AY 2022/2023

Politecnico di Milano

RASD: Requirement Analysis and Specification Document

Alessandro Pignati Federico Sarrocco Alessandro Vacca

Professor
Matteo Camilli

Version 1.0
21 dicembre 2022

Indice

1 Introduction **1**
- 1.1 Purpose . . . 1
  - 1.1.1 Goals . . . 2
- 1.2 Scope . . . 2
  - 1.2.1 Phenomena . . . 2
- 1.3 Definitions, Acronyms, Abbrevations . . . 3
  - 1.3.1 Definitions . . . 3
  - 1.3.2 Acronyms . . . 4
  - 1.3.3 Abbreviations . . . 4
- 1.4 Revision History . . . 4
- 1.5 Reference Documents . . . 4
- 1.6 Document Structure . . . 5

2 Overall Description **6**
- 2.1 Product Perspective . . . 6
  - 2.1.1 Scenarios . . . 6
  - 2.1.2 Class Diagram . . . 7
- 2.2 Product Functions . . . 8
  - 2.2.1 Sign up . . . 8
  - 2.2.2 View the nearby charging stations . . . 9
  - 2.2.3 Charge booking . . . 9
  - 2.2.4 Remotely unlocking a socket . . . 10
  - 2.2.5 Remotely starting a charge . . . 10
  - 2.2.6 Notify of a finished charge . . . 11
  - 2.2.7 Pay for the service . . . 11
  - 2.2.8 Change active vehicle for the suggestions . . . 11
  - 2.2.9 Send suggestions . . . 12
  - 2.2.10 Select the charging station's energy provider . . . 12
  - 2.2.11 Select a battery policy for a charging station . . . 13
  - 2.2.12 Select the charging station's charging cost . . . 13
- 2.3 User Characteristics . . . 14
  - 2.3.1 Unregistered Customer . . . 14
  - 2.3.2 Registered Customer . . . 14
  - 2.3.3 Charging Point Operator . . . 14
- 2.4 Assumptions, Dependencies and Constraints . . . 14
  - 2.4.1 Domain assumptions . . . 14

3 Specific Requirements **16**
- 3.1 External Interface Requirements . . . 16
  - 3.1.1 User Interfaces . . . 16
  - 3.1.2 Hardware Interfaces . . . 16
  - 3.1.3 Communication Interfaces . . . 16
- 3.2 Functional Requirements . . . 16

3.2.1 Use case diagrams 16
3.2.2 Use Case Tables and Sequence Diagrams 19
3.2.3 Requirements 39
3.2.4 Traceability Matrix 42
3.3 Performance Requirements 42
3.4 Design Constraints 42
3.4.1 Standards Compliance 42
3.4.2 Hardware Limitations 42
3.5 Software System Attributes 43

4 Formal Analysis **44**
4.1 Alloy code 44
4.2 Worlds 49

5 Effort spent **52**

1 Introduction

eMall (e-Mobility for All) is an easy-to-use application which intent is to help the user to recharge their electric vehicle in order to reduce our carbon footprint. Users need an application whose main intent is to plan the charging process of an electric vehicle, thus reducing the interference and constraints on our daily schedule. Charging Point Operators need an application to manage their charging stations.

1.1 Purpose

The aim of the product is to simplify the process of electric vehicle charging, improving the customers' experience. Moreover, the Charging Point Operators processes will be facilitated. The experience will be enhanced because many aspects of the electric vehicle charging will be integrated, and they will be located within a single service.

The application should provide two types of accesses for Customers (people who need to charge their vehicles) and Charging Point Operators.

Customers will have the possibility to see which charging stations are available nearby and to also know the charging cost and if they have special offers. They will also have the opportunity to book a charge at a station at a certain timeframe in advance and to start the charging process remotely. They will be notified when the process has finished and the car is fully charged, and they will be able to pay from the application. Moreover, the eMall service has a smart system that suggests to the customer the optimal charging solution, based on their schedule, the charging cost and the current state of charge. On the other hand Charging Point Operators will be able to select energy options and set special offers.
The eMall platform will be composed of two modules, and thus the whole system will be developed modularly:

- **Module 1: e-Mobility Service Provider subsystem:**
  This part of the system will handle all the features regarding a customer, that will be interacting directly with this subsystem.
- **Module 2: Charging Point Management System subsystem:**
  This part of the system will handle all the features regarding a single or multiple charging stations. e-Mobilty Service Providers and check for socket availability and other various informations. Module 1 will interact with this module in order to check for socket availability and to actually start the charge. The Charging Point Operators will interact with the module to manually configure the stations options.

1.1.1 Goals

- **G1: Allow customers to obtain information about nearby charging stations**
  The customer can view information about any nearby charging station, such as charging cost and about special offers, availability of every type of charging socket, and if a certain type of socket is occupied at the time, the estimated time for the first to be free.
- **G2: Allow customers to book a charge for a certain timeframe**
  The customer can select any available charging station and book the charging for the next 15 minutes, if available.
- **G3: Allow customers to start the charge**
  The customer can remotely start the charge once the electric car has been connected to the station's socket, that must be manually unlocked from the service.
- **G4: Allow customers to know when the charging has finished**
  The customer will be notified by the application when the charging of his vehicle has been completed.
- **G5: Allow customers to pay for the charging service**
  The customer has the option to pay directly from the eMall application for the obtained service.
- **G6: Allow customers to receive suggestions on where to charge**
  The customer can receive suggestions from eMall on the optimal station to charge, based on his schedule, his vehicle's state of charge and the stations' prices.
- **G7: Allow Charging Point Operators to decide the energy acquisition options**
  The Charging Point Operator will have the possibility to choose whether to acquire energy automatically or to acquire it manually from a specific energy provider. Another option will be to store or use the energy from the batteries if they are present.
- **G8: Allow Charging Point Operators to dynamically choose the charging cost and to set offers**
  The Charging Point Operator will have the possibility to select the charging cost price based on the energy providers prices. They will also be able to set special offers based on the energy providers prices.

1.2 Scope

1.2.1 Phenomena

According to the paper "The World and the Machine" by M. Jackson and P. Zave, we can identify the application domains. The following table describes

the world, shared and the machine phenomena, including the reference to which part controls the phenomena.

Phenomenon	Controller	Shared
Customer charges his vehicle at a charging station	W	N
Customer books a charge at a charging station	W	Y
Customer connects the vehicle at the charging station	W	N
Customer receives QR code for his booking	M	Y
Customer authenticates with a QR code at the station at the selected timeslot	M	Y
Customer remotely starts the charge	W	Y
Customer registration	M	Y
Customer login	W	Y
Customer views a charging station's information	W	Y
Customer pays for the charge	W	N
Customer receives advice on the optimal place to charge	W	Y
Customer is notified when the charge has been completed	W	Y
Charging Point Operator chooses from which energy provider to acquire energy	W	Y
Charging Point Operator decides to charge the station's battery	W	Y
Charging Point Operator select the cost for charging	W	Y
Charging Point Operator login	W	Y

1.3 Definitions, Acronyms, Abbrevations

1.3.1 Definitions

- **Customers:** The people whom this service is directed. They can belong to any age and gender. Their goal is to efficiently charge their electric vehicle.
- **Charging stations:** Places that offer the service of electric vehicle recharging.
- **e-Mobility Service Providers:** The companies that offer the service of charging at different stations.
- **Charging Point Operators:** The companies that manage charging stations (one or more).
- **Distribution System Operators:** The charging station's energy providers.
- **Charging Point Managment System:** The single CPO's system to manage all the IT infrastructure.
- **Notification:** It's an alert that a certain event occurred. This alert can be a "Push Notification" on the smartphone, an SMS, an email and so on.

- **Push Notification:** It's an automated message sent by an application to a user when the application is not running.

1.3.2 Acronyms

- **eMall:** e-Mobility for all
- **eMSP:** e-Mobility Service Provider
- **CPO:** Charging Point Operator
- **DSO:** Distribution System Operator
- **CPMS:** Charging Point Management System
- **API:** Application Programming Interface
- **UML:** Unified Modeling Language

1.3.3 Abbreviations

- **ID:** Identifier. It's a generally unique sequence of numbers or letters in order to unambiguously identify an entity.
- **Gn:** Goal number n
- **Dn:** Domain assumption number n
- **Rn:** Requirement number n

1.4 Revision History

- December 21, 2022: version 1.0, initial release

1.5 Reference Documents

- Specification document: "R&DD Assignment A.Y. 2022-2023"
- Alloy official documentation: `https://alloytools.org/documentation.html`
- Paper: "Jackson and Zave: the world and the machine"
- UML official specification `http://www.omg.org/spec/UML/`
- BPMN official specification `http://www.omg.org/spec//BPMN/index.html`

1.6 Document Structure

- **Section 1: Introduction**
  This section offers a brief description of the problem and required functionalities. It also contains the list of definitions, acronyms and abbreviations that will be found in this document. Finally, there is the version history of the document, containing the revisions list and their content, and document structure, which describes the main purposes of the sections of this document.

- **Section 2: Overall Description**
  This section shows the possible scenarios of interaction by the user with the system. It also offers a summarized description about the overall organization of the system, the hardware and software constraints and the interfaces needed to get it to work. It also contains a description of all the features offered by the application, and of the actors who use it.

- **Section 3: Specific Requirements**
  This section contains system requirements specification. It includes functional requirements described through some scenarios, use cases and sequence diagrams. In this section non-functional requirements are specified too. Moreover, requirements are mapped to the goals of the system.

- **Section 4: Formal Analysis through Alloy**
  This section contains the description of the analysis' objective and the formal analysis with the use of Alloy.

- **Section 5: Effort Spent**
  This section presents the total effort spent by the project's members.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

1. **Customer wants to start using the eMall service**
   The customer Matteo decides that he wants to use the eMall service to take advantage of the efficiency it offers. He launches the service and selects the option to sign up. Matteo then inputs all the relevant information and he is granted access to the service.

2. **Customer books the recharge for his car**
   After a long drive, the customer Matteo notices that the level of his battery is low. He stops, takes his phone and launches the eMall service, logs in, and then he is presented with all the nearby stations. He selects different stations, checks their availability and views all the stations' information. The Customer then chooses a station according to his preferences and then finalizes the booking. He then receives the booking's summary, including the booked socket. He is also presented with a reminder that the booking will be kept for 15 minutes at most.

3. **Customer starts the charge**
   The customer Matteo drives to a station that he has a charge booked with, he then unlocks the booked socket from the service, he then connects his electric vehicle to the socket and, after he has left the vehicle, he launches the eMall service, navigates to his current booking, and he is presented with the option to start the charge. He selects this option and the station initiates charging on the booked socket. eMall than shows the user the estimated time for the full charge of the vehicle.

4. **Customer is notified of a finished charge**
   After the customer Matteo has started to charging process at the station, he goes to a nearby cafeteria to have a cup of coffee. When the car has fully charge, the eMall service notifies him on his phone with a push notification.

5. **Customer pays for the service**
   The customer Matteo, after he has used the eMall service for a successful charge and provided that he has not paid for the service yet, launches the app and he is presented with the option to pay for the service, he selects this option and then enters his credit card information. The service that processes his payment, and if it is successful he is presented with a success message. The application will now mark that booking as paid.

6. **Customer is reminded to charge his electric vehicle**
   While the customer Matteo is driving, the eMall service monitors the car's battery level, the customer's location and his schedule. When the

service finds an optimal place to charge, related to schedule, location and state of charge it notifies the customer with a push notification. Matteo then stops, launches the eMall service and then he is presented with the optimal station to charge, based on his location, his schedule and the stations' prices. He then books the charge.

7. **CPO chooses from which DSO acquire energy**
   The CPO Evox wants to modify the DSO energy source of a charging station. The CPO's employee Mario launches the eMall service, logs in, and then through the dashboard he is able to select his first station and then selects the DSO MaDistribution from which to acquire energy.

8. **CPO chooses the cost of a charging**
   The CPO Evox decides to modify the cost of a charging type in a specific station. The CPO's employee Mario launches the eMall service, logs in, and then using the dashboard selects the first station and then the charging type "fast", he then sets the charging price to 0.60/kWh.

9. **CPO selects automatic cost calculation of a charging**
   The CPO Evox wants to set the automatic vehicle charging cost calculation. The CPO's employee Mario launches the eMall service, logs in, and then using the dashboard selects the first station and then the charging type "slow", he then set the charging price to automatic.

10. **CPO selects the battery policy discharge**
    The CPO Evox wants to change the battery usage policy, he wants to take half of the energy from the batteries and the other half from the DSO. The CPO's employee Mario launches the eMall service, logs in, and then using the dashboard select the second charging station and then set battery policy to discharge and the discharge factor to 50%.

11. **CPO select the battery policy disabled**
    The CPO Evox wants to disable the usage of batteries, neither to charge them neither to use them to charge the vehicles. The CPO's employee Mario launches the eMall service, logs in, and then using the dashboard selects the third charging station, and then set the battery policy to disabled.

12. **CPO decides to make a special offer**
    The CPO Evox to promote his service decides to make a special 10% discount on the price of fast charging in his second station. The CPO's employee Mario launches the eMall service, logs in, and then using the dashboard selects the station and the charging type. He set the offer percentage to 10%

2.1.2 Class Diagram

This is the Class Diagram of the system. We consider the suggestion, that it's sent to the Customer via push notification, as a prepared booking. For this

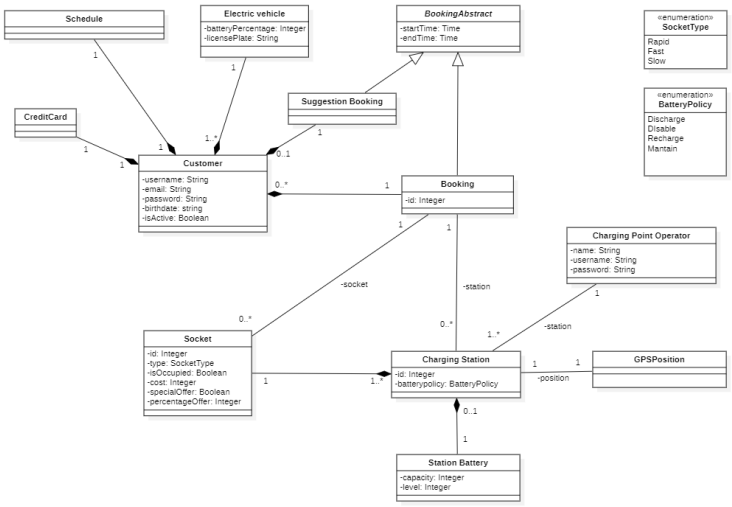


Figura 1: High-level UML diagram with main classes

reason it inherits from the abstract class "BookingAbstract". The customer has to choose only if he wants to accept or not the suggested booking. Assuming that every socket has a unique ID in the single station and charging stations have a unique ID too (w.r.t Domain Assumptions), the "Booking" class is associated both to the "Charging Station" and the "Socket" classes because it would be impossible to know the correspondent Charging station directly from the booked socket.

2.2 Product Functions

In this section the main functionalities of eMall are presented and described.

2.2.1 Sign up

This functionality lets the registration of **Unregistered Customers** (1.3.1) in order to use eMall. The first step is to enter the registration form, then the User inserts his credentials; if correct an e-mail is sent to the inserted address with a link to validate it. Finally, the User is taken to the login page.

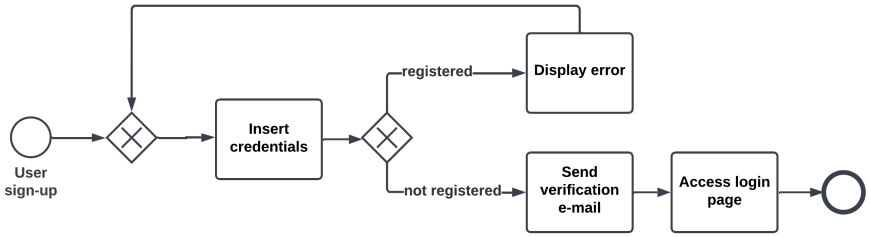


Figura 2: BPMN diagram sign up method

2.2.2 View the nearby charging stations

This functionality allows the **Customers** (1.3.1) to view information about the nearby charging stations, and is available to all the logged Users. The first step is to open the eMall service and the User is presented with the nearby charging stations. If he clicks on one of the stations, the user can view the current station's recharge price and socket availability. The User at any time can go back to view the other stations and he is free to view information about them.

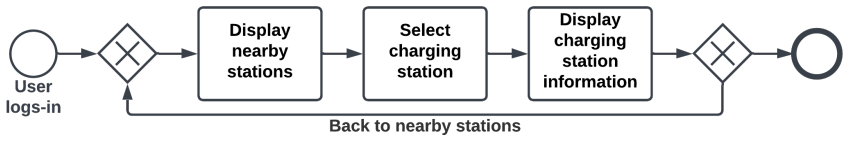


Figura 3: BPMN diagram view nearby stations method

2.2.3 Charge booking

This functionality allows the **Customers** (1.3.1) to book a charge for their electric vehicle, and is available to all of the logged Users. The first step is to open the eMall service and to log in. The second step is to pick a station from the stations view (As described in 2.2.2). Finally, the User can select the "Book" option to reserve a specific type of socket for 15 minutes at the selected station. He then is presented with a brief summary of the booking, like the socket's identification and the station's address. He can also view the summary at anytime in a specific section of the service.

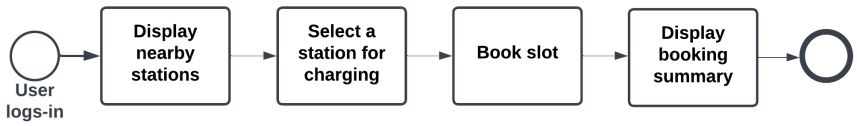


Figura 4: BPMN diagram charge booking method

2.2.4 Remotely unlocking a socket

This functionality allows the **Customers** (1.3.1) to unlock a charging socket to charge their electric vehicle, and is available when a logged User has booked a recharge at a station. The User must access the brief summary of his booking (As described in 2.2.3). He can select the "Unlock" option to unlock his booked socket, in order to connect his car to the socket.

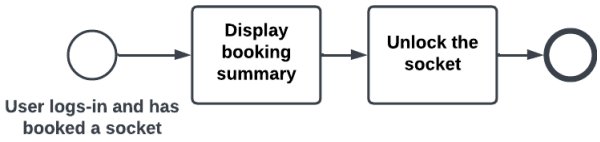


Figura 5: BPMN diagram socket unlocking method

2.2.5 Remotely starting a charge

This functionality allows the **Customers** (1.3.1) to remotely start the charge for their electric vehicle, and is available when a logged User has booked a recharge at a station, and has connected his car to the socket. The User must access the brief summary of his booking (As described in 2.2.3). He can select the "Start charging" button to remotely start the charging process on his booked socket.

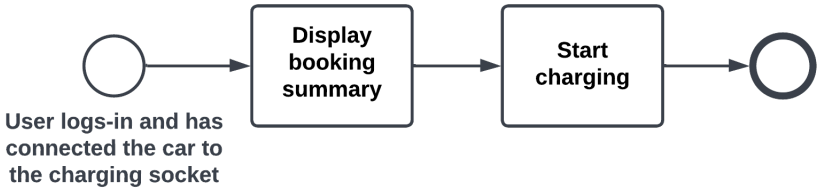


Figura 6: BPMN diagram remotely starting the charge method

2.2.6 Notify of a finished charge

This functionality allows the **Customers** (1.3.1) to be informed when the charging process has finished, and is available when a logged User has been charging his car at a station. When the charging process ends, the user is notified (with a push notification) that his car is fully charged.

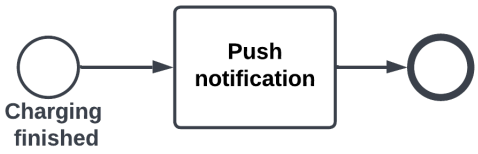


Figura 7: BPMN diagram notify a finished charge method

2.2.7 Pay for the service

This functionality allows the **Customers** (1.3.1) to pay for the charging service, provided that he has used the service. When the charging process ends, the User can open the brief summary of his reservation (As described in 2.2.3) in order to pay for the service. After the opens the summary, he is presented with the option to pay and if it picked, he should enter his credit card's information and click the "Submit" button. If the payment has been successful, the service shows him the success screen and the summary of the reservation will be marked as paid, otherwise he is returned to enter a valid credit card's information.

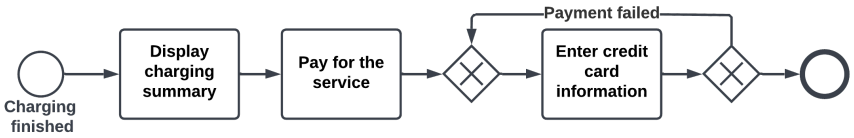


Figura 8: BPMN diagram pay for the service method

2.2.8 Change active vehicle for the suggestions

This functionality allows the **Customers** (1.3.1) to change the active vehicle for suggestions. The Customer must access the "Change Active Vehicle" submenu in the app. He can insert the license plate and clicks "Submit". The system then verifies that the vehicle is actually owner by the current Customer, and then returns a success message. If the process is not successful, an error message is shown.

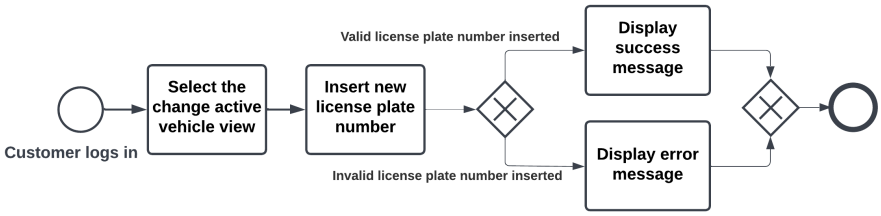


Figura 9: BPMN diagram change active vehicle method

2.2.9 Send suggestions

This functionality allows the **Customer** (1.3.1) to receive optimal suggestions for charging. The system continuously monitors a user's charging level, his schedule and his location. The system has an internal model to determine an optimal place to charge if the battery level is under 50%, based on user location and schedule. The Customer will then be notified at the computed time with a push notification. The next time the user opens the app, the system will check if the Customer's current location is less than 10 km far from the charging station, and it will also check if the station has still available sockets of the type of the suggestion. If both conditions are true, the system will present the Customer with a pre-made booking that he can only accept or discard. If the Customer accepts, the system will perform the booking automatically, equivalent to 2.2.3.

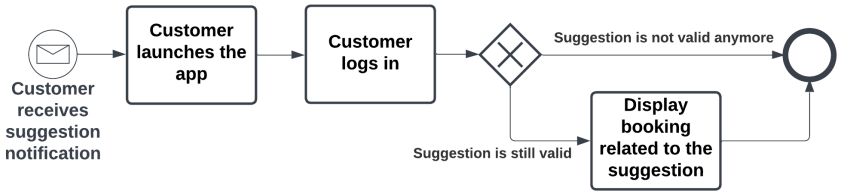


Figura 10: BPMN diagram send suggestions method

2.2.10 Select the charging station's energy provider

This functionality allows the **CPOs** (1.3.1) to manually adjust the energy provider. An energy provider can be select automatically, or a CPO can manually select a specific **DSO** (1.3.1) for the supply of energy. The first step for the CPO is to login in the platform and select a charging station. Then, the CPO can select an energy provider option for the station.

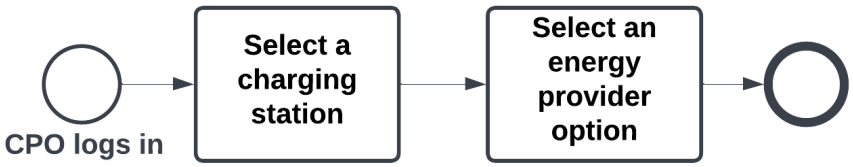


Figura 11: BPMN diagram select energy provider method

2.2.11 Select a battery policy for a charging station

This functionality allows the **CPOs** (1.3.1) to manualy select a battery policy for a charging station. A battery policy can be automatically select, or a charge, discharge and disable policy can be selected. The first step for the CPO is to log in in the platform and select a charging station. Then, the CPO can select a battery policy. If the discharge policy is selected, a CPO can also select a discharging factor, that will determine how much energy from the batteries will be mixed with the energy from the **DSOs** (1.3.1).

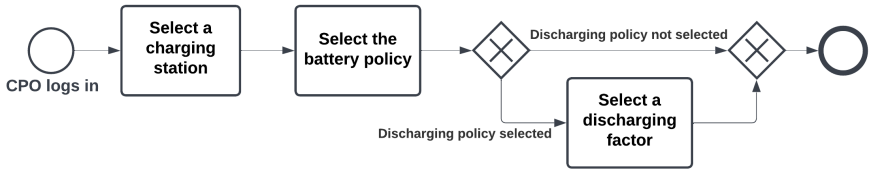


Figura 12: BPMN diagram select battery policy method

2.2.12 Select the charging station's charging cost

This functionality allows the **CPOs** (1.3.1) to manually adjust the charging price at a station. The first step for the CPO is to log in on the platform and to select a charging station. The second step is to select a specific charging type, such as slow, fast and rapid. Finally, the CPO can select a charging cost. Also, an option is present to apply a special offer (in percentage) on the current charging cost.

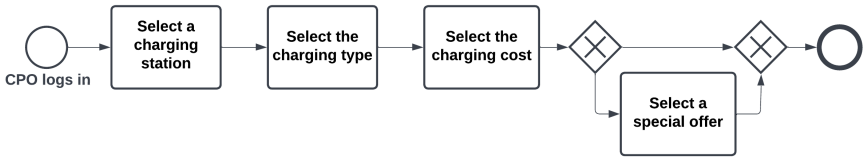


Figura 13: BPMN diagram send suggestions method

2.3 User Characteristics

The following three actors are considered in the e-Mall system.

2.3.1 Unregistered Customer

It's a person that doesn't have a registered account in the e-Mall application. In this case, he can register an account to use the platform.

2.3.2 Registered Customer

It's a person that has a registered account in the e-Mall application. He is able to use the e-Mall service and book a charge for his electric vehicle through the service. He is also able to unlock a charging socket for his reservation and to pay for the service through the platform.

2.3.3 Charging Point Operator

It's the charging station owner and manager. This actor is responsible for the supply of the charging service offered by an eMSP. Every reservation made on the e-Mall platform is fulfilled by CPOs.

2.4 Assumptions, Dependencies and Constraints

2.4.1 Domain assumptions

D1: Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.

D2: Each CPO who wants to access the CPMS platform needs to have a device connected to the Internet (such as PC, Mac, smartphone, etc), with the most common Web Browsers (e.g. Firefox, Google Chrome, Microsoft Edge, Apple Safari, etc).

D3: The customer's mobile device fully supports the push notification technology.

D4: The customer's schedule and location is accessible by the platform.

D5: The charging station checks the charging sockets, making sure that nobody can occupy a reserved socket apart from the booker.

D6: The data automatically obtained by the system in order to send suggestions to the customer is accurate and truthful.

D7: Charging stations have only a single CPMS system.

D8: A fully functioning payment system is present and returns if a transaction has been successful or not.

D9: The vehicle start charging only when it is connected to the booked socket.

D10: The socket notifies the CPMS when a vehicle is attached and ready to charge.

D11: Every CPO is supplied with login credentials when the system is installed.

D12: Every Charging Station has a unique identification number (ID).

D13: Sockets in each station have a unique identification number (ID). Sockets in different charging stations could have the same ID.

D14: Given a license plate number and personal identification, it exists an API that provides the battery level of the vehicle associated with the license plate.

D15: All the sockets present in charging stations feature a system that can retrieve the charging level of a car after it has been connected to the socket.

D16: The socket notifies the CPMS when a vehicle has finished to charge.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface of eMall is a mobile application, with an easy to use design. It should be available on every major mobile operating system. On the other hand, the interface of the CPMS subsystem is a web application with a dashboard with all the available features. It should be available on every major web browser.

3.1.2 Hardware Interfaces

The system is fully operated on the Internet, so it doesn't feature any hardware interface, apart from the device needed to access the service.

3.1.3 Communication Interfaces

The main communication interfaces of the platform are the users and CPOs. The system interacts with users to receive booking requests or to send information, and to CPOs to let them select different charging station options.

3.2 Functional Requirements

3.2.1 Use case diagrams

1. **Unregistered Customer**

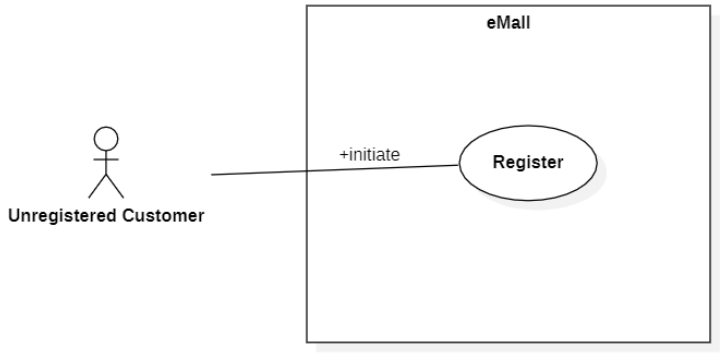


Figura 14: Unregistered Customer - Use Case Diagram

2. **Registered Customer**

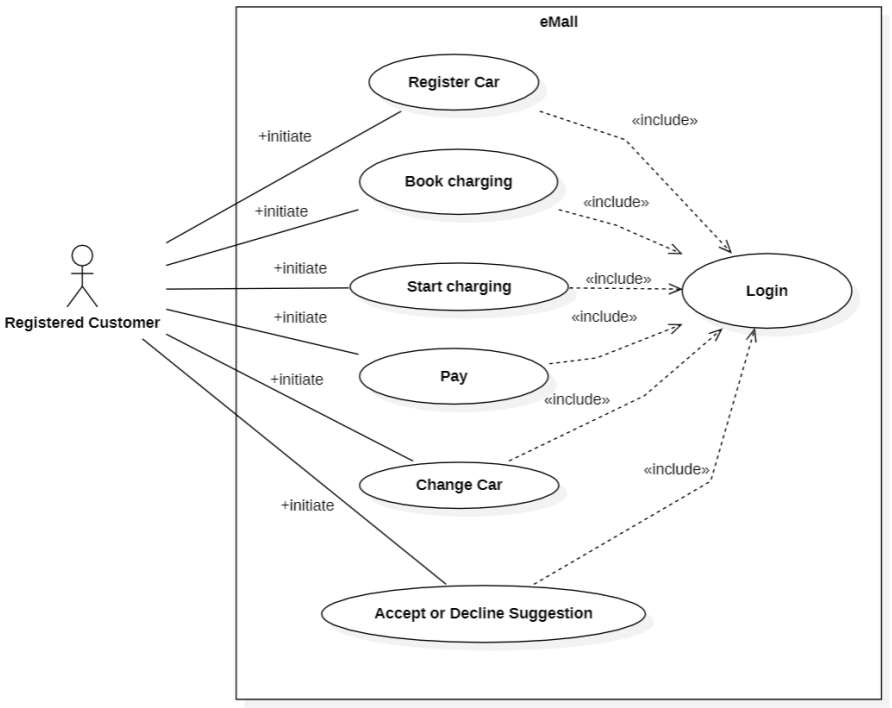


Figura 15: Registered Customer - Use Case Diagram

3. **Charging Point Operator**

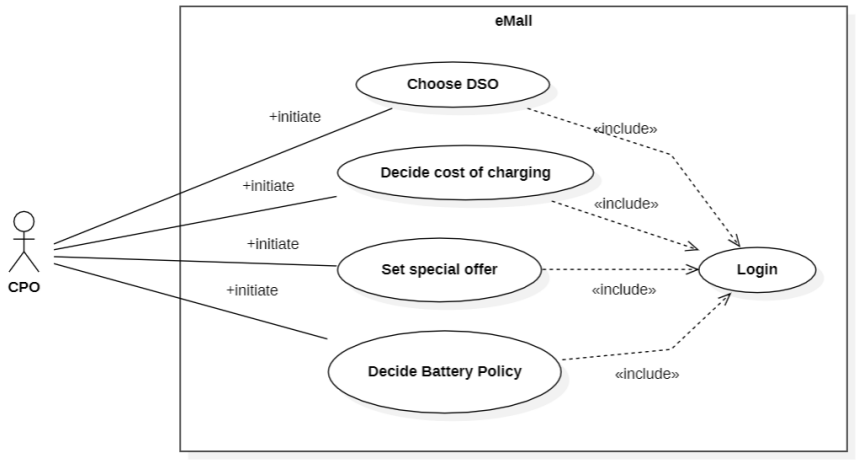


Figura 16: Charging Point Operator - Use Case Diagram

3.2.2 Use Case Tables and Sequence Diagrams

ID	1
Name	Sign Up Customer
Actor	Unregistered Customer
Entry conditions	Unregistered Customer has downloaded and opened the application on his smartphone
Input	• username • email • password • birthdate

Events flow	• The application displays the Sign In screen • Unregistered Customer clicks on "Sign Up" • The application displays a list of fields that User must compile: username, email, password, birthdate • Unregistered Customer inserts the mandatory data and accepts the "Terms of Services" • Unregistered Customer clicks on confirm button • The application displays the acceptance of registration and invites the Customer to go on his inbox in order to confirm the registration • Customer opens his inbox, checks the e-mails and clicks on confirmation link
Exit conditions	Customer registration has been successful: user data are stored in the database of the system. Customer can now Login with his credentials.
Output	• The email of the Customer is stored in the database of the application • The Customer receives the confirmation email
Exception 1	Customer inserts an e-mail which is already stored in the database. So, after he inserts his data and clicks on confirm, the application displays an error page which tells him that he is already registered to the service and invites him to login with that e-mail.
Exception 2	Customer inserts an invalid e-mail. So, after the Customer clicks on the confirm button, the application displays the same page and an error message, which suggests to the Customer to check the e-mail inserted or to change it.

Tabella 1: Sign Up Customer

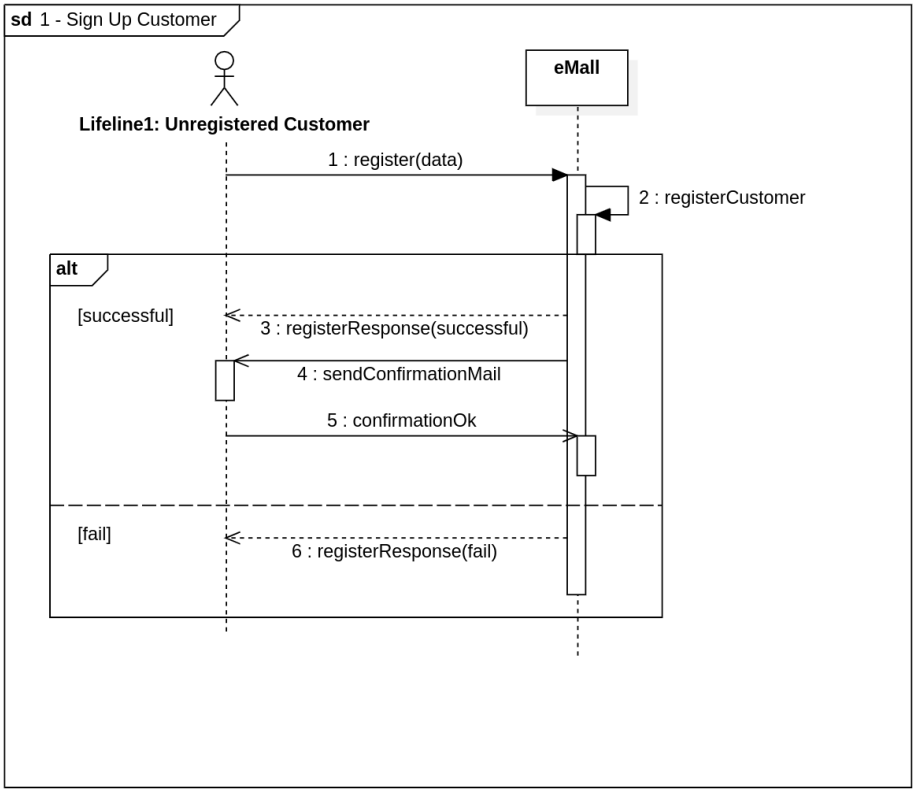


Figura 17: Sign Up Customer

ID	2
Name	Login Customer on the application
Actor	Customer
Entry conditions	• Customer has downloaded and opened the application on his smartphone • Customer has registered to the service
Input	Customer email and password associated to a valid registration
Events flow	• The system displays the Login page • Customer inserts in apposite fields the credentials for logging in and presses the Login button. • The system checks the correctness of the credential inserted. • The system displays the home page of the application.

Exit condition	User is logged in
Exception 1	Customer inserts a wrong combination of credentials and presses Login button. In this case, the system detects the error and the application displays the Login page with an error.

Tabella 2: Login Customer

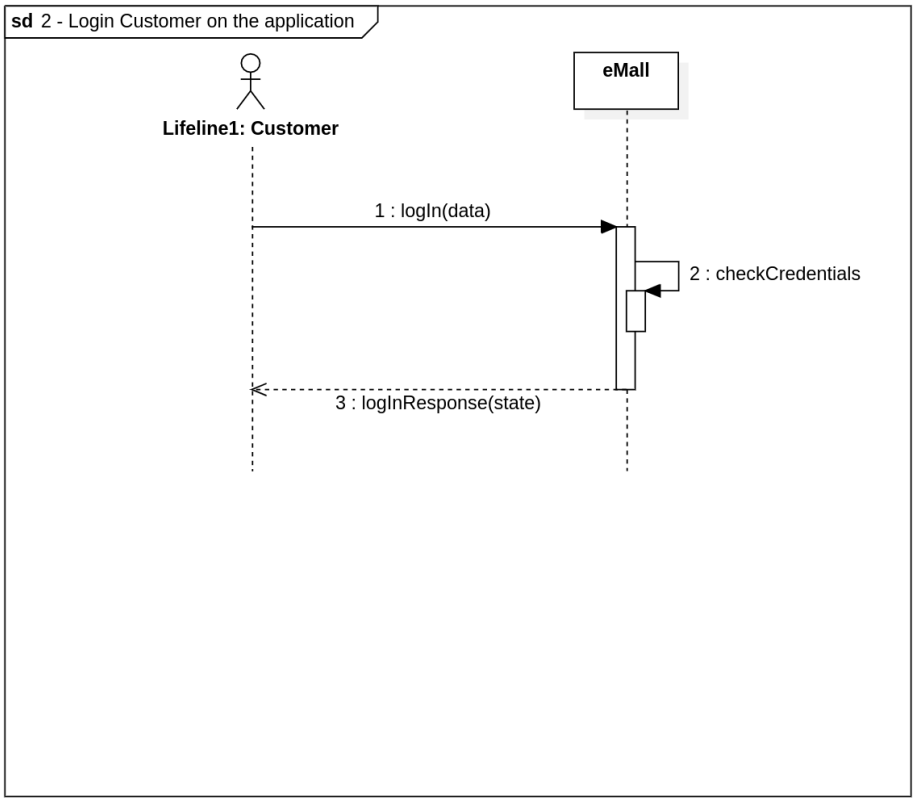


Figura 18: Login Customer

ID	3
Name	Book a Charge
Actor	Customer
Entry conditions	Customer has logged in

Events flow	• The application displays the home page, which is a map with all the nearby charging stations. • The Customer chooses a charging station. • The application displays all information about a charging station, including its current availability. • The Customer selects an available station and type of socket for the recharge and clicks the confirm button. • The application displays a confirm popup, with the essential details of the upcoming booking. • Customer clicks the confirm button.
Exit condition	The application displays a brief summary of the booking, including the unique ID number of the booking.
Output	• The system has the received the booking, and will mark the socket as booked for 15 minutes, and then either the booking is fulfilled by the Customer, or the system will mark the socket as available. • The Customer sees the summary in his bookings section, with a reminder that the booking only lasts 15 minutes.
Exception 1	Customer selects an unavailable charging station. In this case, the application displays an error message stating the situation, with the time for the first socket of the selected type to be available.
Exception 2	Customer presses on confirm button, but he has already booked a socket currently. In this case, the application displays an error page, and redirects the Customer to the home page.

Tabella 3: Book a charge

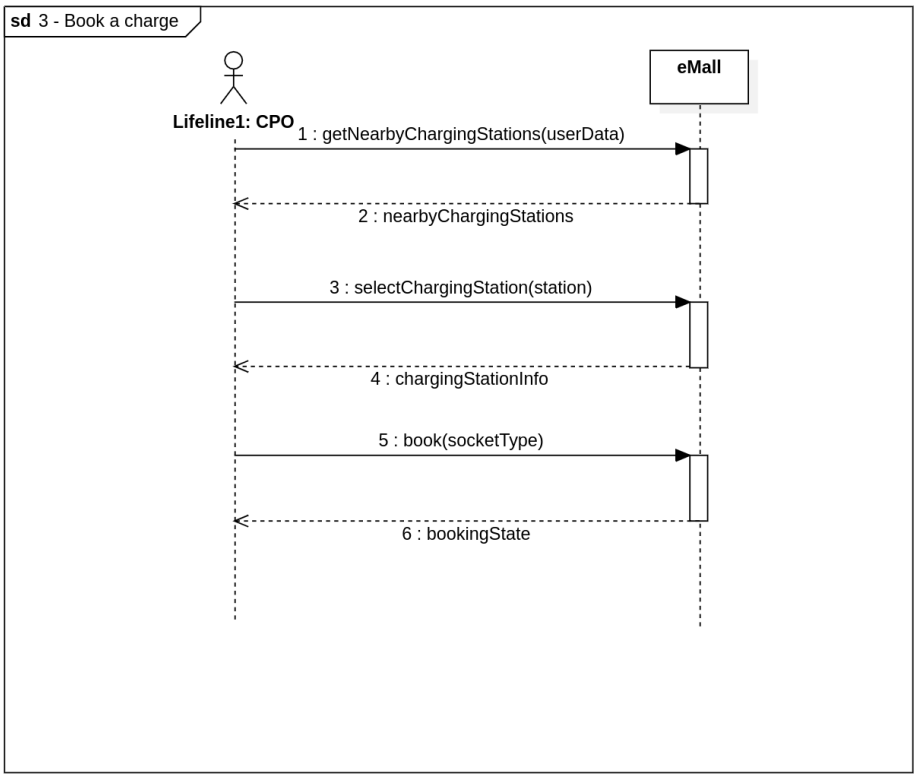


Figura 19: Book a charge

ID	4
Name	Remotely Start a Charge
Actor	Customer
Entry conditions	• Customer has booked a charge and has driven his car to the socket of the said booking in time. • Customer has opened the application and logged in.

Events flow	• The application displays a map with all the nearby charging stations. • The Customer selects the "Bookings" menu from the side menu. • The application displays all his current and previous bookings. • The Customer selects the booking related to current station, socket and time. • The application displays the brief summary of his booking, and also an "Unlock Socket" button. • Customer clicks the button, and then the application displays a confirmation message. • The Customer connects his electric vehicle to the socket. • The application displays an updated version of the summary of the booking with a "Start Charging" button. • The Customer clicks the said button.
Exit condition	The application displays a confirmation message with the estimated time of charge completion, and then redirects the user to the home page.
Output	The Customer's vehicle recharges.
Exception 1	Customer clicks on the "Start Charging" button while his vehicle is not connected. In this case, the application displays an error message that says the there isn't any vehicle to the socket, and redirects the Customer to the current booking view.

Tabella 4: Remotely start a charge

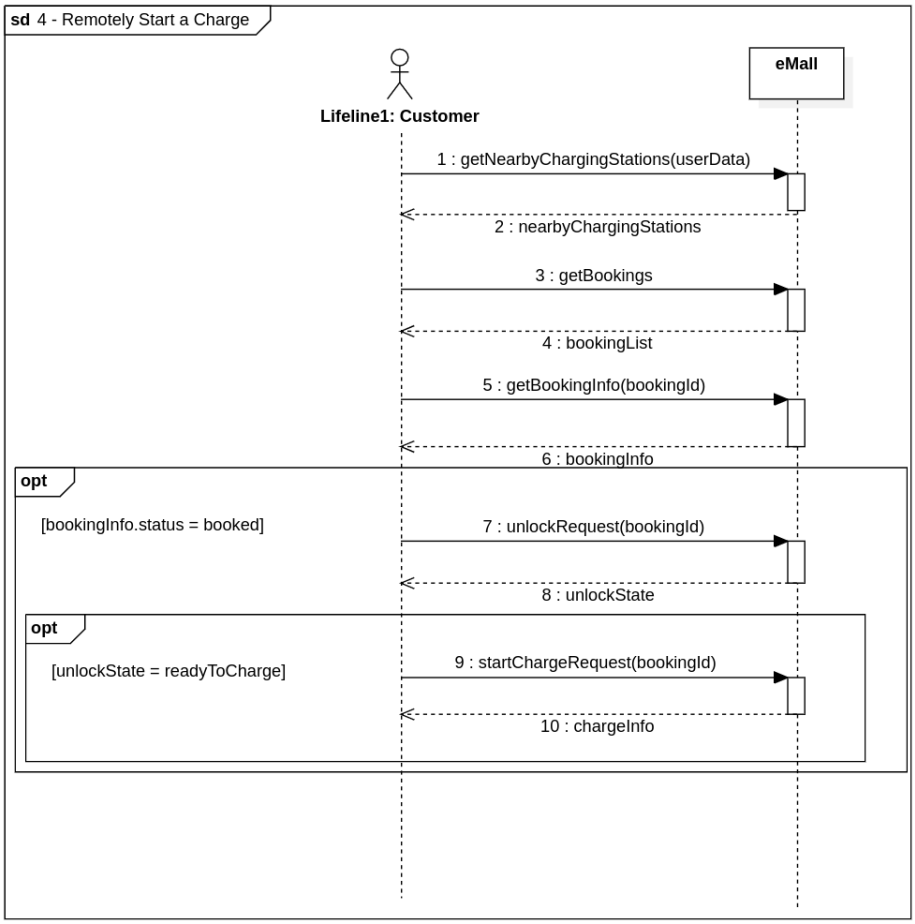


Figura 20: Remotely start a charge

ID	5
Name	Pay for the Charging Service
Actor	Customer
Entry conditions	• Customer has used the service for a charge. • Customer has opened the application and logged in.

Events flow	• The application displays a map with all the nearby charging stations. • The Customer selects the "Bookings" menu from the side menu. • The application displays all his current and previous bookings. • The Customer selects the booking related to the one that he wants to pay. • The application displays the brief summary of his booking, and also a "Pay" button. • Customer clicks the button, and then the application redirects him to an external payment platform, where he can input his preferred payment method and then pay.
Exit condition	The Customer will then be returned to the booking summary once the payment has been finished, with a success message.
Output	The Customer's booking has been paid.
Exception 1	Customer inputs an invalid payment method. In this case, the external services closes and the he is returned to the application. The application then displays an error message that says the payment method is invalid.

Tabella 5: Pay for the Charging Service

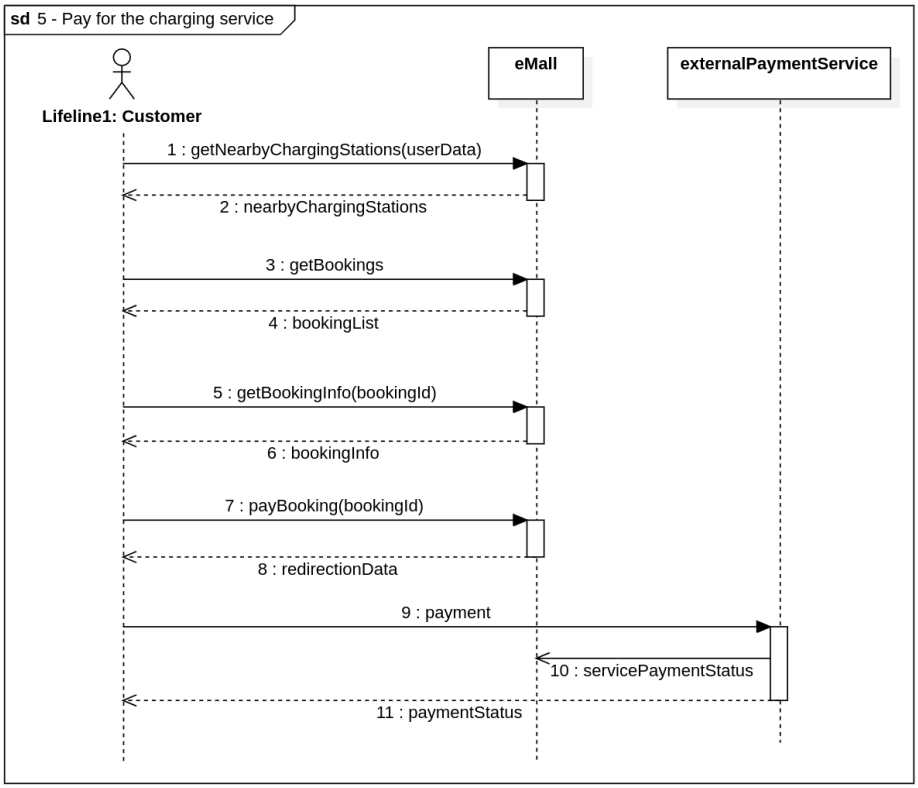


Figura 21: Pay for the Charging Service

ID	6
Name	View Suggestions
Actor	Customer
Entry conditions	• Customer has logged in. • Customer has received a push notification about the availability of a suggestion.
Events flow	• The application displays a summary of the suggestion with all the information about the booking to make. • The Customer selects the "Accept" option.
Exit condition	The application displays a brief summary of the booking, including the unique ID number of the booking.

Output	• The system has the received the booking, and will mark the socket as booked for 15 minutes, and then either the booking is fulfilled by the Customer, or the system will mark the socket as available. • The Customer sees the summary in his bookings section, with a reminder that the booking only lasts 15 minutes.
Exception 1	Customer logs in after some time since the notification. The system will check if his current position is close enough to the suggestion and if the type of socket of the suggestion is still available at the station. If it is not available, the system won't show the suggestion and the user will be directed to the home page.
Exception 2	The user selects the "Decline" option. The system will cancel the suggestion and redirect the user to the home page.

Tabella 6: View Suggestions

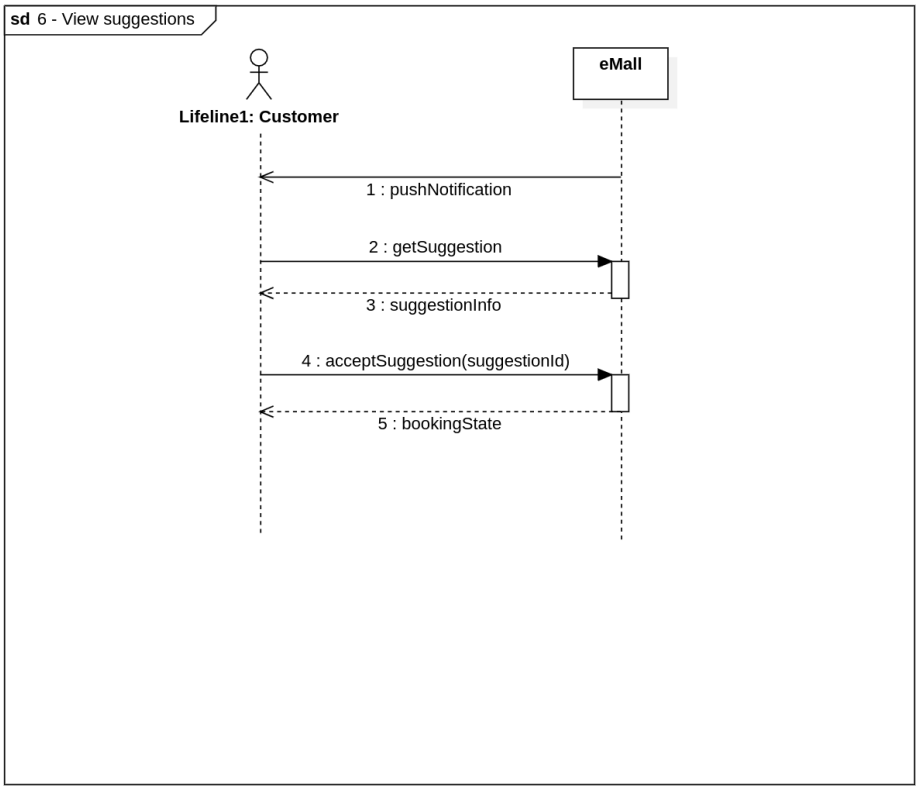


Figura 22: View Suggestions

ID	7
Name	CPO Login
Actor	CPO
Entry conditions	CPO has opened the web portal of his CPMS subsystem.
Input	The CPO credentials associated with a valid registration.
Events flow	• The website displays the Login screen. • The CPO inputs the credentials. • The system checks for the correctness of the credentials.
Exit condition	CPO is logged in.
Exception 1	CPO inserts a wrong combination of credentials and presses Login button. In this case, the system detects the error and the application displays the Login page with an error.

Tabella 7: CPO Login

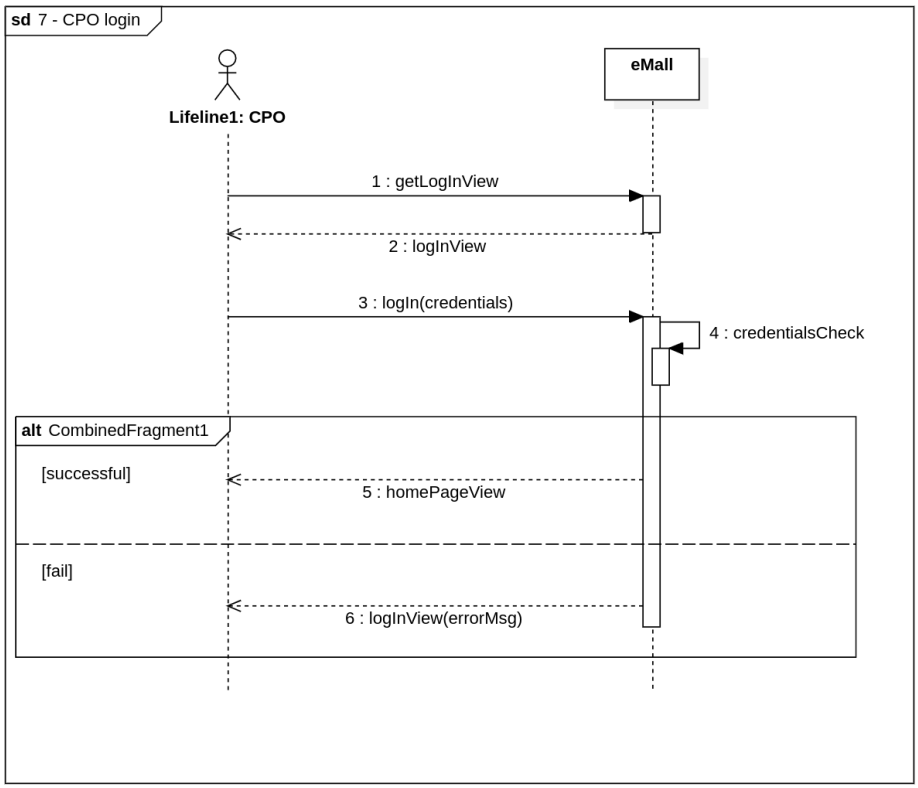


Figura 23: CPO Login

ID	8
Name	Select a DSO option
Actor	CPO
Entry conditions	CPO has logged in

Events flow	• The web portal displays the home page, which is a list with all the charging station that are owned by the current CPO. • The CPO chooses a charging station. • The portal displays the dashboard of the previously selected station, that features information about a charging station, including its current price, special offers, battery level (if present) and battery policy, with options to change them. • The CPO clicks on the "Update provider" button. • The application displays a popup with all the available DSOs and another option to select the DSO automatically. • The CPO picks his preferred option. • CPO clicks the "Confirm" button.
Exit condition	The application displays a confirmation popup, and after the CPO closes it, he is redirected to the view of the updated dashboard of his previously selected station.
Output	The system has the received the change in the DSO acquirement process, and the station will acquire the energy from the selected DSO. If the "Auto" option is selected, the system will automatically pick the cheapest DSO.
Exception 1	While updating the provider, the CPO clicks on the "Discard" button. He returns to the dashboard and no change is made.

Tabella 8: Select a DSO option

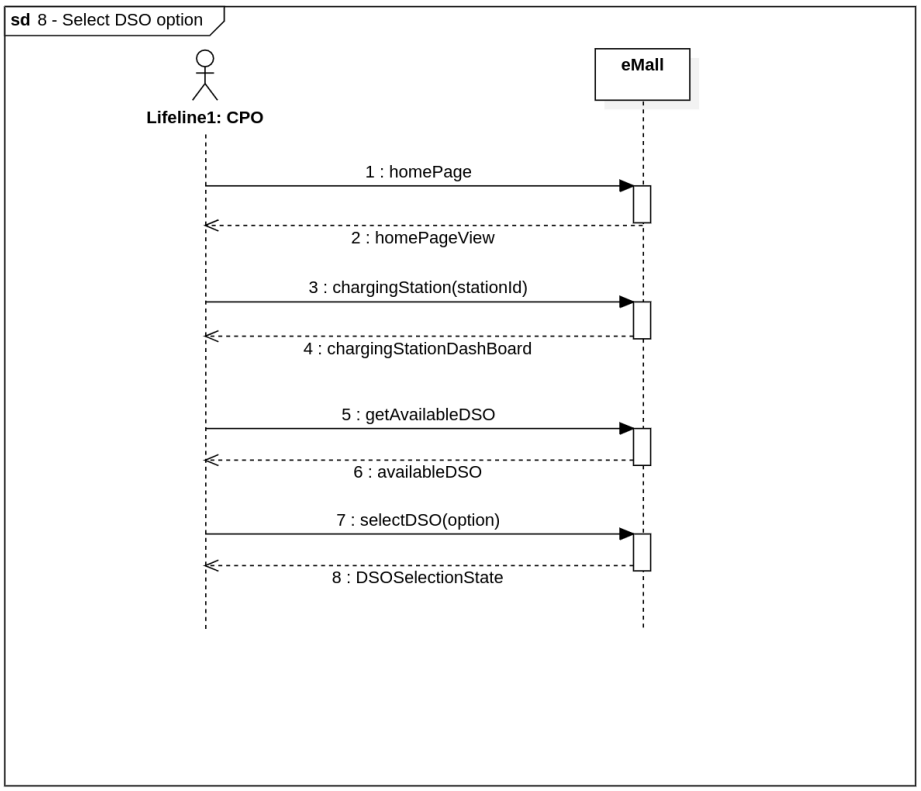


Figura 24: Select a DSO option

ID	9
Name	Change station energy price
Actor	CPO
Entry conditions	CPO has logged in

Events flow	• The web portal displays the home page, which is a list with all the charging station that are owned by the current CPO. • The CPO chooses a charging station. • The portal displays the dashboard of the previously selected station, that features information about a charging station, including its current price, special offers, battery level (if present) and battery policy, with options to change them. • The CPO clicks on the "Update price" button. • The application displays a popup where the user can input the desired price in /kWh. • CPO clicks the "Confirm" button.
Exit condition	The application displays a confirmation popup, and after the CPO closes it, he is redirected to the view of the updated dashboard of his previously selected station.
Output	The system has updated the station price with the new value.
Exception 1	While updating the price, the CPO clicks on the "Discard" button. He returns to the dashboard and no change is made.
Exception 2	If the user inputs an invalid price (such as negative or zero) and clicks the "Submit" button the portal will show an error message, and then he will be returned to the dashboard.

Tabella 9: Change station energy price

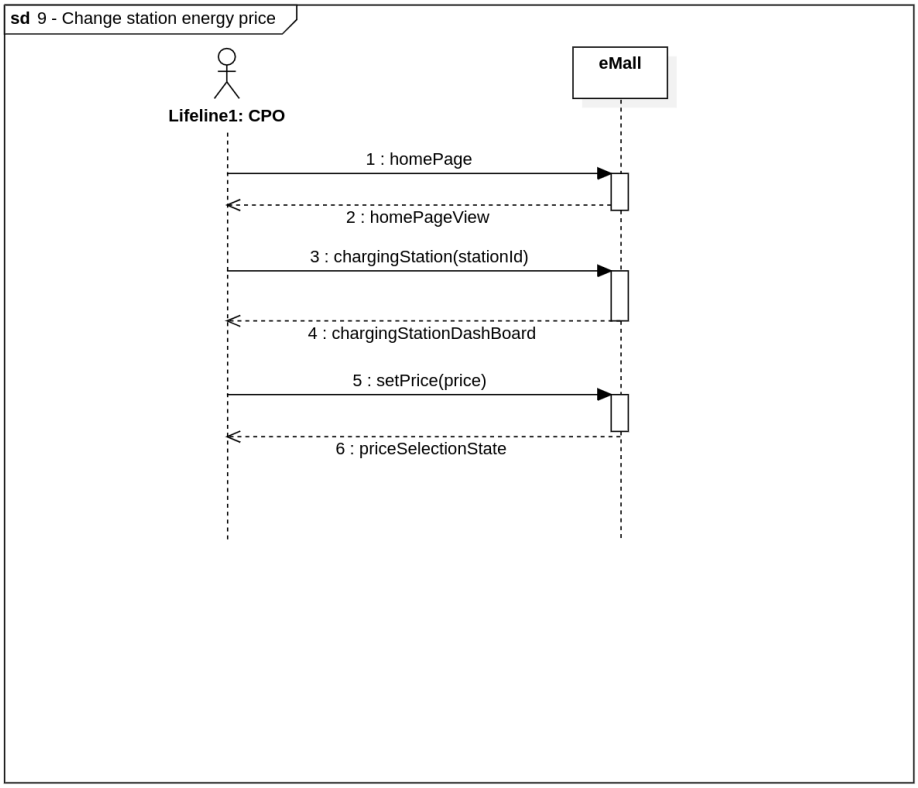


Figura 25: Change station energy price

ID	10
Name	Set special offers
Actor	CPO
Entry conditions	CPO has logged in

Events flow	• The web portal displays the home page, which is a list with all the charging station that are owned by the current CPO. • The CPO chooses a charging station. • The portal displays the dashboard of the previously selected station, that features information about a charging station, including its current price, special offers, battery level (if present) and battery policy, with options to change them. • The CPO clicks on the "Set special offer" button. • The application displays a popup where the user can input the desired discount in percentage with respect to the current station price, or set the discount to 0% (Offer deactivated). • CPO clicks the "Confirm" button.
Exit condition	The application displays a confirmation popup, and after the CPO closes it, he is redirected to the view of the updated dashboard of his previously selected station.
Output	The system has updated the station price with the new value.
Exception 1	While updating the discount factor, the CPO clicks on the "Discard" button. He returns to the dashboard and no change is made.
Exception 2	If the user inputs an invalid factor (such as negative or with characters) and clicks the "Submit" button the portal will show an error message, and then he will be returned to the dashboard.

Tabella 10: Set special offers

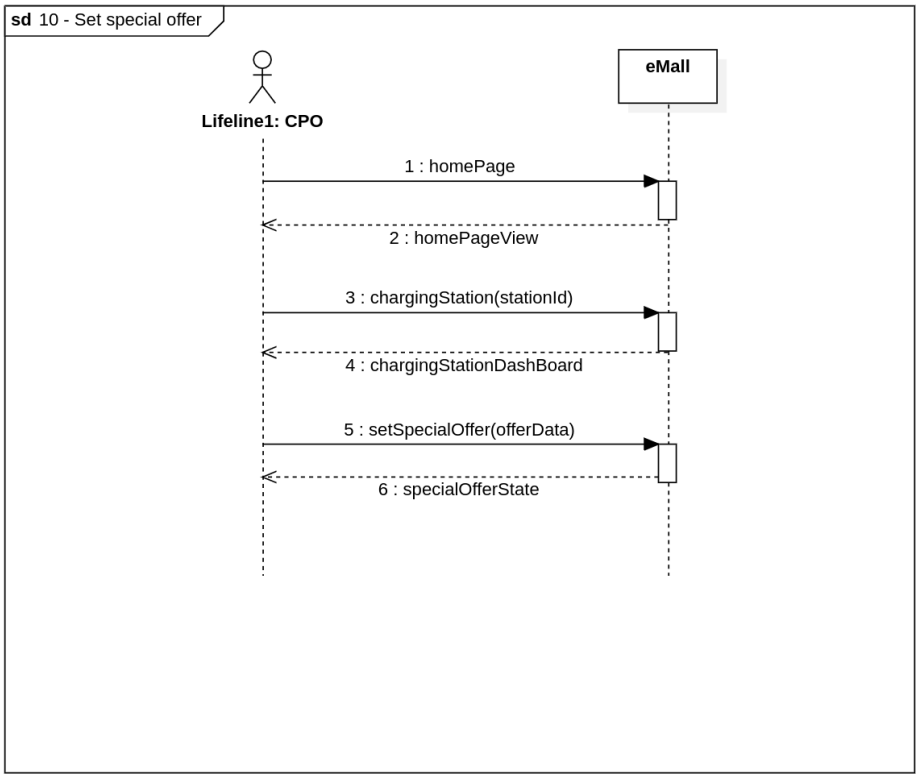


Figura 26: Set special offers

ID	11
Name	Decide battery policy
Actor	CPO
Entry conditions	• CPO has logged in • The station has a battery

Events flow	• The web portal displays the home page, which is a list with all the charging station that are owned by the current CPO. • The CPO chooses a charging station. • The portal displays the dashboard of the previously selected station, that features information about a charging station, including its current price, special offers, battery level (if present) and battery policy, with options to change them. • The CPO clicks on the "Set Battery Policy" button (only present if station has a battery). • The application displays a popup where the user can select his desired battery policy. The available policies are: – Maintain: the battery will stay at its current charge level. – Charge: the battery will charge up to 100%. – Discharge: the battery will be discharged to 0%. The CPO can also input the factor of battery energy to mix with the network acquired one. • CPO clicks the "Confirm" button.
Exit condition	The application displays a confirmation popup, and after the CPO closes it, he is redirected to the view of the updated dashboard of his previously selected station.
Output	The system has updated the internal battery management of the station.
Exception 1	While updating the battery policy, the CPO clicks on the "Discard" button. He returns to the dashboard and no change is made.
Exception 2	If the user inputs an invalid factor (such as negative or with characters) and clicks the "Submit" button the portal will show an error message, and then he will be returned to the dashboard.

Tabella 11: Decide battery policy

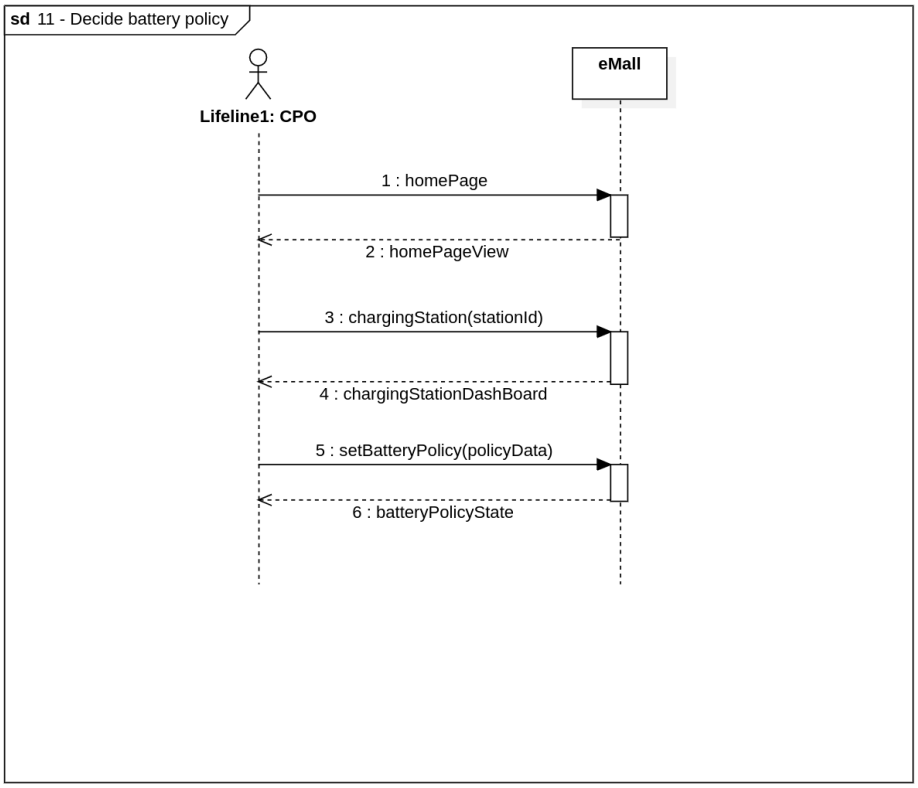


Figura 27: Decide battery policy

3.2.3 Requirements

-G1: Allow customers to obtain information about nearby charging stations

- **R1:** The system allows customers to view nearby station.
- **R2:** The system allows customer to select a charging station to view its information.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D4:** The customer's schedule and location is accessible by the platform.

-G2: Allow customers to book a charge for a certain timeframe

- **R1:** The system allows customers to view nearby station.
- **R2:** The system allows customers to select a charging station to view its information.

- **R3:** The system shows available socket for each charging station.
- **R4:** The system allows customer to book an available socket in a charging station.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D4:** The customer's schedule and location is accessible by the platform.

-G3: Allow customers to start the charge and show predicted charging time

- **R5:** The system allows customers to view personal bookings.
- **R6:** The system allows customers to select a specific booking.
- **R7:** The system allows to start a charge for a booking when the vehicle is connected and ready to charge.
- **R8:** The system shows to the user the predicted charging time when the charge is started.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D9:** The vehicle start charging only when it is connected to the booked socket.
- **D10:** The socket notifies the CPMS when a vehicle is attached and ready to charge.
- **D15:** All the sockets present in charging stations feature a system that can retrieve the charging level of a car after it has been connected to the socket.

-G4: Allow customers to know when the charging has finished

- **R9:** The system show the binary charging state (in charge/finished).
- **R10:** The system recommends alternative day/time slots or store/-chains to a user.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D16:** The socket notifies the CPMS when a vehicle has finished to charge.

-G5: Allow customers to pay for the charging service

- **R5:** The system allows customers to view personal bookings.
- **R6:** The system allows customers to select a specific booking.

- **R11:** The system allows customers to pay a booked charge.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D8:** A fully functioning payment system is present and returns if a transaction has been successful or not.

-G6: Allow customers to receive suggestions on where to charge

- **R12:** The system periodically searches possible nearby station to charge based on the customer car's battery SoC (<50%), customer's location, customer's calendar.
- **R13:** The system sends a push notification if a possible nearby station is found and no suggestions were sent in the previous 1h.
- **D1:** Each customer who wants to use eMall needs to have a mobile device with the most common mobile OSes (e.g. iOS, Android), and also a reliable Internet connection with that device.
- **D3:** The customer's mobile device fully supports the push notification technology.
- **D4:** The customer's schedule and location is accessible by the platform.
- **D6:** The data automatically obtained by the system in order to send suggestions to the customer is accurate and truthful.
- **D14:** Given a license plate number and personal identification, it exists an API that provides the battery level of the vehicle associated with the license plate.

-G7: Allow Charging Point Operators to decide the energy acquisition options

- **R14:** The system allows CPO to view all their the charging stations
- **R15:** The system allows CPO to view information about a single charging station
- **R16:** The system allows CPO to select energy acquisition battery policy (auto, charge, discharge) for a selected charging stations, if the station has batteries
- **R17:** The system allows CPO to select an energy provider (DSO) or set the selection to auto-mode
- **D2:** Each CPO who wants to access the CPMS platform needs to have a device connected to the Internet (such as PC, Mac, smartphone, etc), with the most common Web Browsers (e.g. Firefox, Google Chrome, Microsoft Edge, Apple Safari, etc).
- **D11:**Every CPO is supplied with login credentials when the system is installed.

-G8: Allow Charging Point Operators to dynamically choose the charging cost and to set offers

- **R14:** The system allows CPO to view all their the charging stations
- **R15:** The system allows CPO to view information about a single charging station
- **R18:** The system allows CPO to set the charging prices for a selected charging station
- **R19:** The system allows CPO to set a special offer for a selected charging station and a specific charging type (slow, fast, rapid)
- **D2:** Each CPO who wants to access the CPMS platform needs to have a device connected to the Internet (such as PC, Mac, smartphone, etc), with the most common Web Browsers (e.g. Firefox, Google Chrome, Microsoft Edge, Apple Safari, etc).
- **D11:** Every CPO is supplied with login credentials when the system is installed.

3.2.4 Traceability Matrix

3.3 Performance Requirements

Performance requirements are not particularly critical for the system, but it is anyway desirable that the system has a good response time, which could be included between 0.1 and 2 seconds. Otherwise, Customers and CPOs may think that the service is interrupted or does not work. Moreover the system has to guarantee a good experience for all users, taking into account that not everyone has access to high speed internet. Therefore, the system should minimize the amount of broadband needed in order to fully download the content. The server infrastructure will be designed to be scalable so that it will be possible to adapt it to the increment of users when the application diffusion will increase. Finally, the system should be able to handle many concurrent users, at least 100 000.

3.4 Design Constraints

3.4.1 Standards Compliance

The system must be compliant to the GDPR law, as it will mainly used in EU.
Secure connections must be used for data comunicaction.
Customers must accept and read the privacy policy during the sign-up phase.

3.4.2 Hardware Limitations

Each customer is required to have a smartphone device on which it is possible to install the eMall application. The device must have access to the internet.
The CPO must have access to the internet using a web browser on a PC.

3.5 Software System Attributes

NFR1: **Reliability**
The system must prevent downtime, in order to let Customers to charge their vehicle in every moment. At the same time CPO must be granted to access the system and perform their actions all the time. To guarantee that the system is up most of the time, preventive regular maintanance should be perfomed. To make the system 24/07 accesible the server could be duplicated and the duplicated one run in parallel. So that during failures and maintanance, of one of them, users can access the other one.

NFR2: **Availability**
Given the fact that eMall is not an emergency service or anything related to critical situations, the system must provide availability of 99.9%. This means that the average time between the occurrence of a fault and service recovery (MTTR) has to be contained at around 0.365 days per year.

NFR3: **Security**
The data provided by the users contain some sensitive information, so the security aspect cannot be underestimated. The central database must be protected with all the available measures to avoid any external or internal attack. The passwords inside the data store have to be encrypted and in case of password recovery, this must never be sent in clear. Communication between parties are encrypted and goes on a secure channel (through SSL protocol) to avoid traffic sniffing and spoofing, thus avoiding cheating attacks and guaranteeing privacy and consistency.

NFR4: **Maintainability**
The system must guarantee a high level of maintainability. It must be designed in such a way that permits future addition of functionalities with minimum effort. Design techniques must in fact guarantee an high reusability. The code must be well documented and hard-coding must be avoided. A testing routine has to be provided and it has to cover at least 75% of the entire codebase, excluding interfaces code.

NFR5: **Portability**
The application must be developed for two different platforms: iOS and Android smartphones. The web application must run on any OS (like Windows, Mac OS, Linux, etc) that supports a web browser. Even mobile devices like iPads and Android tablets must be able to access the web app.

4 Formal Analysis

4.1 Alloy code

```
 //Signatures

    abstract sig Bool {}
    one sig TRUE extends Bool {}
    one sig FALSE extends Bool {}

    abstract sig BatteryPolicy {}
    one sig Discharge extends BatteryPolicy {}
    one sig Disable extends BatteryPolicy {}
    one sig Recharge extends BatteryPolicy {}
    one sig Mantain extends BatteryPolicy {}

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


    sig StationBattery{
        capacity: one Int,
        level: one Int
    }{
capacity ≥ 0
level ≥ 0}

    sig Socket{
        socketid: one Int,
        type: one SocketType,
        cost: one Int,
        isOccupied: one Bool,
        specialOffer: one Bool,
        percentageOffer: one Int

    }{      socketid > 0
percentageOffer > 0
cost > 0
   }

    sig ElectricVehicle{}

    abstract sig AccountRegistration {
    username: one Username,
    password: one Password,
    email: one Email
    }

    sig Email {}
    sig Username{}

    sig Password{}
    { //each password is associated to a registration (some registration may
        ↪ have the same password)
        all p: Password | (some r: AccountRegistration | r.password = p)
    }

sig Birthday{}
    sig Customer extends AccountRegistration{
            electricvehicle: some ElectricVehicle,
            birthday: one Birthday,
            isActive: one Bool,
        booking: set Booking,
```

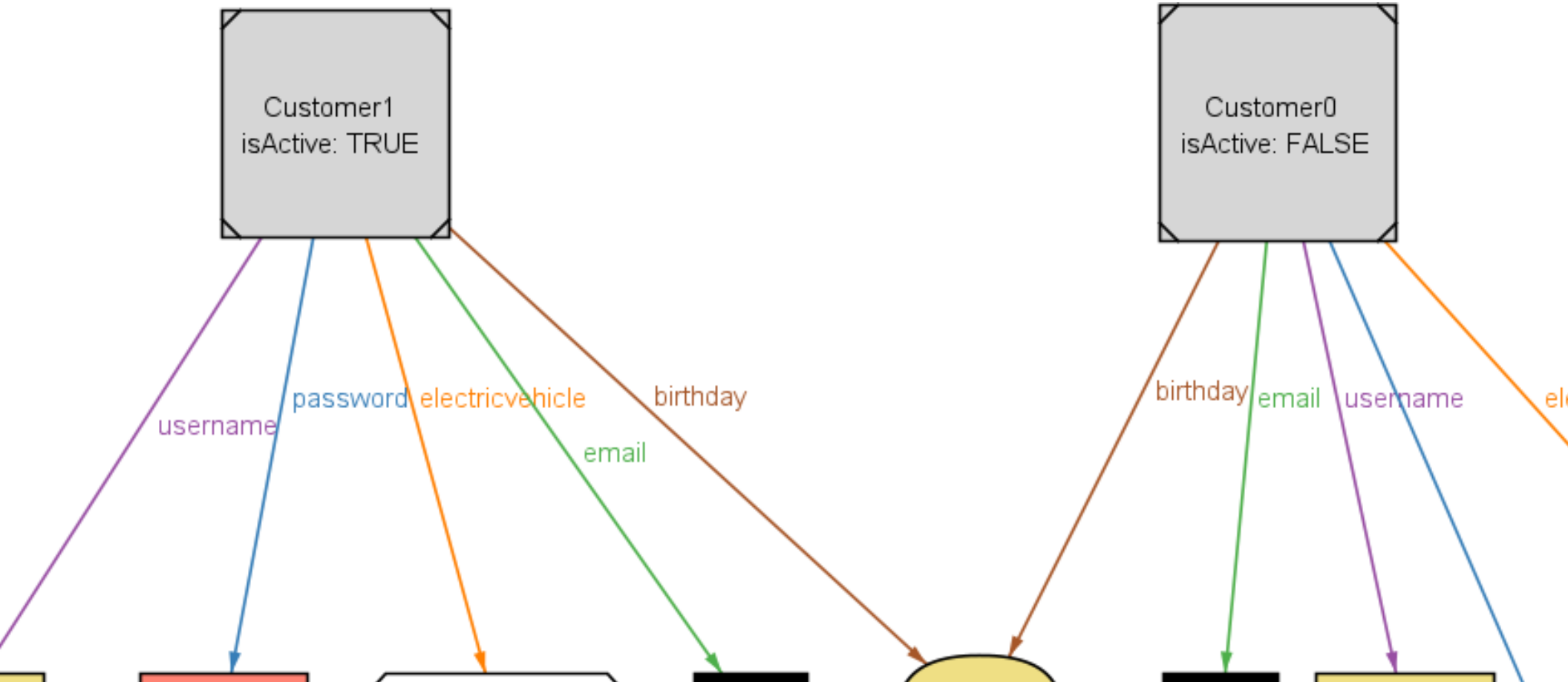
```
        suggestionbooking: lone SuggestionBooking


    }

    sig CPO extends AccountRegistration{
        name: one Name  ,
        stations: some ChargingStation
    }

    sig GPSPosition {}
    sig Name{}

    //We use a simplified model of a time
    sig Time {
            time: one Int
    }{time ≥ 0}

    abstract sig BookingAbstract {
         startTime: one Time,
         endTime: one Time,
        socket: one Socket ,
        station: one ChargingStation
    }
{startTime.time < endTime.time}

    sig ChargingStation{
        id: one Int,
        batterypolicy: one BatteryPolicy,
        location: one GPSPosition,
        socket: some Socket,
        battery: one StationBattery
    }{id > 0}

    sig Booking extends BookingAbstract {
            id: one Int
   }
{id>0}

    sig SuggestionBooking extends BookingAbstract {}


    ------------------------------------------------------------------

    // Facts

        fact{ //a unique username is associated to each registration
                no disj r1, r2: AccountRegistration | r1.username = r2.
                    ↪ username
        }

        fact{ //a unique email address is associated to each registration
                no disj r1, r2: AccountRegistration | r1.email = r2.email
        }

        fact { //The CPO name must be unique
                no disj c1, c2: CPO | c1.name = c2.name
        }

        fact{  //each socket in the same charging station has a different id
                all c: ChargingStation | (no disj s1, s2: Socket | s1.
                    ↪ socketid = s2.socketid
                and (s1 in c.socket) and (s2 in c.socket))
         }

        fact{ //all locations connected
                all g: GPSPosition | one c: ChargingStation | g= c.location
        }
```

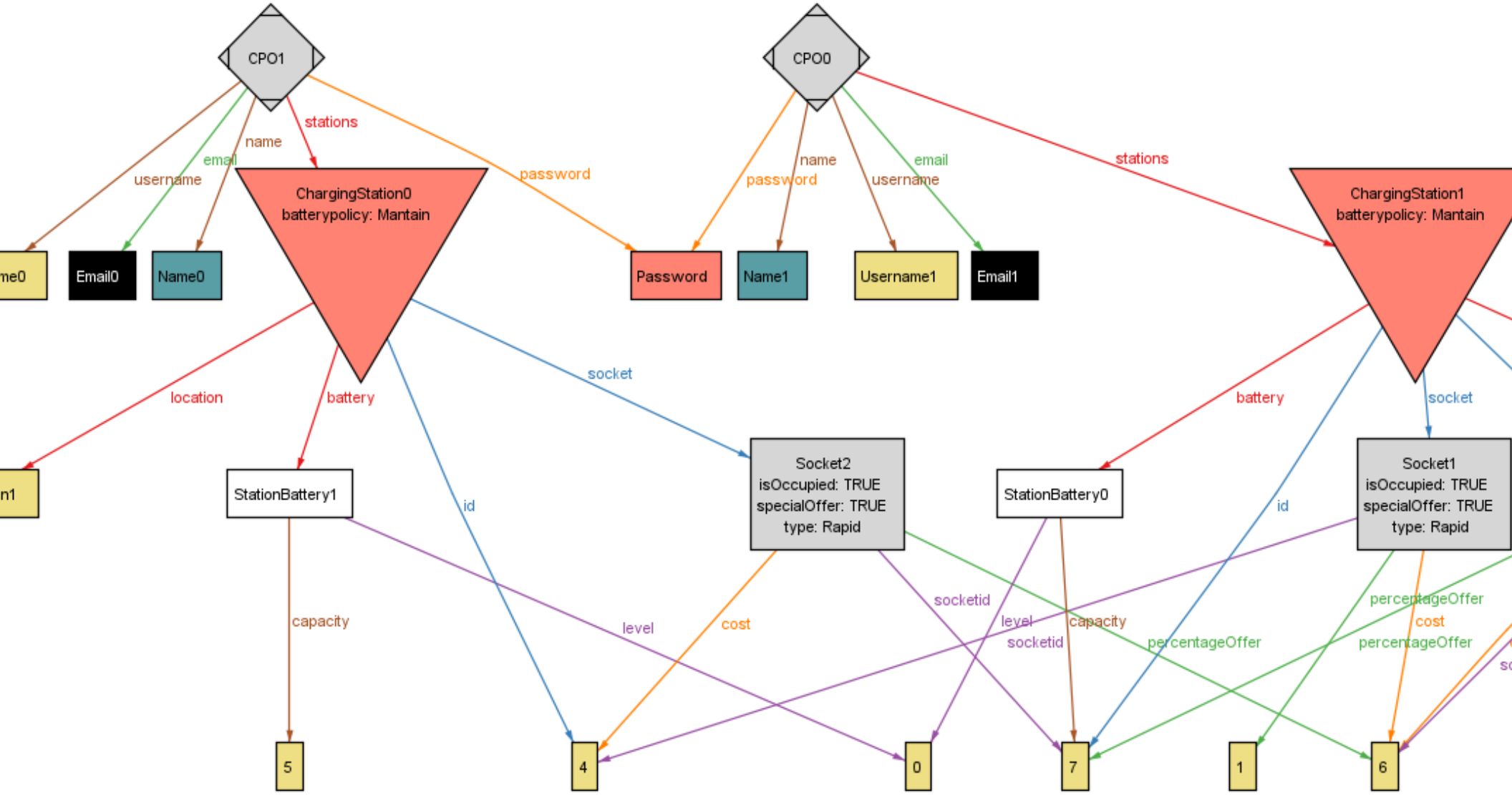
```
fact{ //all bookings connected
        all b: Booking | one c: Customer | b in c.booking
}
fact{ //all suggested bookings connected
        all s: SuggestionBooking | one c: Customer | s = c.
            ↪ suggestionbooking
}

fact{ //all email connected
        all e: Email | one a:  AccountRegistration | e= a.email
}

fact{ //each charging station has a different station battery
        all c: ChargingStation | no disj s1, s2: StationBattery | c.
            ↪ battery = s1
        and c.battery = s2
}

fact { //Each ChargingStation has a unique CPO
        all s: ChargingStation | no disj c1, c2: CPO | s in (c1.
            ↪ stations) and s in (c2.stations)
}

fact{ //all electric vehicle connected
        all e: ElectricVehicle | one c: Customer | e in (c.
            ↪ electricvehicle)
}

fact{ //all names connected
        all n: Name | one c: CPO | n= c.name
}

fact{ //all sockets connected
        all s: Socket | one c: ChargingStation | s in ( c.socket)
}

fact{ //all station batteries connected
        all b: StationBattery | one c: ChargingStation | b= c.battery
        }

fact{ //all usernames connected
        all u: Username | one a:  AccountRegistration| u = a.username
}

fact{ //all birthdays connected
                all b: Birthday | some c: Customer | b = c.birthday
}

fact{ //all charging stations connected
        all cs: ChargingStation | one c:  CPO| cs in c.stations
}

fact{  //different Customers have a different set of electric
    ↪ vehicles
        no disj c1, c2 : Customer | c1.electricvehicle = c2.
            ↪ electricvehicle
}


fact{  //the booked socket must be in the corresponding charging
    ↪ station
        all b: BookingAbstract |  ((b.socket) in (b.station.socket))
}



fact{  // Cannot exist two bookings at the same time in the same
```

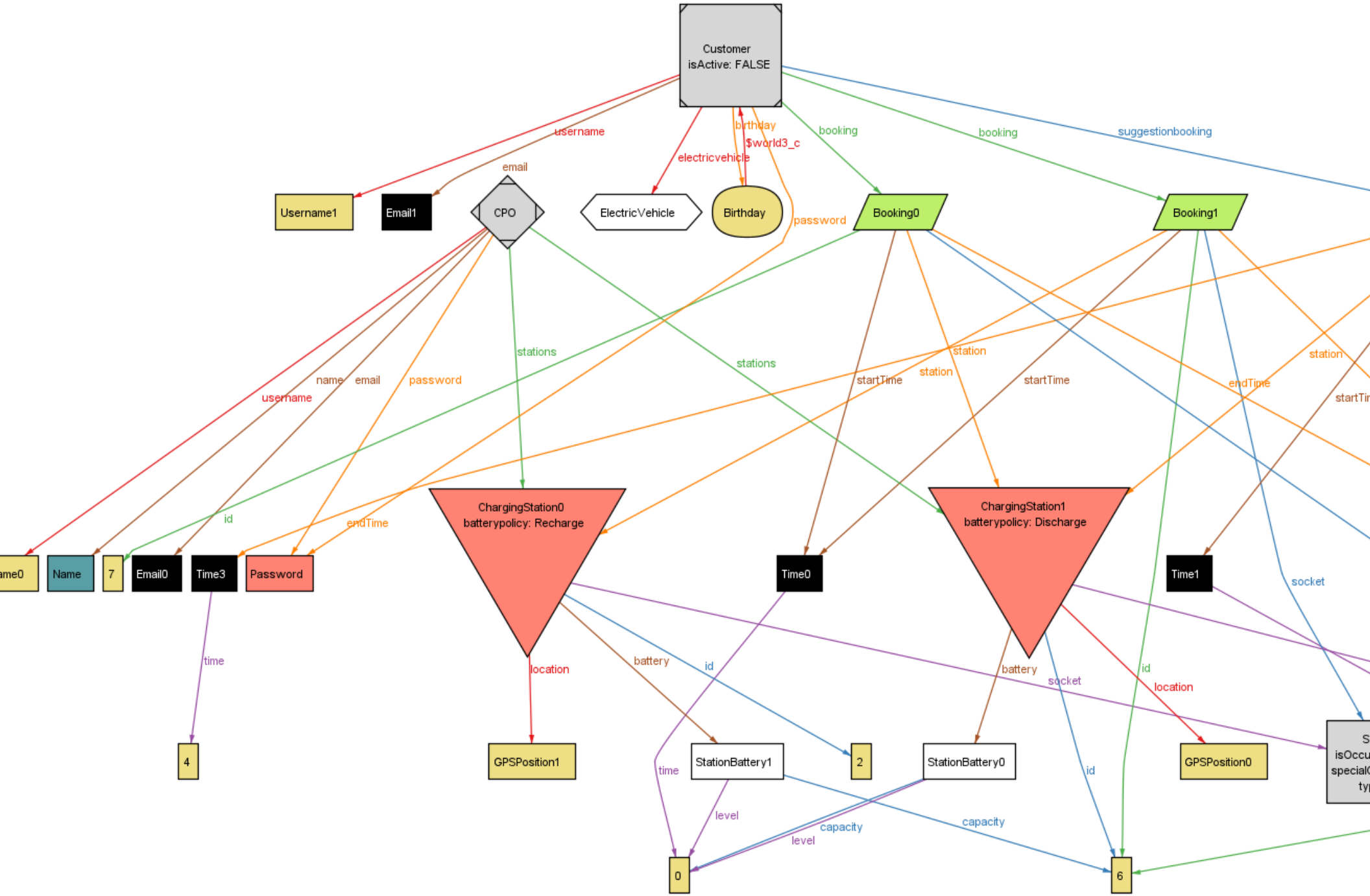
```
            ↪ socket
                no disj b1, b2: Booking | b1.socket = b2.socket and b1.
                    ↪ station= b2.station and
                ((b1.startTime.time > b2.startTime.time and b2.endTime.time >
                    ↪  b1.endTime.time)
                or (b2.startTime.time < b1.endTime.time) or
                (b2.endTime.time > b1.startTime.time) )

        }


        fact{ //Customer can have just one booking at the time
                all c: Customer | (no disj b1, b2: Booking| (b1 in c.booking
                    ↪ and b2 in c.booking) and
                ((b1.startTime.time > b2.startTime.time and b2.endTime.time >
                    ↪  b1.endTime.time)
                or (b2.startTime.time < b1.endTime.time and b2.endTime.time >
                    ↪  b1.endTime.time ) or
                (b2.endTime.time > b1.startTime.time and b1.endTime.time > b2
                    ↪ .endTime.time ) )

                and  b1.startTime.time = b2.startTime.time and b1.endTime.
                    ↪ time = b2.endTime.time)
} //non funziona bene


        fact{ //All bookings have a different id
                no disj b1, b2: Booking | b1.id = b2.id

        }


//      fact{ //Customer doesn't have a suggested booking during a booking
         //     }



-----------------------------------------------------------------
//predicates

// simulation  that  shows  the  result  of some customer registrations
pred world1 {
        #Customer = 3
        #CPO = 0
        #Password > 1
        #Time = 0
        #ElectricVehicle > 2
        #SuggestionBooking = 0
        #Booking = 0
}
run world1 for 7

// simulation  that  shows  the  result  of some CPOs and the corresponding
    ↪ Charging Stations
pred world2 {
        #Customer = 0
        #CPO = 2
        #ChargingStation = 2
        #Time = 0
        #SuggestionBooking = 0
        #Booking = 0
        #Socket > 2
}
run world2 for 7

// simulation  that  shows  the  result  of some bookings and suggested
    ↪ bookings
```

```
pred world3 {
        #Customer = 1
        #ChargingStation = 2
        #Socket = 2
        #Booking  = 2
        #SuggestionBooking > 0
}
run world3 for 7
```

4.2 Worlds


Customer1
isActive: TRUE
Customer0
isActive: FALSE
username
password
electricvehicle
email
birthday
birthday
email
username


CPO1
CPO0
stations
name
email
username
password
ChargingStation0
batterypolicy: Mantain
ChargingStation1
batterypolicy: Mantain
Email0
Name0
Password
Name1
Username1
Email1
location
battery
socket
StationBattery1
StationBattery0
Socket2
isOccupied: TRUE
specialOffer: TRUE
type: Rapid
Socket1
isOccupied: TRUE
specialOffer: TRUE
type: Rapid
id
capacity
level
cost
socketid
percentageOffer
5
4
0
7
1
6


Customer
isActive: FALSE
username
email
birthday
$world3_c
electricvehicle
booking
booking
suggestionbooking
Username1
Email1
CPO
ElectricVehicle
Birthday
password
Booking0
Booking1
stations
stations
name
email
password
username
startTime
station
station
startTime
endTime
station
id
endTime
ChargingStation0
batterypolicy: Recharge
ChargingStation1
batterypolicy: Discharge
ame0
Name
7
Email0
Time3
Password
Time0
Time1
socket
time
location
battery
id
battery
socket
id
location
4
GPSPosition1
time
StationBattery1
2
StationBattery0
id
GPSPosition0
level
capacity
capacity
level
0
6

5 Effort spent

Student	Time for S.1	S.2	Time for S.3	Time for S.4
Alessandro Pignati	3h	5h	13h	1h
Federico Sarrocco	3h	8h	10h	8h
Alessandro Vacca	3h	6h	11h	8h