POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

eMall – e-Mobility for all

Design Document - DD

Author(s): **Fabio Lusha - 10882532**
Bianca C. Savoiu Marinas - 10684465

Academic Year: 2022-2023

Contents

1 | Introduction

1.1. Purpose

Widespread electrification of transport is the most efficient way to reach Europe's climate objectives for the sector and electric charging is the main asset to overcome the obstacles of the take-up of electric vehicles (EVs). EVs can reduce CO2 by an estimated annual 600,000 tons by 2030, going towards a carbon neutral Europe, and the importance of this aim raises the problem of having efficient systems that manage the charging services. The eMall is thought as an all-encompassing application that oversees the entire process from the user interaction to the effective recharge of the EV's battery.

The main goal we want to achieve with the eMall software is to help the EVDs (electric vehicle drivers) to have better access to recharge and to be able to book a charging point in order to avoid interference with his daily plans. Another important purpose of the system is to safeguard not only the users but also the providers of the service and this is made through privacy agreements and the actual interaction, that guarantees to supervise both interested parts, in order to get the best possible service and pay for it accordingly, having also a technical and economic exploitation of the charging infrastructures.

In this context there is an increase in the requested electric energy, but large amounts of power in short periods would require investments in the reinforcement of the distribution networks, which have not been designed to accommodate such load. It becomes necessary to introduce new systems and solutions to optimize the operation of the distribution networks. In this context we can identify the DSOs as the suppliers of electricity through the distribution networks. The DSOs interact with the eMall, and in particular with the CPMS (Charging Point Management System) module of the system to be. The CPMS, then, gives the information about the DSO's supply to the CPOs, which are important actors, that use the system in order to manage the charging service. A CPO is represented by an employee or a software, part of the business that owns some charging stations and wants to manage them through the eMall, deciding from where to acquire energy, and how to establish the prices, the special offers and other details about the stations.

The eMall is thought as a software that manages both the interaction with the businesses that offer the charging service and the interaction with the EVDs which want to use these services in order to charge their EVs. Therefore, the eMall provides a mobile application (eMma), which through its interface allows to the EVD to obtain the service, and provides, also, a web application that the CPOs use to manage the charging stations. The EVD interacts, as well, with the charging point interface (eMci), that communicates with the CPMS part of the eMall, in order to start the charging session from the station, plugging then the car to the compatible connector to effectively charge the EV.

By the official definition of the IEEE Std 1016™-2009 standard, the DD is a representation of a software design that is to be used for recording design information, addressing various design concerns, and communicating that information to the design's stakeholders. So, in this document we focus on the design of the system to be, describing the components and the interaction among them and with external systems, trough interfaces, in order to achieve the goals and satisfy the requirements explained in the RASD document.

1.2. Scope

The eMall has two main stakeholders: the CPOs and the EVDs. There are different goals to satisfy respectively for the CPOs and the EVDs and different associated requirements to develop. For a detailed description of the domain in which the e-Mall will operate, and to see all the goals and the requirements, the reader should refer to the RASD. While, in this document, we briefly present the main design concerns of the stakeholders, and the main architectural styles.

The stakeholders have goals that, in order to be satisfied, need the storage of a large volume of data, therefore scalable technologies are required to manage the amount of information used by the system. Also, the stakeholders, especially the EVDs, are not expert users, thus in order to have an user-friendly system is necessary to create a graphical interface with which the users will interact. The system will have a GUI for the eMma and a GUI for the web application used by the CPOs, compatible with any browser, in order to facilitate the access to the system. We want also for the user interfaces to be compliant with different devices, regardless of the screen size.

This document will present as main adopted architectural style the 3-layered architecture, that we combine with other architectural choices. For the business logic layer, in fact, we

will present a micro-services architecture, while for the presentation layer we will adopt a client-side rendering architecture, in which the client (the web browser and the mobile app) is responsible for rendering the GUI of the application. These architectural choices will be further explained in the following chapter of the DD.

1.3. Definitions, Acronyms, Abbreviations

1.3.1. Abbreviations

- **eMall**: e-Mobility for all
- **eMma**: e-Mall mobile application
- **eMci**: e-Mall charger interface
- **CPMS**: Charging Point Management System
- **CPO**: Charge Point Operator
- **eMSP**: Electric Mobility Service Providers
- **DSO**: Distribution System Operator
- **OCPI**: Open Charge Point Interface
- **DBMS**: Database Management System
- **DBAL**: Database Abstraction Layer
- **OS**: Operating System
- **EV**: Electric Vehicle
- **EVD**: Electric Vehicle Driver
- **EVSE**: Electric Vehicle Supply Equipment
- **HV**: High Voltage
- **LV**: Low Voltage
- **MV**: Medium Voltage
- **AC**: Alternating current
- **DC**: Direct current
- **RASD**: Requirements Analysis and Specification Document

- **DD**: Design Document
- **UI**: User Interface
- **GUI**: Graphical User Interface

1.3.2. Definitions

- **DSO**: typically the entity responsible for the operation and management of distribution networks – High, Medium and Low Voltage networks.
- **CPO**: entity that technically manages all the EV infrastructure assets, depending of existing country regulation – this role can be assured by the DSO or other entity.
- **eMSP**: is the entity that can explore the economic side of the EV charging infrastructure, namely by selling energy for charging purposes.
- **CPMS**: is a software system that manages the charge point infrastructure – can manage the technical and economic aspects of the charging infrastructures.
- **EVD**: person or entity who owns an EV car and can use the public or private facilities for charging purposes.
- **EVSE (Electric Vehicle Supply Equipment)**: It is an equipment that is able to charge EV batteries with AC or DC loads and with different rated powers depending on the type of equipment.
- **Socket outlet**: the port on the electric vehicle supply equipment (EVSE) that supplies charging power to the vehicle
- **Plug**: the end of the flexible cable that interfaces with the socket outlet on the EVSE
- **Cable**: a flexible bundle of conductors that connects the EVSE with the electric vehicle
- **Connector**: the end of the flexible cable that interfaces with the vehicle inlet
- **Vehicle inlet**: the port on the electric vehicle that receives charging power
- **Rectifier**: an electrical device that converts alternating current (AC) to direct current (DC)
- **eMma**: the eMSP subsystem responsible for the EVD interaction from the mobile app

- **eMci**: the eMSP subsystem responsible for the EVD interaction at the charging point
- **additional costs**: overtime penalty, deposit for unregistered users
- **Status of the charger**: can be free, occupied, booked and in maintenance
- **Smart meter**: is an electronic device that records information such as consumption of electric energy, voltage levels, current, and power factor; allow the reading of energy flow and real-time usage, and consequently permit the identification of interruptions in energy flow
- **DD**: is an SDD (Software Design Description), which is a representation of a software design to be used for communicating design information to the stakeholders and also to guide the development of the system. The standard refers to this document as SDD, but in the following presentation we will call it DD
- **RASD**: is the document that analyzes and presents all the requirements of the system to be, explaining the domain in which the software will operate under some assumptions, and its interactions with the users

1.4. Reference Documents

- `IEEE Std 1016™-2009 International Standard for Information Technology - Systems Design - Software Design Descriptions`: describes how to structure a DD, giving a representation of a software design to be used for communicating design information to its stakeholders. The requirements for the design languages (notations and other representational schemas) to be used for conformant design documents are specified
- `ISO/IEC/IEEE 42010 International Standard - System and Software engineering - Architecture description`: this International Standard addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions. This International Standard provides a core ontology for the description of architectures. The provisions of this International Standard serve to enforce desired properties of architecture descriptions and also specifies architecture frameworks and architecture description languages (ADLs), in order to usefully support the development and use of architecture descriptions
- `RDD` assignment document
- `RASD` the document written for the e-Mall, which specifies the goals and the require-

ments the system to be has to achieve and analyzes the domain of the system

- Electric Vehicle CPMS and Secondary Substation Management by F. Campos, Efacec, Portugal; L. Marques, Efacec, Portugal and K. Kotsalos, Efacec, Portugal (15 October 2018): used to define the interactions between the different parts of the system and the actors; models the EV public infrastructures, the eMSP, the DSO and the CPMS together with the APIs and protocols that allow their communication

1.5. Document Structure

This document mainly follows the guidelines of the IEEE Std 1016™-2009 International Standard for Information Technology - Systems Design - Software Design Descriptions, especially the sections 4 and 5 of the document, ordering the parts in a way that fits best the topics of this document. The document is composed by the following sections:

- the first section, to which this part belongs, provides an introduction of the eMall, similar to the RASD document, introducing also the main architectural choices further explained in the following parts of the document
- the second section provides a specific description of the architecture, specifying the design decisions regarding the software to be; the section contains a formal description using UML diagrams to show the components of the eMall, the interfaces, and the interactions of the components among them and with external systems; we start with a high-level view of the system and then we decompose it in smaller parts showing their connections and dependencies; the section contains a component interface diagram, a deployment view and sequence diagrams describing the main interactions explained with the use cases of the RASD interactions between components.
- the third section provides mockups of the user interface, explaining the design of the GUI developed for the mobile app and the design of the GUI developed for the web application, describing the interaction of the stakeholders with the interfaces
- the fourth section provides a mapping between the components described in the previous sections and the requirements specified in the RASD, and a component is mapped on the requirement when it contributes to its fulfillment; in order for the document to be self contained we also report the requirements identified in the RASD, and we explain the mapping, thus the reader can understand how the provided architecture reaches the requirements

- the fifth section provides a plan to follow during the implementation of the eMall, specifying from which components to start, which components can be developed in parallel and provides, also, a plan for the integration of the components and the testing, from unit testing, to integration testing and finally the testing of the entire system
- the sixth section reports the effort spent by the components of the group to complete this document
- the last section contains the references used, beyond the ones already specified in this chapter

2 | Architectural design

2.1. Overview

In this section we will provide a high-level view and description of the components that our system is made of. The architecture chosen for our system is a three-tier one. The major advantages of this architectural style is the decoupling of the application logic from the presentation logic and the data persistence concerns. Further details about the characteristics of this architectural style will be given in section 2.6, for now we will proceed with the general overview of the system. In the picture below is illustrated a high-level view of the system with an informal notation, where each rectangular box represents a high-level computational unit of the system, meanwhile the double-edged arrow represents the interaction between two components.

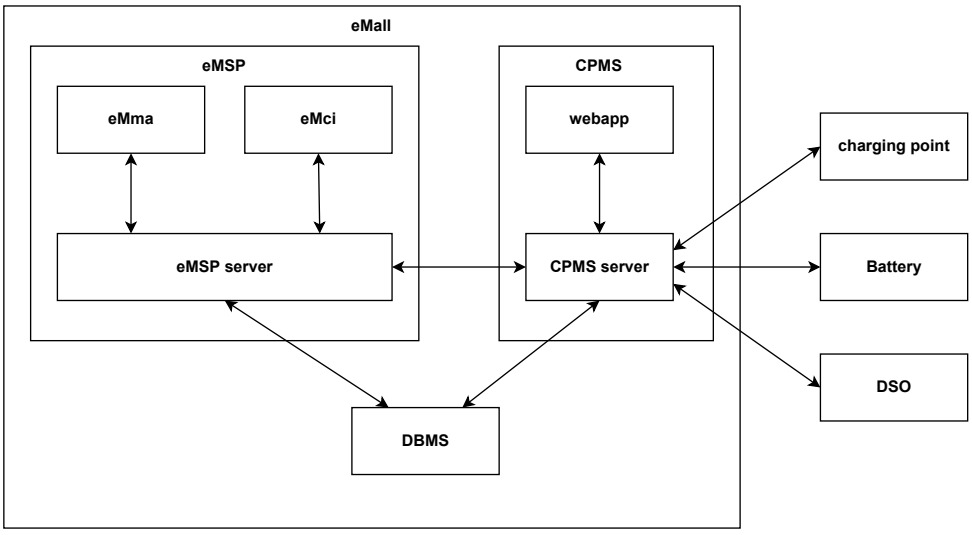

Figure 2.1: High level description of the components and their interactions

The eMall system, illustrated in the figure, is the objective of this design document. It is clear that the system is divided in two main sub-systems: eMSP and CPMS. This choice is driven by an interoperability requirement of the eMSP and the CPMS with different

CPMS and eMSP systems respectively, offered by other companies. Nonetheless, this low-coupling of the CPMS with the eMSP doesn't preclude us from reusing components that have the same functionality in both sub-systems. It is also important to point out that our system, specifically the CPMS sub-system, must be able to interface with the system responsible for managing the technical aspect of the charging point, the system that manages the battery, if present, and the DSO's software system.

As we stated previously, the system has a three-tier architecture. In particular the three tiers are:

Client tier It's the tier closest to the user and its duty is to manage the user interaction. This means that it must handle the visualization of the content to the user and interpret and translate the user interaction in requests to be forwarded to the application tier. We'd like to remark that this tier doesn't contain any application (or business) logic. We will use a client-side rendering software architecture to design this layer.

Application tier This is the part where the core and the business logic of the system is implemented, consequently, this second layer realizes the functionalities required to the system, like the booking service or the charging station management service for the CPO. All this functionalities shall be discussed in more detail in the upcoming sections. As we will see in the following section, a micro-services approach has been used to build such layer.

Data tier The third, and bottom tier, of our system is the data tier, where the persistence concerns of our system are met. The eMall system, both CPMS and eMSP sub-systems, has to handle a large amount of data, which must be carefully stored in order to have a properly working system. The data management is an aspect of software systems that has been thoroughly studied and developed, so the obvious choice for our system is to use an already implemented and tested Database Management System (DBMS).

2.2. Component view

In the section we will discuss and elaborate on the components that compose our system in order to implement the functionalities listed in the RASD document. We will start from a higher level of abstraction to provide a grasp of how the system works, and then we will proceed to further analyze the system at finer levels.

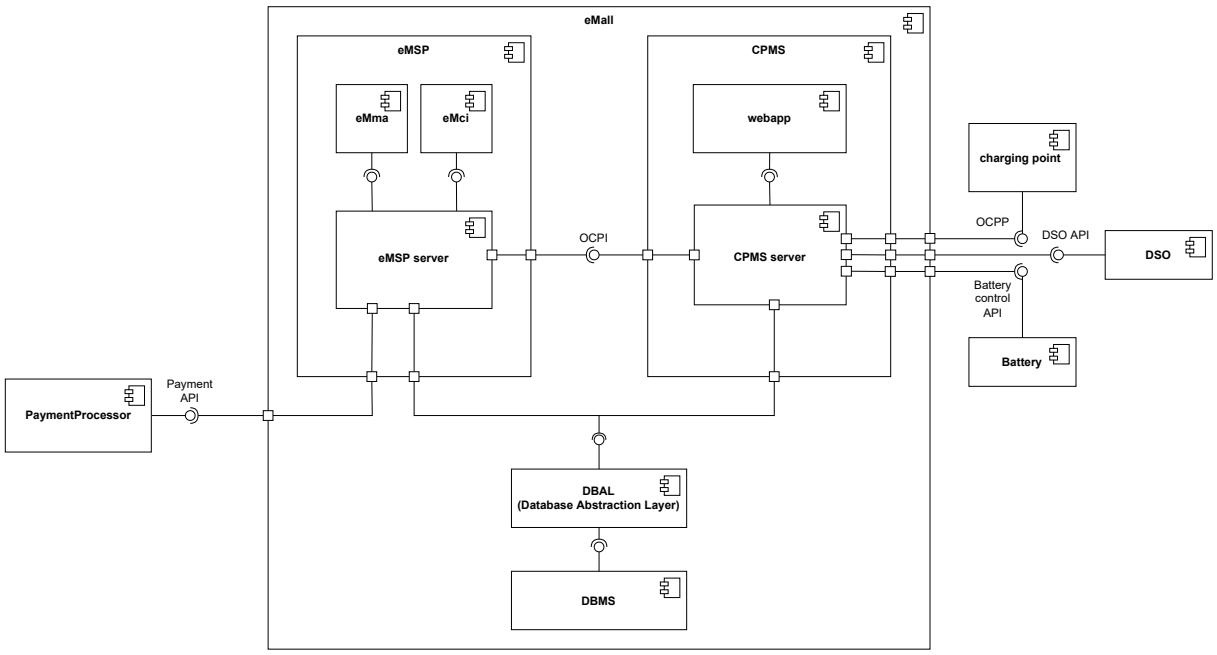

Figure 2.2: Component view of the system

This higher level view of the components gives a simple overview

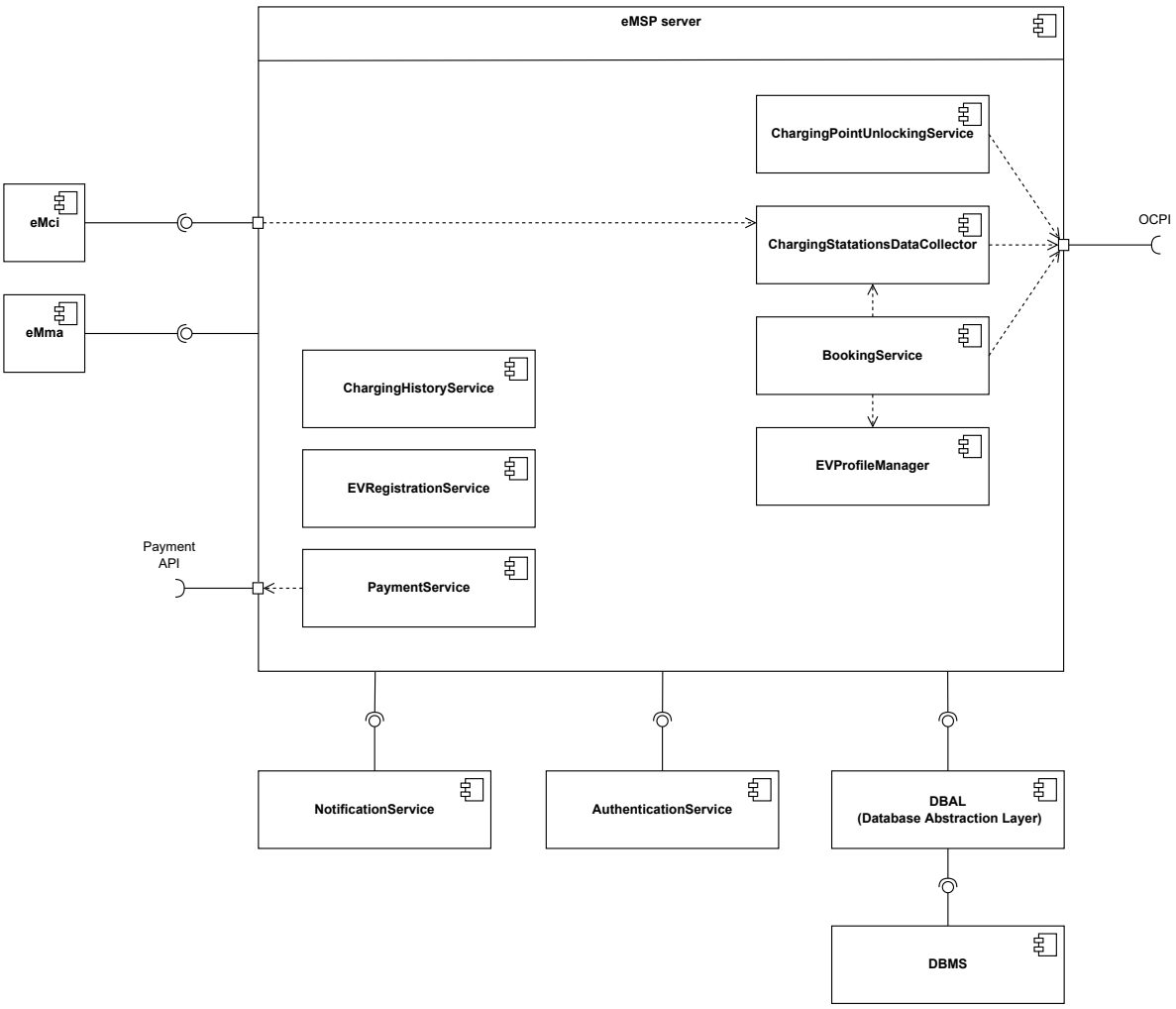

Figure 2.3: eMSP composite structure

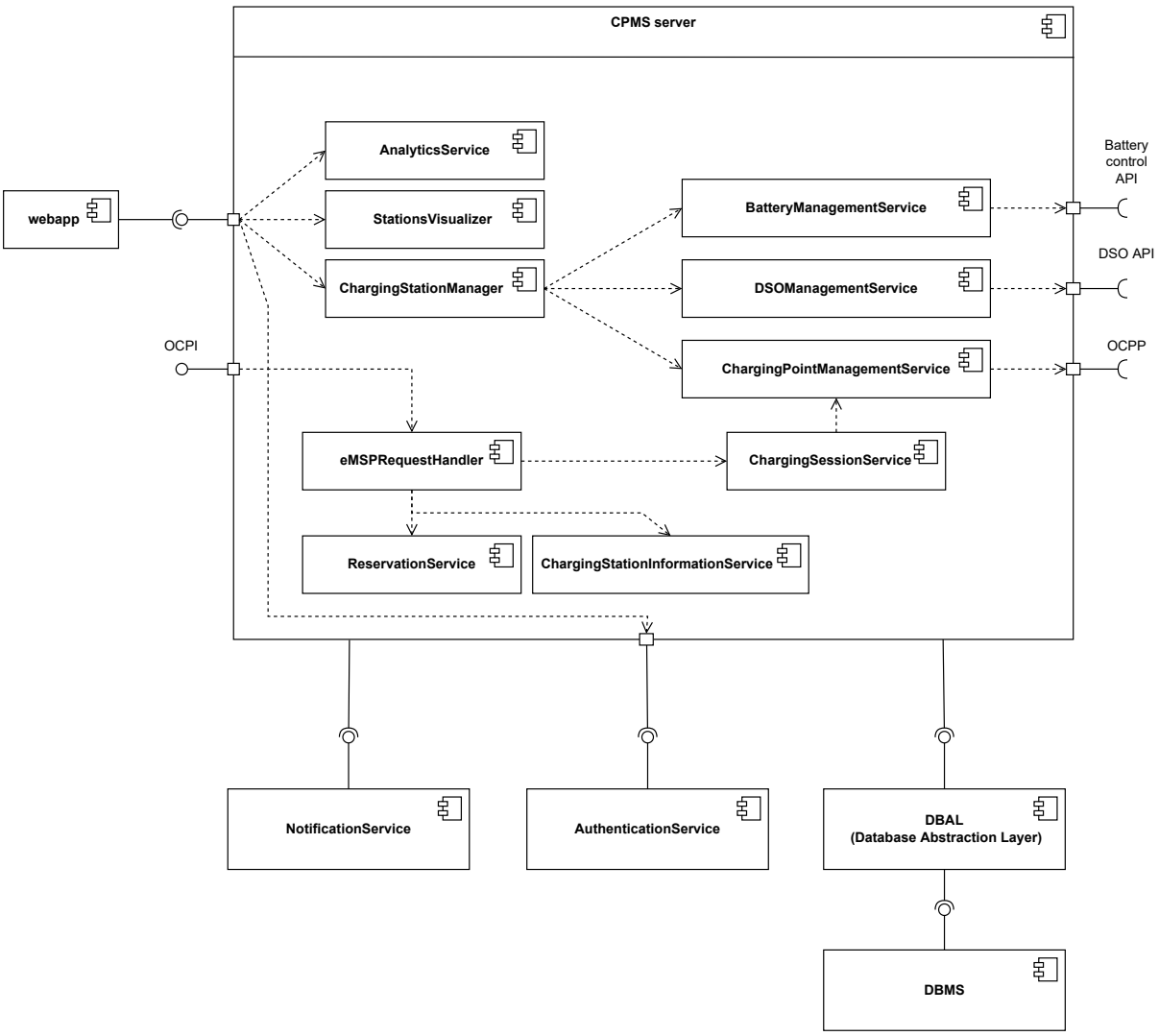

Figure 2.4: CPMS composite structure

2.3. Deployment view

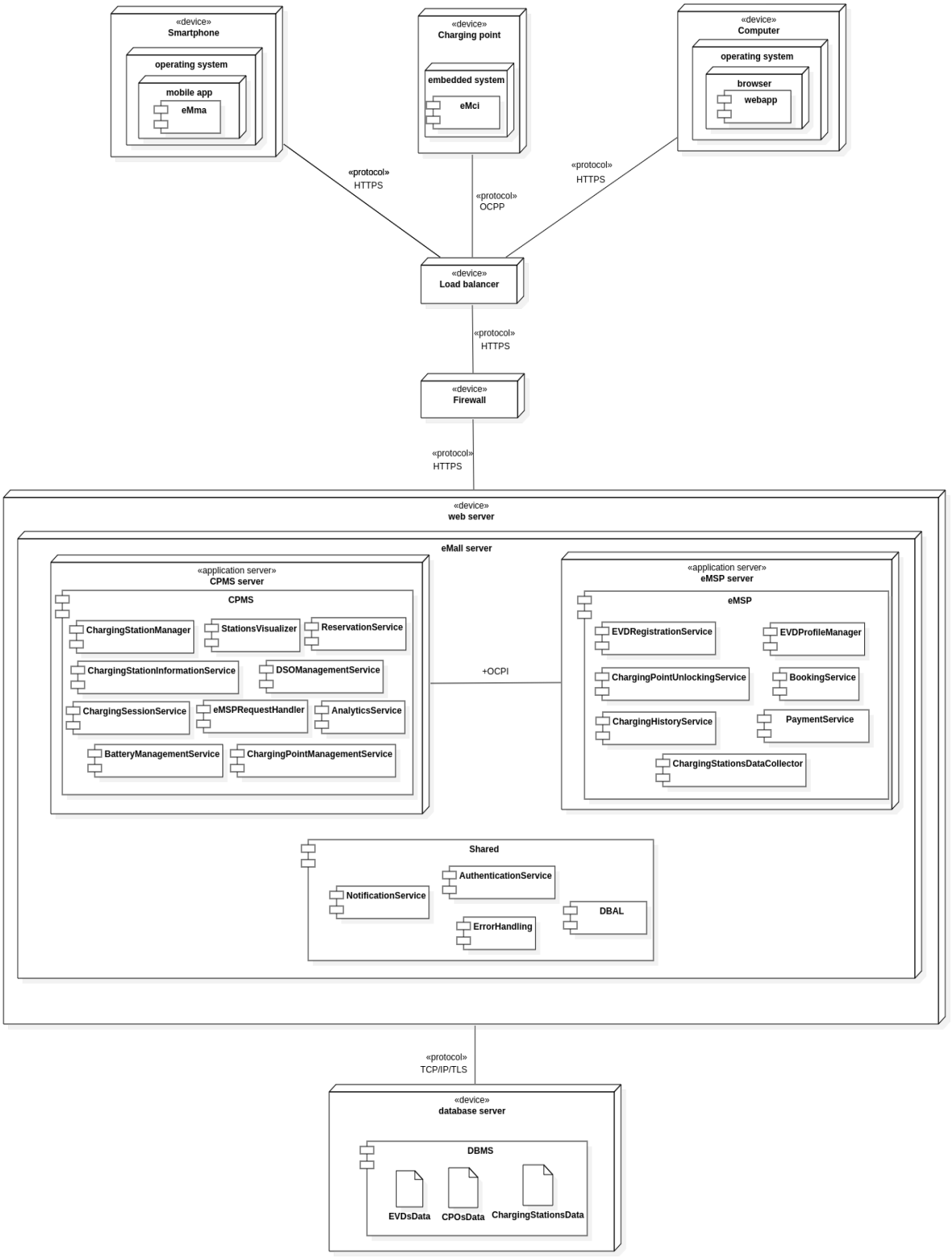

Figure 2.5: Deployment view of the system

At the client level we can see three different devices interacting with the system:

- **Smartphone** - The smartphone runs the mobile application of the eMall, the eMma, and has internet access in order to send the HTTPS requests to the system. This is the kind of device used by the EVDs that use the eMall
- **Computer** - The computer runs the web application of the eMall, and also has internet access to manage the service through HTTPS requests to the system. This is the kind of device used by the CPOs of the companies that use the eMall to manage their charging stations
- **Charging Point** - The charging point is the specific device used by the EVDs to charge the EVs and it has an embedded system, which through the OCPP protocol communicates with the CPMS part of the eMall system, in order to correctly provide the charging service

Between the client level and the application level, we have some architectural elements that allow to achieve some non-functional requirements, such as better performance, scalability, availability and security:

- **Load balancer** - The load balancer is a network device that distributes incoming requests across a group of servers to help improve the performance and availability of the application. A load balancer can help to improve the performance of the application by distributing incoming requests across multiple servers, rather than routing all requests to a single server. This can help to prevent any single server from becoming overloaded, which can improve the overall responsiveness and performance of the application. A load balancer can make it easier to scale the application horizontally by allowing to add or remove servers as needed and this can be useful if we need to add more capacity to handle a growing number of users. A load balancer can, also, help to improve the availability of the application by routing traffic to a healthy server in the event that one of the servers becomes unavailable. We can see that the load balancer is useful to improve different non-functional requirements of the eMall, that can be implemented using different servers to have better characteristics
- **Firewall** - The firewall allows to protect the network from external threats and unauthorized access, blocking incoming traffic that does not meet the security rules. The firewall is necessary in order to comply with certain regulations and industry standards, because we are handling sensitive data (financial information, personal data), so is necessary to protect the data. The firewall can also help to improve the

performance of the application by blocking traffic that is not necessary or relevant to the application, improving the overall responsiveness of the eMall

The eMall web application and mobile application provide both static and dynamically generated content, so the system runs web servers for the static content and application servers to generate content dynamically. The load balancer sends to the web server the HTTPS requests that need only static content, and on the other side sends to the correct application server the requests to generate the dynamic content and accomplish more complex functionalities due to the interaction of the eMall components. At the application level the deployment diagram shows in a simplistic way the following elements:

- **Web server** - For the web server we have a computer that stores software and website raw data, such as HTML files, images, text documents, and JavaScript files. The hardware of the web server is connected to the web and supports the data exchange with different devices connected to the Internet
- **Application server** - In the deployment diagram on the same hardware we also have the application servers, one for the CPMS and one for the eMSP, with also the shared services. The application servers contain different micro-services, that interact among them and with external APIs, as shown in the previous component diagram. The micro-services could also be implemented on more servers, splitting the eMSP and CPMS application servers in more servers, or creating a redundancy of the available services on different machines to improve performance and availability, exploiting even better the load balancer
- **Shared components** - The shared components shown in the diagram are components that belong to the two application servers, but also to the web server. The web server of the eMall handles part of these functionalities, sending the rest to the application servers

Finally at the lowest level we have the persistent part of the system, which interacts with the eMall trough TCP/IP and is hidden to the higher levels due to the use of the DBAL. The Database level is composed by:

- **Database server** - It hosts the database of the system and manages the different data through a DBMS
- **Database artifacts** - The different database artifacts shown in the diagram represent physical implementations of the DB, an implementation for the data regarding the EVDs, one for the companies and CPOs data and finally an implementation for the data regarding the charging stations and any other related data

For a more secure system we consider not only a DB, but also some replicas, distributed on different machines to guarantee more availability, fault tolerance and disaster recovery.

2.4. Runtime view

2.5. Component interfaces

2.6. Selected architectural styles and patterns

2.7. Other design decisions

3 | User Interface Design

3.1. EVD user interface

3.1.1. Log in

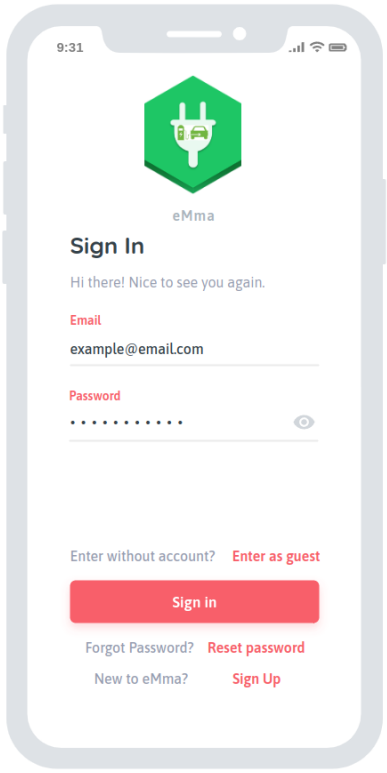

Figure 3.1: UI for the log in

We can see from the UI for the log in that the operation is very simple. To sign up is enough to enter the email and the password used during the registration. The EVD can enter into eMma as a guest, if he doesn't want to sign up, but in that case it will perceive only some limited functionalities of the application. It can also be useful to set a mechanism to manage the case in which the user forgets his password.

3.1.2. Register

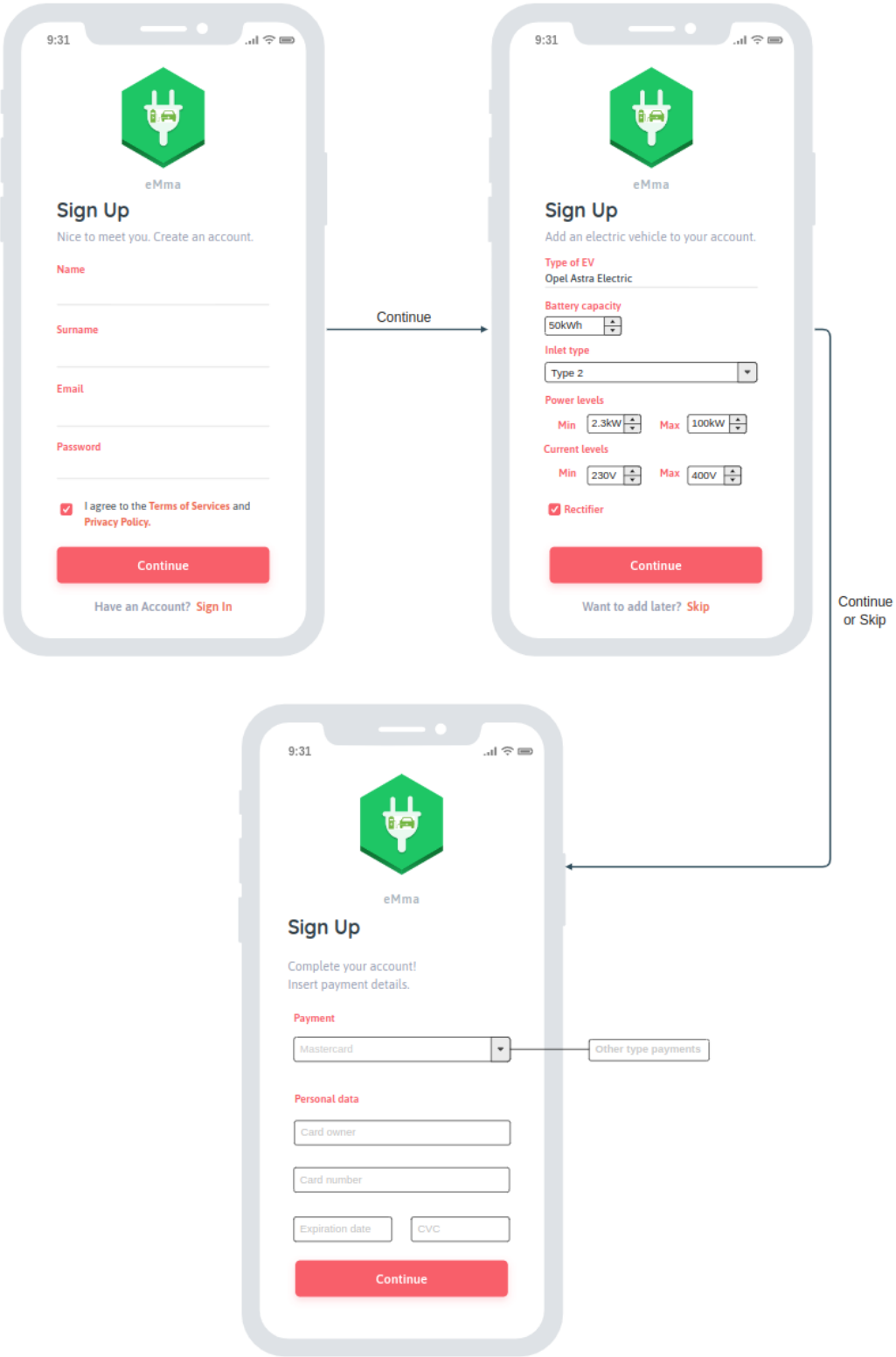

Figure 3.2: UI for registration

The UI for the registration was thought in order to be as user-friendly as possible. First, to sign up is necessary to insert some personal data, such as the name, the surname, and the email and password that will be used for the log in, as previously explained. It is also necessary to accept the 'Term of Service' in order to continue with the registration. The second step asks the user to add some data about his EV and we can see that the insertion is guided by the interface in order to avoid as much as possible data errors. The EVD has to add the type of the EV, the inlet type, the power levels and the current levels, and has to check or not the final box to inform if the car has or not the rectifier. Finally, to complete the sign up process the user is requested to insert the payment details. After registration, from the personal profile the EVD will be able to add new data to his profile, such as new payment methods and new EVs, and he will be able to modify these ones.

3.1.3. Charge now

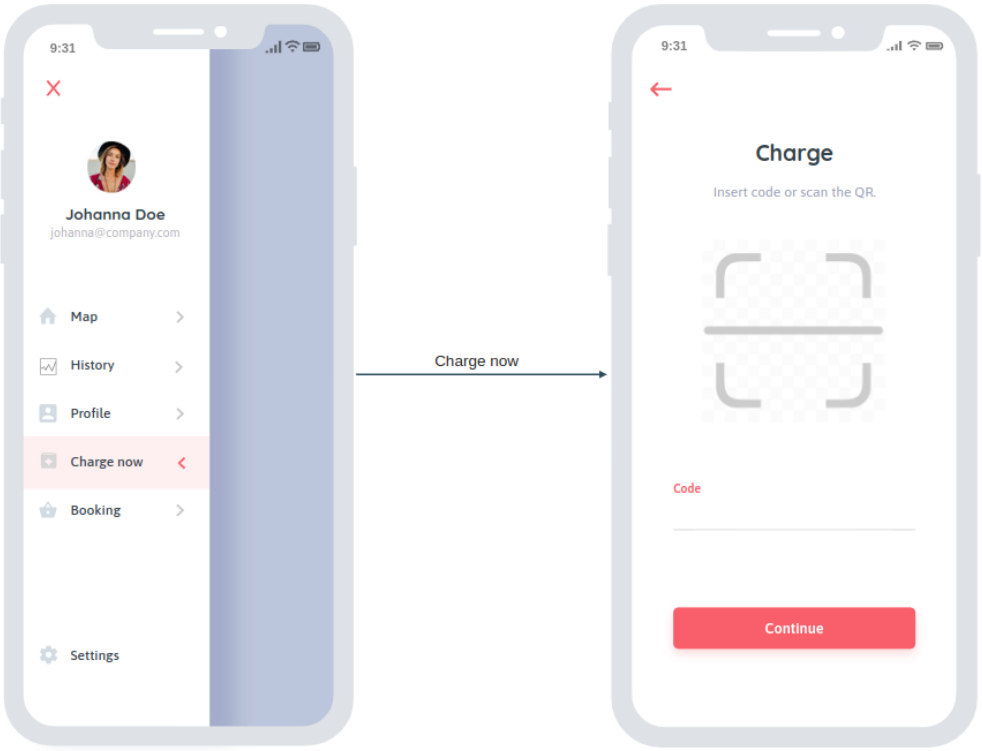

Figure 3.3: UI for starting the charging session

To start the charging process the EVD has to insert the code or scan the QR present on the charging point. This can be started from the menu or from the 'Charge now' button present in the charging station page, as we will see in the following mock-ups.

3.1.4. Visualize stations

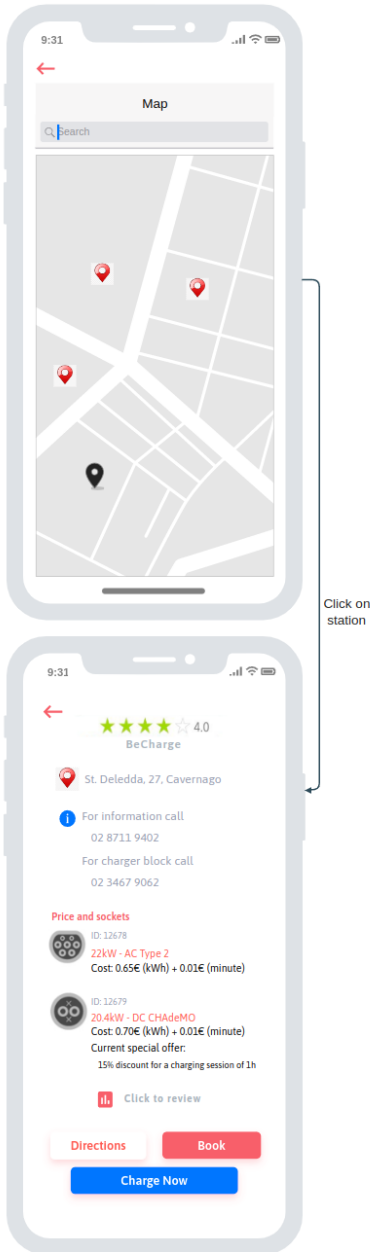

Figure 3.4: UI to visualize stations on the map and their information

The main page of the eMma shows to the user the map and permits to visualize the charging stations nearby or to insert a position in which to search for new stations. Clicking on one of the stations, the application shows the charging station page with the most important information: the rating, the name of the station, the address and the contacts, the available sockets and the respective prices and special offers. Finally, it is possible to leave a review or to see the ones left by other users. From this page is

possible to get the directions to reach the station and it is also possible to start the main operations of the system: the charging session and the booking of a charging point.

3.1.5. Booking charging point

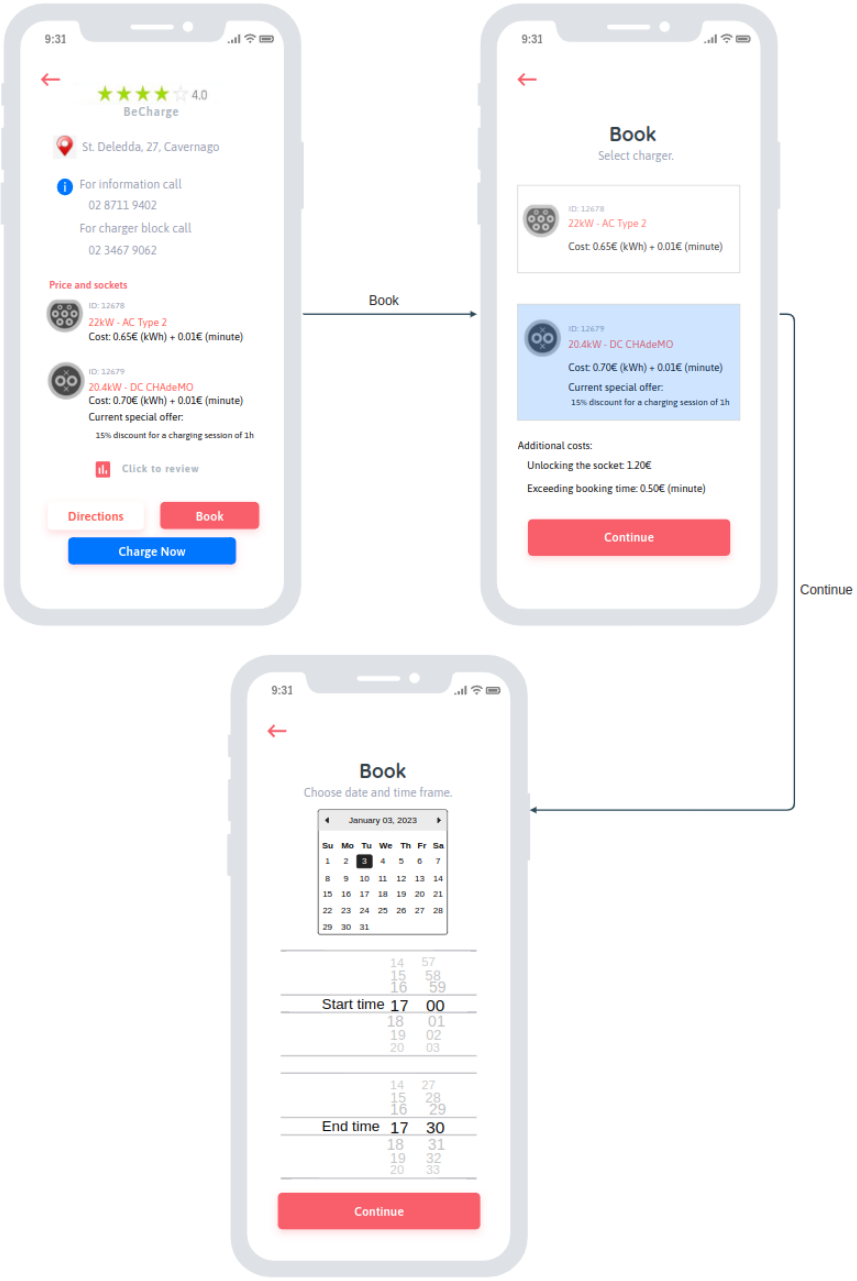

Figure 3.5: UI to book the charging point in a certain time frame

During the booking operation the EVD has to select the charging point he wants to book from the selected station and also to insert the date and the time frame of the booking.

3.1.6. Terminate charging

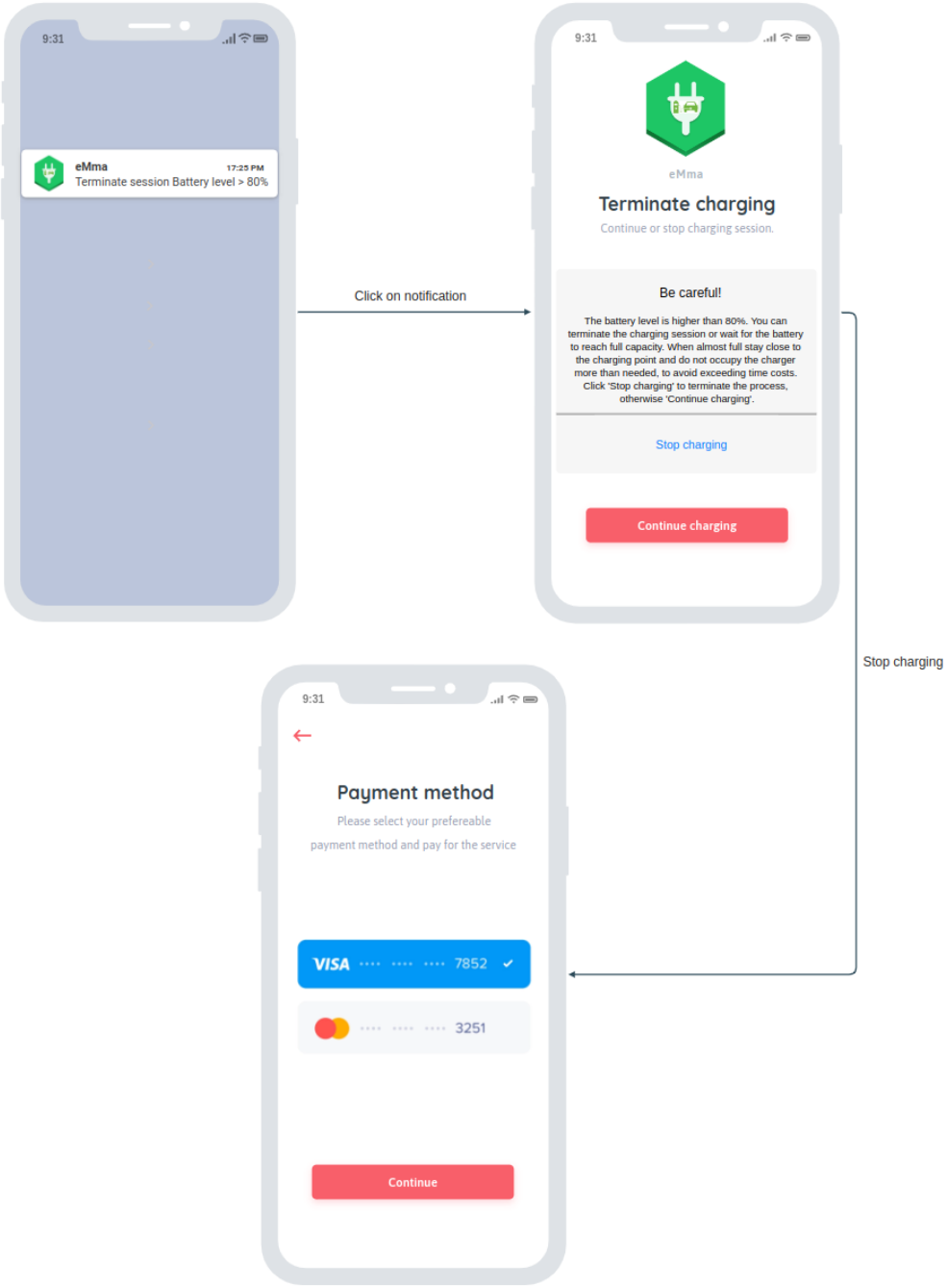

Figure 3.6: UI to terminate the charging session

We can see from this UI, that the eMma sends a notification to the user smartphone when the battery level reaches 80%. The user can decide if he wants to stop the charging session or proceed with the charging until full battery. When the EVD decides to terminate the charging the eMma asks to select the payment method in order to pay for the service.

3.2. CPO user interface

4 | Requirements traceability

4.1. Requirements

Requirement	Description
R1	The system shall allow an unregistered EVD to register an account
R2	The system shall allow a registered EVD to insert data about his EVs
R3	The system shall allow a registered EVD to update EV's details
R4	The system shall allow a registered EVD to add new EVs
R5	The system shall allow a registered EVD to delete an EV
R6	The system shall allow a registered EVD to insert personal data (name, surname, email, payment details)
R7	The system shall allow a registered EVD to update personal data
R8	The system shall allow an EVD to view the charging stations on the map
R9	The system shall allow an EVD to view relevant data about the charging stations
R10	The system shall allow an EVD to view relevant data about a specific charging point
R11	The system shall allow an EVD to view the prices of the charging points of the stations
R12	The system shall allow an EVD to view the special offers of the charging points of the stations
R13	The system shall allow a registered EVD to review a charging station
R14	The system shall allow an EVD to share his location through the GPS module of his mobile phone
R15	The system shall allow an EVD to choose the area in which to visualize the charging stations on the map, if different from the actual GPS location

R16	The system by default must show on the homepage the charging stations on the map territory, centered on the EVD's location
R17	The system shall allow a registered EVD to book a charge from an available charging point in a charging station
R18	The system shall allow a registered EVD to choose the time-frame (date and time) in which to book a charge
R19	The system shall allow a registered EVD to cancel a charging point booking
R20	The system shall allow a registered EVD to visualize his profile data
R21	The system shall allow a registered EVD to view the charging history
R22	The system shall allow a registered EVD to choose a visualization criteria for the charging history
R23	The system shall allow an EVD to start a charging session
R24	The system shall allow a registered EVD to insert the charging point code, in order to start a charging session
R25	The system shall allow an EVD to choose the payment method to use in order to pay for the obtained service
R26	The system must allow the registered EVD to log in
R27	The system must allow the CPO to log in with the company credentials
R28	The system must allow the EVD to access the mobile application, eMma, with or without an account
R29	The system must store the charging history: previous charges and bookings related to the specific EV
R30	The system must collect electric energy data from the DSOs
R31	The system must collect the data about the charging stations from the CPOs
R32	The system must notify the EVD with a specific message, that clarifies the problem, if an error occurs (login error, update error, payment error, ecc.)
R33	The system must notify the EVD with a success message if the operation terminated without errors (successful registration, successful profile modification, ecc.)
R34	The system must ask for EVD's consent to use the location information given by the GPS module

R35	The system must ask the EVD, during registration, to agree to the terms of service
R36	The system must check for the correctness of the data inserted by the EVD (login details, charging point code, ecc.)
R37	The system must show only the free charging points to book
R38	The system must show for each available charging point the relevant information (type of charging, type of socket, charging speed, price, special offers, etc.)
R39	The system must show a summary of the successful booking operation
R40	The system must change the status of the charging point during charging
R41	The system must unlock the charging point if the code is correct
R42	The system shall allow the CPO to view the charging stations
R43	The system shall allow the CPO to view any notification regarding the charging stations
R44	The system shall allow the CPO to update the details of a charging station
R45	The system shall allow the CPO to delete a charging station
R46	The system shall allow the CPO to add a charging station
R47	The system shall allow the CPO to delete a charging point from a charging station
R48	The system shall allow the CPO to add a charging point to a charging station
R49	The system shall allow the CPO to set the price of the charging point
R50	The system shall allow the CPO to set a special offer for the charging point
R51	The system must check the correctness of the data inserted by the CPO
R52	The system must store the data of the charging stations
R53	The system must notify the CPO with a specific message if an error occurs during an operation
R54	The system must notify the CPO with a success message if the operation terminates without errors
R55	The system shall allow the CPO to select a charging station
R56	The system shall allow the CPO to view the DSO's updated prices for the energy sources

R57	The system shall allow the CPO to view the DSO's special offers for the energy sources
R58	The system shall allow the CPO to change the DSO of a charging station
R59	The system shall allow the CPO to choose the DSO's energy source for the charging station
R60	The system shall allow the CPO to set a special offer for the charging station
R61	The system shall allow the CPO to select some criteria to graphically visualize aspects of the charging stations
R62	The system must show a graphical representation of some aspects of the charging stations
R63	The system shall allow the registered EVD to view his upcoming bookings
R64	The system shall allow the CPO to select a notification to solve it
R65	The system shall allow the CPO to decide if for a charging station wants to store energy from the DSO in the battery of the station
R66	The system shall allow the CPO to decide if for a charging station wants to use the battery or acquire energy from the DSO
R67	The system shall allow an EVD to charge the battery until full capacity
R68	The system must communicate with external APIs to complete the payment

Table 4.1: Requirements

4.2. EVD requirements traceability

4.3. CPO requirements traceability

Components:

- ANS - AnalyticsService
- SV - StationsVisualizer
- CSM - ChargingStationManager
- BMS - BatteryManagementService

- DMS - DSOManagementService
- CPS - ChargingPointManagementService
- NS - NotificationService
- AS - AutheticationService
- DBAL - Database Abastraction Layer
- DBMS - Database Management System

R	**AS**	**ANS**	**SV**	**CSM**	**DMS**	**CPS**	**NS**	**DBAL**	**DBMS**
R27									
R30									
R31									
R42									
R43									
R44									
R45									
R46									
R47									
R48									
R49									
R50									
R51									
R52									
R53									
R54									
R55									
R56									
R57									
R58									
R59									
R60									
R61									
R62									
R64									
R65									

R66									

Table 4.2: CPO requirements traceability

5 | Implementation, integration and test plan

6 | Effort spent

Activity	Time spent
Organization	3h
Introduction	3h
Architectural design	4h
User interface design	5h
Requirements traceability	1h
Implementation, integration and test plan	h
Total time spent	h

Table 6.1: The time Bianca Savoiu has spent working on this project

Activity	Time spent
Organization	3h
Introduction	h
Architectural design	h
User interface design	h
Requirements traceability	h
Implementation, integration and test plan	h
Total time spent	h

Table 6.2: The time Fabio Lusha has spent working on this project

7 | References

Fowler, M. (2003). *UML Distilled*. Addison-Wesley.

Larman, C. (2005). *Applying UML and Patterns*. Prentice Hall PTR.

List of Figures

List of Tables