where to get energy for charging. Operators are expert people who know details not known to drivers, know how to efficiently use a dashboard to modify and obtain the information they need

2.4 Assumptions, dependencies and constraints

2.4.1 Domain Assumptions

D1	An EV driver arrives at the charging station at a time close to its reservation starting time
D2	An EV driver leaves the charging station when the charge is finished
D3	An EV driver doesn't occupy an already booked charging spot
D4	An EV driver provides correct information when registering
D5	At least one DSO can always provide energy to the CPOs
D6	An user that books a charge has an electric vehicle to charge
D7	An user that books a charge is always reliable

2.4.2 Dependencies

Dep1	The system requires access to a third party maps API
Dep2	The system will use the GPS of the driver's computer or smartphone
Dep3	The system will require internet connection to interact with all the users
Dep4	The system will use an external API to retrieve the prices of energy by the available DSOs
Dep5	The system will use an external API to retrieve the EV battery status and a list of EV available on the market
Dep6	The system will use an external API to retrieve data or send data to the CPs
Dep7	The system will use an external API to access the calendar of the users
Dep8	The system will use a payment gateway to perform payment operations
Dep9	The system will use an external API to send push notification to the users
Dep10	The system will use a third party API to send SMS to customers phones

2.4.3 Constraints

- The system shall be compliant to local laws and regulations, in particular users data should be treated according to the GDPR. This means that users should be always able to request their data
- The system should collect only necessary data, such as the user telephone number
- To better protect the users' sensitive information, such as their telephone number and credentials, their data should be encrypted
- The external APIs, especially those critical for the correct functioning of the system, must be chosen among those with the highest availability and reliability

3 Specific requirements

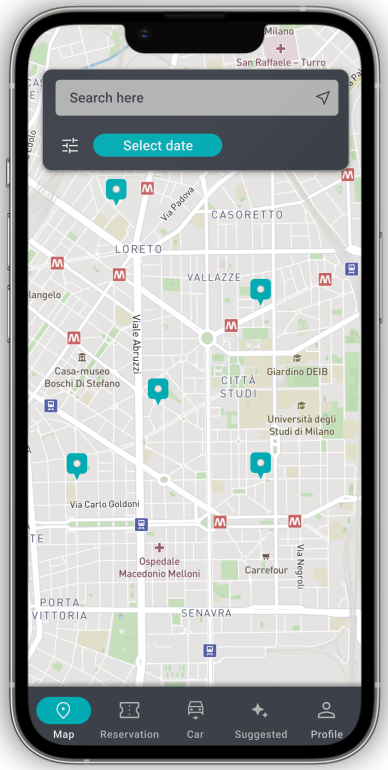
3.1 External Interface Requirements

In this section details about user interfaces, hardware and application programming interfaces are described.

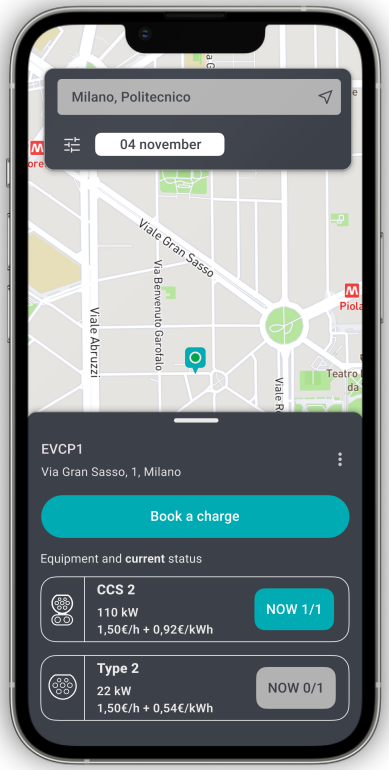
3.1.1 User Interfaces

In this subsection we present user interfaces for both types of users, drivers and CPOs

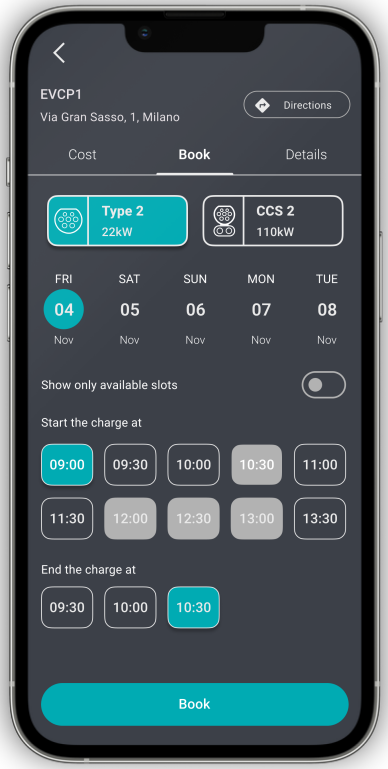
(a) Home



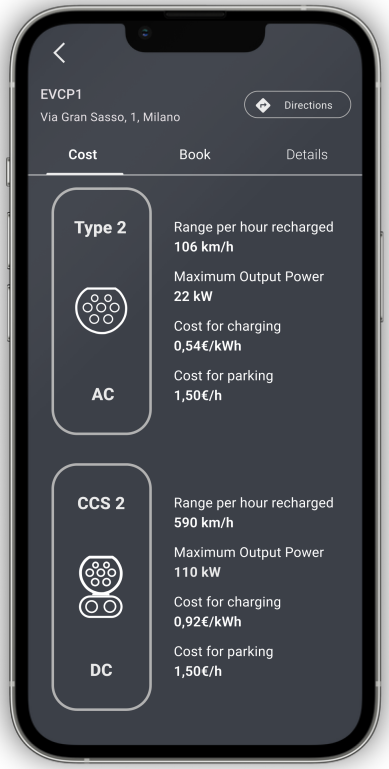
(b) Selected EVCP: drawer



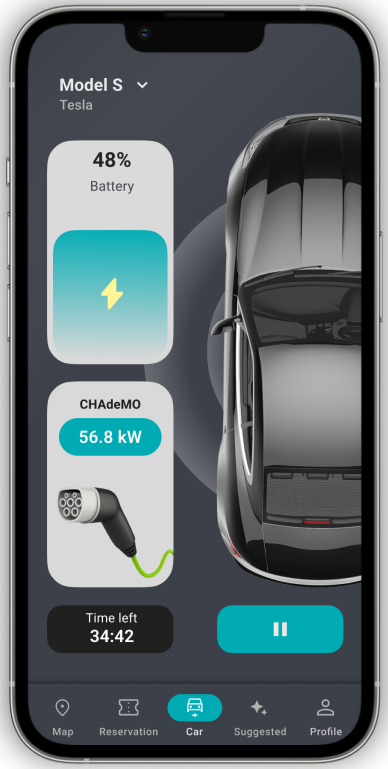
(c) Book charge tab



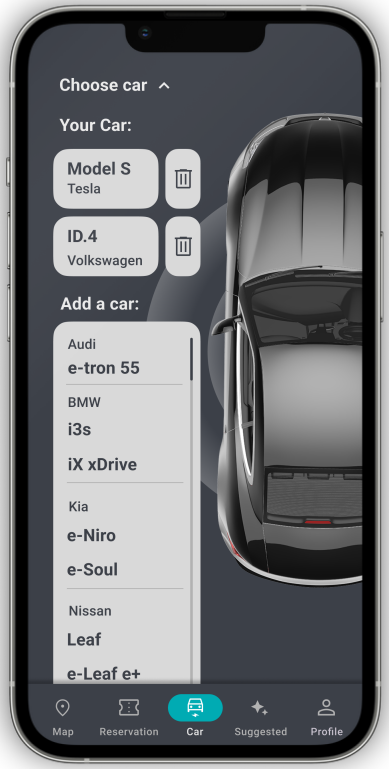
(d) Book cost tab



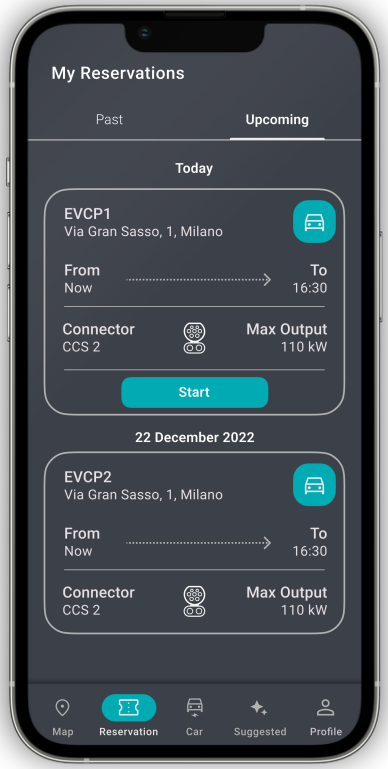
(e) Car status



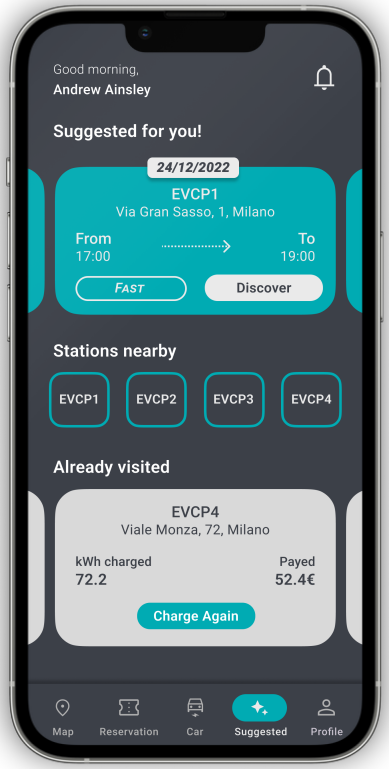
(f) Add a car



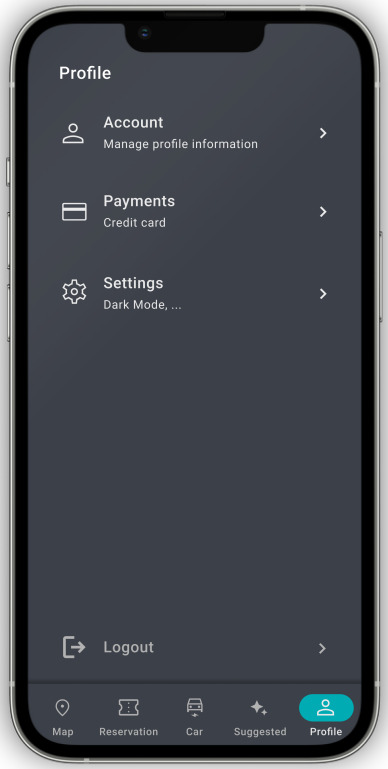
(g) Reservations



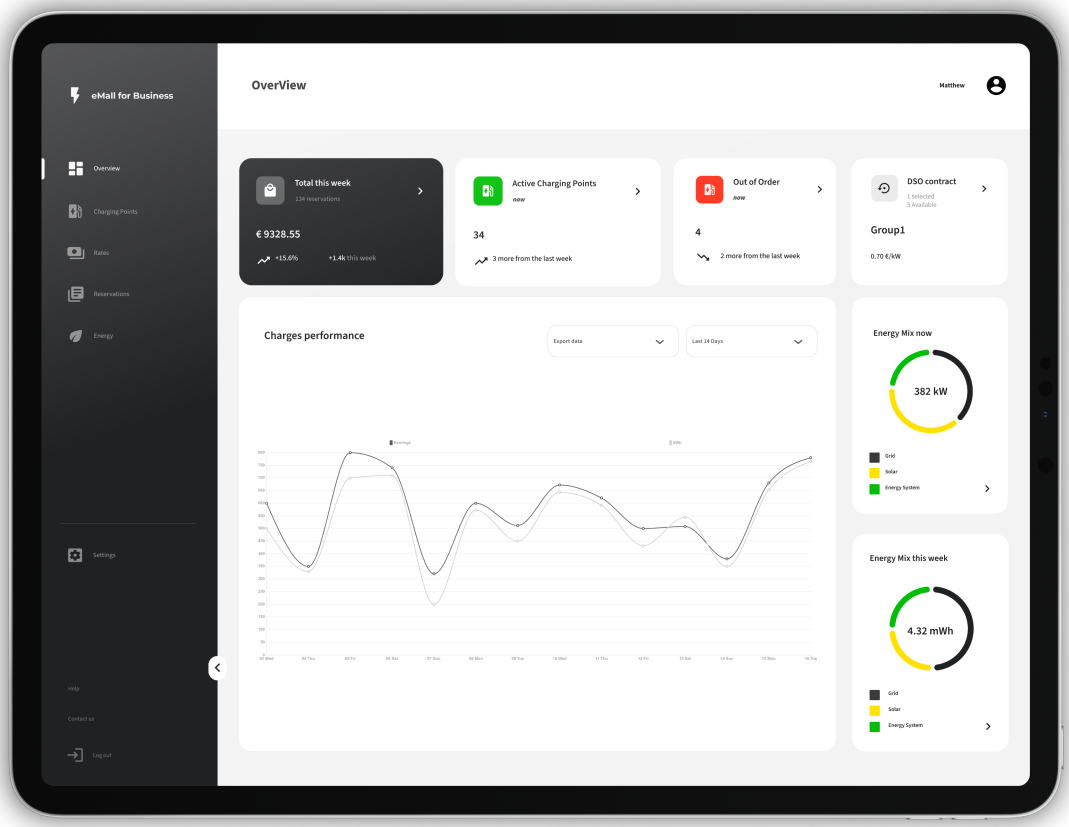
(h) Suggestions



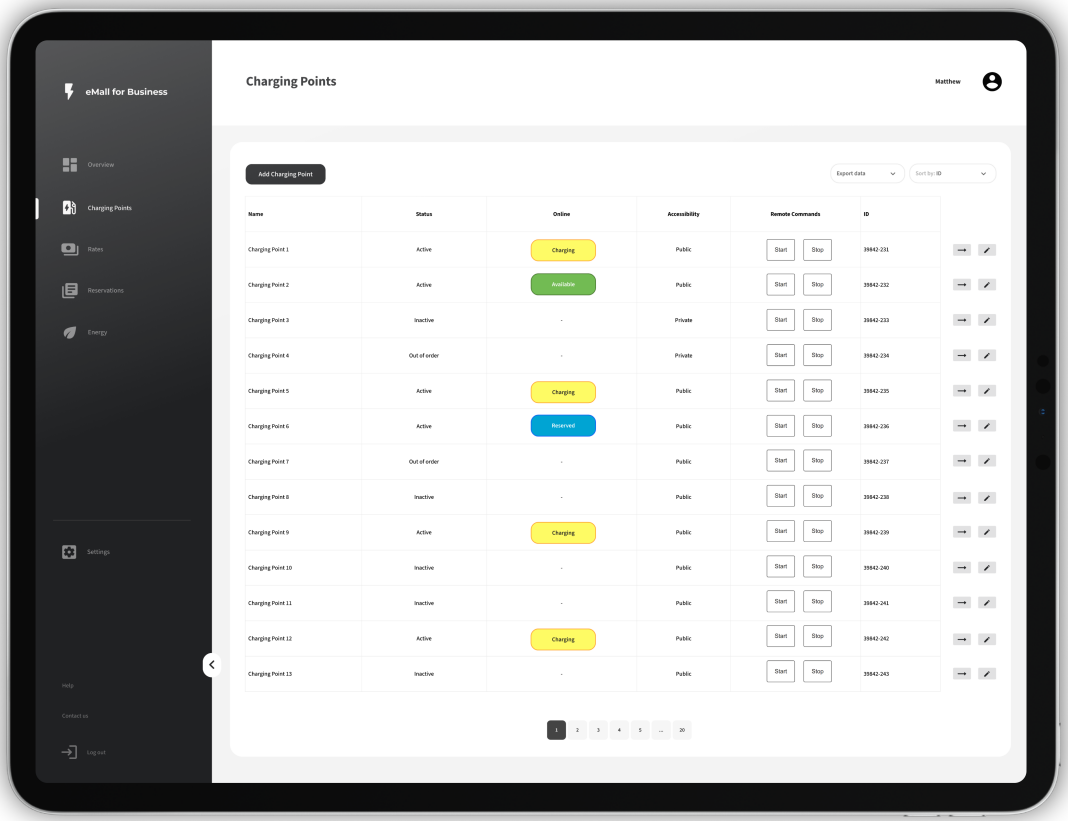
(i) Profile



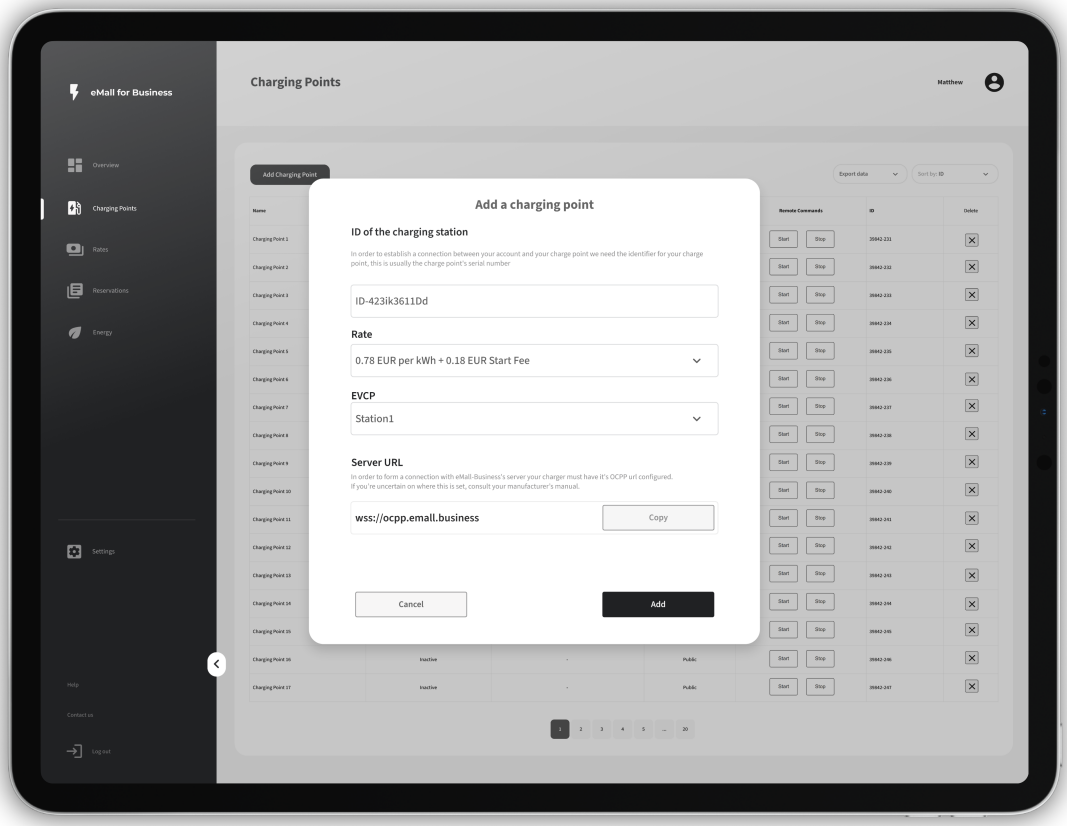
(j) Dashboard



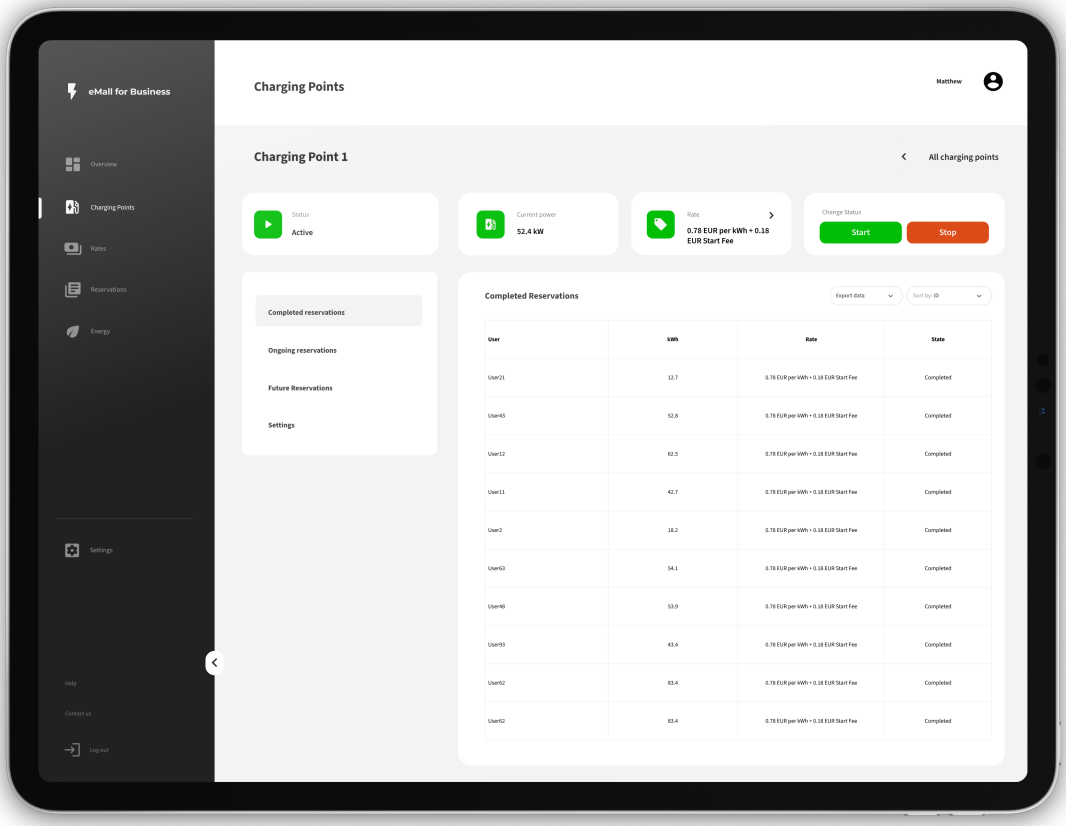
(k) Charging Points



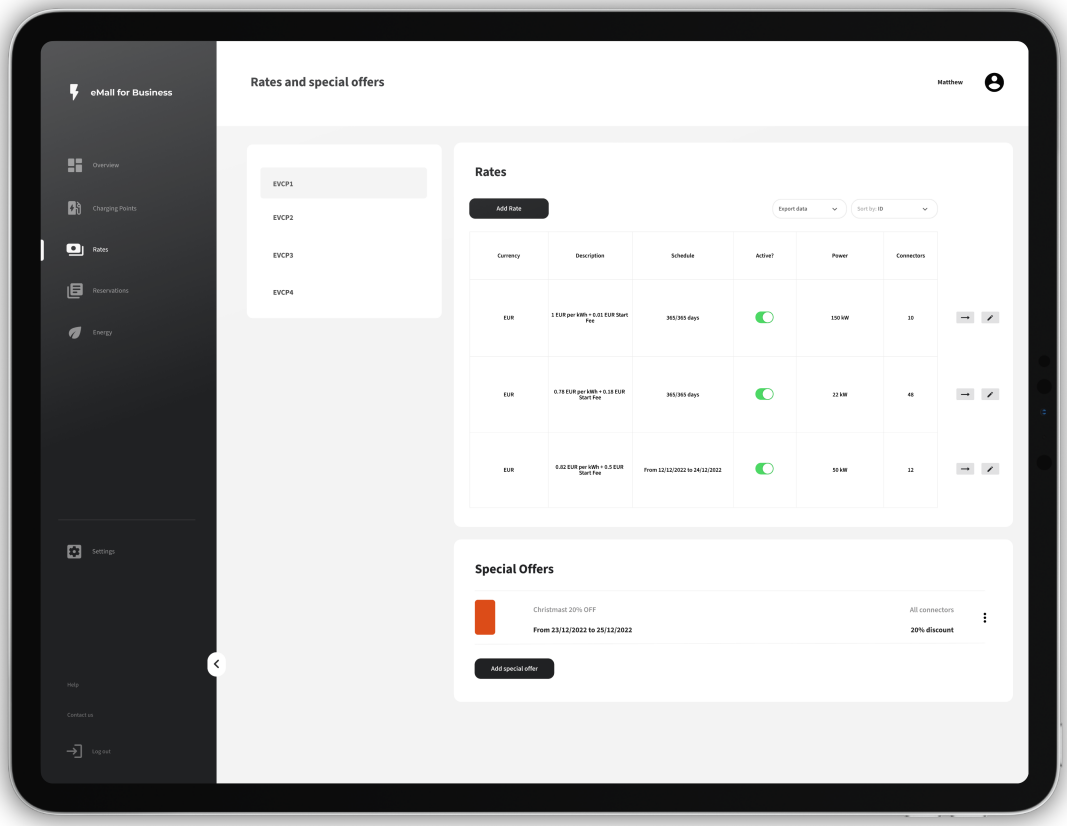
(l) Add a Charging Point



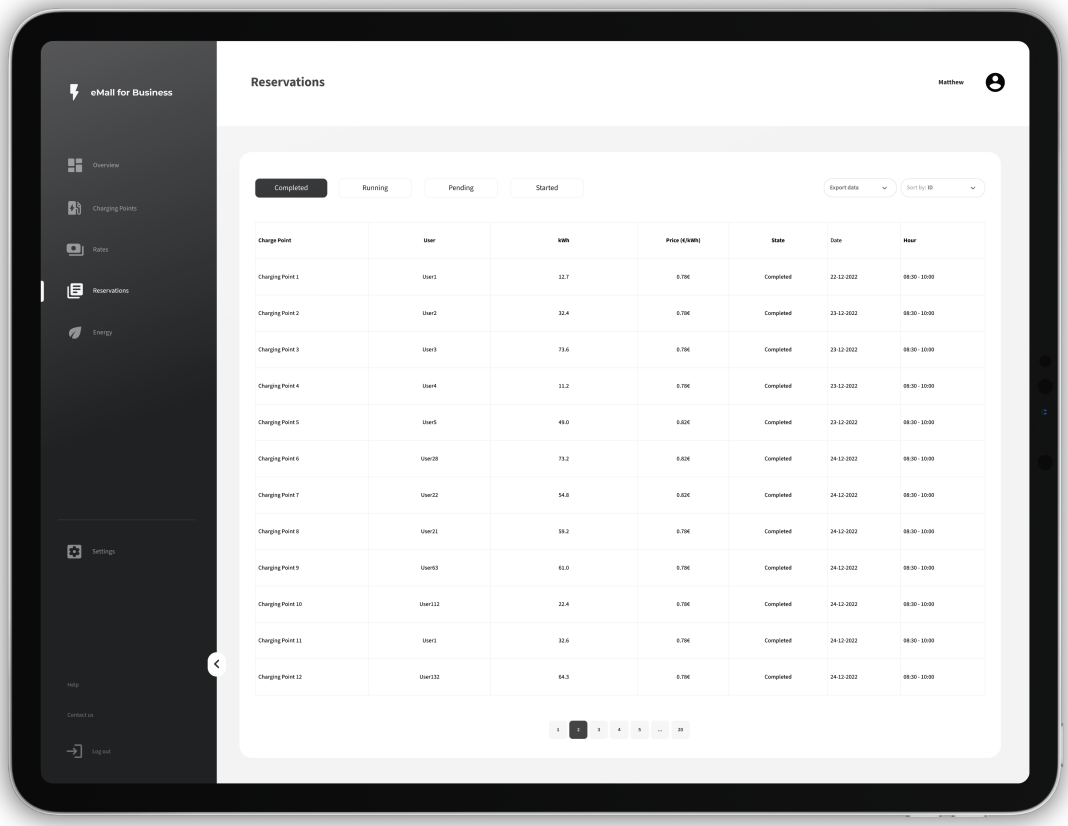
(m) Charging Point



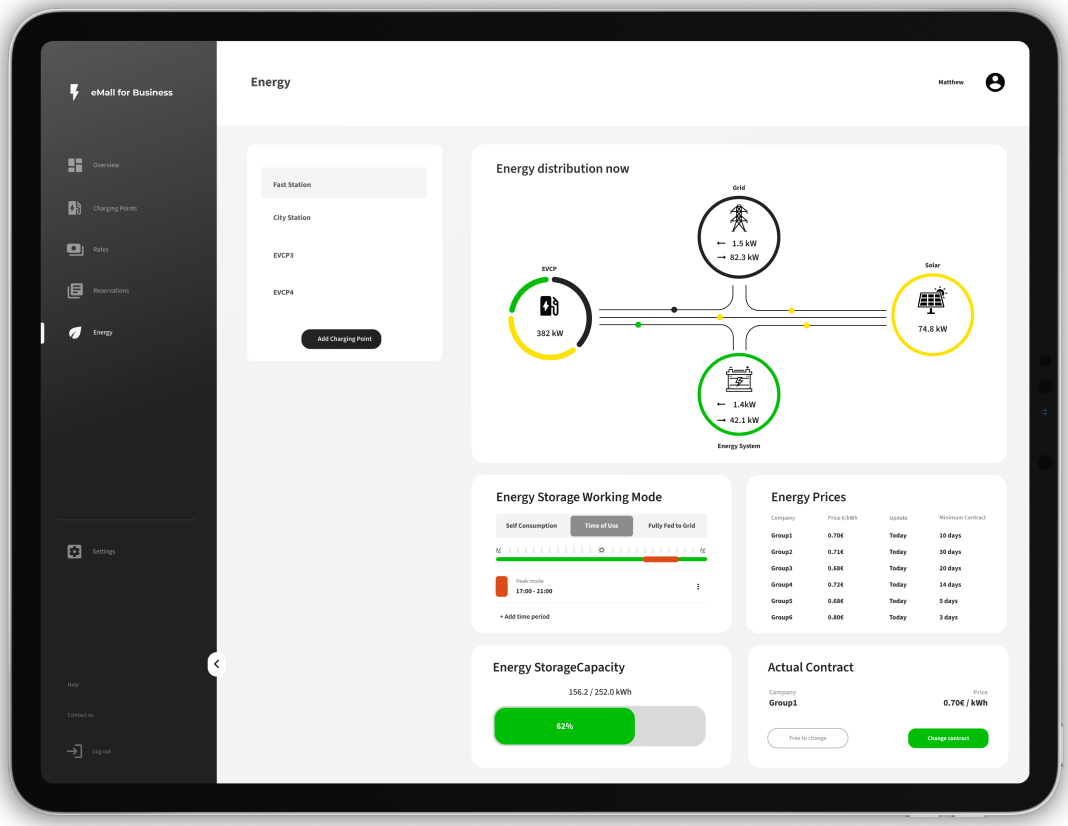
(n) Rates



(o) Reservations list



(p) Energy management



3.1.2 Hardware Interfaces

To use the system, both EV drivers and CPOs must have a mobile device or a personal computer. Due to the outdoor expected use of the system, a smartphone will be a more suitable device for an EV driver, instead for a CPO is suggested to use the system from a personal computer. The system is based on the ability of gathering data from different sources, such as the CP, the battery storage that may be present in an EVCP and the battery status of an EV. These data are provided through hardware interfaces that are external to our system, and we assume that an API let the system access to these data

3.1.3 Software Interfaces

eMall is a web application, then a modern web browsers is the only software required

3.1.4 Communication Interfaces

The system requires a stable internet connection to work properly. The backend of the system will expose a unified RESTful API to communicate with all clients using HTTPS and TCP/IP.
Furthermore, the system relies on various external interfaces accessible via uniform web API. These services are:

- **Map Service**: eMall system relies on a map service to get the map and to do operations on the map which is the central component of the application.
- **GPS Service**: eMall system relies on GPS to obtain the location of the user and to show their position on the map
- **DSO Energy Pricing Service**: eMall system use this interface to obtain updated prices of the different DSOs available for the location of the EVCPs. The interface also, permits to select from which DSO acquire energy.
- **EV Service**: eMall system uses this interface that has a register that contains data about all the EVs around the world as the model, the manufacturer, the battery capacity, the efficiency, the range, the connector type and the maximum input power. It is also used to show the battery status of the user's car
- **OCPP compliant CP client**: eMall system relies on it to retrieve data or send data to the CPs
- **Calendar OS API**: The interface is used to identify which are the commitments and which are the free slots and suggest proactively when is more convenient to charge
- **Payment Gateway**: eMall system relies on it for processing every payment operation on the application
- **Push Notification Service**: eMall notifications will be sent to update in real time the EV driver that the charging shift will begin shortly, when the charging process has ended and of special offer in nearby CPs
- **SMS Service**: eMall system relies on it to send verification code SMS to customers phones.

3.2 Functional Requirements

In this section a top-down approach is followed. We provide first use cases diagrams, for each use case we provide a detailed description of the interaction and a sequence diagram. Then the list of functional requirements is built along with the mapping on both goals and use cases for traceability reasons.

3.2.1 Use case diagrams

Note: in the following diagrams, unless otherwise specified, it is the user who initiates the use case

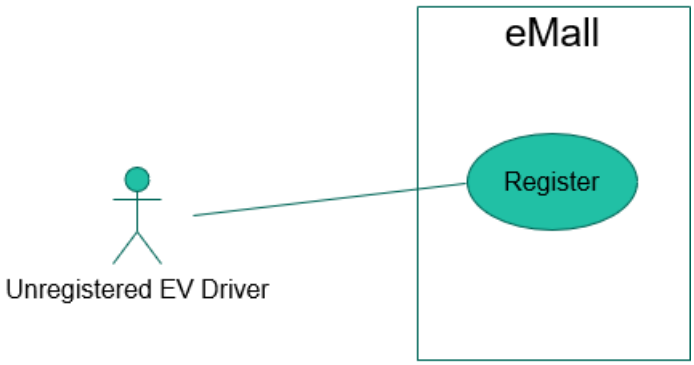


Figure 3: Unregistered EV Driver use case diagram

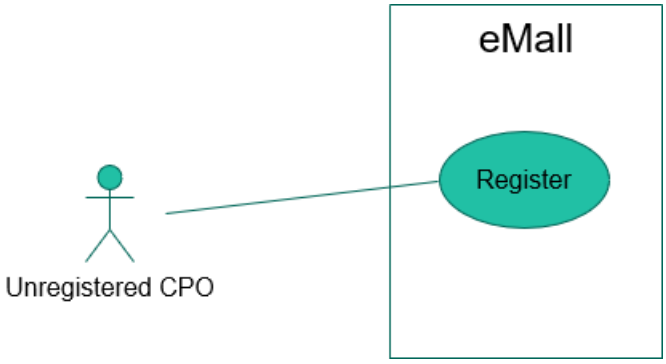


Figure 4: Unregistered CPO use case diagram

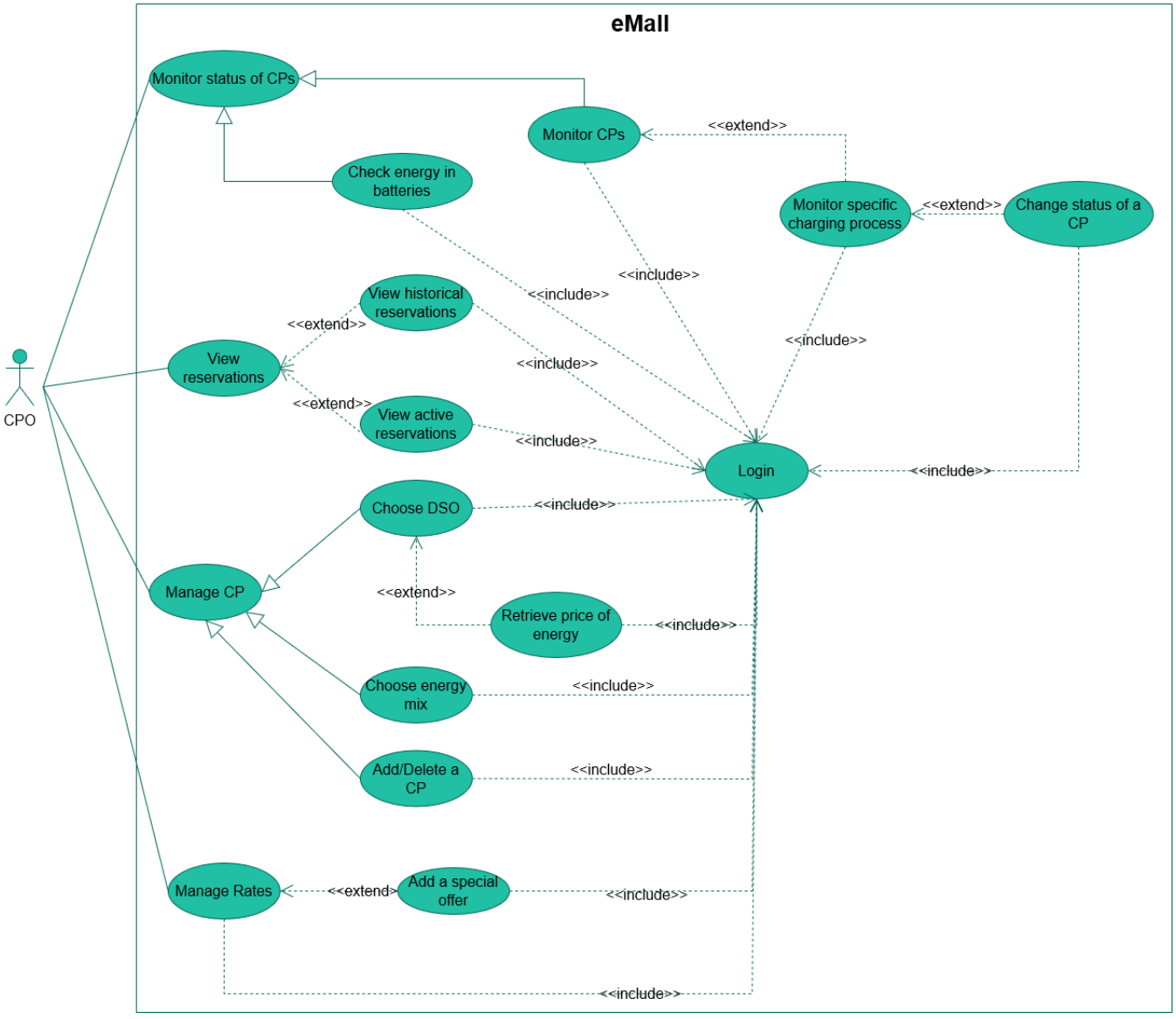


Figure 5: CPO use case diagram

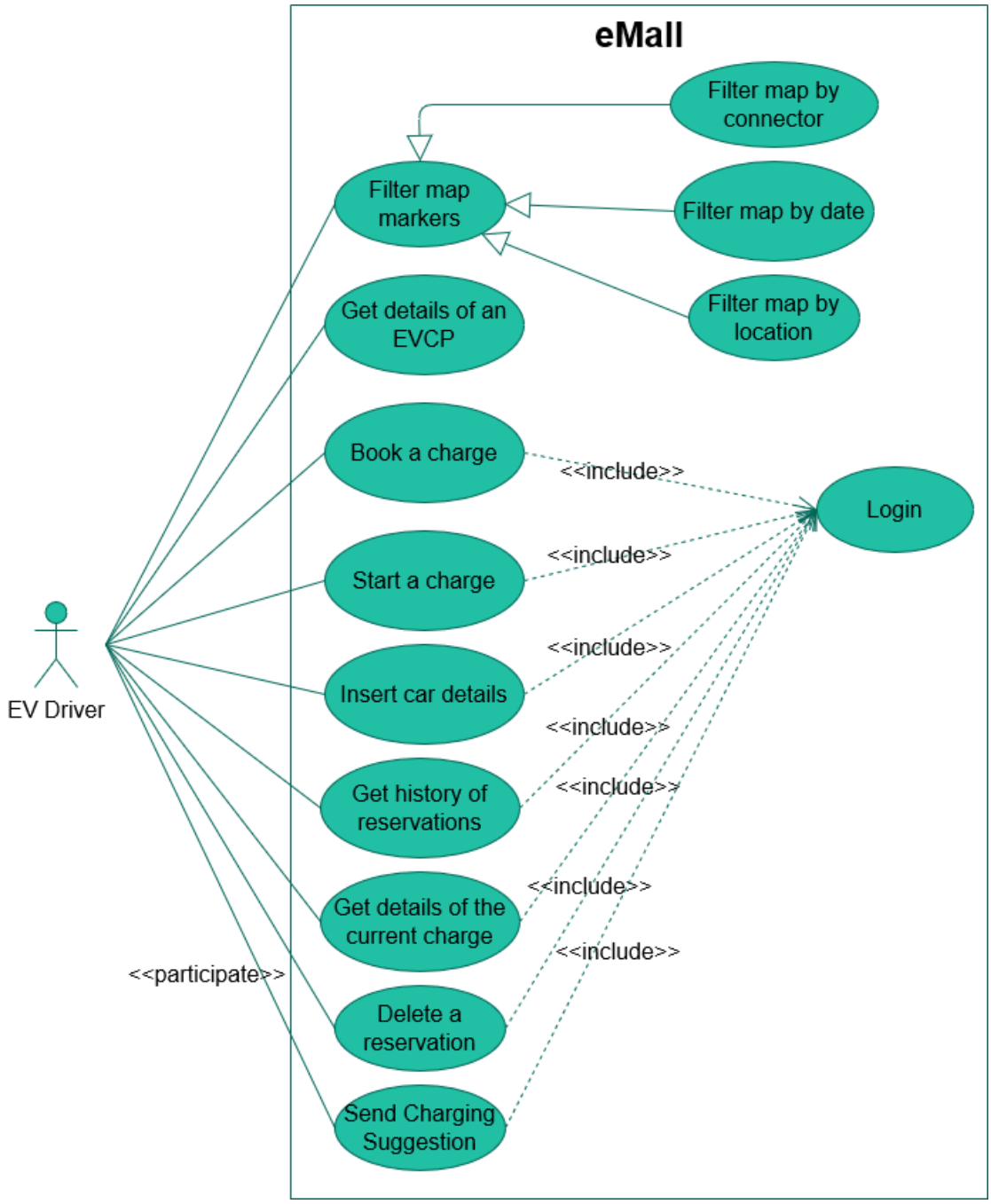


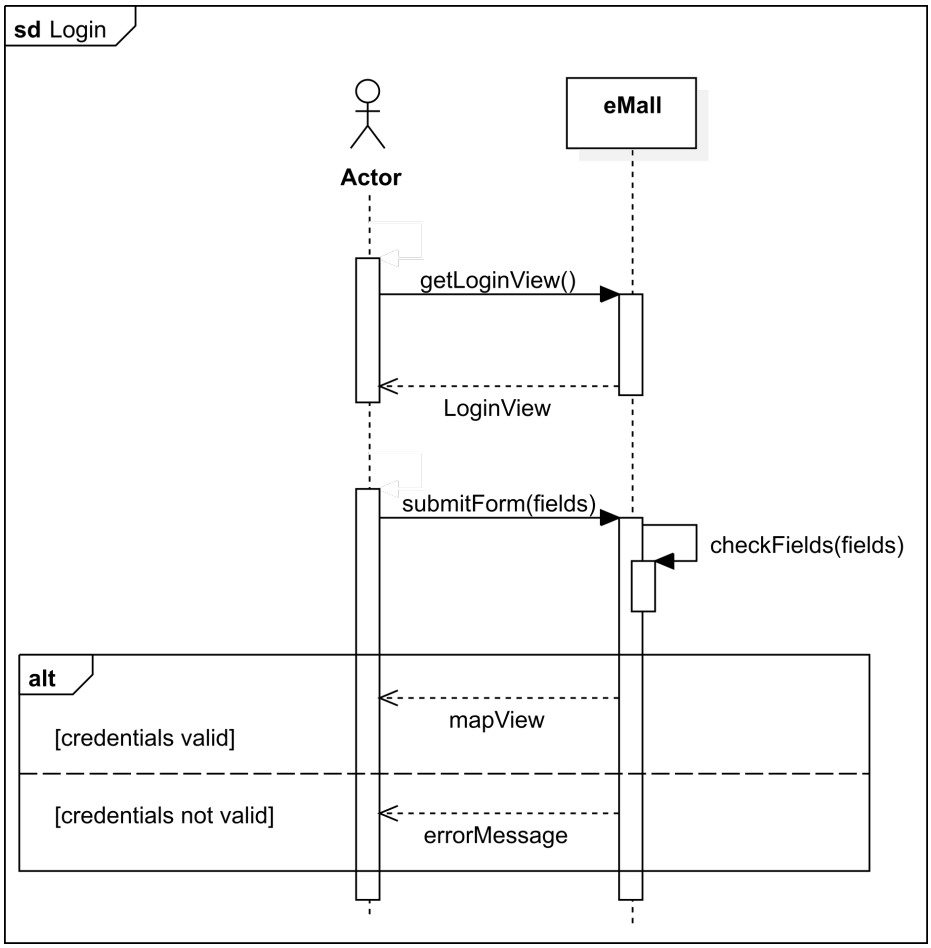
Figure 6: EV Driver use case diagram

3.2.2 Use cases

In all the following use cases and sequence diagrams, the System To Be is shown as a black box interacting with the world, that is users and external API.

Use case [U1]: Login

Actor(s)	EV driver, CPO
Entry Condition	The actor is already registered in the system
Event Flow	1. The actor requires the Login Page 2. The system shows the Login Page to the actor 3. The actor inserts credentials and send it to the system 4. The system processes the information and shows a success message redirecting the user to the homepage
Exit Condition	The actor is logged and the homepage is displayed
Exceptions	• A wrong username or password is submitted
Notes	In case of exception the system will notify user with a human-readable message

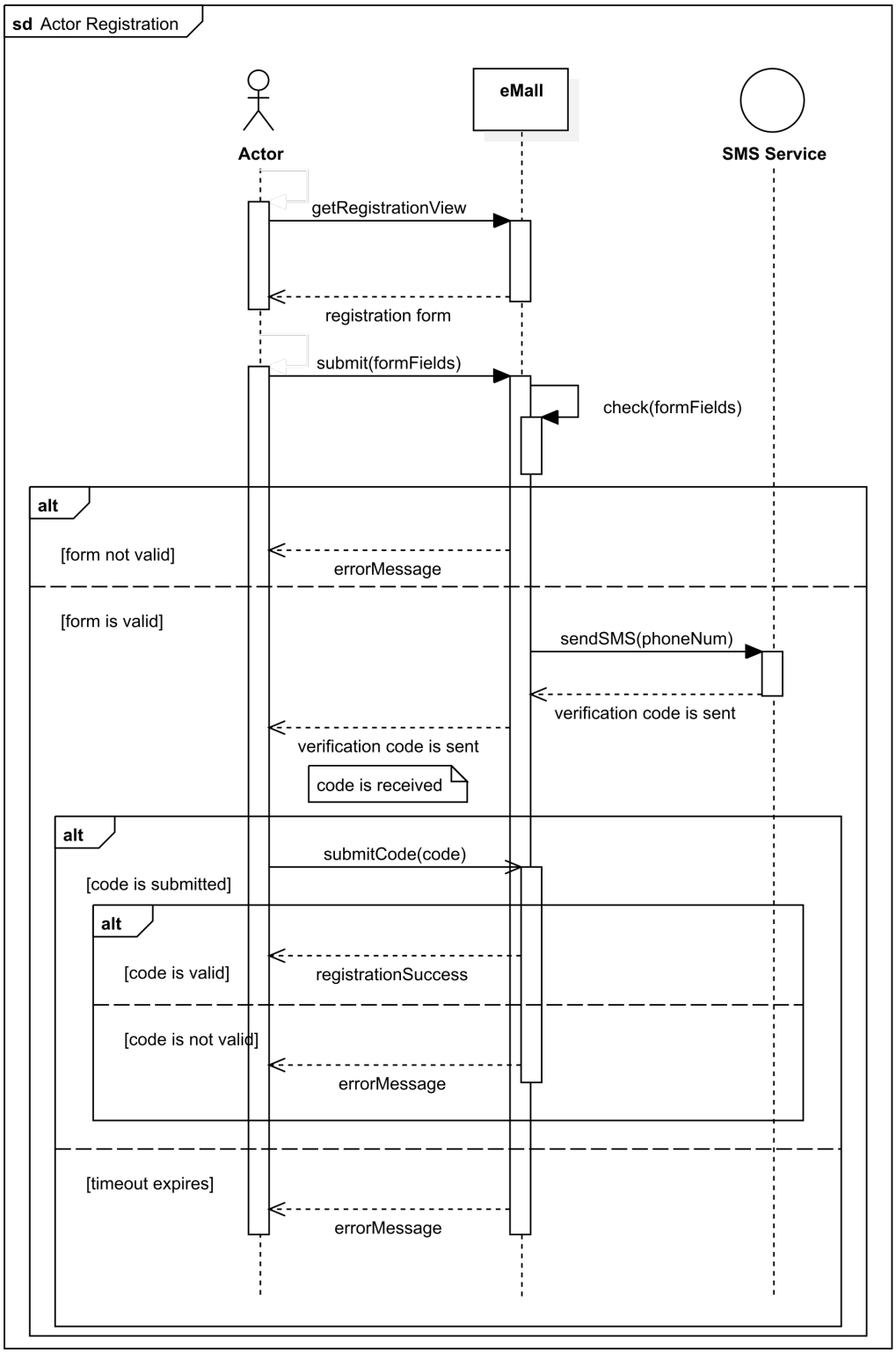


Use case [U2.1]: CPO registration

Actor(s)	CPO, SMS Service
Entry Condition	CPO clicks 'Sign Up' in the business dedicated application homepage
Event Flow	1. The system sends operator the registration form 2. Operator enters company name, VAT, IBAN, and password. Then submits the data upon reading and accepting the Privacy Policy and the Terms of Service 3. The system calls SMS Service API to send to the CPO an SMS containing a secret code 4. The SMS Service sends the SMS to the operator 5. The operator submits the received verification code 6. The system processes the provided information and displays a success message
Exit Condition	A new operator account is created
Exceptions	• A required registration field is missing when the form is submitted • The operator is not associated to the given VAT • A wrong verification code is submitted • The timeout of verification expires
Notes	All the exception are treated the same: the system will notify operator with a human-readable message and the system asks to retry

Use case [U2.2]: EV driver registration

Actor(s)	EV driver, SMS Service
Entry Condition	EV driver clicks 'Sign Up' in the application homepage
Event Flow	1. The system sends user the registration form 2. Driver enters name, surname, birth date, telephone number and password. Then submits the data upon reading and accepting the Privacy Policy and the Terms of Service 3. The system calls SMS Service API to send to the driver an SMS containing a secret code 4. The SMS Service sends the SMS to the driver 5. The driver submits the received verification code 6. The system verifies the code and displays a success message
Exit Condition	A new EV driver account is created
Exceptions	• A required registration field is missing when the form is submitted • A wrong verification code is submitted • The timeout of verification expires
Notes	All the exception are treated the same: the system will notify user with a human-readable message and the user is redirected to the homepage


sd Actor Registration
Actor
eMall
SMS Service
getRegistrationView
registration form
submit(formFields)
check(formFields)
alt
[form not valid]
errorMessage
[form is valid]
sendSMS(phoneNum)
verification code is sent
verification code is sent
code is received
alt
submitCode(code)
[code is submitted]
alt
[code is valid]
registrationSuccess
[code is not valid]
errorMessage
[timeout expires]
errorMessage

Use case [U3.1]: Map filtering by date

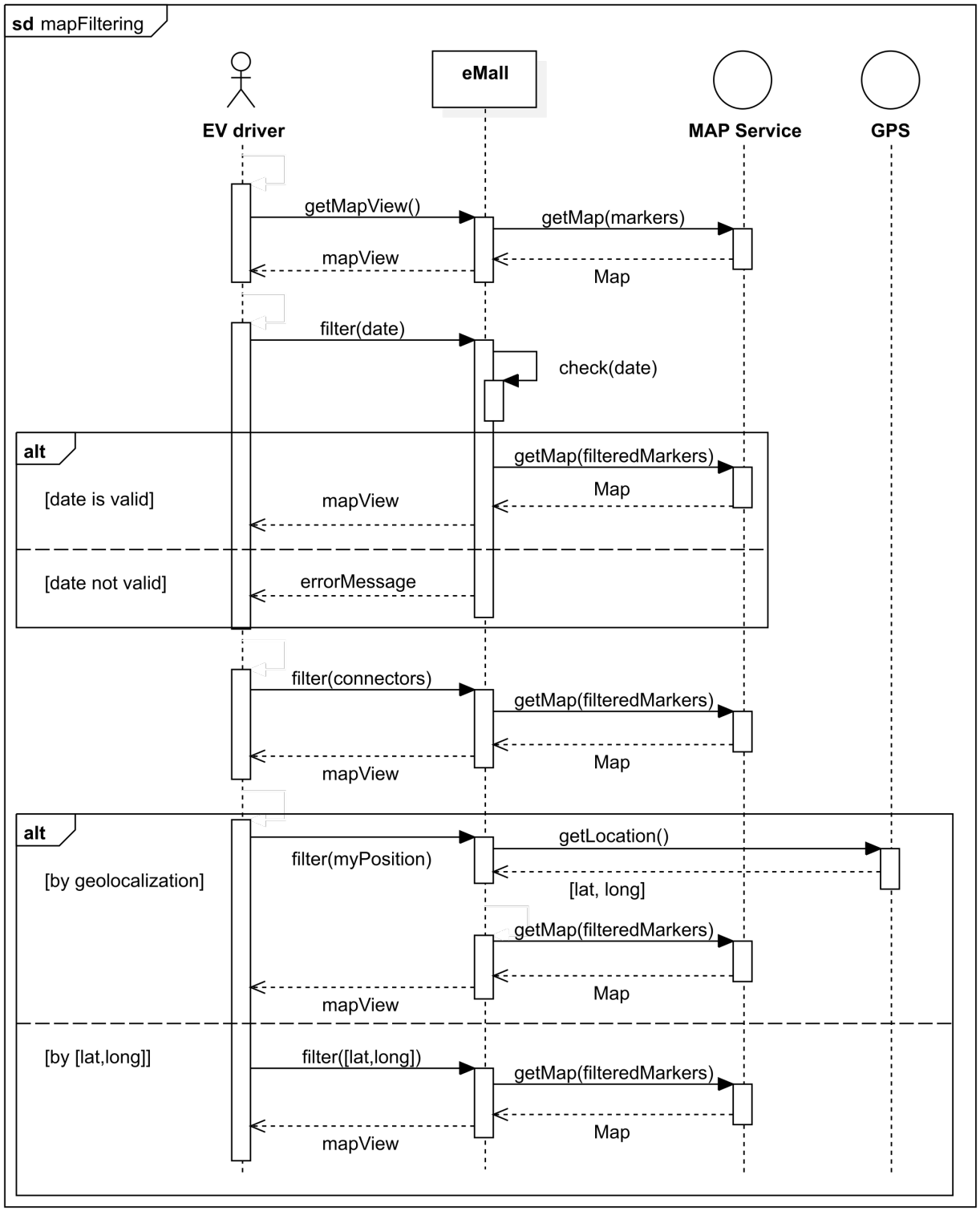
Actor(s)	EV driver, Map Service
Entry Condition	EV driver is in the map page
Event Flow	1. Driver selects a date from the search bar calendar 2. The system filters the CPs that have at least one available spot in the provided date and calls the Map Service 3. The Map Service updates the map
Exit Condition	The filtered markers are displayed
Exceptions	• The provided date is before the actual date of searching
Notes	In case of exception the system ignores the filtering and continues showing the map as it was

Use case [U3.2]: Map filtering by connector

Actor(s)	EV driver, Map Service
Entry Condition	EV driver is in the map page
Event Flow	1. Driver selects a list of connectors from the search bar connector dropdown 2. The system filters the CPs that have at least one available spot with one of the chosen connectors and calls the Map Service 3. The Map Service updates the map
Exit Condition	The filtered markers are displayed
Exceptions	• The provided connector is not recognized by the system
Notes	In case of exception the system ignores the filtering and continues showing the map as it was

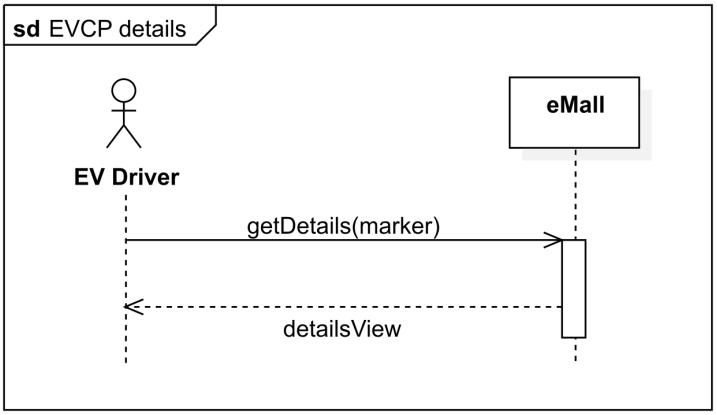
Use case [U3.3]: Map filtering by location

Actor(s)	EV driver, Map Service, GPS Service
Entry Condition	EV driver is in the map page
Event Flow	1. Driver specifies an address where to search for CPs or their current position 2. The system shows the CPs that are nearby the specified address and calls the Map Service 3. The Map Service updates the map
Exit Condition	The filtered markers are displayed
Exceptions	No exception in this use case
Notes	


sd mapFiltering
EV driver
eMall
MAP Service
GPS
getMapView()
getMap(markers)
mapView
Map
filter(date)
check(date)
alt
[date is valid]
getMap(filteredMarkers)
Map
mapView
[date not valid]
errorMessage
filter(connectors)
getMap(filteredMarkers)
Map
mapView
alt
[by geolocalization]
filter(myPosition)
getLocation()
[lat, long]
getMap(filteredMarkers)
Map
mapView
[by [lat,long]]
filter([lat,long])
getMap(filteredMarkers)
Map
mapView

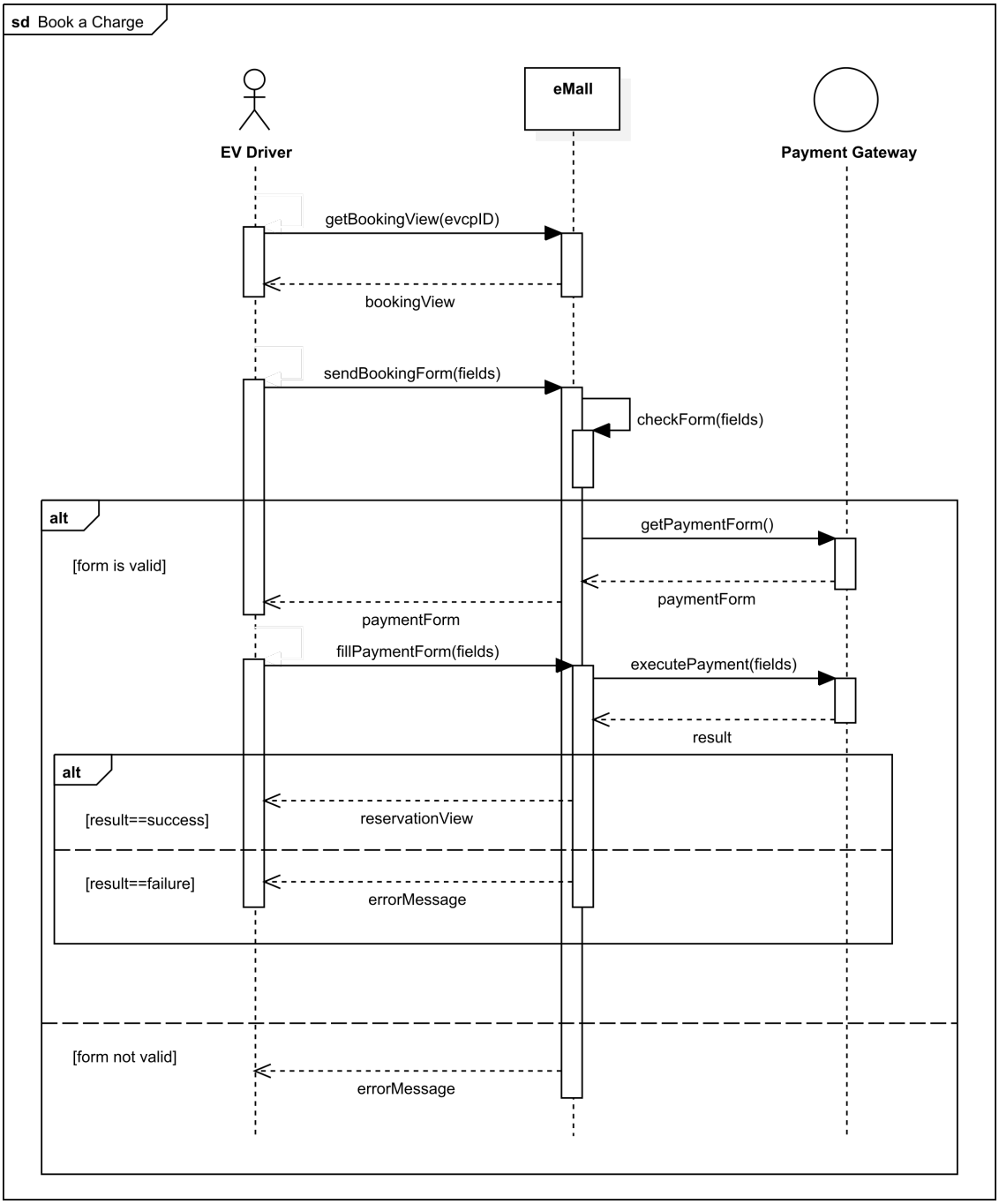
Use case [U4]: Get details of an EVCP

Actor(s)	EV driver
Entry Condition	EV driver is in the map page
Event Flow	1. Driver selects a marker representing an EVCP on the map 2. The system shows the details of the EVCP associated to the marker
Exit Condition	A page with EVCP details is shown
Exceptions	No exception in this use case
Notes	



Use case [U5]: Book a charge

Actor(s)	EV driver, Payment Gateway
Entry Condition	Authenticated EV driver is in the EVCP details page
Event Flow	1. User clicks 'Book a Charge' button or one among the 'Now' buttons related to connector types (if available) 2. The system show a booking page containing a form with connector, date and time frame selection 3. User fills the form and clicks 'Book' button 4. The system processes the information and redirects the user to the payment gateway interface 5. The Payment Gateway sends user a payment form 6. The user fills the form with payment information and clicks 'Confirm' 7. The Payment Gateway processes the information and replies to the system with a notification 8. The System processes the information and show a success message redirecting the user to the reservation tab
Exit Condition	The spot is booked and the transaction is successfully executed
Exceptions	• At least one of the inserted booking data is not valid • The payment gateway return an invalid payment
Notes	• First Exception: the system shows an error and redirect user to EVCP details page • Second Exception: the system shows an error during the payment process and asks to retry

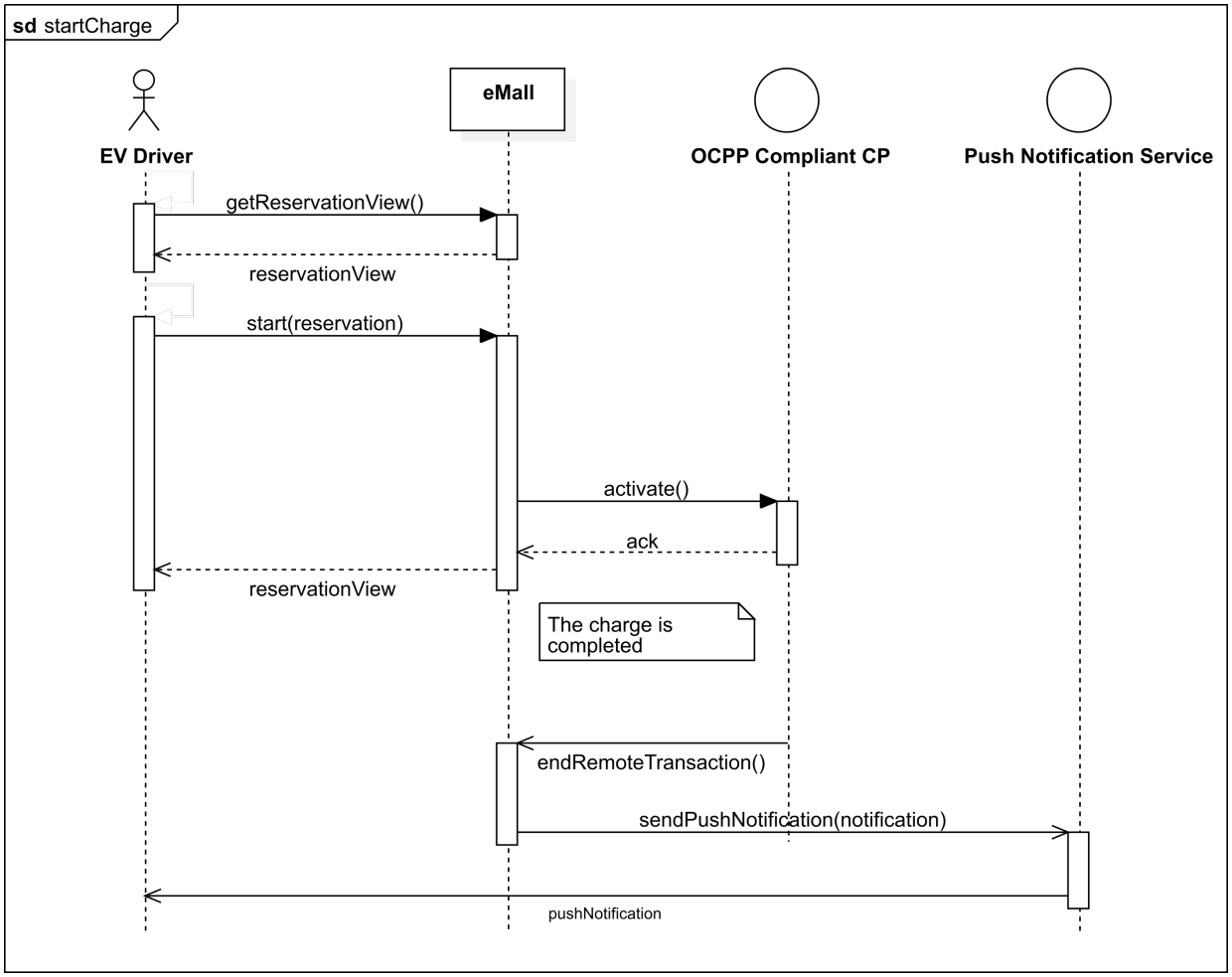

sd Book a Charge
EV Driver
eMall
Payment Gateway
getBookingView(evcpID)
bookingView
sendBookingForm(fields)
checkForm(fields)
alt
[form is valid]
getPaymentForm()
paymentForm
paymentForm
fillPaymentForm(fields)
executePayment(fields)
result
alt
[result==success]
reservationView
[result==failure]
errorMessage
[form not valid]
errorMessage

Use case [U6.1]: Start a charge

Actor(s)	EV driver, OCPP Compliant CP client
Entry Condition	Authenticated EV driver is in the reservation page
Event Flow	1. The system shows all the reservations with the ”pending” reservation in evidence 2. User clicks the button 'Start' in the reservation in evidence to start the charge 3. The system receives the input and activates the CP through the OCPP protocol 4. The CP replies acknowledging the system that its status has changed from ”reserved” to ”charging” 5. The system changes the status of the reservation in ”running”
Exit Condition	The charging is started
Exceptions	No exception in this use case
Notes	

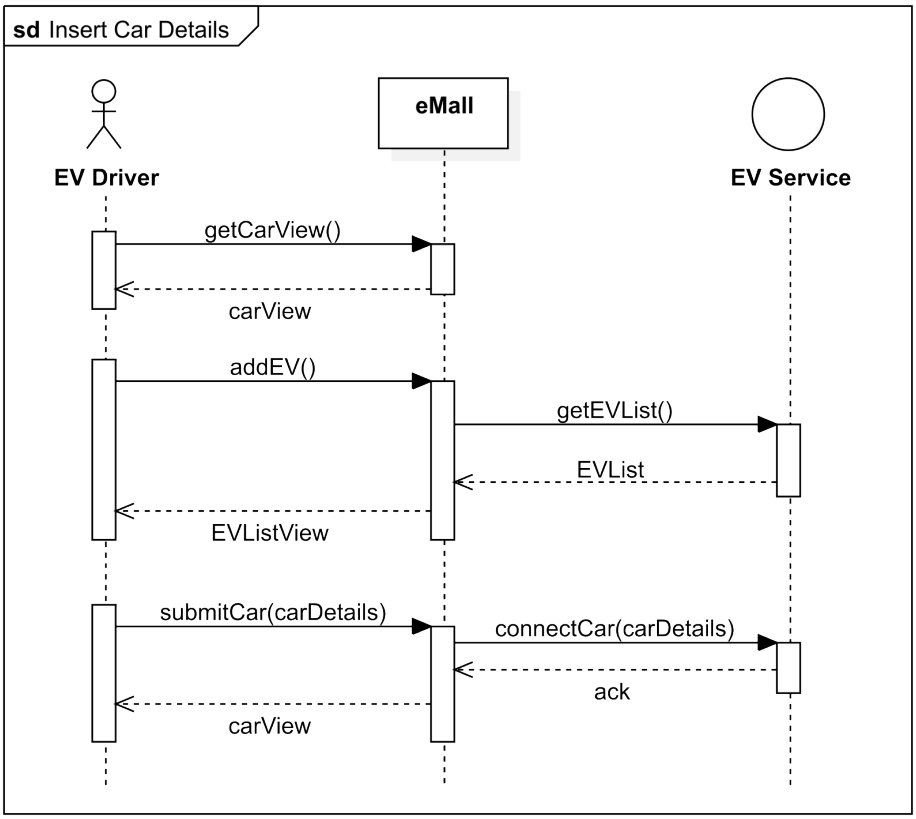
Use case [U6.2]: End of Charge Notification

Actor(s)	EV driver, Push Notification Service
Entry Condition	The charge is completed
Event Flow	1. At the end of the charge, The system asks Push Notification Service to notify the user about the status of the charge 2. The Push Notification Service sends the notification to the user
Exit Condition	The notification is sent to the user
Exceptions	No exception in this use case
Notes	



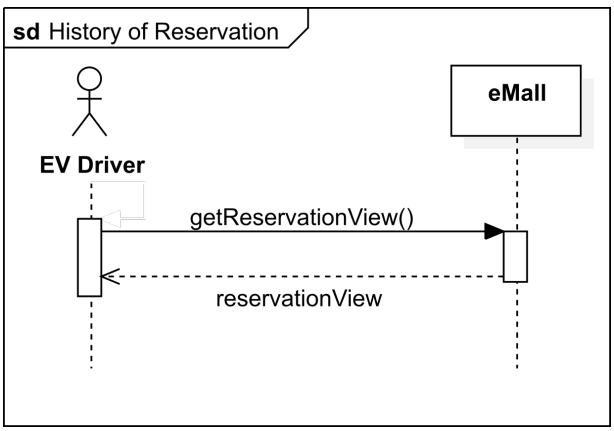
Use case [U7]: Personalize experience by inserting car details

Actor(s)	EV driver, EV Service
Entry Condition	Authenticated EV driver is in homepage
Event Flow	1. The user ask the car view 2. The system shows user the car page 3. The user clicks 'Add an EV' button 4. The system ask EV Service an updated list of EVCP 5. The EV Service replies with an updated list of EV 6. The systems shows the list of EV to user 7. The user select one of the EV and connects the personal EV to the EV Service API 8. The system processes the information and shows a success message
Exit Condition	The user has inserted car details
Exceptions	No exception in this use case
Notes	



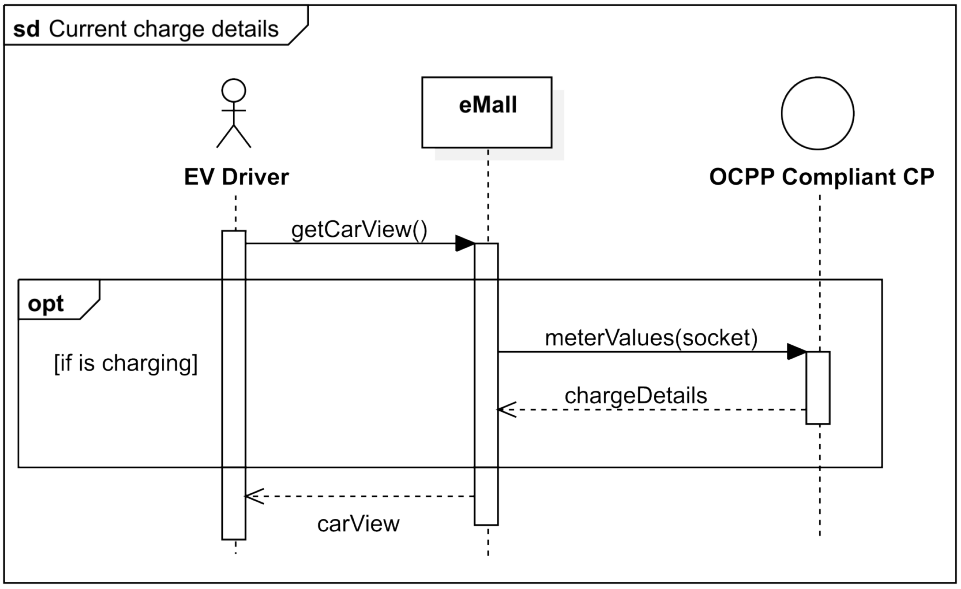
Use case [U8]: Get history of reservations

Actor(s)	EV driver
Entry Condition	Authenticated EV driver is in the home page
Event Flow	1. The user switch to the reservation tab 2. The system shows a page with a list of past reservations and active ones
Exit Condition	A list of reservations is shown
Exceptions	No exception in this use case
Notes	



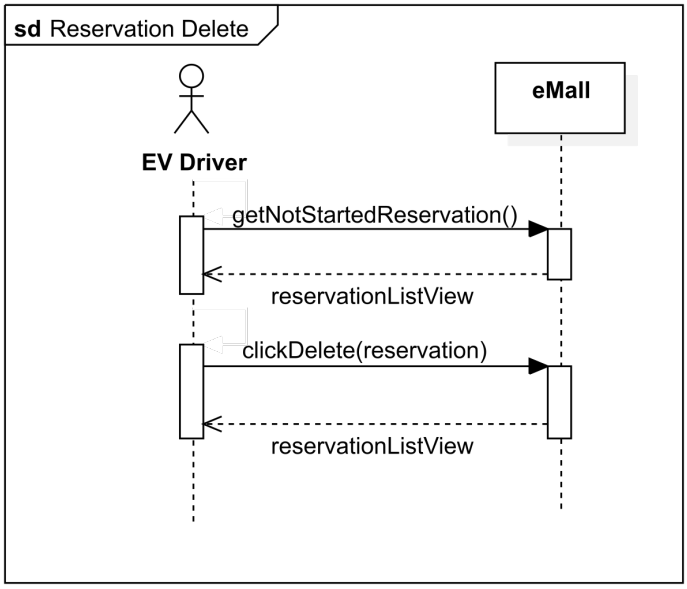
Use case [U9]: Get details of the current charge

Actor(s)	EV driver, OCPP Compliant CP Client
Entry Condition	Authenticated EV driver is in the car tab
Event Flow	1. The system asks the CP, through OCPP, the details of the current charge of the user 2. The CP replies, through OCPP, with the status 3. The system shows all the details of the current charge
Exit Condition	The details of the current charge are shown
Exceptions	No exception in this use case
Notes	



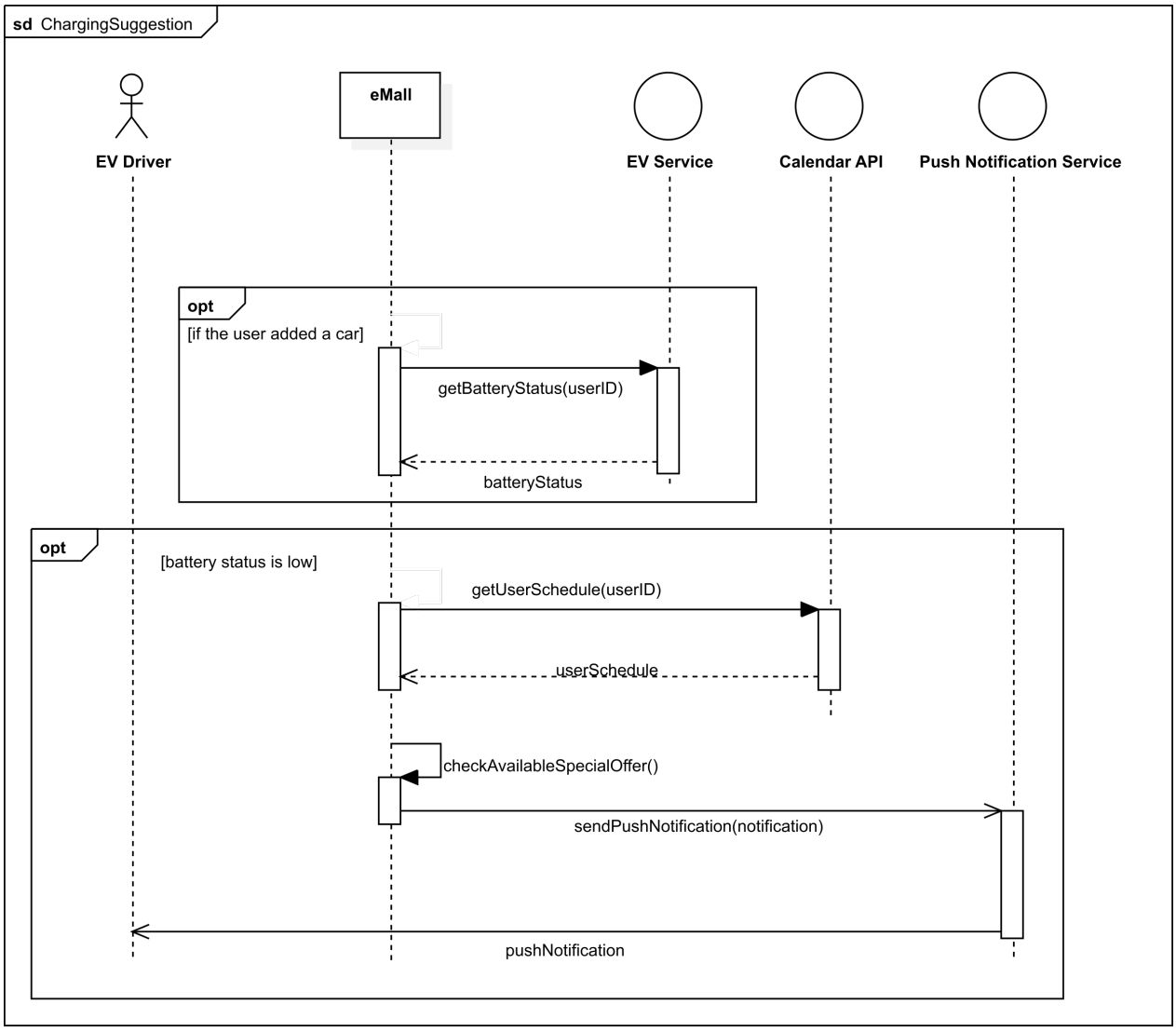
Use case [U10]: Delete a reservation

Actor(s)	EV driver
Entry Condition	Authenticated EV driver is in reservation tab
Event Flow	1. The system shows a list with all the "not started" reservation 2. User selects a reservation and click the associated button 3. The system receives the input and shows details on the selected reservation 4. User clicks "delete" button in current reservation 5. The system displays a success message
Exit Condition	The user has deleted a reservation
Exceptions	• At least one of the inserted data is not valid
Notes	The system shows an error asking to retry



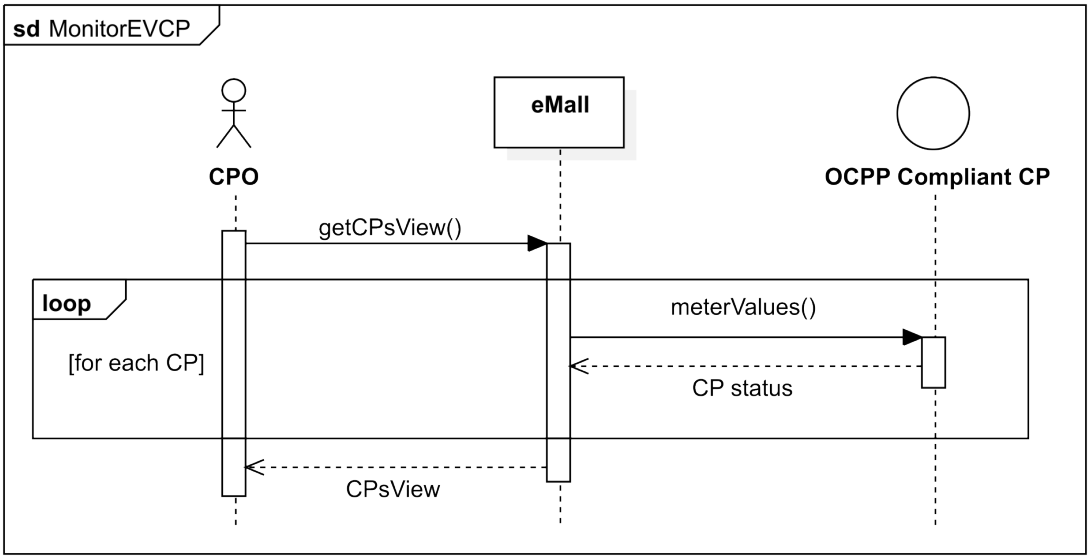
Use case [U11]: Send charging suggestion

Actor(s)	EV driver, Push Notification Service, Calendar API, EV Service
Entry Condition	Authenticated EV driver (not necessarily using the system)
Event Flow	1. The system calls the EV Service asking for the EV battery status 2. The EV Service replies with the battery status 3. The system calls the Calendar API asking for the user schedule 4. The Calendar API replies with the user schedule 5. The system asks Push Notification Service to suggest the user when and where to charge 6. The Push Notification Service sends the notification
Exit Condition	The notification is pushed to the user
Exceptions	No exception in this use case
Notes	



Use case [U12]: Monitor CPs

Actor(s)	CPO, OCPP compliant CP client
Entry Condition	Authenticated CPO
Event Flow	1. The operator clicks "Charging Points" tab 2. The system asks all the CPs, through OCPP, the details of their status 3. The CPs reply, through OCPP, with their status 4. The system shows a table with detailed views of CPs
Exit Condition	The aggregate CPs details charts are displayed
Exceptions	No exception in this use case
Notes	

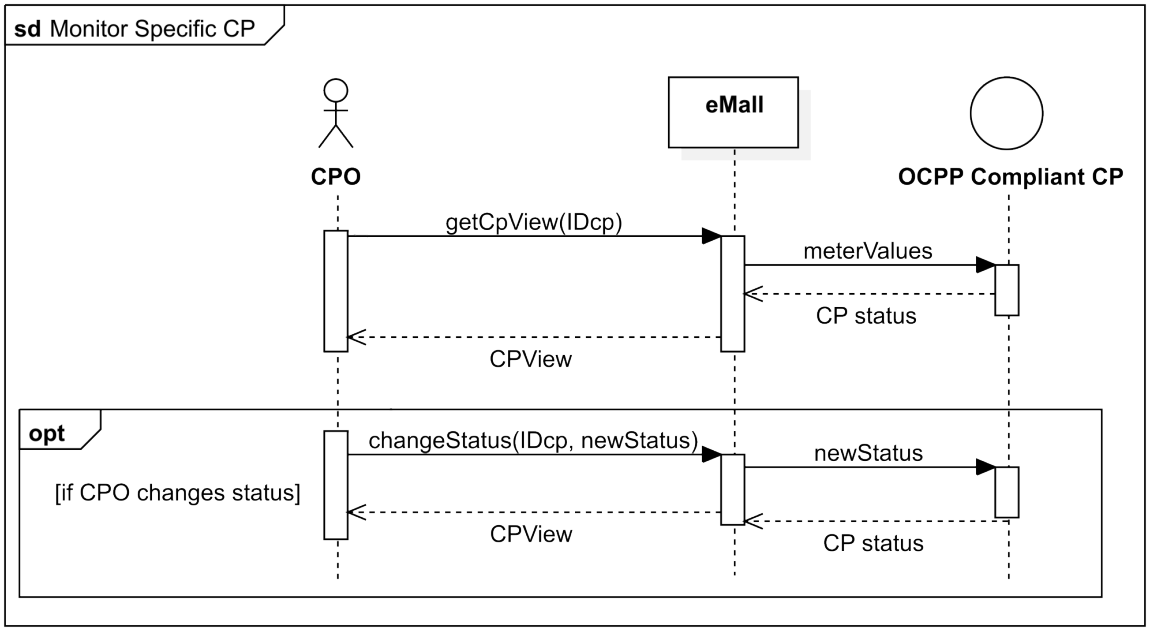


Use case [U13.1]: Monitor specific CP

Actor(s)	CPO, OCPP compliant CP client
Entry Condition	Authenticated CPO is in "Charging Points" tab
Event Flow	1. The operator selects a specific CP 2. The system asks CP, through OCPP, details of its status 3. The CP replies, through OCPP, with the status 4. The system shows the details of the chosen CP
Exit Condition	The details of the specific CP are displayed
Exceptions	No exception in this use case
Notes	

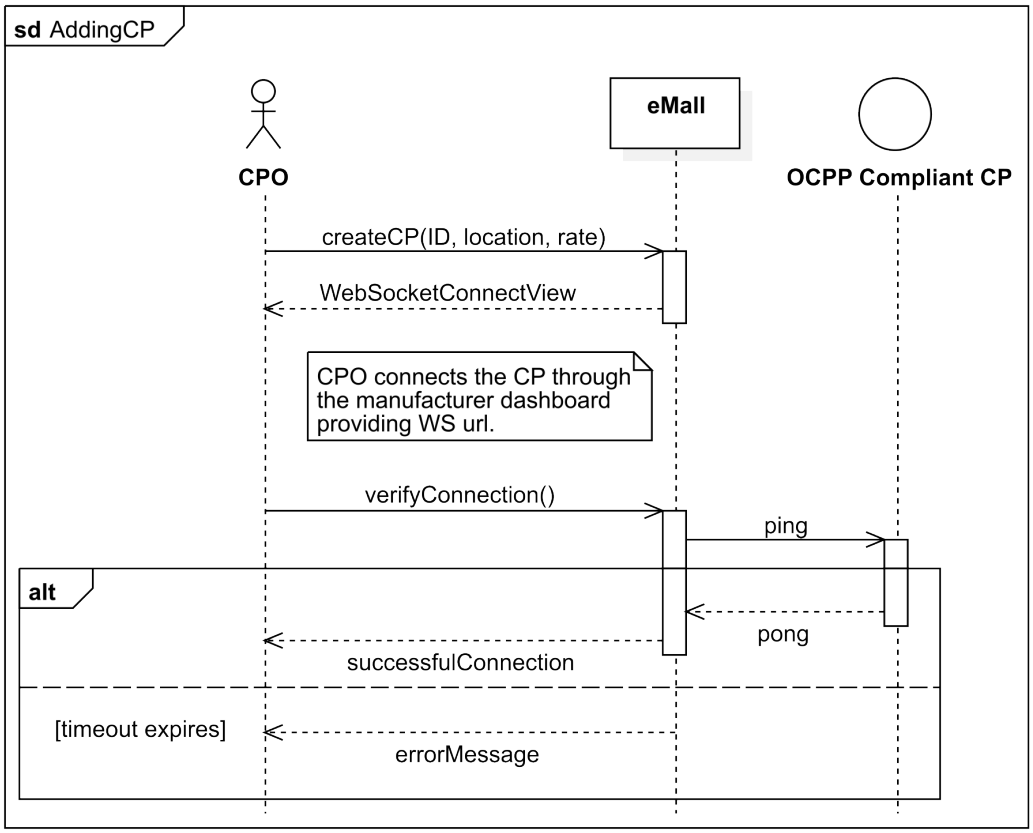
Use case [U13.2]: Change Status of a CP

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Charging Point" tab
Event Flow	1. The operator changes the state of the specific CP 2. The system asks CP, through OCPP, to change to the desired status 3. The CP replies, through OCPP, with the new status 4. The system shows the details of the chosen CP
Exit Condition	The change of status of the specific CP has been planned
Exceptions	No exception in this use case
Notes	



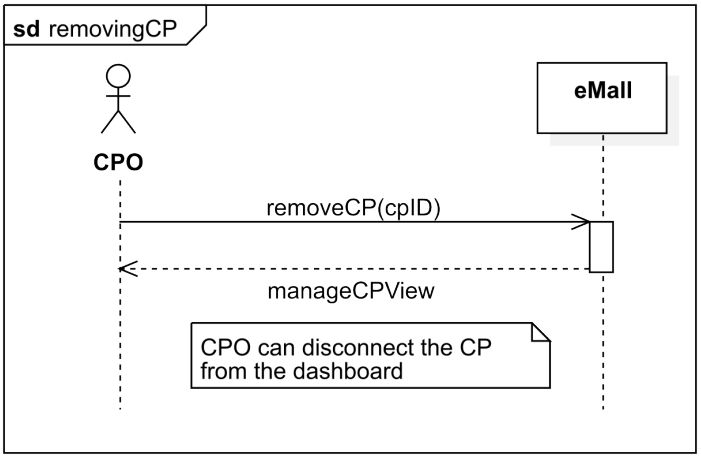
Use case [U14]: Adding a CP

Actor(s)	CPO, OCPP compliant CP client
Entry Condition	Authenticated CPO is in "Charging Points" tab
Event Flow	1. The operator clicks "Add a new CP" button 2. The system provides a popup to insert details of the new CP 3. The operator specifies the ID (serial number of CP), the EVCP in which it is, then clicks "connect" button 4. The system shows an OCPP URL to connect the CP to the CPMS and waits for a connection 5. The operator connects the CP using the CP's manufacturer platform 6. The system shows a message when the connection is established
Exit Condition	The new CP is connected
Exceptions	1. The user cancels the operation
Notes	In case of exception the system close the pupup



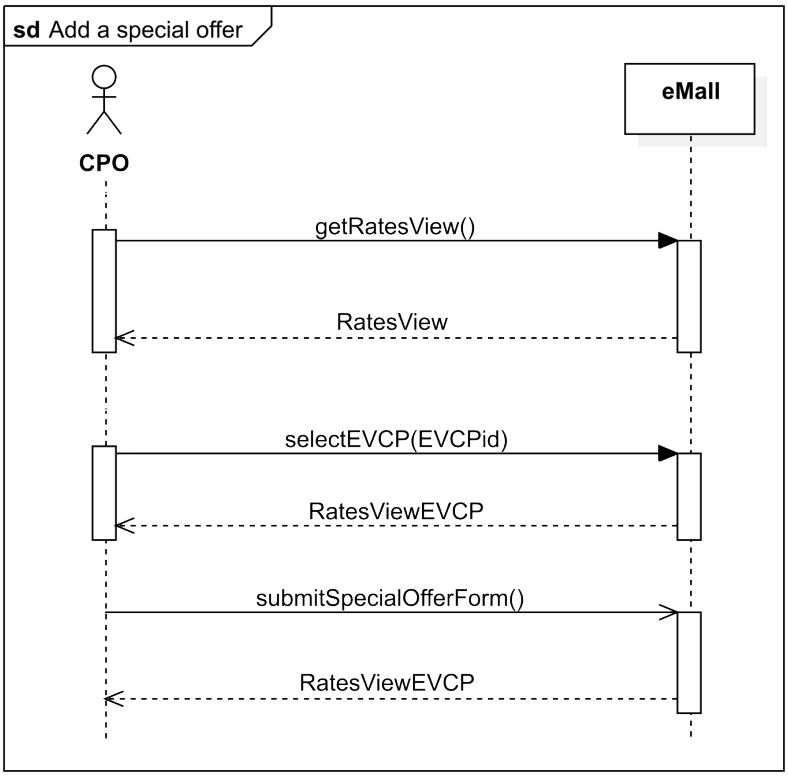
Use case [U15]: Removing a CP

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Charging Points" tab
Event Flow	1. The operator selects a CP and clicks "Remove CP" button 2. The system removes the association with the CP
Exit Condition	The CP is removed
Exceptions	No exception in this use case
Notes	



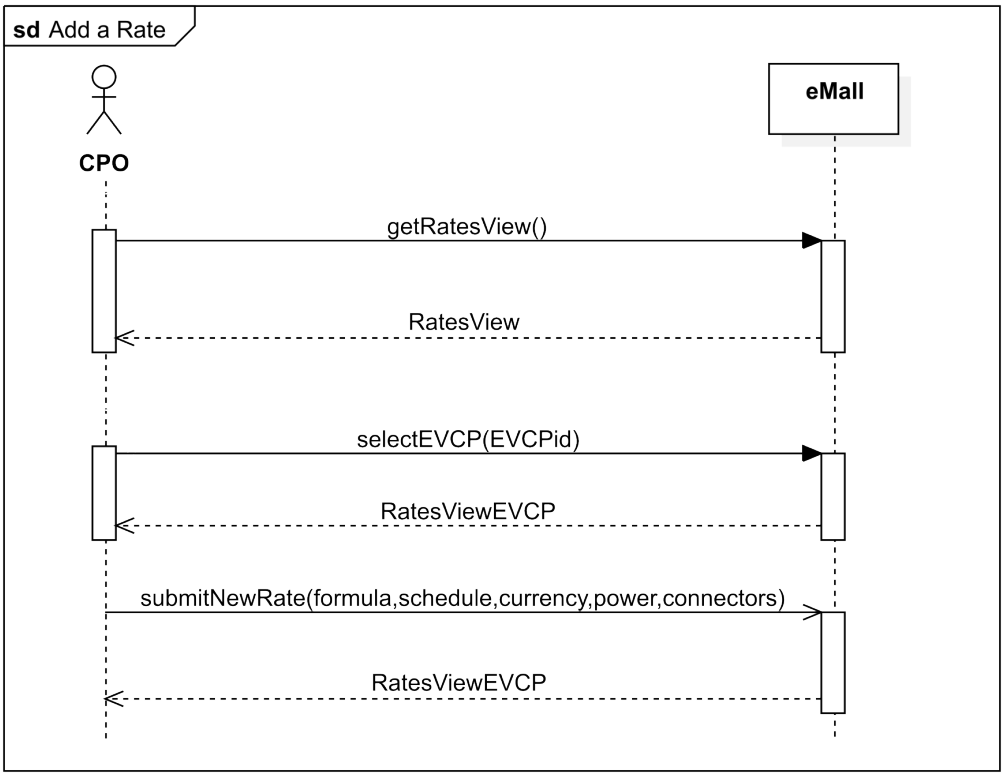
Use case [U16.1]: Add a special offer

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Rates" tab
Event Flow	1. The operator chooses a specific EVCP 2. The system shows the rates about the EVCP and a form to add a special rate 3. The operator fills the form by specifying a time slot and the percentage of discount then submits 4. The system processes the information and shows a success message
Exit Condition	The special offer is created
Exceptions	No exception in this use case
Notes	



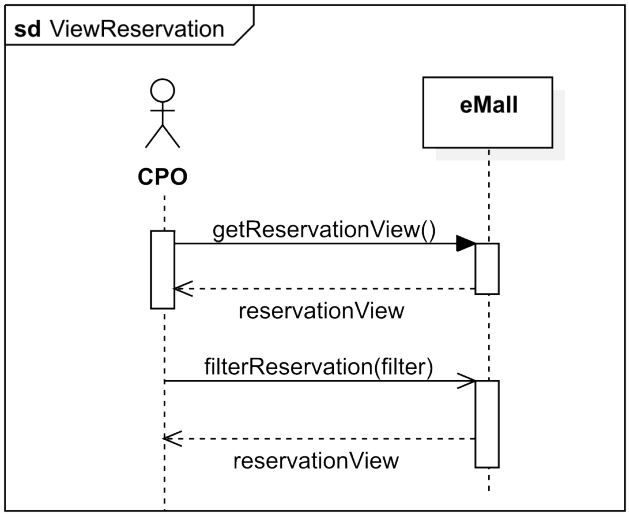
Use case [U16.2]: Add a Rate

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Rates" tab
Event Flow	1. The operator chooses a specific EVCP 2. The system shows the rates about the EVCP 3. The operator clicks "Add a rate" button 4. The system shows a form to fill to create a new rate 5. The operator fills the form with the rate formula, the schedule of the rate, the currency, the power of the connector and the sockets that will use that rate, the submits 6. The system shows the rates about the EVCP
Exit Condition	The special offer is created
Exceptions	• The operator cancels the operation
Notes	In case of exception the system closes the form



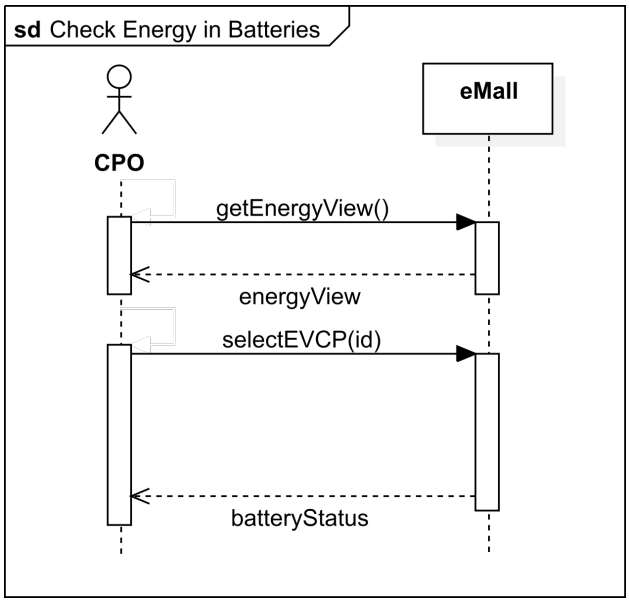
Use case [U17]: View reservations

Actor(s)	CPO
Entry Condition	Authenticated CPO
Event Flow	1. The operator clicks "Reservations" tab 2. The system shows the reservations' table 3. The operator choose a filter (completed, running, pending, upcoming) or to sort reservations (by CP, EVCP, ...)
Exit Condition	The reservations table is displayed
Exceptions	No exception in this use case
Notes	



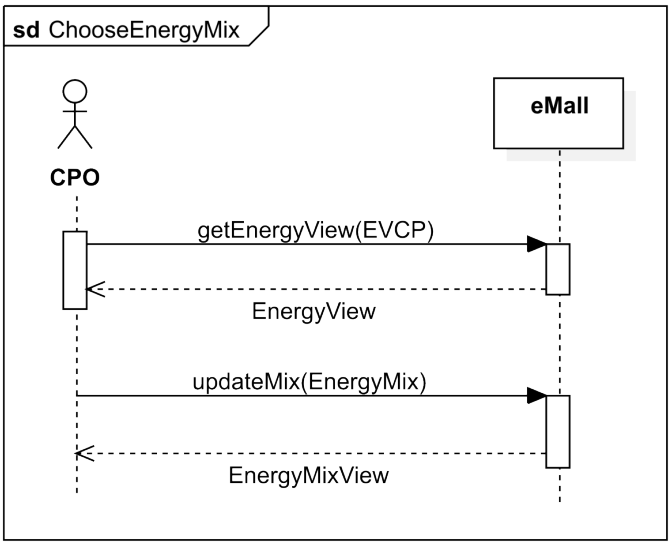
Use case [U18]: Check energy in batteries

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Energy" tab
Event Flow	1. The operator choose a specific EVCP 2. The system shows the energy storage status of the selected EVCP, if any
Exit Condition	The 'Check energy in batteries' chart is shown
Exceptions	No exception in this use case
Notes	



Use case [U19]: Choose energy mix

Actor(s)	CPO
Entry Condition	Authenticated CPO is in "Energy" tab
Event Flow	1. The operator chooses a specific EVCP and configure a specific Energy Storage Working Mode 2. The system processes the information and shows a success message
Exit Condition	The choice of an EVCP energy storing mode is completed
Exceptions	No exception in this use case
Notes	

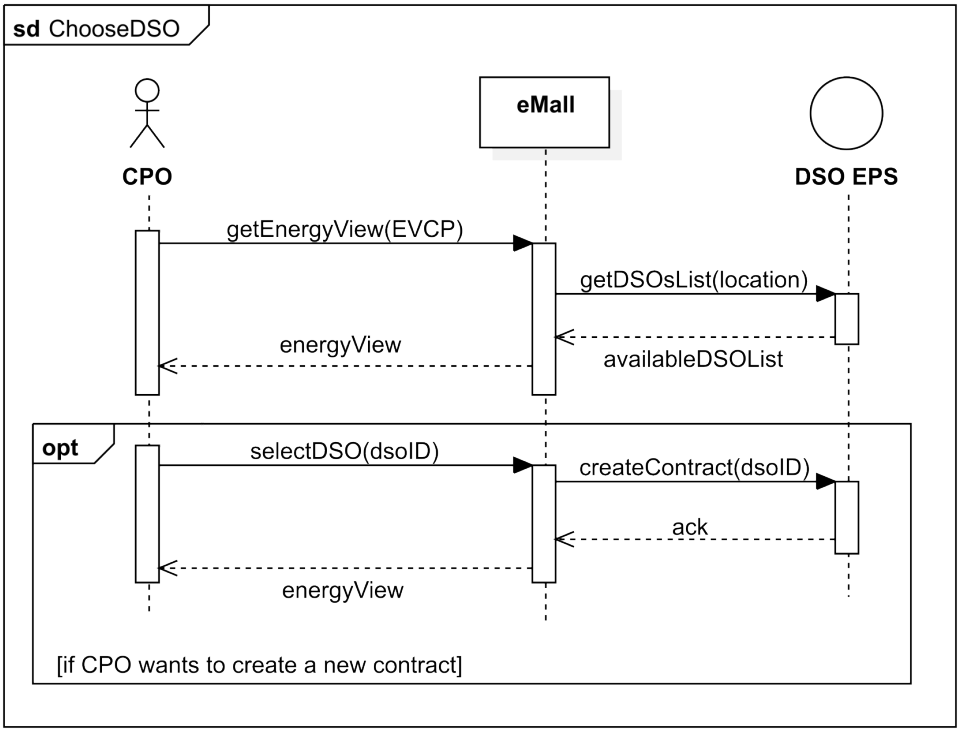


Use case [U20.1]: Retrieve price of energy

Actor(s)	CPO, DSO Energy Price Service
Entry Condition	Authenticated CPO is in "Energy" tab
Event Flow	1. The operator choose a specific EVCP 2. The system asks DSO Energy Pricing Service the list of available DSOs with their prices 3. The DSO Energy Pricing Service replies with the list requested 4. The system shows the list of available DSOs with their prices
Exit Condition	The details of price energy are displayed
Exceptions	No exception in this use case
Notes	

Use case [U20.2]: Choose DSO

Actor(s)	CPO, DSO Energy Pricing Service
Entry Condition	Authenticated CPO is in "Energy" tab
Event Flow	1. The operator choose a specific EVCP, clicks "Change contract" 2. The operator selects a DSO and submit its choice to the system 3. The system asks DSO Energy Pricing Service to acquire energy from the chosen DSO for the specified EVCP 4. The DSO Energy Pricing Service replies with an ACK 5. The system shows a success message
Exit Condition	The choice of DSO is completed
Exceptions	No exception in this use case
Notes	



3.2.3 CPMS subsystem Functional Requirements

Rx	Description	Priority Level
R1	The system must allow unregistered CPO to register an account	Must Have
R2	The system must allow registered CPO to login	Must Have
R3	The system must allow authenticated CPOs making a special offer on their CPs prices	Must Have
R4	The system must allow authenticated CPOs monitoring the charging process to infer when the battery is full	Must Have
R5	The system must allow authenticated CPOs retrieving the amount of energy available in their EVCPs batteries	Must Have
R6	The system must allow authenticated CPOs retrieving the number of vehicle being charged in their EVCPs and for each vehicle the amount of absorbed power	Must Have
R7	The system must allow authenticated CPOs retrieving the remaining charge time for each connected vehicle	Should Have
R8	The system must allow authenticated CPOs retrieving details on active and historical reservations on their EVCPs	Should Have
R9	The system must allow authenticated CPOs acquiring information from the DSOs about the current price of energy	Must Have
R10	The system must allow authenticated CPOs deciding from which DSO to acquire energy from	Must Have
R11	The system must dynamically decide where to get energy for charging (electrical grid, battery or a mixture)	Must Have
R12	The system must allow authenticated CPOs statically deciding where to get energy for charging (electrical grid, battery or a mixture)	Must Have
R13	The system must allow authenticated CPOs adding, modifying and deleting CPs	Must Have
R14	The system must allow authenticated CPOs changing availability status of their CPs	Must Have

3.2.4 eMSP subsystem Functional Requirements

Rx	Description	Priority Level
R15	The system must allow unregistered users to register an account	Must Have
R16	The system must allow registered users to login	Must Have
R17	The system must allow authenticated users to personalize their experience by providing information of their EV	Should Have
R18	The system must allow users to search for CPs in the map	Must Have
R19	The system must show to the users CPs nearby their current position	Must Have
R20	The system must allow retrieving details on a given CP regarding connector types supported and cost of the charge	Must Have
R21	The system must allow authenticated user to book a CP for a certain time interval	Must Have
R22	The system must allow booking a CP if and only if it is free for the specified time interval	Must Have
R23	The system must notify users when the charging shift is about to start	Nice to Have
R24	The system must allow authenticated users to start the charge	Must Have
R25	The system must suggest users when and where to charge based on daily schedule, special offers and availability	Must Have
R26	The system must allow authenticated users to monitor the charging status	Nice to Have
R27	The system must notify authenticated users when the charging process is completed	Must Have
R28	The system must allow authenticated users to pay for the charge	Must Have
R29	The system must allow authenticated users to delete a reservation	Should Have
R30	The system must allow authenticated users to view historical reservations	Should Have

3.2.5 Mapping on goals

G1 Allow EV driver to plan efficiently their charging process
Requirements
R1 The system must allow unregistered CPO to register an account
R13 The system must allow authenticated CPOs adding, modifying and deleting CPs
R14 The system must allow authenticated CPOs changing availability status of their CPs
R15 The system must allow unregistered users to register an account
R16 The system must allow registered users to login
R18 The system must allow users to search for CPs in the map
R19 The system must show to the users CPs nearby their current position
R21 The system must allow authenticated user to book a CP for a certain time interval
R22 The system must allow booking a CP if and only if it is free for the specified time interval
R25 The system must suggest users when and where to charge based on daily schedule, special offers and availability
R28 The system must allow authenticated users to pay for the charge
R29 The system must allow authenticated users to delete a reservation
Domain Assumptions
D1 An EV driver arrives at the charging station at a time close to its reservation starting time
D2 An EV driver leaves the charging station when the charge is finished
D3 An EV driver doesn't occupy an already booked charging spot
D5 At least one DSO can always provide energy to the CPOs

G2 Allow EV driver to have a single application for all the processes involving the charge with a personalized experience based on the car and the user commitments

Requirements

R15 The system must allow unregistered users to register an account

R16 The system must allow registered users to login

R17 The system must allow authenticated users to personalize their experience by providing information of their EV

R18 The system must allow users to search for CPs in the map

R19 The system must show to the users CPs nearby their current position

R20 The system must allow retrieving details on a given CP regarding connector types supported and cost of the charge

R21 The system must allow authenticated user to book a CP for a certain time interval

R23 The system must notify users when the charging shift is about to start

R24 The system must allow authenticated users to start the charge

R25 The system must suggest users when and where to charge based on daily schedule, special offers and availability

R26 The system must allow authenticated users to monitor the charging status

R27 The system must notify authenticated users when the charging process is completed

R28 The system must allow authenticated users to pay for the charge

R29 The system must allow authenticated users to delete a reservation

R30 The system must allow authenticated users to view historical reservations

Domain Assumptions

Dep1 The system requires access to a third party maps API

Dep2 The system will use the GPS of the driver's computer or smartphone

Dep3 The system will require internet connection to interact with all the users

Dep5 The system will use an external API to retrieve the EV battery status and a list of EV available on the market

Dep7 The system will use an external API to access the calendar of the users

Dep8 The system will use a payment gateway to perform payment operations

Dep9 The system will use an external API to send push notification to the drivers

Dep10 The system will use a third party API to send SMS to customers phones

G3 Allow CPOs to be reached by EV drivers looking for charging points

Requirements

R1 The system must allow unregistered CPO to register an account

R3 The system must allow authenticated CPOs making a special offer on their CPs prices

R4 The system must allow authenticated CPOs monitoring the charging process to infer when the battery is full

R7 The system must allow authenticated CPOs retrieving the remaining charge time for each connected vehicle

R8 The system must allow authenticated CPOs retrieving details on active and historical reservations on their EVCPs

R13 The system must allow authenticated CPOs adding, modifying and deleting CPs

R14 The system must allow authenticated CPOs changing availability status of their CPs

R15 The system must allow unregistered users to register an account

R16 The system must allow registered users to login

R18 The system must allow users to search for CPs in the map

R21 The system must allow authenticated user to book a CP for a certain time interval

R24 The system must allow authenticated users to start the charge

R25 The system must suggest users when and where to charge based on daily schedule, special offers and availability

R27 The system must notify authenticated users when the charging process is completed

R28 The system must allow authenticated users to pay for the charge

Domain Assumptions

D2 An EV driver leaves the charging station when the charge is finished

D3 An EV driver doesn't occupy an already booked charging spot

D5 At least one DSO can always provide energy to the CPOs

D6 An user that books a charge has an electric vehicle to charge

D7 An user that books a charge is always reliable

G4 Provide smart managing of charging stations, including the register of reservations

Requirements

R1 The system must allow unregistered CPO to register an account

R4 The system must allow authenticated CPOs monitoring the charging process to infer when the battery is full

R6 The system must allow authenticated CPOs retrieving the number of vehicle being charged in their EVCPs and for each vehicle the amount of absorbed power

R7 The system must allow authenticated CPOs retrieving the remaining charge time for each connected vehicle

R8 The system must allow authenticated CPOs retrieving details on active and historical reservations on their EVCPs

R13 The system must allow authenticated CPOs adding, modifying and deleting CPs

R14 The system must allow authenticated CPOs changing availability status of their CPs

Domain Assumptions

D5 At least one DSO can always provide energy to the CPOs

Dep3 The system will require internet connection to interact with all the users

Dep6 The system will use an external API to retrieve data or send data to the CPs

G5 Allow CPOs to choose between contracts of energy providers and to determine the energy source mix

Requirements

R5 The system must allow authenticated CPOs retrieving the amount of energy available in their EVCPs batteries

R9 The system must allow authenticated CPOs acquiring information from the DSOs about the current price of energy

R10 The system must allow authenticated CPOs deciding from which DSO to acquire energy from

R11 The system must dynamically decide where to get energy for charging (electrical grid, battery or a mixture)

R12 The system must allow authenticated CPOs statically deciding where to get energy for charging (electrical grid, battery or a mixture)

Domain Assumptions

D5 At least one DSO can always provide energy to the CPOs

Dep3 The system will require internet connection to interact with all the users

Dep4 The system will use an external API to retrieve the prices of energy by the available DSOs

3.2.6 Mapping on use cases

Use Case	Requirements
U1 - Login	R2 R16
U2 - Registration	R1 R15
U3 - Map Filtering	R18 R19
U4 - EVCP Details	R20
U5 - Book a Charge	R21 R22 R28
U6 - Start a Charge	R24 R27
U7 - Insert Car Details	R17
U8 - (EV Driver) History of Reservation	R26 R30
U9 - Details of the current charge	R26
U10 - Delete a reservation	R29
U11 - Send charging suggestion	R25
U12 - Monitor CPs	R4 R6 R7
U13 - Monitor Specific CP and Change Status	R14
U14 - Adding a CP	R13
U15 - Removing a CP	R13
U16 - Add a special offer and Add a Rate	R3
U17 - (CPO) View reservation	R8
U18 - Check energy in batteries	R5
U19 - Choose energy mix	R11 R12
U20 - Retrieve price of energy and Choose DSO	R9 R10

3.3 Performance Requirements

To guarantee the best possible experience to both EV drivers and CPOs, the system should:

- provide a scalable, reactive and capable of load-balancing backend
- assure protection against DDoS attacks
- provide a responsive and fluid frontend. The frontend must handle correctly asynchronous interactions with the server, even when the connection quality is bad, in order to provide the best possible experience
- push notification with a delay that is imperceptible to the user
- be able to handle a number of users that increases with the growth of the EV market. As shown in previous statistics in the Introduction, in 2021 the number of EV in use worldwide was about 16498750 units. About 6 million EV are on the roads in Europe. The system must be capable of handling initially at least 500000 users and 75000 peak concurrent users

3.4 Design Constraints

3.4.1 Standards compliance

Specifications described in this document must be respected by the system. Source code of the application must be commented on and documented adequately. The system should respect the guidelines described by the GDPR.

3.4.2 Hardware limitations

The system requires any device and a stable internet connection.

3.5 Software System Attributes

3.5.1 Reliability

In order to guarantee better reliability performances, all the scheduled maintenance of the system should be done during the night. It should also be noted that some functionality of the system relies on external APIs, though the system should not completely fail because of failure in one of those. It's also important to avoid data loss through redundant storage methods.

3.5.2 Availability

The system should offer its functionalities with an availability equal to 99.5%, or more. In other words, the system must be inaccessible for less than two days every year. To achieve this goal, the system should provide a high redundancy for the most critical components.

3.5.3 Security

To protect users sensible data the system must collect the minimum data possible, and store them in an encrypted database. The connection between the application must follows modern standard protocols as HTTPS. Do note that not only phone number and password should be considered as sensible data, but also users location, EV details and reservations. Moreover, all passwords must be salted and encrypted.

3.5.4 Maintainability

Source code and correlated documentation must be commented and kept updated. Modularity, low coupling and high cohesion between components must be a focus during the designing and developing phases. In particular modularity of the frontend and backend must allow developers to update the backend without users even noticing it.

3.5.5 Portability

The application must be available on major mobile OSs targeting at least the current release. The web application must run at least on the most recent browsers. It would be a smart choice to reuse most of the codebase across mobile and web by using cross platform tools.

Summary of Non Functional Requirements

NFR1	The backend of the system must be scalable and flexible (e.g. allocate resources based on the users to be served) to support a variable number of users over time
NFR2	The frontend of the system must be responsive, fluid and easy to use
NFR3	The system must respect the guidelines of the GDPR
NFR4	The availability of the system must be greater or equal to 99.5%
NFR5	The system must collect the minimum amount of data possible and encrypt those data
NFR6	The system must be easily maintainable
NFR7	The system must be usable on modern browsers and major mobile OSs

4 Formal Analysis Using Alloy

Alloy is a specification language for describing, designing, and verifying the behavior of complex systems. The language is based on first-order logic, and has a mathematical foundation that allows for automated reasoning about the correctness of designs. Once a model has been defined, it will be analyzed using a tool that supports Alloy, such as AlloyAnalyzer. This tool can automatically check whether the model satisfies the specified constraints and properties, and can also be used to explore the space of possible behaviors to find designs that meet the desired criteria.

4.1 Objectives of the analysis

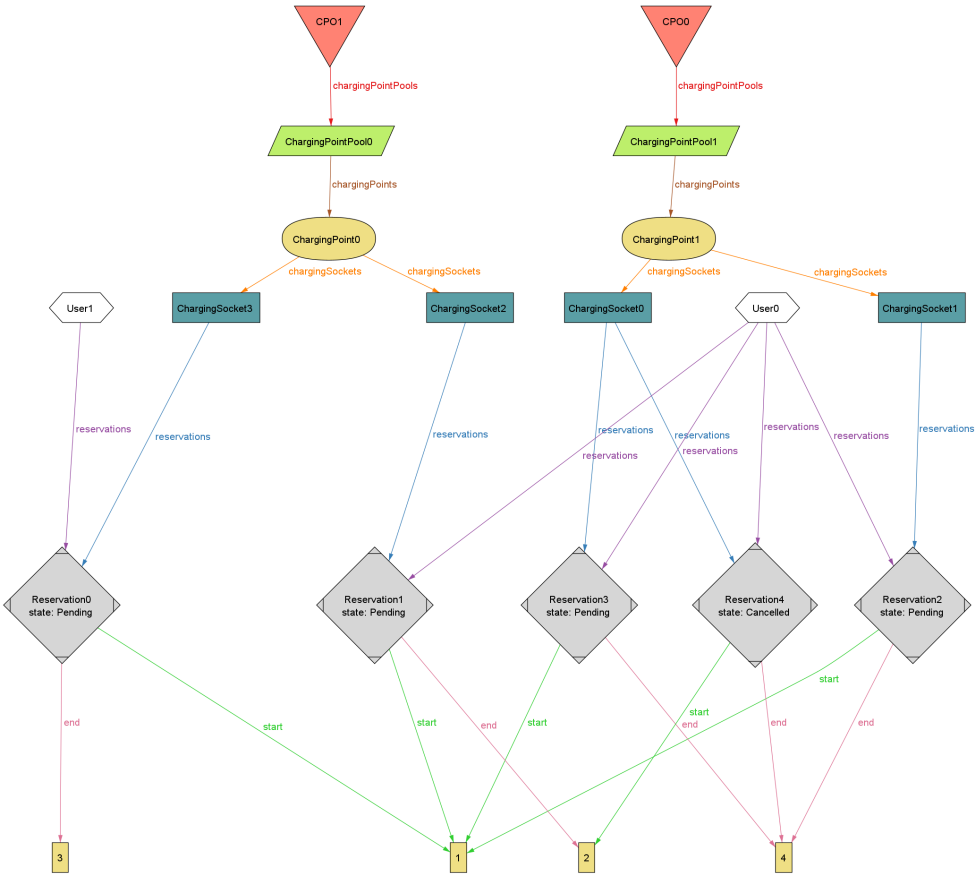
The main goal of the formal analysis is to formally describe the domain and properties of the system to be.

4.1.1 Reservation Model

The reservation of a CP is a core part of the System to Be. The Alloy model for verifying reservations of charging points includes abstractions for the entities involved in a reservation, such as charging points, charging sockets and reservation times. It also includes constraints on the relationships between these entities, such as the fact that a charging socket can be reserved by one vehicle at a time, or that a vehicle can be charged by one charging socket at a time.

```
// a charging station with multiple charging points
sig ChargingPointPool {
        chargingPoints: some ChargingPoint
}

// a single charging column with 1 or 2 connector
sig ChargingPoint {
        chargingSockets: some ChargingSocket
}
{
        #chargingSockets <= 2
}

sig ChargingSocket {
        reservations: set Reservation
}

abstract sig State {}
one sig Cancelled, Pending extends State {}

sig Reservation {
        state: one State,
    start: one Int,
    end: one Int
}{
        start > 0;
        end > start
}

// an operators: has 0 or more CP stations
sig CPO {
        chargingPointPools: set ChargingPointPool
}

// an EV Driver
sig User {
    reservations: set Reservation
}

// all charging point pools are managed by a CPO
fact EVCPofCPO{
        all evcp: ChargingPointPool | one cpo: CPO | evcp in cpo.chargingPointPools
}

// all charging point are in a charging point pool (and only in one)
fact CPinPool {
        all cp: ChargingPoint | one evcp: ChargingPointPool | cp in evcp.chargingPoints
```

```
}

// all charging socket are in a charging point (and only in one)
fact SocketinChargingPoint {
        all s: ChargingSocket | one cp: ChargingPoint | s in cp.chargingSockets
}

// each reservation is for one and only one charging socket
fact reservationSocket {
        all r: Reservation | one cp: ChargingSocket | r in cp.reservations
}

// each reservation is done by a user
fact reservationByUser {
        all r: Reservation | one u: User | r in u.reservations
}

// No two overlapping reservation for the same socket in PENDING state
fact noOverlappingReservation {
        all s: ChargingSocket | no disjoint r1, r2: s.reservations |
        (r1.state = Pending and r2.state = r1.state and r2.start >= r1.start and r2.start < r1.end)
}

pred showReservationModel {
        #Reservation = 5
        #User = 2
        #CPO = 2
}

// if two reservations on the same charging socket overlap then one of them has been cancelled
assert reservationCancelledIfOverlapping {
        all s: ChargingSocket | all disjoint r1, r2: s.reservations |
        (r2.start >= r1.start and r2.start < r1.end)
                                implies
        (r1.state = Cancelled or r2.state = Cancelled)
}

check reservationCancelledIfOverlapping for 35
run showReservationModel for 10
```




```
Executing "Check reservationCancelledIfOverlapping for 35"
   Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 Mode=batch
   221358 vars. 7630 primary vars. 451292 clauses. 6894ms.
   No counterexample found. Assertion may be valid. 709ms.
```

5 Effort Spent

Pair working

Project Specification Analysis, Brainstorming, RASD structure	**4h**
Study of the context and existing protocols	**5h**
Goals and Phenomena elicitation	**3h**
Use cases diagrams and description	**7h**
Sequence diagrams	**8h**
Final Review	**5h**

Giovanni

Definitions, Acronyms	**1.5h**
Assumptions, dependencies	**2.5h**
Mapping on goals, on use cases	**3h**
Non-functional requirements	**3.5h**
Formal Analysis with Alloy	**8h**
Review: traceability	**1.5h**
Review: use cases exceptions	**1.5h**

Matteo

User characteristics
Product Functions
Hardware Interfaces, Software Interfaces
Performance Requirements, Standards Compliance, Hardware Limitations, Reliability, Availability, Sec
UML

Lorenzo

Scenarios	**3h**
Class Diagram	**1h**
State Diagrams	**2.5h**
User Interfaces	**17h**
Review	**4h**

6 References

- Software Engineering II course slides
- Use Case Diagrams were made using draw.io
  `https://app.diagrams.net/`
- All other diagrams were made using StarUML
  `https://staruml.io/`
- Alloy Analyzer
  `https://github.com/AlloyTools/org.alloytools.alloy`
- OCPP Protocol
  `https://www.openchargealliance.org/protocols/ocpp-201/`
- OCPI Protocol
  `https://evroaming.org/ocpi-background/`