

POLITECNICO DI MILANO  
Department of Electronics, Information, and Bioengineering  
Computer Science Engineering



## Software Engineering 2

eMall - e-Mobility for All

Group components:

Roberto **CIALINI** 933385

Umberto **COLANGELO** 935073

Vittorio **LA FERLA** 224509

Academic Year 2022-2023

## Contents

1	Introduction . . . . .	1
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Definitions, Acronyms, Abbreviations . . . . .	2
1.3.1	Definitions . . . . .	2
1.3.2	Abbreviations . . . . .	2
1.4	Revision History . . . . .	3
1.5	Reference Documents . . . . .	3
1.6	Document Structure . . . . .	3
2	Architectural Design . . . . .	4
2.1	Overview: High-level components and their interaction . . . . .	4
2.2	Component view . . . . .	4
2.2.1	eMSP . . . . .	5
2.2.2	CPMS . . . . .	7
2.3	Deployment view . . . . .	8
2.4	Runtime view . . . . .	8
2.4.1	Registration Runtime . . . . .	8
2.4.2	Login Runtime . . . . .	9
2.4.3	View Map Page . . . . .	10
2.4.4	View Stations Page . . . . .	10
2.4.5	Make a reservation . . . . .	11
2.4.6	Monitor the charge . . . . .	12
2.4.7	Start the charge . . . . .	13
2.4.8	Change Energy Mix . . . . .	14
2.4.9	Change the <i>DSO</i> . . . . .	15
2.4.10	Accept a recommendation . . . . .	16
2.5	Component interfaces . . . . .	16
2.6	Selected architectural styles and patterns . . . . .	16
2.7	Other design decisions . . . . .	16

3	User Interface Design . . . . .	17
3.1	Driver . . . . .	17
3.1.1	Login . . . . .	17
3.1.2	Register . . . . .	18
3.1.3	User Profile . . . . .	19
3.1.4	Vehicles . . . . .	20
3.1.5	Payment Methods . . . . .	21
3.1.6	Map and Stations . . . . .	22
3.1.7	Reservations . . . . .	23
3.1.8	Recommendations . . . . .	24
3.1.9	Create Reservation . . . . .	25
3.1.10	Charging Status . . . . .	26
3.2	Administrator . . . . .	27
3.2.1	Login . . . . .	27
3.2.2	Stations . . . . .	27
3.2.3	Station Settings . . . . .	28
3.2.4	DSO . . . . .	29
4	Requirements traceability . . . . .	30
5	Implementation, Integration and test Plan . . . . .	31
5.1	Implementation . . . . .	31
5.2	Integration & Test plan . . . . .	33
6	Effort Spent . . . . .	35
6.0.1	Roberto Cialini . . . . .	35
6.0.2	Umberto Colangelo . . . . .	35
6.0.3	Vittorio La Ferla . . . . .	35
7	References . . . . .	36

## 1 Introduction

### 1.1 Purpose

Nowadays sustainability is one of the most important and debated topics in our society. In fact, in the next few years we are going to deal with a huge green transition to limit our carbon footprint on the planet, such as in the transportation field, which finds itself as one of the main contributors of global warming. In this direction, in recent years the old motor vehicles running on gasoline are leaving space for electricity-powered vehicles, even though there are several central aspects to deal with in order to let the electric vehicles be competitive with the old vehicle generation. In this direction, the goal is to create a fully operative and diffused infrastructure for the fast charging of the batteries, which is one of the main limitations for the final customer. In fact, the batteries need to be charged often and nowadays the task of finding an available charging spot is not as easy as it seems.

With this issue in mind, *eMall* is an operating system, itself composed of two subsystems, whose goal is to offer a way to find available charging stations for electric vehicles, offering at the same time to the user the possibility to access to several features such as the reservation of a specific socket at a certain timeframe or the reception of personalised proactive suggestions by the system.

This document contains a description of the architectural design for the system, including the components involved and how they interact. Additionally mockups of the user interface are presented and a plan for the implementation, testing and integration of the system. Therefore, this document should guide the development of the system.

### 1.2 Scope

While there are several stakeholders to consider, this document is only concerned about two actors: Drivers and Administrators. The Driver is the final user, the one who interacts with the *eMSP* to have the possibility to book the battery recharge of his vehicles. Instead the role of the system Administrator mainly concerns to monitor the correct behaviour of the system and to take strategic decisions.

The main system is divided into two subsystems: the *eMSP* and *CPMS*. The *eMPS*

is designed to be an interface and to communicate both with the Driver and the Administrator, driving their requests. The *CPMS*, instead, is modelled on the *OCPI 2.2.1* protocol and is referred to as a specific *CPO*. The main task of the *CPMS* is to supply information about its *CPO* charging stations to the *eMSPs* it is linked to, both for the Driver and the Administrator usage.

Although *CPO* and *DSO* are mentioned in this document along with the other entities described before, we will not consider either their internal system or their decision making.

The architecture of *eMall* follows the three-tier pattern with a presentation layer, business logic layer and a data layer.

### 1.3 Definitions, Acronyms, Abbreviations

#### 1.3.1 Definitions

Definitions	Description
<b>Driver Identifier</b>	To identify a specific driver, this could be an identification number such as her/his SSN
<b>Car Identifier</b>	To identify a specific car, this could be the licence plate
<b>Station Identifier</b>	To identify a specific charging station

#### 1.3.2 Abbreviations

Abbreviations	Definitions
<b>RASD</b>	Requirements Analysis and Specification Document
<b>API</b>	Application Programming Interface
<b>RX</b>	Requirements number X
<b>GPS</b>	Global Positioning System
<b>CLI</b>	Command Line Interface
<b>DBMS</b>	Database Management System
<b>DB</b>	Database
<b>MVC</b>	Model View Controller
<i><b>eMall</b></i>	e-Mobility for All
<b>EV</b>	Electric Vehicle
<b>Driver</b>	Electric vehicle driver
<b>Administrator</b>	<i>CPO</i> administrator

## 1.4 Revision History

Version 1.1.0

## 1.5 Reference Documents

- The specification document "Assignment RDD AY 2022-2023.v3.pdf"
- RASD

## 1.6 Document Structure

This document is composed of seven sections, detailed below.

In the first section the problem is introduced together with the purpose of this specific report and a recap of the context. Additionally, some necessary information in order to read the report is given, such as definitions and abbreviations.

Section two contains the description of the architectural design of the system together with motivations and reasons that led to opting for these solutions. It starts with a high-level overview of the architecture and then breaks each part down into components. The components are described and their interdependence are shown in the component diagram. Moreover, the section contains a component interface diagram, a deployment view and sequence diagrams describing the interactions between components in the run-time view.

In section three design mockups of the user interface is presented.

Section four contains the requirement traceability matrix, where each of the components described in section two is mapped to the requirements specified in the *RASD*. The mapping is based on whether the component contributes to the fulfilment of the requirement.

Section five describes the suggested implementation order and test plan of the system.

In section six is shown the total effort spent by each of the project members.

Section seven contains the references used.

## **2 Architectural Design**

### **2.1 Overview: High-level components and their interaction**

### **2.2 Component view**

In this section the component view is presented with a component diagram and corresponding descriptions.

## 2.2.1 eMSP

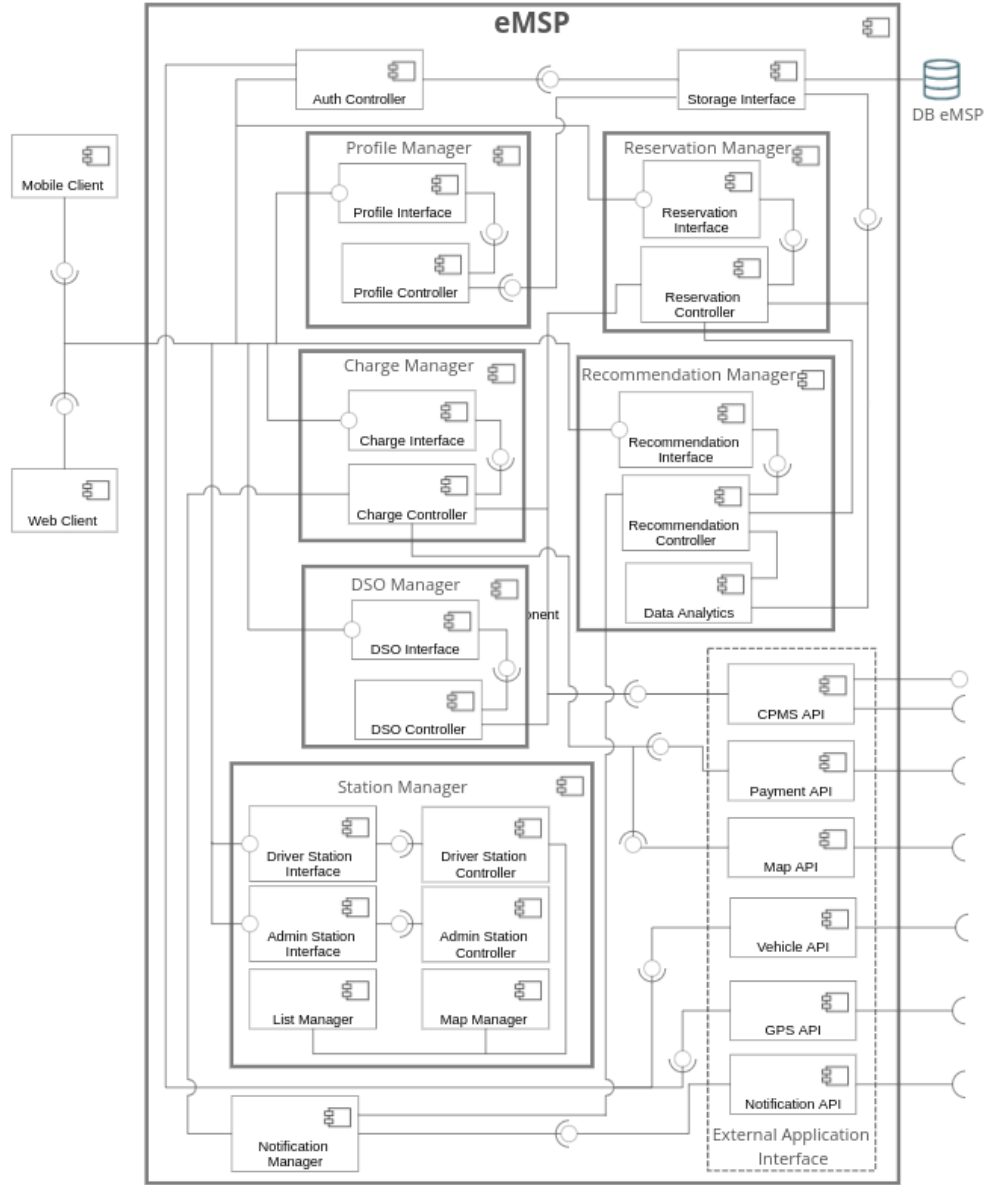


Fig. 1: Component View of eMSP

- **Mobile Client:** Mobile Client Application. E.g application on a mobile phone.
- **Web Client:** Web Client Application. E.g web browser
- **Auth Controller:** this component handles all the operations for the authentication such as registration process and login process. It communicates with the



DBMS through Storage Interface in order to retrieve and insert user credentials.

- **Auth Controller:** this component handles all the operations for the authentication such as registration process and login process. It communicates with the DBMS through Storage Interface in order to retrieve and insert user credentials.
- **Storage Interface:**Storage Interface: it provide methods to access to database. This interface is needed in order to decouple all the components from the DBMS technology.
- **Notification Manager:**his component provides Notifications to the Customer, such as new Recommendation or to notify the end of the charge.
- **Profile Manager:** this component handles all changes to the user profile, such as change the Active Vehicle or enter a new payment method.
- **Station Manager:** this components is needed to handle all the operations which require the station. Basically provides all the views in which are involved the stations.
- **Charge Manager:** this components is needed to handle all the operations which require the station. Basically provides all the views in which are involved the stations.

## 2.2.2 CPMS

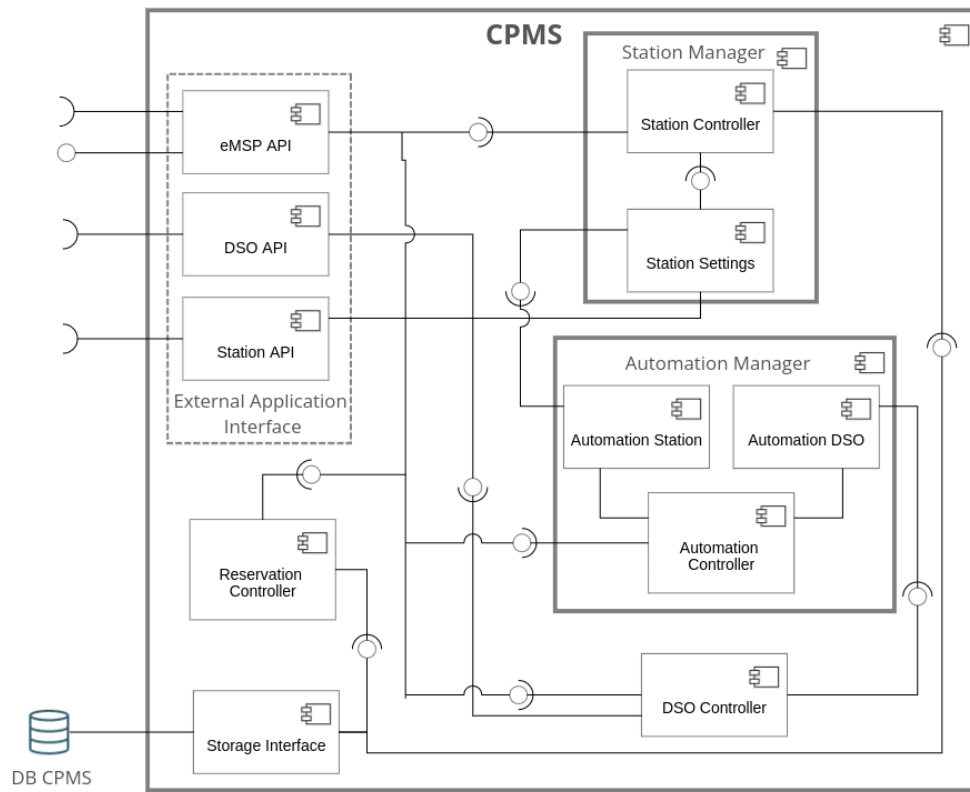


Fig. 2: Component View of CPMS

## 2.3 Deployment view

## 2.4 Runtime view

### 2.4.1 Registration Runtime

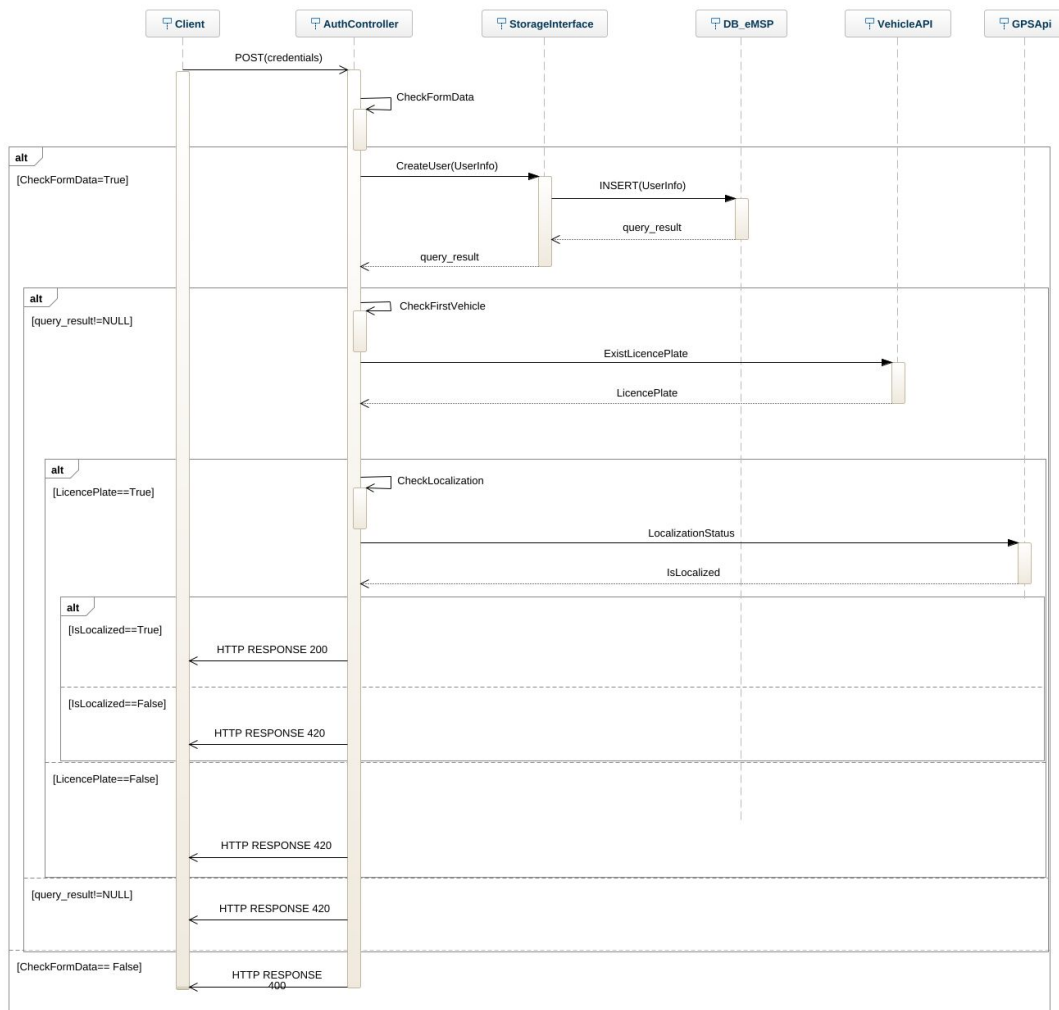


Fig. 3: Sequence diagram of the registration process for a Driver

## 2.4.2 Login Runtime

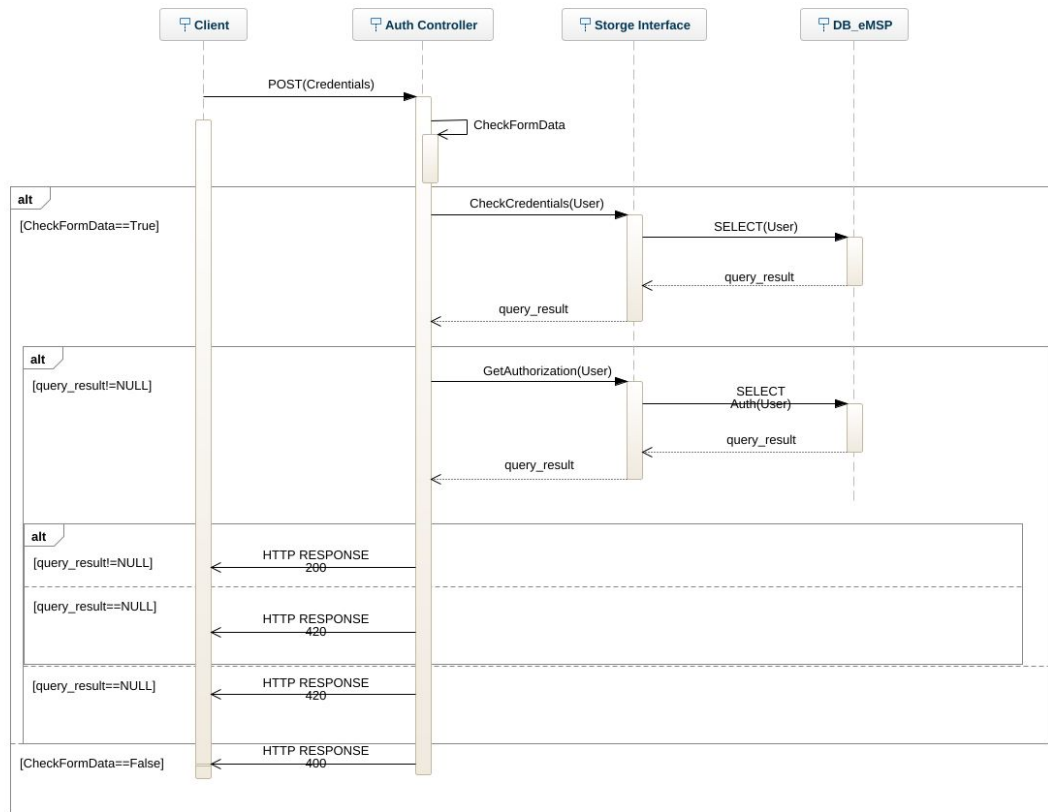


Fig. 4: Sequence diagram of the login to eMail

### 2.4.3 View Map Page

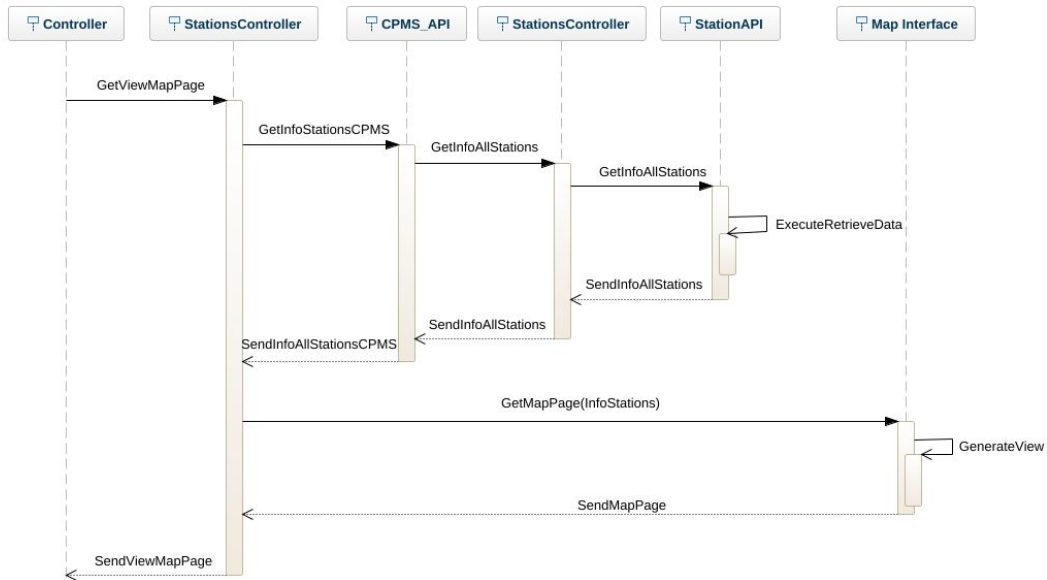


Fig. 5: Sequence diagram of the visualization of Map Page

### 2.4.4 View Stations Page

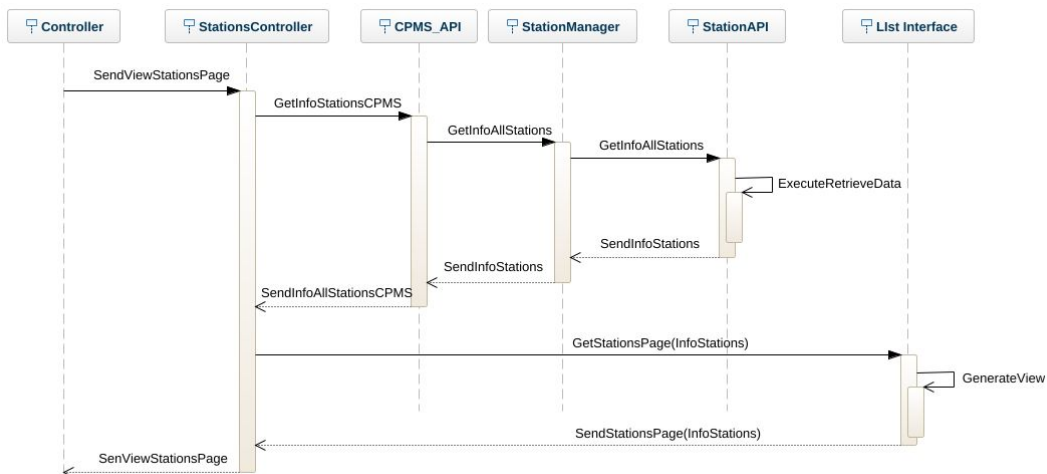


Fig. 6: Sequence diagram of the visualization of the list in Stations Page

## 2.4.5 Make a reservation

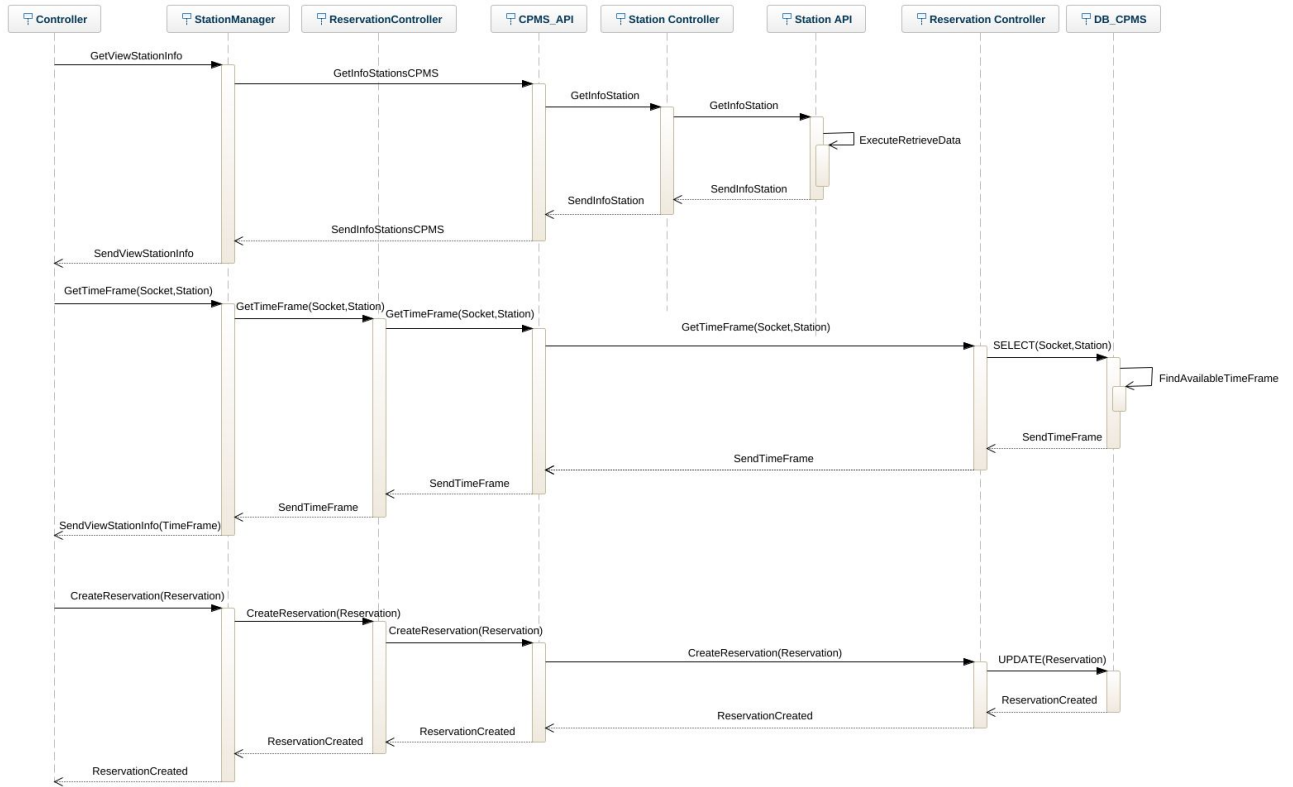


Fig. 7: Sequence diagram for making a reservation

## 2.4.6 Monitor the charge

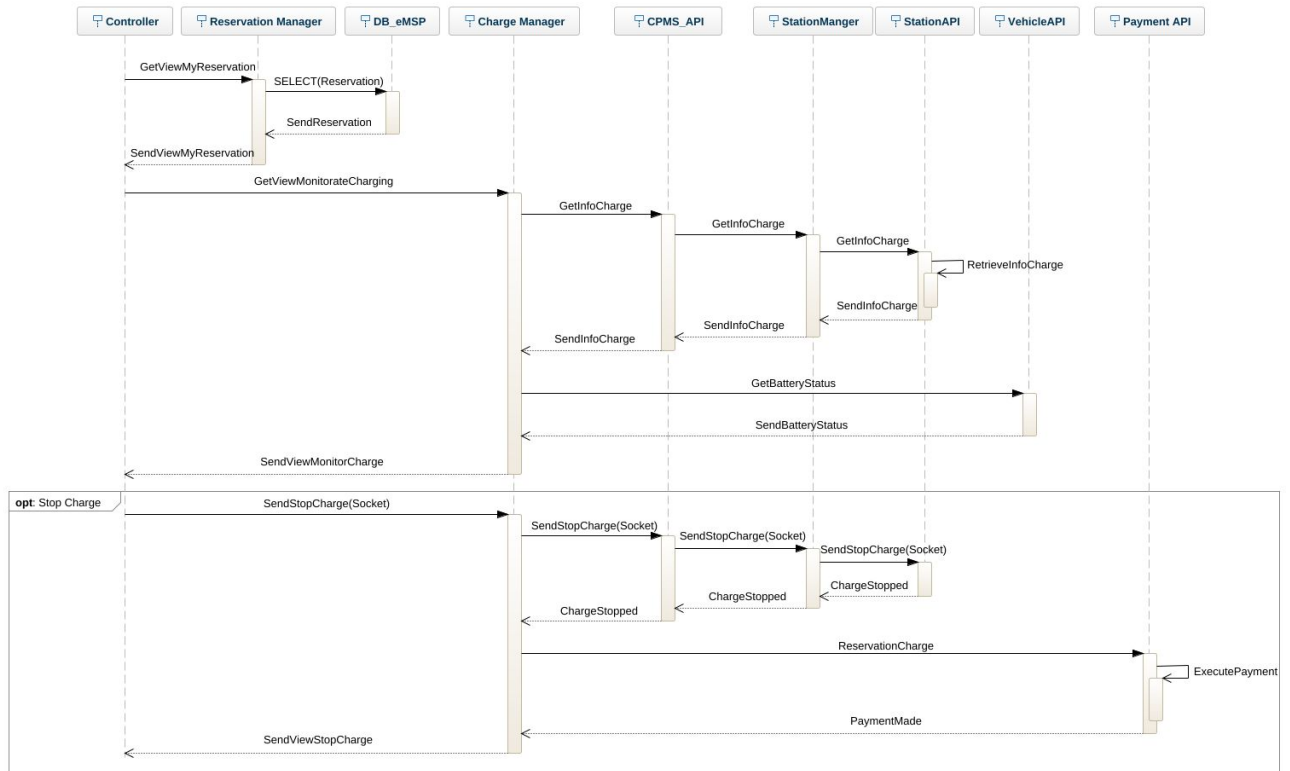


Fig. 8: Sequence diagram for monitoring the charge

## 2.4.7 Start the charge

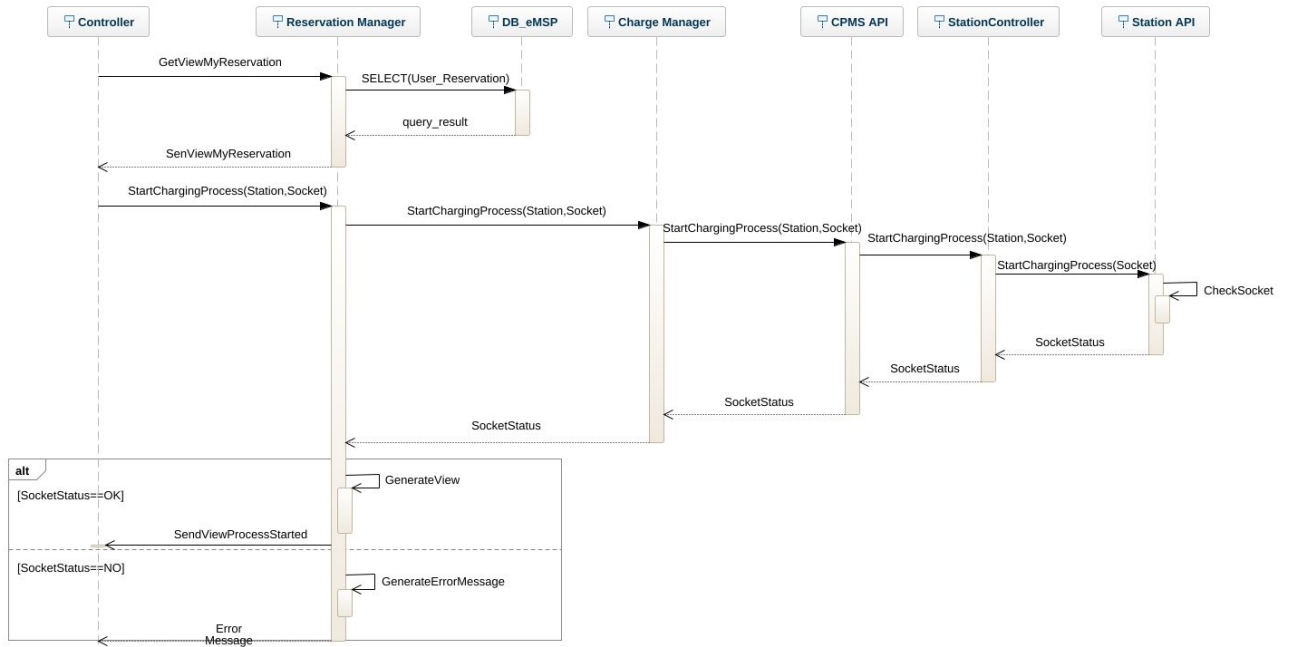


Fig. 9: Sequence diagram for starting the charge



## 2.4.8 Change Energy Mix

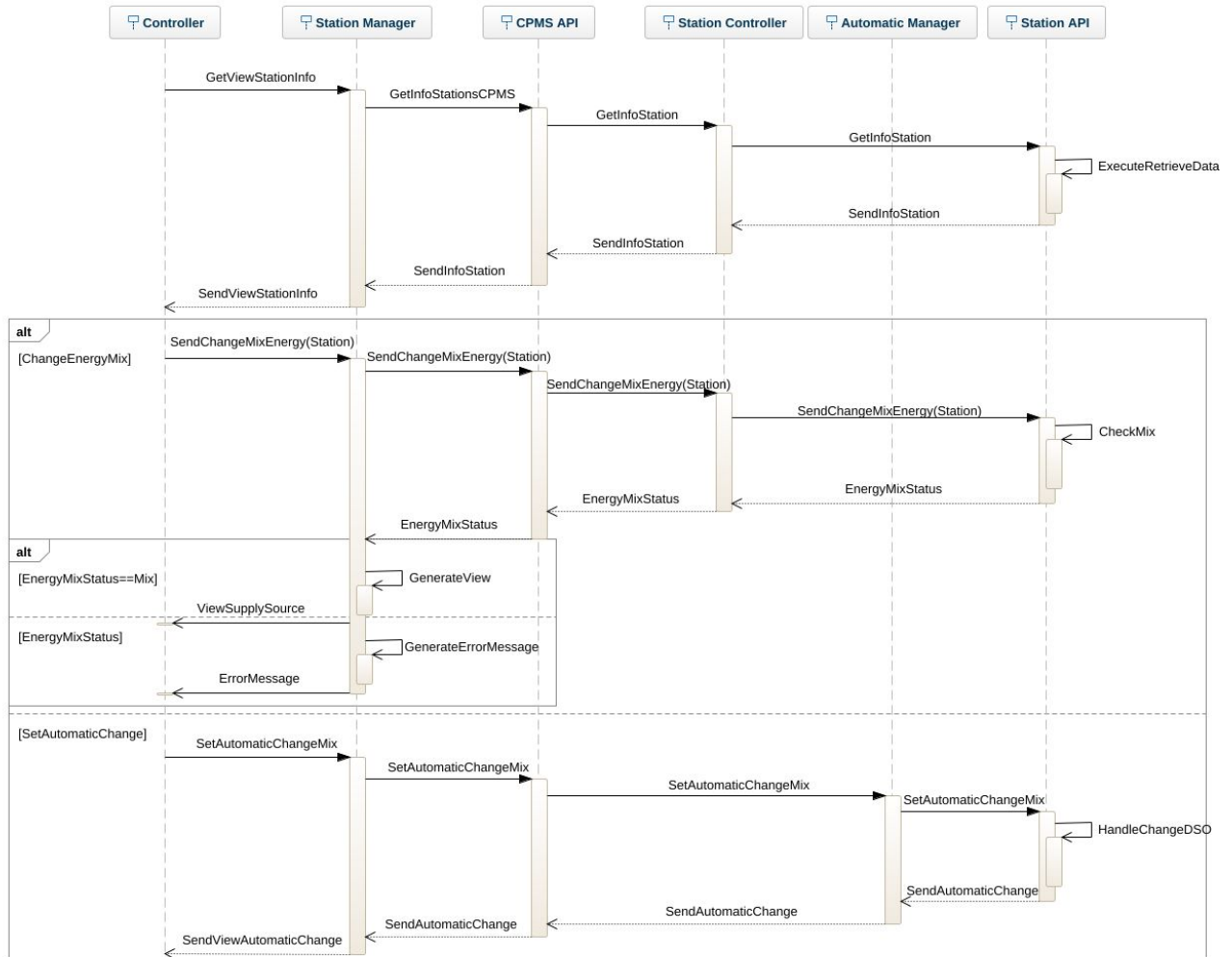


Fig. 10: Sequence diagram for energy mix settings

## 2.4.9 Change the DSO

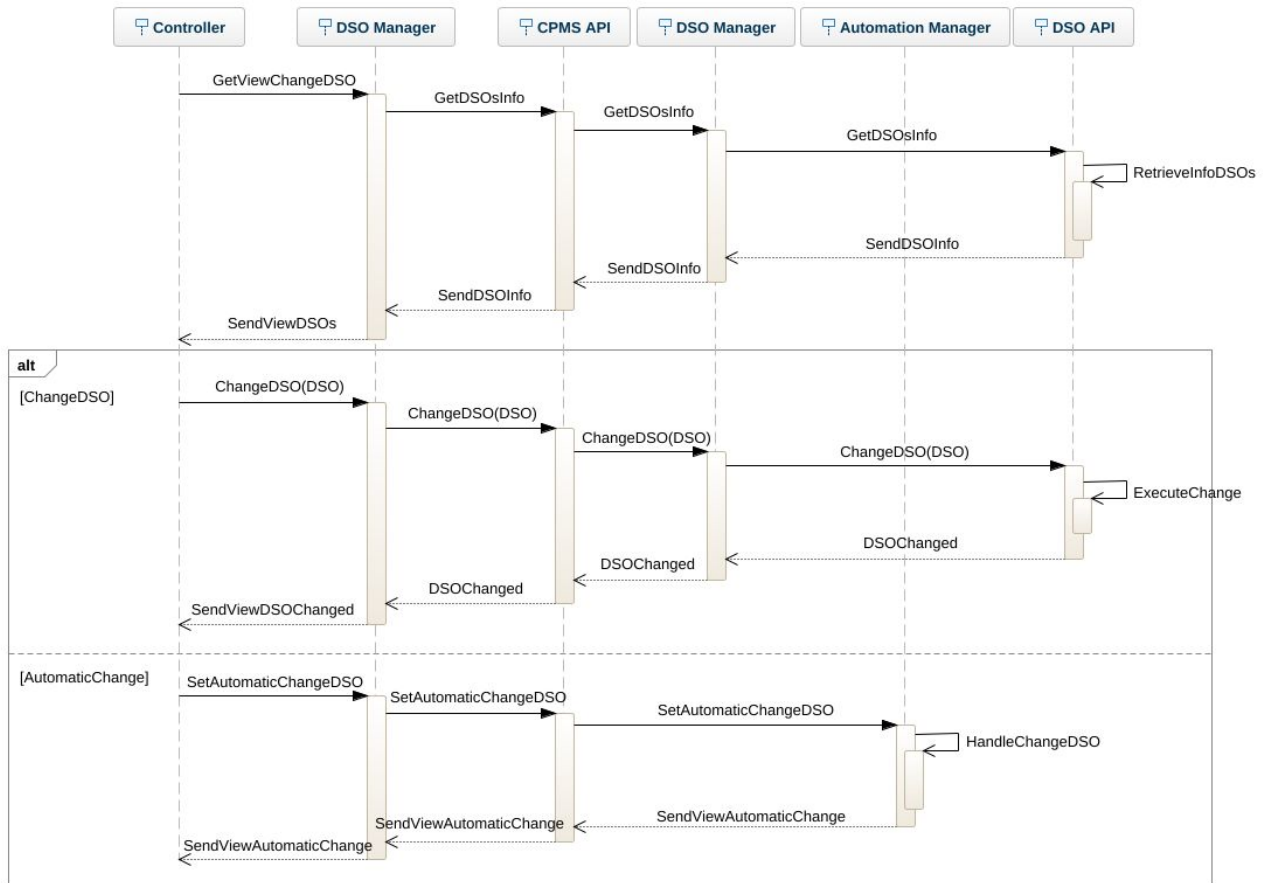


Fig. 11: Sequence diagram for DSO settings

### 2.4.10 Accept a recommendation

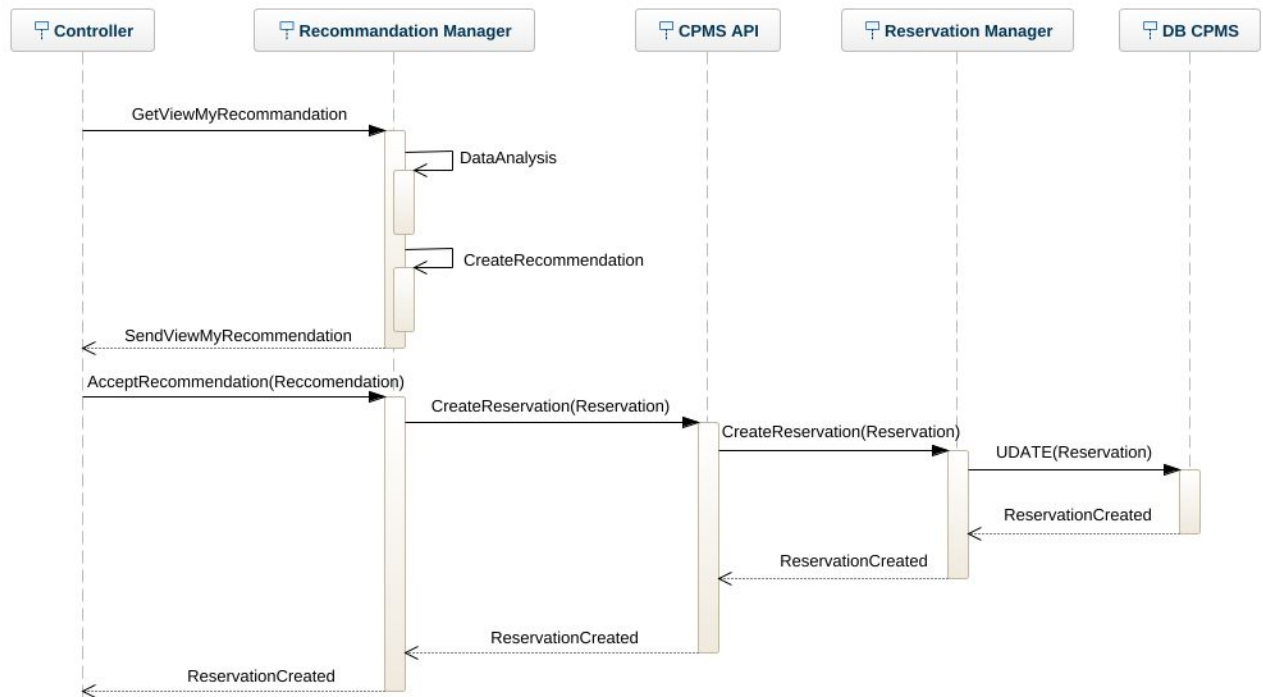


Fig. 12: Sequence diagram of the acceptance of a recommendation

## 2.5 Component interfaces

## 2.6 Selected architectural styles and patterns

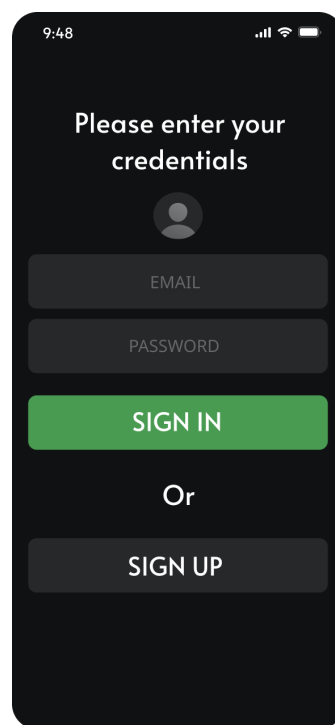
## 2.7 Other design decisions

### 3 User Interface Design

In this section are presented the mockups of the user interface. In particular, it is shown a mobile app view for the Drivers, while for the Administrators it is shown a PC view.

#### 3.1 Driver

##### 3.1.1 Login



**Fig. 13:** *UI of the Login Page*

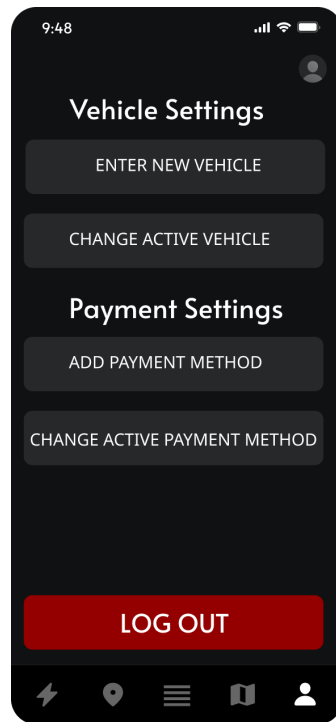
### 3.1.2 Register

The figure displays three sequential mobile app screens for driver registration, each with a dark background and white text. The top status bar of each screen shows the time as 9:48 and signal indicators.

- Screen 1: Create New Account**
  - Title: **Create New Account**
  - Input fields: NAME, SURNAME, EMAIL, PASSWORD, DOCUMENT ID
  - Button: **CONTINUE** (green)
- Screen 2: Vehicle**
  - Title: **Vehicle**
  - Input fields: VEHICLE NAME, LICENCE PLATE
  - Title: **Payment Method**
  - Input fields: CARD NUMBER, MM/YY, CVV
  - Button: **CONTINUE** (green)
- Screen 3: Accept terms and conditions**
  - Title: **Accept terms and conditions**
  - Text: By clicking on, "Create new account" button, you will allow the system to use the geolocalization service of your device and your daily schedule information
  - Button: **Create new account** (green)

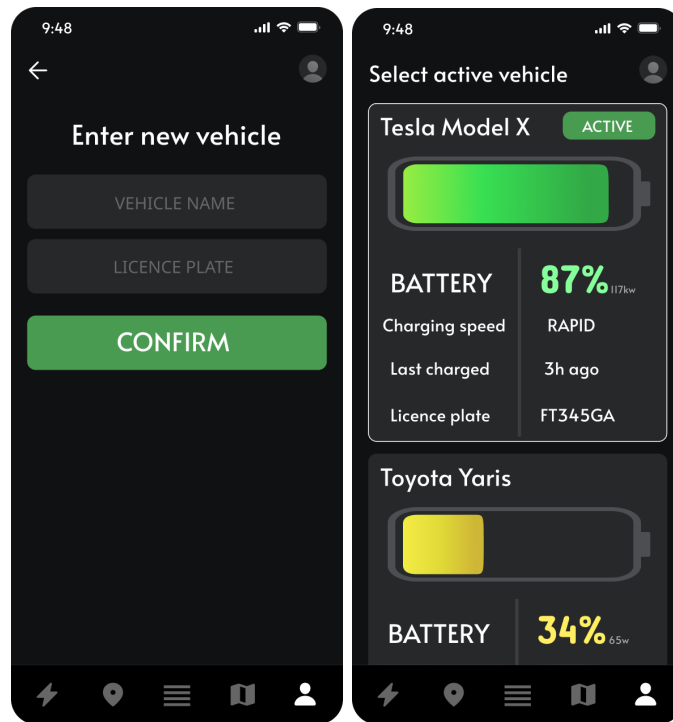
**Fig. 14:** UI for registering as a Driver, divided in three separate steps

### 3.1.3 User Profile



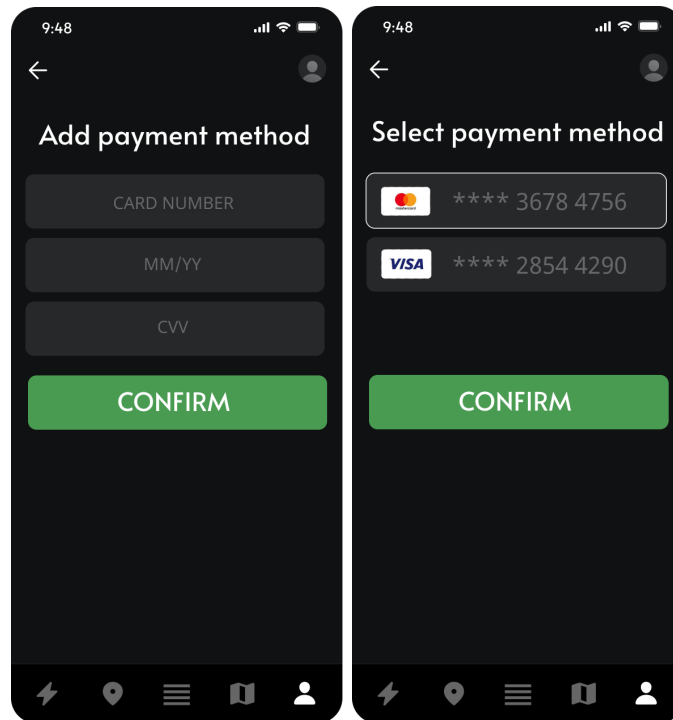
**Fig. 15:** *UI of the User Profile Page*

### 3.1.4 Vehicles



**Fig. 16:** UI of Driver entering a new vehicle (on the left) and changing the active vehicle (on the right)

### 3.1.5 Payment Methods



**Fig. 17:** UI of Driver adding a new payment method (on the left) and changing the active payment method (on the right)



### 3.1.6 Map and Stations

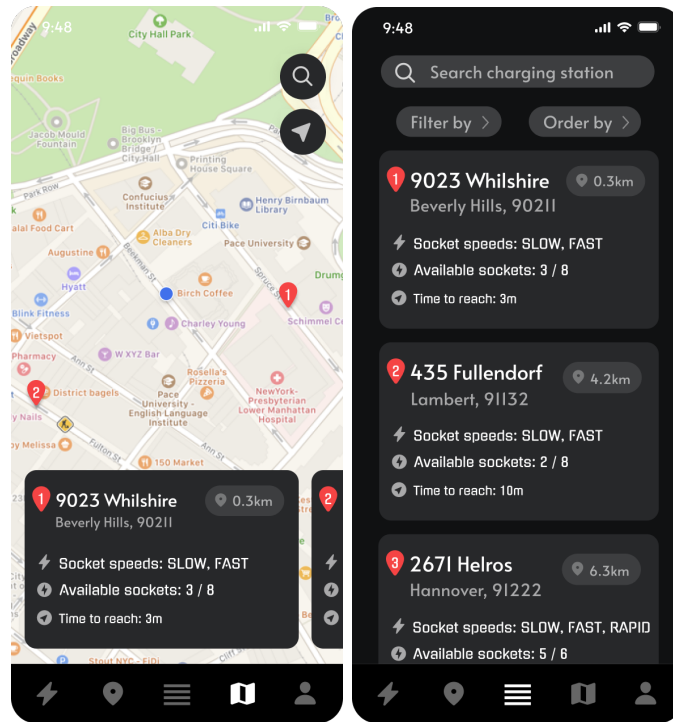


Fig. 18: UI of Map Page (on the left) and Stations Page (on the right)

### 3.1.7 Reservations

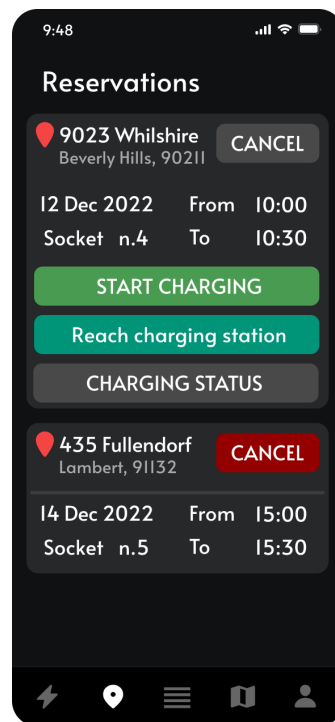


Fig. 19: UI of the MyReservations Page

### 3.1.8 Recommendations

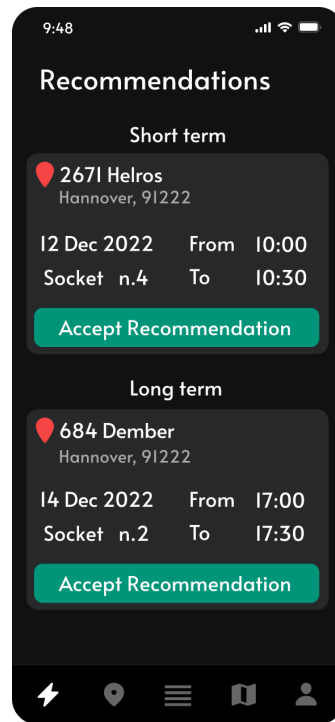


Fig. 20: UI of the MyRecommendations Page

### 3.1.9 Create Reservation

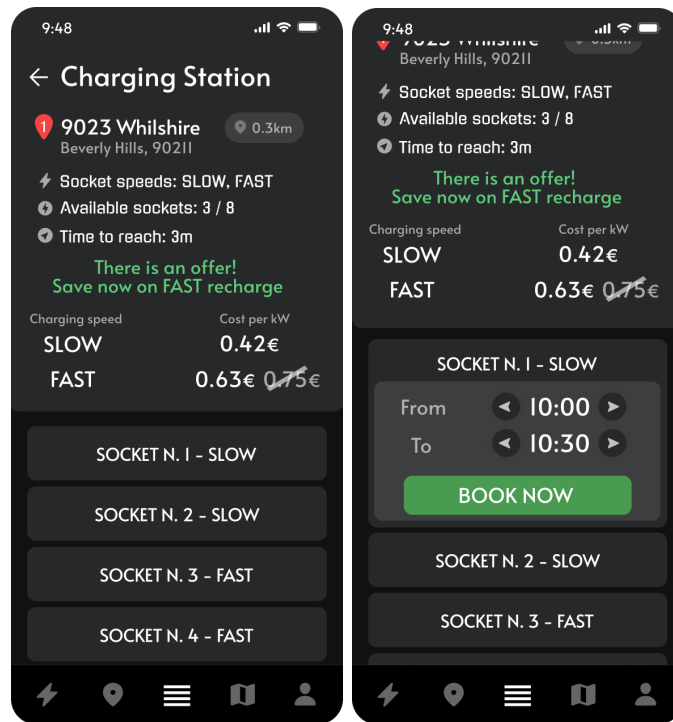


Fig. 21: UI after selecting a station (on the left) and after selecting a socket (on the right)

### 3.1.10 Charging Status

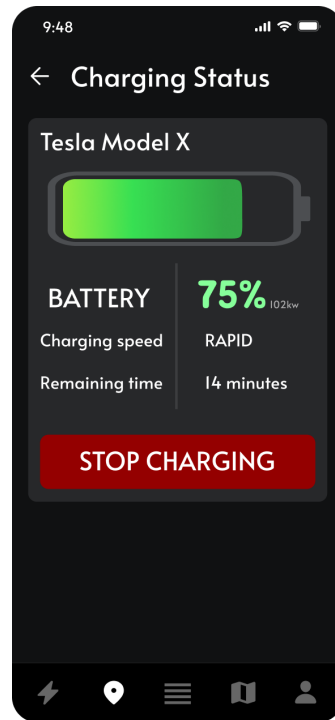
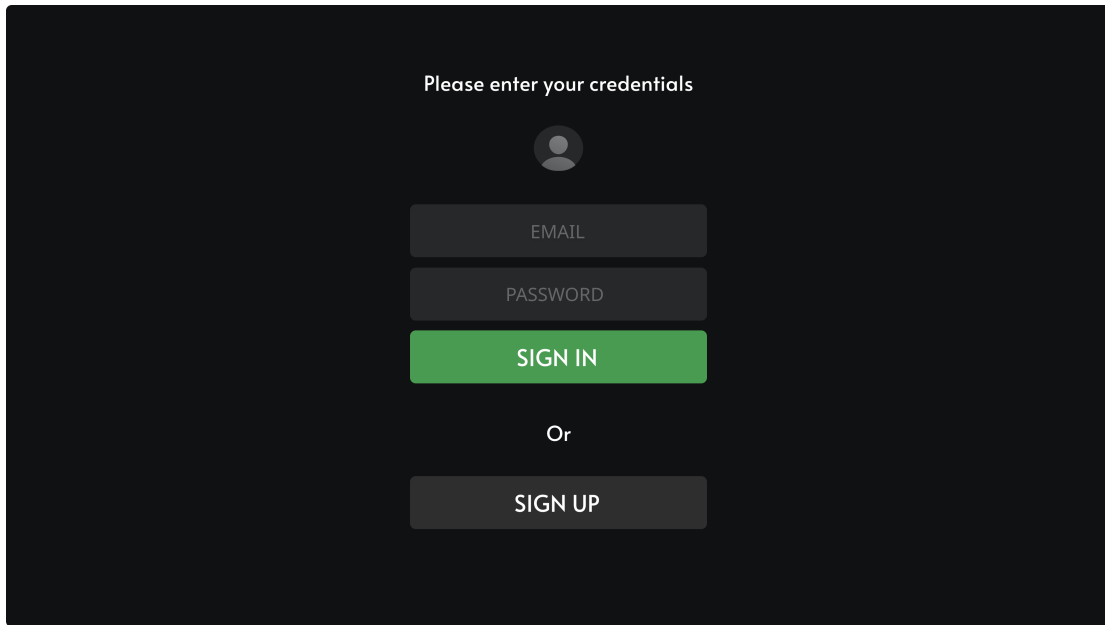



Fig. 22: UI of the ChargingStatus Page

## 3.2 Administrator

### 3.2.1 Login



Please enter your credentials



EMAIL

PASSWORD

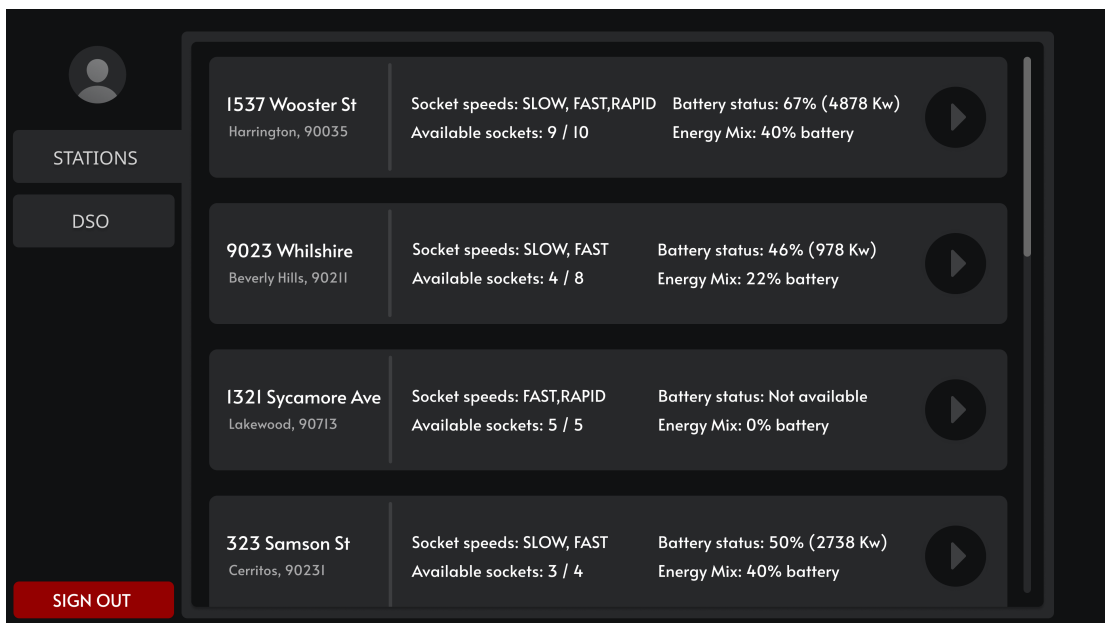
SIGN IN


Or

SIGN UP

Fig. 23: UI of the Login Page

### 3.2.2 Stations





STATIONS

DSO

SIGN OUT





<b>1537 Wooster St</b> Harrington, 90035	Socket speeds: SLOW, FAST, RAPID Available sockets: 9 / 10	Battery status: 67% (4878 Kw) Energy Mix: 40% battery	
<b>9023 Whilshire</b> Beverly Hills, 90211	Socket speeds: SLOW, FAST Available sockets: 4 / 8	Battery status: 46% (978 Kw) Energy Mix: 22% battery	
<b>1321 Sycamore Ave</b> Lakewood, 90713	Socket speeds: FAST, RAPID Available sockets: 5 / 5	Battery status: Not available Energy Mix: 0% battery	
<b>323 Samson St</b> Cerritos, 90231	Socket speeds: SLOW, FAST Available sockets: 3 / 4	Battery status: 50% (2738 Kw) Energy Mix: 40% battery	

Fig. 24: UI of the Stations Page

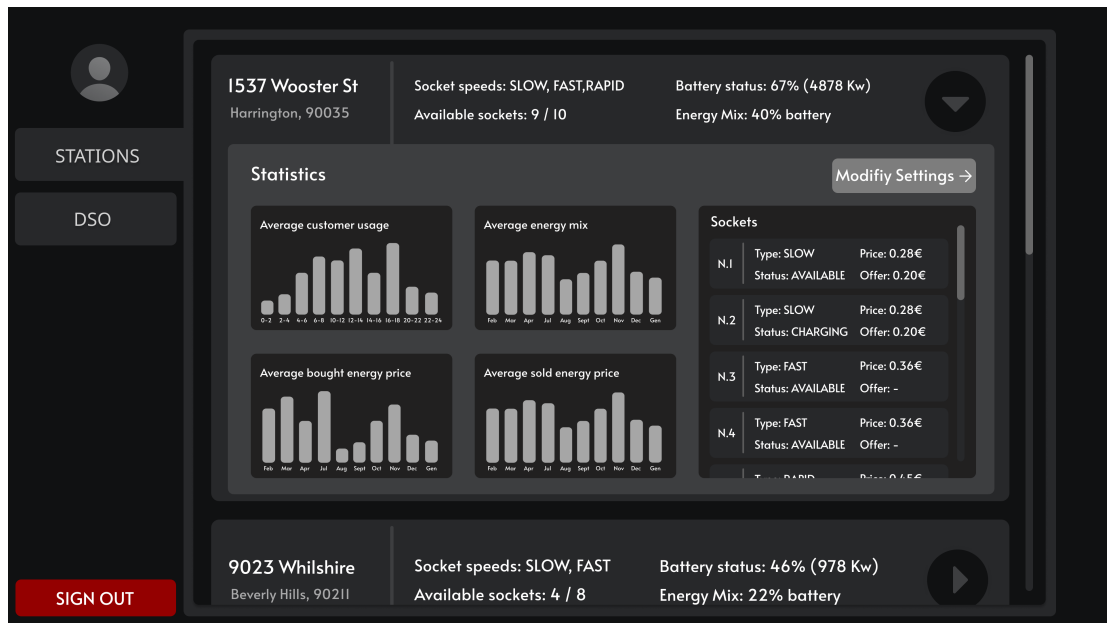


Fig. 25: UI of the Stations Page after selecting a station

### 3.2.3 Station Settings

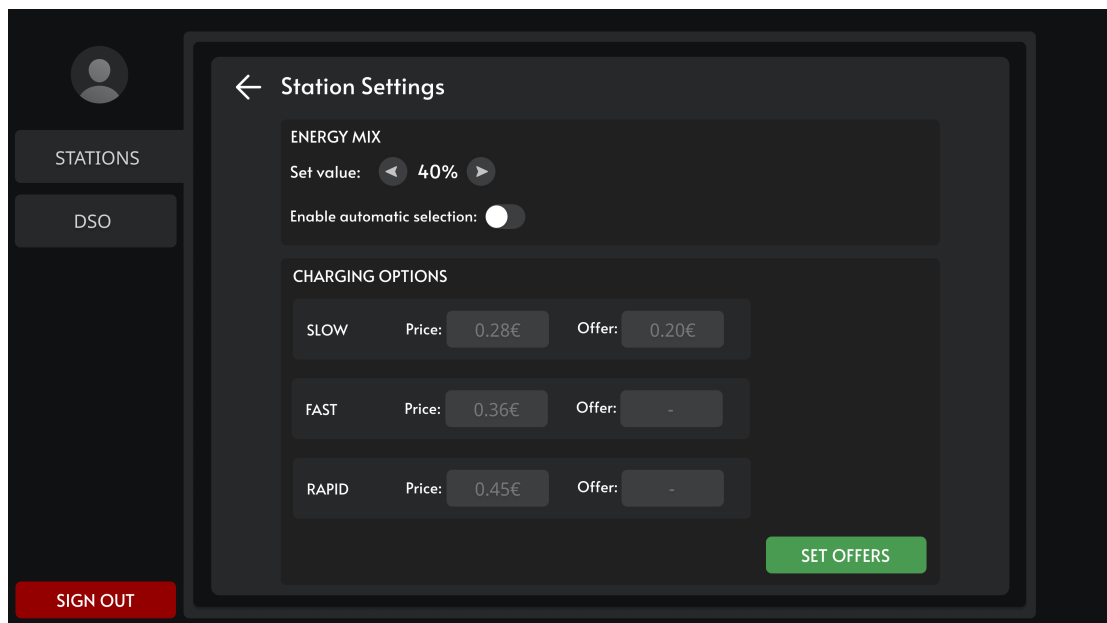


Fig. 26: UI of the Station Settings Page

## 3.2.4 DSO

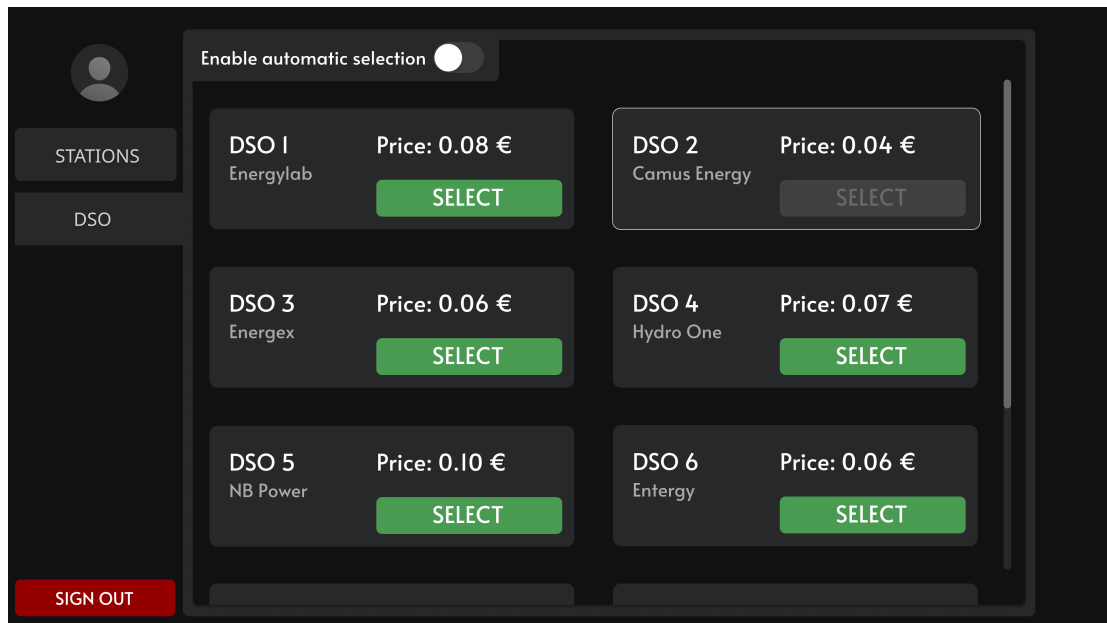


Fig. 27: UI of the DSO Page



## 4 Requirements traceability

This section contains a table explaining what components, according to their abbreviations specified in the list below, are required in order to fulfil each of the requirements specified in the *RASD*.

- **MC** - Mobile Client
- **WC** - Web Client
- **ACC** - Authentication Controller
- **SI** - Storage Interface eMSP
- **SI\*** - Storage Interface CPMS
- **DB** - Database eMSP
- **DB\*** - Database CPMS
- **PM** - Profile Manager
- **SM** - Station Manager eMSP
- **SM\*** - Station Manager CPMS
- **RSM** - Reservation Manager eMSP
- **RSM\*** - Reservation Manager CPMS
- **CM** - Charge Manager
- **REM** - Recommendation Manager
- **DM** - DSO Manager eMSP
- **DM\*** - DSO Manager CPMS
- **NM** - Notification Manager
- **AM** - Automation Manager
- **EAI** - External API Interfaces eMSP
- **EAI\*** - External API Interfaces CPMS

R	MC	WC	ACC	SI	SI*	DB	DB*	PM	SM	SM*	RSM	RSM*	CM	REM	DM	DM*	NM	AM	EAI	EAI*
R1	X		X	X		X													X	
R2	X			X		X		X											X	
R3	X			X		X		X											X	
R4	X								X	X									X	X
R5	X								X	X									X	X
R6	X								X	X									X	X
R7	X				X		X		X	X	X	X							X	X
R8	X				X		X		X	X	X	X							X	X
R9	X			X		X					X									
R10	X			X		X			X	X	X								X	X
R11	X									X			X						X	X
R12	X									X			X						X	X
R13	X									X			X				X		X	X
R14	X													X			X		X	X
R15	X													X			X		X	X
R16	X				X		X		X	X	X	X							X	X
R17		X		X	X	X	X		X	X									X	X
R18		X		X	X	X	X		X	X									X	X
R19		X		X	X	X	X		X	X									X	X
R20		X		X	X	X	X		X	X									X	X
R21		X		X	X	X	X		X	X					X	X		X	X	X
R22		X		X	X	X	X		X	X					X	X			X	X
R23		X		X		X			X											
R24																	X		X	
R25																	X		X	
R26				X	X	X	X												X	X
R27	X	X	X		X		X													
R28	X	X	X		X		X													

## 5 Implementation, Integration and test Plan

### 5.1 Implementation

The implementation of this system would follow a planned structure, dividing the components into various groups in order to maintain a clear development path.

The components that can be divided into the following categories:

- **Frontend components:** Mobile Client, Web Client.
- **Backend components:** Authentication Controller, Storage Interface eMSP, Storage Interface CPMS, Notification Manager, Profile Manager, Station Manager eMSP, Station Manager CPMS, Reservation Manager eMSP, Reservation Manager CPMS, Charge Manager, Recommendation Manager, DSO Manager eMSP, DSO Manager CPMS, Automation Manager.
- **External components:** Database eMSP, Database CPMS, eMSP API, CPMS API, Map API, Vehicle API, Station API, DSO API, GPS API, Payment API, Notification API.

In order to implement, integrate and test the System, a bottom-up strategy will be used. As previously explained, to support this strategy, the system has been divided

into smaller subsystems in order to implement and test their components separately. From the implementation and test of each subsystem is possible to obtain a reliable and modular product. External APIs are supposed to be reliable and without the necessity to be tested. The development process will follow this path:

1. Implementation and integration of different components of the same subsystem.
2. Integration of different subsystems (client and server application).

The subsystems (and the relative components) to be developed are:

- **Storage subsystem:** Storage Interface eMSP and Storage Interface CPMS.
- **Station subsystem:** Driver Station Interface, Driver Station Controller, Admin Station Interface, Admin Station Controller, Map Manager, List Manager, Station Controller and Station Settings.
- **DSO subsystem:** DSO Interface, DSO Controller eMSP and DSO Controller CPMS.
- **Automation subsystem:** Automation Controller, Automation Station, Automation DSO.
- **Reservation and Charge subsystem:** Reservation Interface, Reservation Controller, Charge Interface and Charge Controller.
- **Recommendation subsystem:** Recommendation Interface, Recommendation Controller and Data Analytic.
- **User subsystem:** Profile Interface, Profile Controller, Authorization Controller and Notification Manager.
- **Client:** Web Client and Mobile Client.

Moreover, the final subsystem integration that will be performed is between Business Logic and Client subsystem. It's important that the verification and validation processes starts as soon as the development of the system begins in order to find errors as quickly as possible. As specified before, will be used an incremental approach for integration

process in order to isolate bugs in single subsystems and don't propagate them. Each component of each subsystem is tested using Unit Testing.

## 5.2 Integration & Test plan

The first subsystem implemented is the backend. It contains all the subsystems previously introduced except for the Client. All the components are implemented and unit tested in parallel, following the group division previously described.

Each group is integrated and tested with the required external services, according to the Component View path, to check the behavior of the components when they work together.

Once finished this phase, components are integrated and will be performed integration testing.

Finally, the frontend will be integrated and tested with the backend. In particular the subsystem to be integrated with the Client Application are User subsystem, Station subsystem, Reservation subsystem and Recommendation subsystem.

The following diagrams describe the process of implementation, integration and testing.

- **Backend**

**Fig. 28:** *Storage subsystem*

**Fig. 29:** *Station subsystem*

**Fig. 30:** *DSO subsystem*

**Fig. 31:** *Automation subsystem*

**Fig. 32:** *Reservation and Charge subsystem*

**Fig. 33:** *Recommendation subsystem*

**Fig. 34:** *User subsystem*

- Client

**Fig. 35:** *Mobile and Web Client*

- System Integration

**Fig. 36:** *System final integration*

## 6 Effort Spent

### 6.0.1 Roberto Cialini

Section	Time spent
Introduction	0
Architectural design	4
User Interface Design	10
Requirements Traceability	0
Implementation, Integration and Test Plan	0
Reasoning	5
<b>Total time</b>	<b>19</b>

### 6.0.2 Umberto Colangelo

Section	Time spent
Introduction	2
Architectural design	6
User Interface Design	0
Requirements Traceability	2
Implementation, Integration and Test Plan	3
Reasoning	5
<b>Total time</b>	<b>18</b>

### 6.0.3 Vittorio La Ferla

Section	Time spent
Introduction	1
Architectural design	11
User Interface Design	0
Requirements Traceability	0
Implementation, Integration and Test Plan	0
Reasoning	5
<b>Total time</b>	<b>17</b>

## 7 References