DD

Langze YE，January 2023

Contents

1 Introduction 2
  1.1 Purpose 2
  1.2 Scope 2
  1.3 Definitions, Acronyms, Abbreviations 2
  1.3.1 Definitions 2
  1.3.2 Abbreviations 3
  1.4 Revision History 3
  1.5 Reference Documents 3
  1.6 Document Structure 3
2 Architectural designs 4
  2.1 Overview: High-level components and their interaction 4
  2.2 Component view 5
  2.3 Deployment view 6
  2.4 Runtime view 7
  2.4.1 Sequence diagrams for eMSP 7
  2.4.2 Sequence diagrams for CPMS 10
  2.5 Component interfaces 11
  2.6 Selected architectural styles and patterns 12
  2.7 Other design decisions 13
3 User interface design 13
  3.1 Users 13
  3.2 CPO 16
4 Requirements traceability 17
5 Implementation, integration and test plan 18
  5.1 Implementation 18
  5.2 Integration & Test plan 19
6 Effort spent 21
7 References 21

1 Introduction

1.1 Purpose

With the development of new means of transportation, electric vehicles have become one of the most popular choices of more people. At the same time, more and more charging stations come into our cities and villages.

To reduce the carbon footprint caused by our urban and sub-urban mobility needs and to facilitate the life of electric vehicles drivers, our new system, E-Mall will play an important role which can help the drivers solving their charging problems. What's more, it will also provide Charging Point Operators (CPOs) the information of charging station, such as the number of charging sockets available. That means E-Mall will also contain the Charge Point Management System.

This document contains a description of the architectural design for the system, including the components involved and how they interact. What's more, some mockups of the user interfaces and the plan for the implementation and integration will also be presented.

1.2 Scope

As mentioned above, E-Mall contains both e-Mobility Service Provider (eMSPs) and the Charge Point Management System (CPMS). Although sometimes Distribution System Operator will be mentioned but we won't consider his interests.

Additionally, both systems will use a three-tier architecture so that the business logic of the system will be separated from the data. Therefore, each layer in the overall structure can be independently developed by different teams. At the same time, future updates of the entire system will also be easier.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Definition	**Description**
Users	Normally it means the electric vehicle drivers.
Operators	Normally it means Charging Point Operators.

1.3.2 Abbreviations

Abbreviation	**Description**
RASD	Requirements Analysis and Specification Document
WP	World Phenomena
SP	Shared Phenomena
eMSPs	e-Mobility Service Providers
CPOs	Charging Point Operators
CPMS	Charge Point Management System
DSOs	Distribution System Operators
DX	Domain assumption number X
GX	Goal number X
RX	Requirement number X

1.4 Revision History

1.5 Reference Documents

- The specification document "01. Assignment RDD AY 2022-2023.pdf"
- RASD.pdf

1.6 Document Structure

The following file contains the remaining 6 sections (section 2 to section 7):

- Section 2 focus on the system architecture design. It starts with an overview of high-level components and their interaction. Then all the components will be described and a component view will show their inter independence. What's more, this section also contains a deployment view, the component interfaces and some sequence diagrams.
- Section 3 will present some mockups of the user interfaces and their connection.
- Section 4 will talk about the requirements traceability by a table which shows the components required for fulfilling each requirement.
- Section 5 describes the suggested implementation and test plan.
- Section 6 contains my effort spent on this report.
- Section 7 presents the reference used.

2 Architectural designs

2.1 Overview: High-level components and their interaction

As mention above in 1.2, the architecture of E-Mall follows the three-tier pattern with a presentation layer, business logic layer and a data layer. With the development of electric vehicles, the number of charging stations in cities and villages will increase day by day. The functions of the charging station will also be more abundant. Therefore, frequent system updates are inevitable. Using a three-tier architecture can make subsequent system updates easier. At the same time, each layer can be independently developed. Moreover, the three-tier architecture also contributes to the improvement of system security.
It is worth noting that eMSP and CPMS are two different systems but they will achieve data interaction through database sharing.
Following is a short description of each tier and an overview of the two systems' three-tier architecture:

• **Presentation tier**

The presentation tier, which also means the client tier, focuses on the interaction with the user. This layer is just a simple which will only communicate with the business logic server.

• **Business logic tier**

The business logic tier contains all the business logic of the system, for example, the calculation of charging time. When it receives instructions from the presentation layer, it will connect the data layer and perform an operation on specific issues. In other words, it will realize the data connection and instruction communication between the three layers.
What's more, some external services, for example the remaining energy of every socket in a charging station, will be also taken care of by the business logic layer.

• **Data tier**

The data layer is mainly responsible for processing the data of the system and it should be able to feedback the result of the operation to the business logic layer.

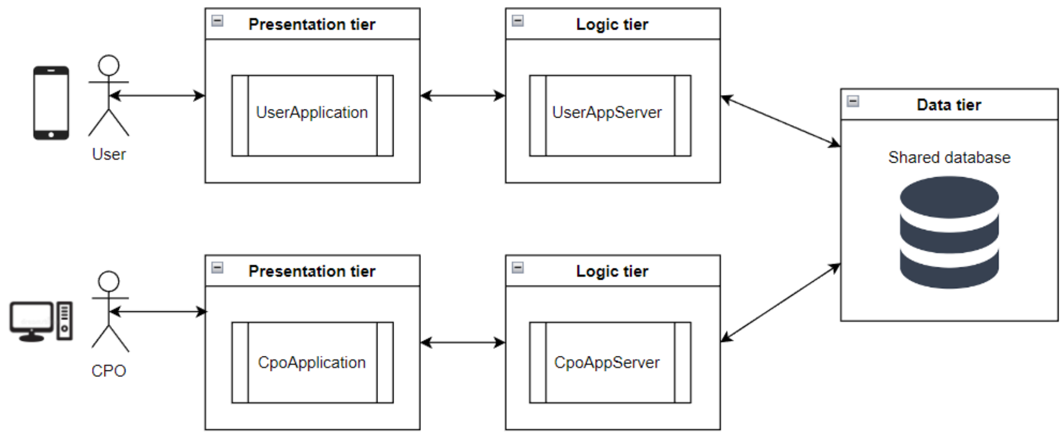


Figure 1:Overview of the systems' three-tier architecture

2.2 Component view

In this section, the component view will be presented with diagrams. The component diagrams are based on the overview diagram which show the two systems' three-tier architecture and the shared database. What's more, the component diagram will pay more attention to the components of business logic tier.

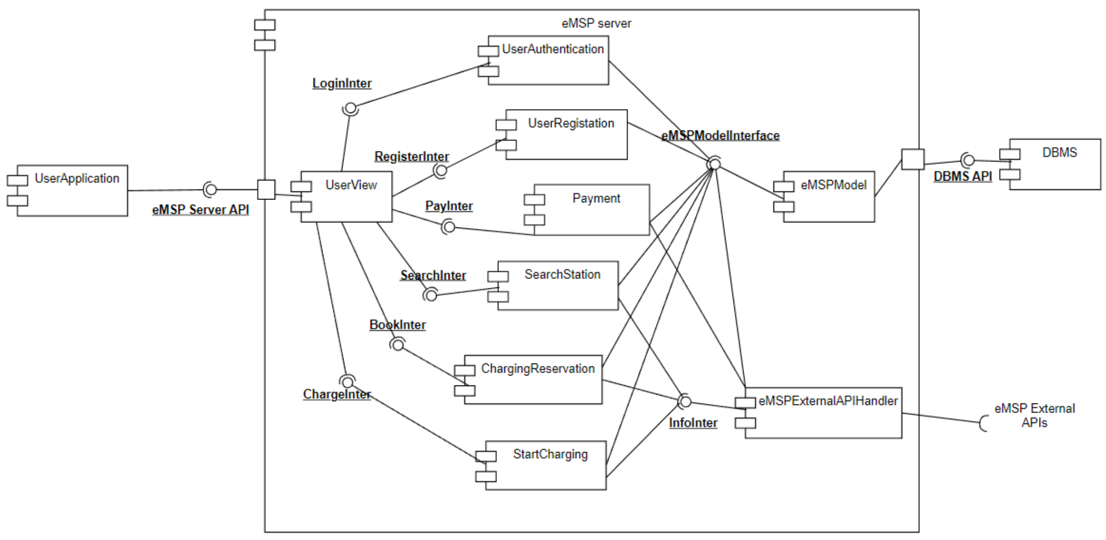


Figure 2: Component diagram of eMSP

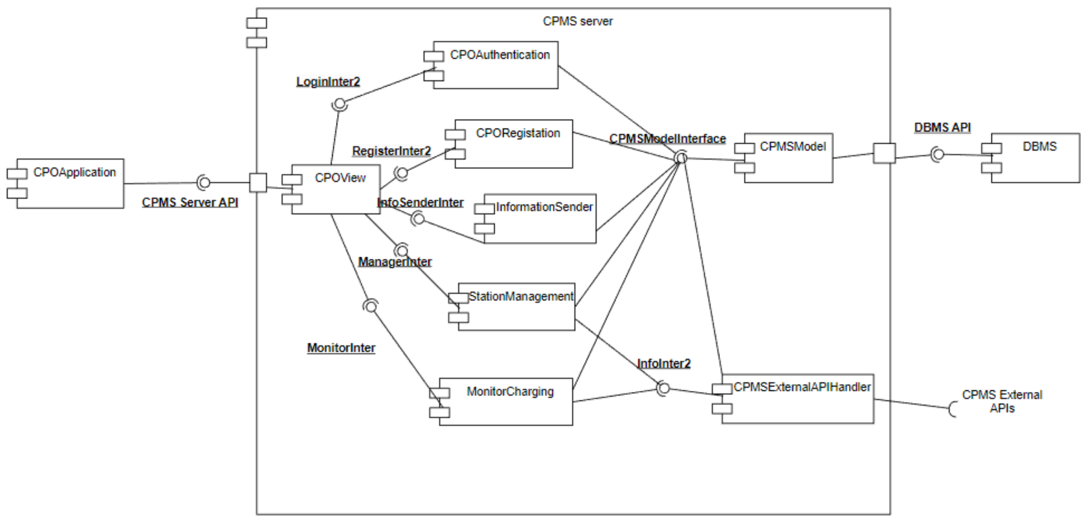


Figure 3: Componenet diagram of CPMS

It should be noted that although the component diagrams of eMSP and of CPMS are presented separately, they have the same DBMS.

In the list below, all components are briefly described:

-User/CPO Application: Client application component, like a web browser on a mobile phone or a PC.

-User/CPO View: User view component, like a web browser shows the main page of E-mall.

-User/CPO Authentication: The component which handles the login of a user/CPO.

-User/CPO Registration: The component which handles the registration of a user/CPO.
-Payment: The component responsible for enabling users to pay the order.
-SearchStation: The component responsible for enabling users to search a charge station.
-ChargingReservation: The component responsible for handling tasks about booking a charge in a specific charging station.
-StartCharging: The component responsible for enabling users to start charging in s charge station.
-InformationSender: The component responsible for enabling CPOs to send information to database, for example, give a special offer to users.
-StationManagement: The component containing the service used to check the status of a charging station (number of charging sockets available, their cost and so on).
-MonitorCharging: The component responsible for enabling CPOs to monitor the charging process in their stations
-eMPS/CPMS model: The component solely responsible for communicating with the data
tier.
- eMPS/CPMS ExternalAPIHandler: This component is responsible for handling communication with external services. In our case, it is used as a data source of the remaining energy of a socket or an electric vehicle battery. And it's also an API enables users to pay the service by other application.
-DBMS: Database Management System

2.3 Deployment view

This section presents a deployment diagram of the system. It presents the required environments and tools to build the system, and the protocol needed for the communication between different parts.

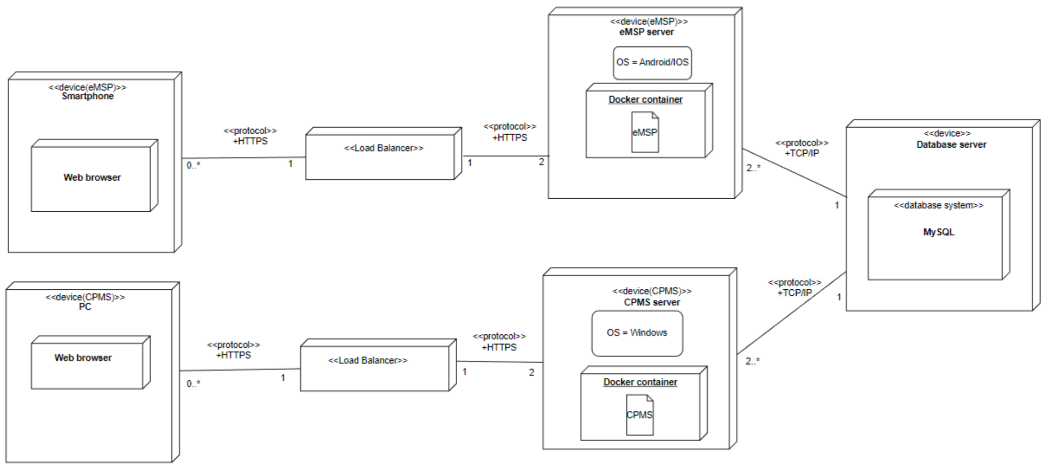


Figure 4: Deployment diagram

• **Smartphone and PC:** These devices should be able to run a web browser with internet access. In general, we think that most of the users/drivers will use a mobile phone and most of the CPOs will use a PC.

• **Load Balancers:** They are used to distribute service requests evenly to the actually executed services, thus ensuring the response speed of the entire system.

• **CPMS/eMSP server:** They focus on the application logic of the system and will also interact with the database server when receiving instructions. More than two servers can increase the stability, the reliability and the carrying capacity of the server in order to cope with the possible increase in the number of users in the future.

• **Database server:** It hosts the database of the two systems and it's a shared database for them.

2.4 Runtime view

In this section, some sequence diagrams will be presented to describe the way components interact to accomplish specific tasks.

For all the sequence diagrams, except for Login, it is assumed that the user or CPO has already logged in. What's more, the interaction between the "UserApplication" and "UserView" is less important than others, sometimes it will be omitted in order to simplify the sequence diagrams.

2.4.1 Sequence diagrams for eMSP

-User registering

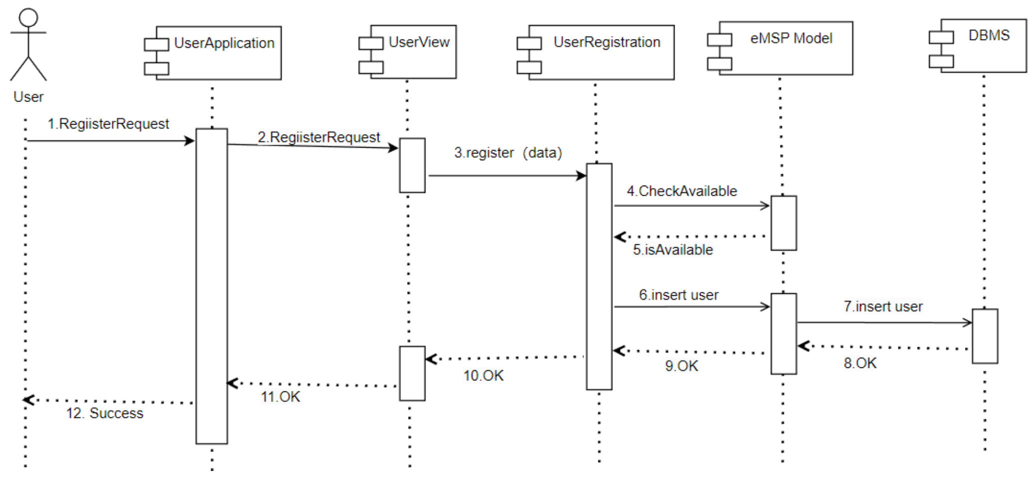


Figure 5: Sequence diagram for user registering

-User login to eMall

Following sequence diagrams will omit "UserView".

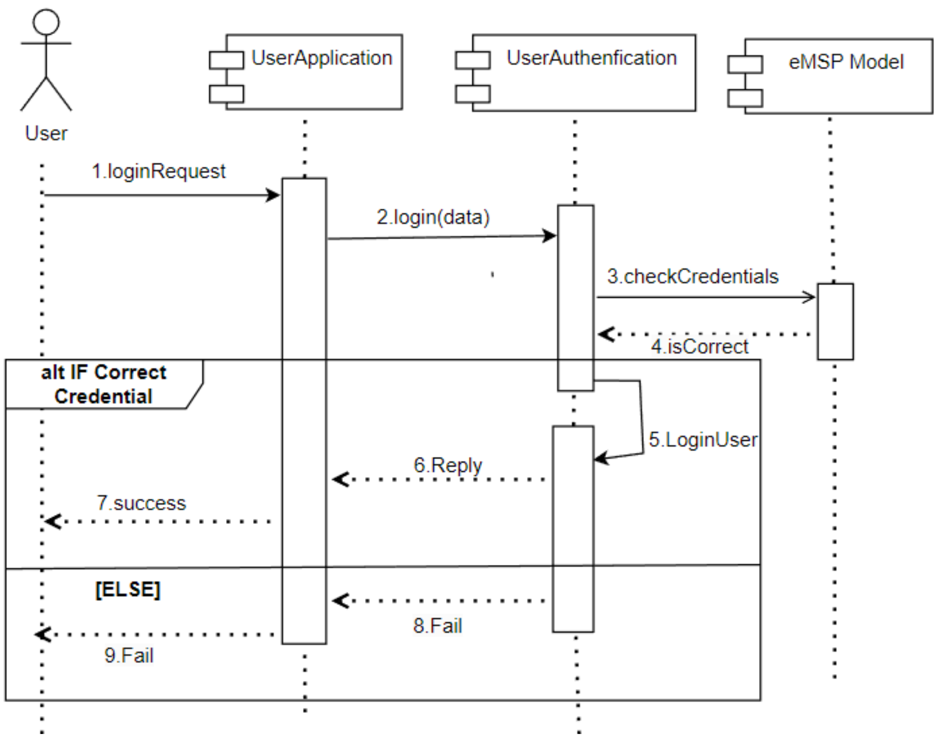


Figure 6: Sequence diagram for login

-User searches a station

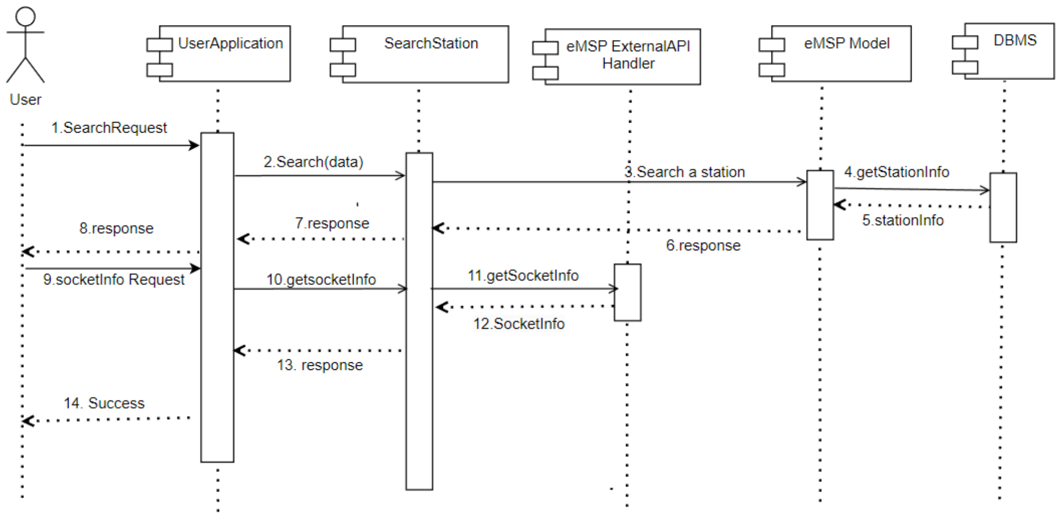


Figure 7: Sequence diagram for searching a station

In this case, the "ExternalAPI Handle" is a data source for more information about the socket, for example, the remaining energy of every socket in a station.

-User books a charge in a station

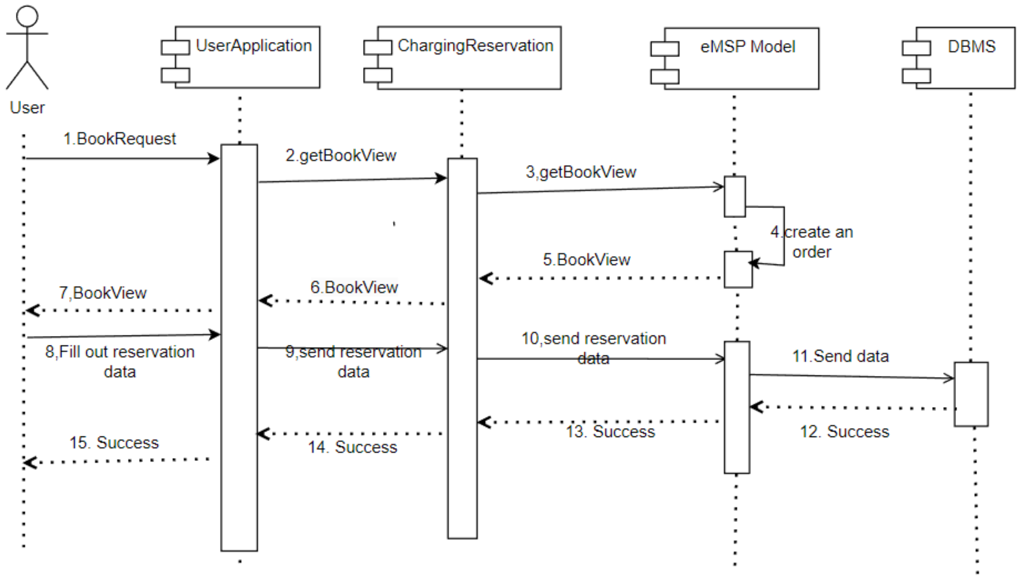


Figure 8: Sequence diagram for booking a charge in a station

In this case, it is assumed that the user has already decided to book a charge in which station and book which socket.

-User charges in a station

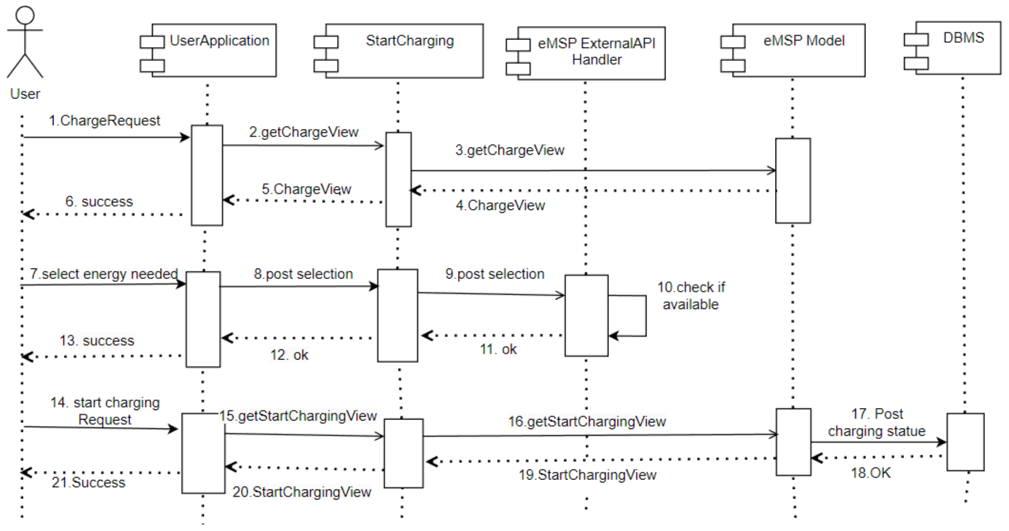


Figure 9: Sequence diagram for charging in a station

In this case, it is assumed that the user has already decided the station and the socket to charge.

-User pays for the service

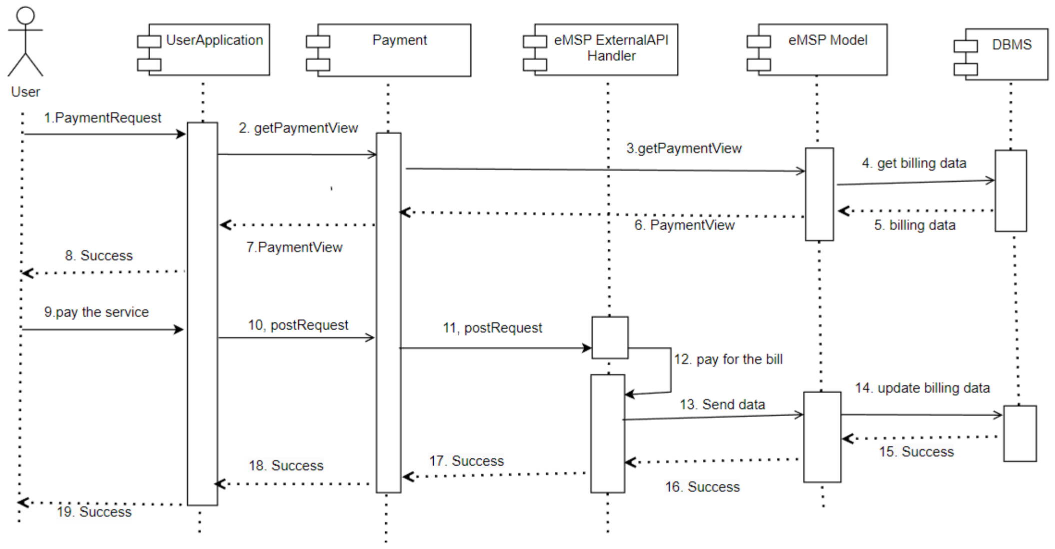


Figure 10: Sequence diagram for paying for a service

2.4.2 Sequence diagrams for CPMS

These similar sequence diagrams, like the CPO login diagram, are omitted.

-CPO checks the status of a charging station

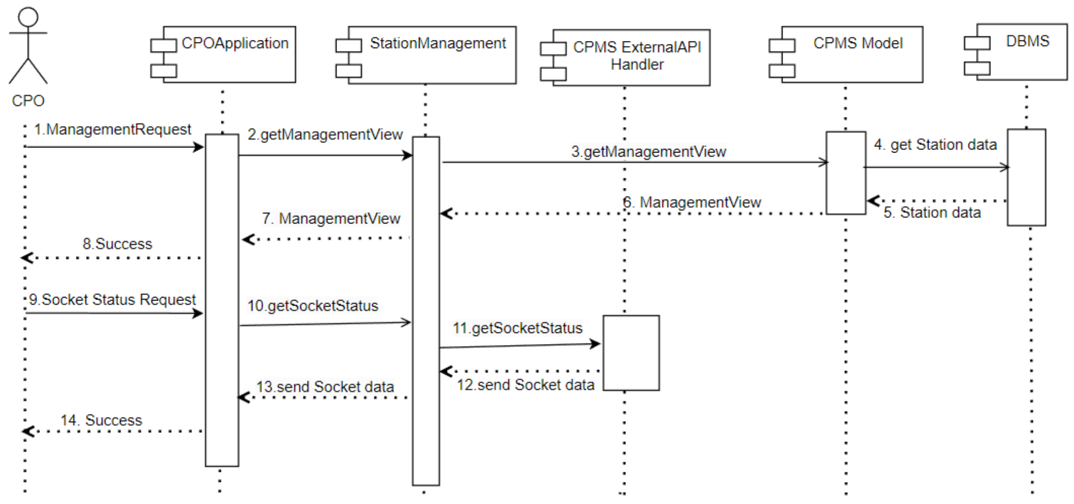


Figure 11: Sequence diagram for checking the status of a station

- CPO monitors a charging process

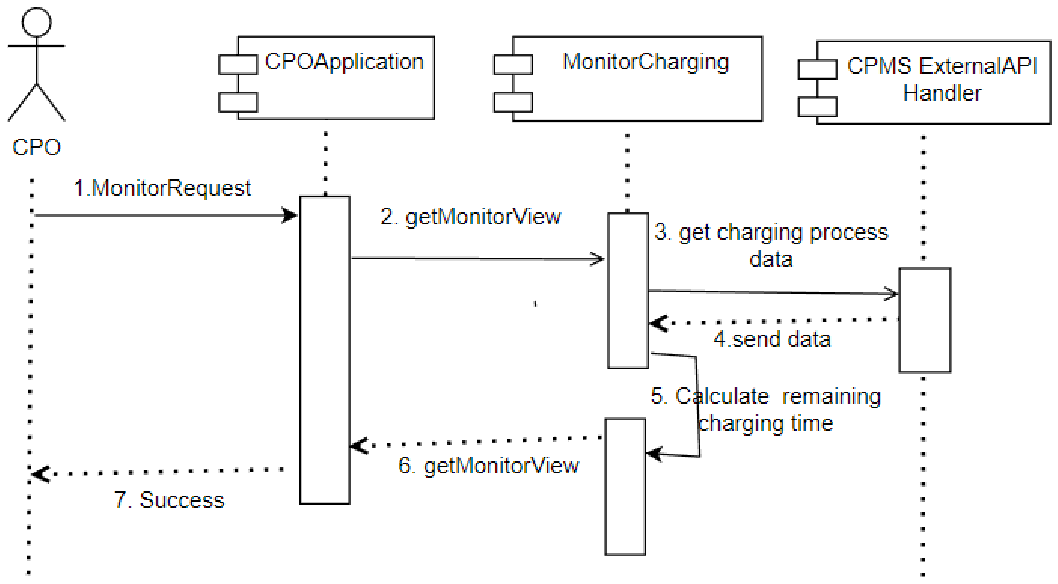


Figure 12: Sequence diagram for monitoring a charging process

2.5 Component interfaces

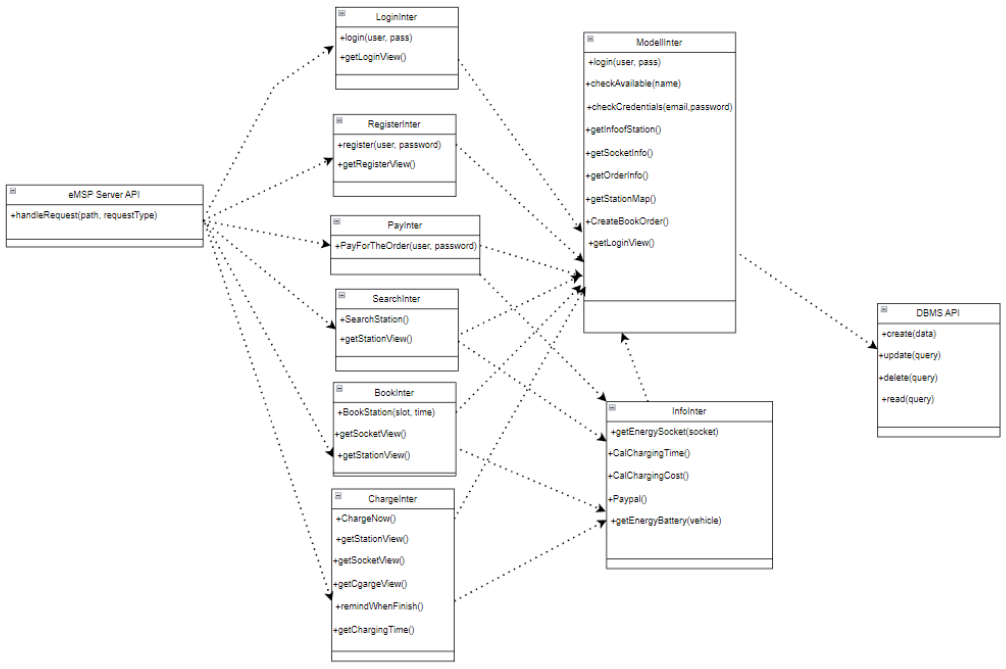


Figure 13: Diagram describing the component interfaces of eMSP

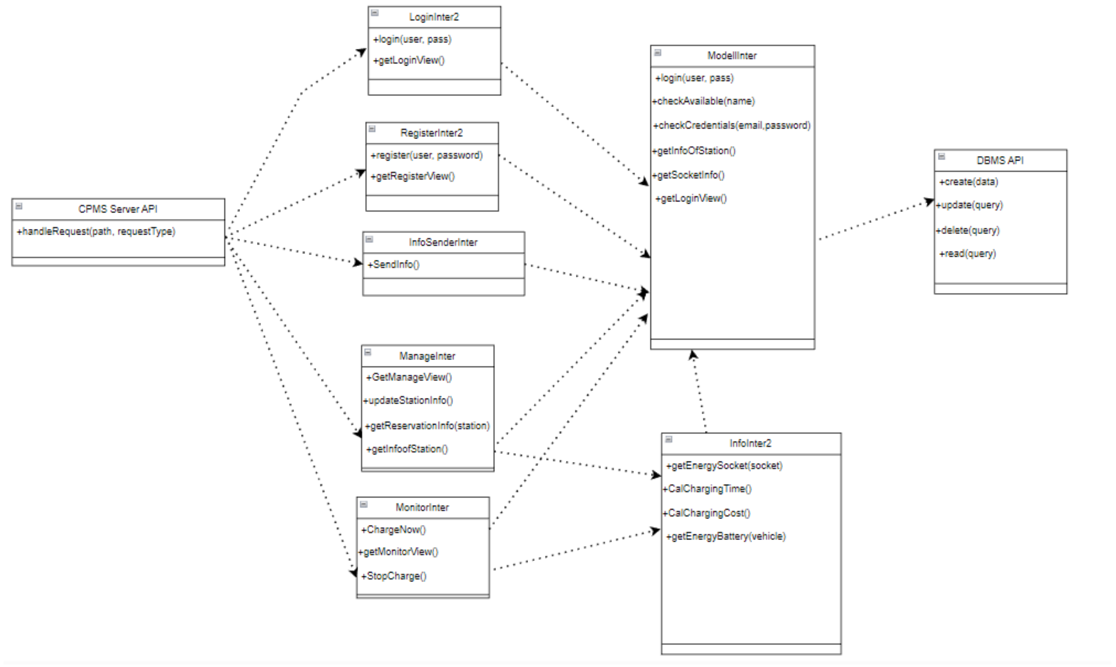


Figure 14: Diagram describing the component interfaces of CPMS

As mentioned above, although the two diagrams are presented separately, they use the same DBMS.

2.6 Selected architectural styles and patterns

• **Three tier architecture**
Both two systems are structured in three tiers, a presentation tier, business logic tier and a (shared) data tier. As previously mentioned, The reason why choosing three tier architecture is to separate the business logic of the system from the data. This has many advantages. First of all, each layer can be independently developed. Secondly, compared to a traditional client-service architecture, it makes sense to decouple the data and business tier. That means the data can be used for other applications if needed and subsequent updates will mainly focus on the business logic layer. Therefore, subsequent system maintenance and updates will be less complicated. Finally, three-tier architecture helps strengthen system security.

• **Model View Controller (MVC)**
We choose MVC as the design pattern for the E-mall system. The MVC pattern separates the components into Model, View and Controller:

Model: Realize business logic processing and database access, etc. (In our case, the DBMS and the Model).

View: Realize the interface for interacting with users and realize the function of input and output of data. (In our case, the User/CPO Application)

Controller: Contact and control the model layer and view layer to complete the user's needs. (In our case, most of the controllers, like “SearchStation”, “ChargeReservation”, etc)

Same reason as choosing a three-tier architecture, MVC will help the system have the characteristics of "high cohesion, low coupling".

2.7 Other design decisions

Firstly, it is worth mentioning that both systems use the same data layer. This is because that the databases required by eMPS and CPMS have a certain overlap. For example, they both need the information of the remaining energy of every socket in a charging station. What's more, a shared database can enable a CPO to give more information about his station, like a special offer, to users. He can upload information to the shared database and users can directly access information.

Additionally, it will be necessary for E-mall that some data will be acquired from external APIs, for example, the remaining energy data of the socket and the vehicle battery. Although it will bring some risks to the security of the system, it contributes to system flexibility and low coupling. Just updating the business logic layer can bring more data sources to the system. At the same time, The independence between each task is also conducive to the maintenance and development of the system.

3 User interface design

Taking into account the user's usage habits, the user interfaces for eMSP will be smartphone interfaces, and these for CPMS will be PC interfaces.

3.1 Users

1. Login

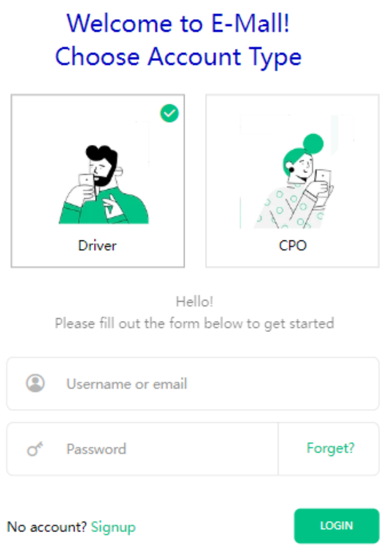
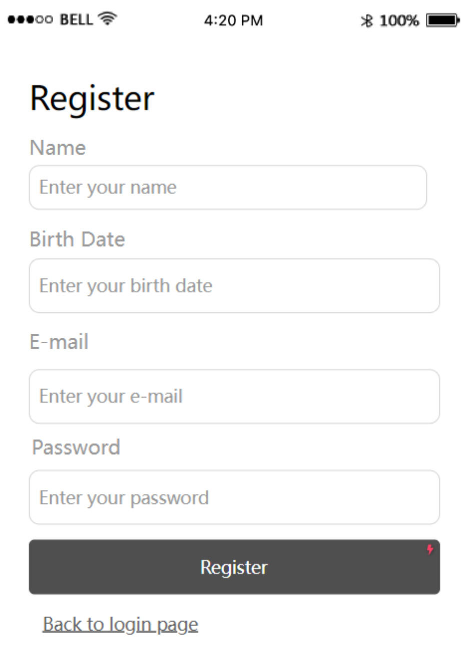


Figure 15: UI for signing in

2. Registration



●●●○○ BELL 4:20 PM 100%

Register

Name

Enter your name

Birth Date

Enter your birth date

E-mail

Enter your e-mail

Password

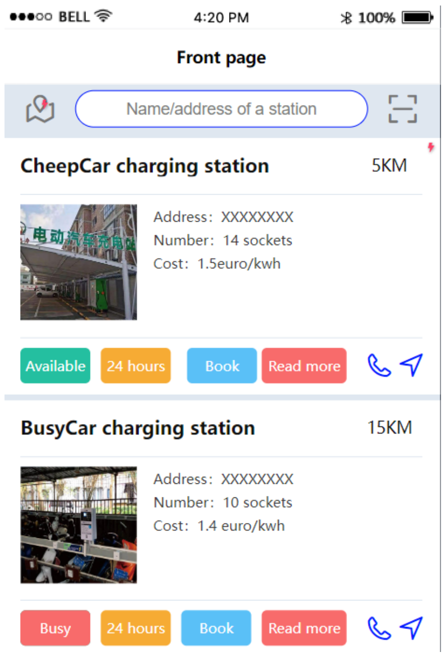
Enter your password

Register

Back to login page

Figure 16: UI for registration

3. Front page: Search a station



●●●○○ BELL 4:20 PM 100%

Front page

Name/address of a station

CheepCar charging station 5KM

Address: XXXXXXXX
Number: 14 sockets
Cost: 1.5euro/kwh

Available 24 hours Book Read more

BusyCar charging station 15KM

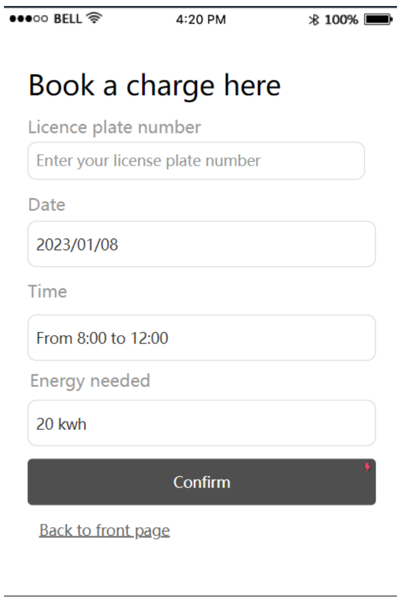
Address: XXXXXXXX
Number: 10 sockets
Cost: 1.4 euro/kwh

Busy 24 hours Book Read more

Figure 17: UI for searching a station

In the front page, user can search a charging station via the search box or just choose a station nearby. He can also click “book” to book a charge or click “read more” to get more information and choose a socket to start charging now. There is a function of scanning the QR code in the upper right corner to start charging more conveniently.

4. Book a charge


●●●○○ BELL 4:20 PM 100%

Book a charge here

Licence plate number

Enter your license plate number

Date

2023/01/08

Time

From 8:00 to 12:00

Energy needed

20 kwh

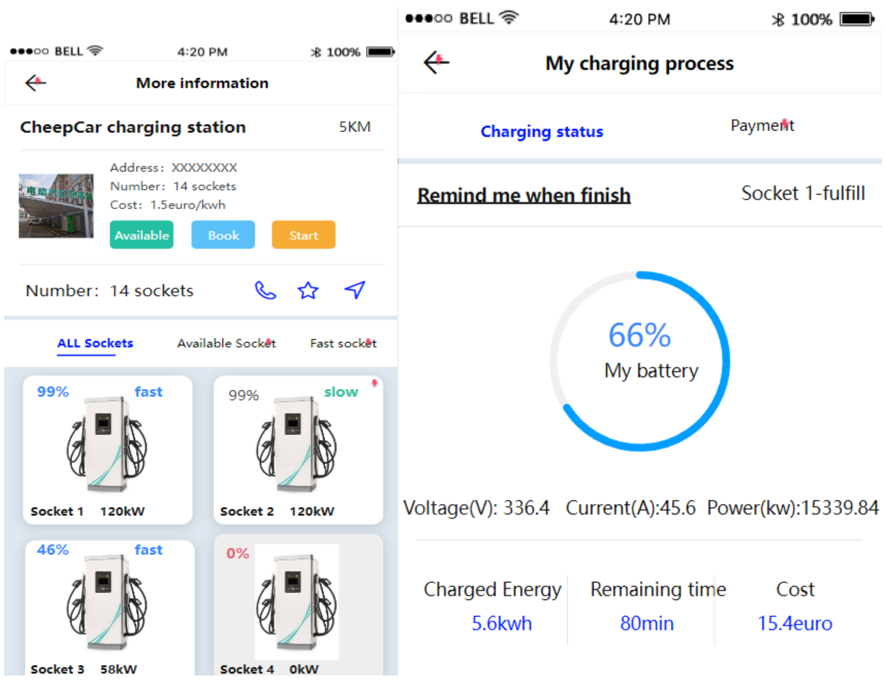
Confirm

Back to front page

Figure 18: UI for booking a charge

If a user has chosen a station and clicks "book", he needs to fill out a form to book a charge.

5. Start a charging process


●●●○○ BELL 4:20 PM 100%

More information

CheepCar charging station 5KM

Address: XXXXXXXX
Number: 14 sockets
Cost: 1.5euro/kwh

Available Book Start

Number: 14 sockets

ALL Sockets | Available Socket | Fast socket

99% fast Socket 1 120kW

99% slow Socket 2 120kW

46% fast Socket 3 58kW

0% Socket 4 0kW

●●●○○ BELL 4:20 PM 100%

My charging process

Charging status | Payment

Remind me when finish | Socket 1-fulfill

66% My battery

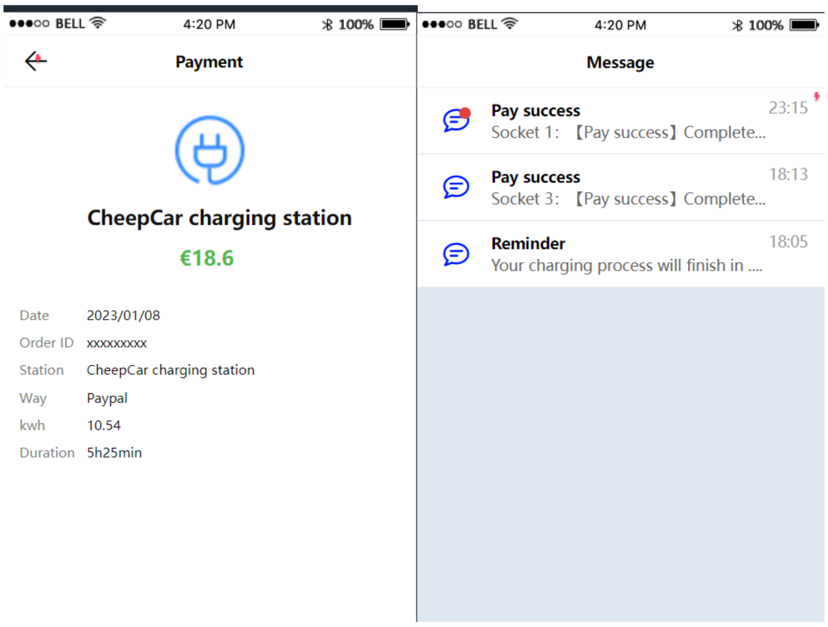
Voltage(V): 336.4 Current(A):45.6 Power(kw):15339.84

Charged Energy	Remaining time	Cost
5.6kwh	80min	15.4euro

Figure 19: UI for charging a station

When a user starts a charging process, the systems will show the charging status and the remaining time for charging. There is also a function which can remind the user when the charging is finished.

6. Pay for the service


●●●○○ BELL 4:20 PM 100%

Payment

CheepCar charging station

€18.6

Date	2023/01/08
Order ID	xxxxxxxxx
Station	CheepCar charging station
Way	Paypal
kwh	10.54
Duration	5h25min

●●●○○ BELL 4:20 PM 100%

Message

Pay success 23:15
Socket 1: 【Pay success】Complete...

Pay success 18:13
Socket 3: 【Pay success】Complete...

Reminder 18:05
Your charging process will finish in ....

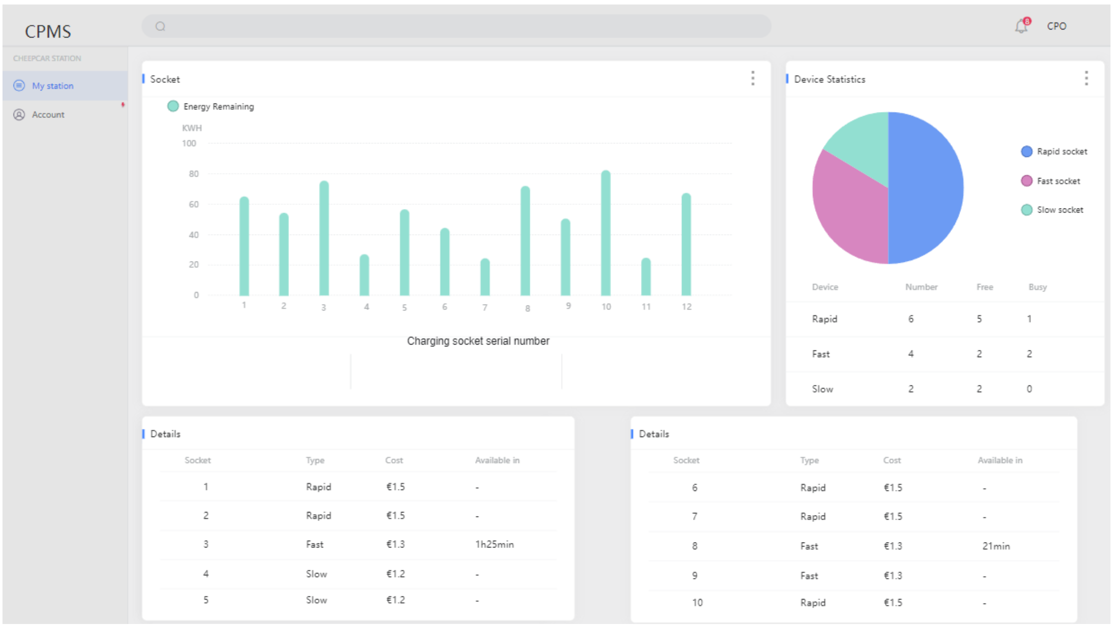
Figure 20: UI for payment

A user can pay for the service when the charging process is finished and he will receive a message after payment.

3.2 CPO

The similar login interface and registration interface are omitted here

-CPO check the status of a charging station and the charging process of sockets



Device	Number	Free	Busy
Rapid	6	5	1
Fast	4	2	2
Slow	2	2	0

Details

Socket	Type	Cost	Available in
1	Rapid	€1.5	-
2	Rapid	€1.5	-
3	Fast	€1.3	1h25min
4	Slow	€1.2	-
5	Slow	€1.2	-

Details

Socket	Type	Cost	Available in
6	Rapid	€1.5	-
7	Rapid	€1.5	-
8	Fast	€1.3	21min
9	Fast	€1.3	-
10	Rapid	€1.5	-

Figure 21: UI for checking the status of a station and monitoring the charging process

This management page enables CPO to know the "external" status of a charging station, like the number of charging sockets, their type, they are available or not, etc. What 's more, CPO can also know about the charging process of every socket.

4 Requirements traceability

This section will talk about the requirements traceability by a table which shows the components required for fulfilling each requirement.

The components have been given abbreviations, see list below:

Abbreviations	**Components**
UA	User/CPO Application
UV	User/CPO View
UAT	User/CPO Authentication
UR	User/CPO Registration
PM	Payment
SS	SearchStation
CR	ChargingReservation
SC	StartCharging
IS	InformationSender
SM	StationManagement
MC	MonitorCharging
Model	eMPS/CPMS model
EAH	eMPS/CPMS ExternalAPIHandler
DMBS	DMBS

Following is a table presents the requirements:

Requirement	**Description**
R1	The system shall allow an unregistered user to register an account
R2	The system shall allow a registered user to know about the information of the charging stations.
R3	The system shall allow a registered user to search the charging station nearby.
R4	The system shall allow a registered user to book a charge in a specific charging process at a certain station.
R5	The system shall allow a registered user modify or cancel the reservation
R6	The system shall allow a registered user to start the charging process at a certain station
R7	The system shall be able to estimate time required for charging
R8	The system shall notify a registered user when the charging process is finished.
R9	The system shall allow a registered user to pay the obtained service
R10	The system shall allow a registered CPO to know the location and status of a charging station.
R11	The system shall detect the remaining power of every socket.
R12	The system shall detect the remaining power of user’s battery.
R13	The system shall allow a registered CPO to monitor the charging process

R14	The system shall allow a registered CPO to update charging station information and send it to eMSPs.
R15	The system shall allow a user/CPO to log in.
R16	The system shall protect user data and privacy

Following is a table presents the requirements traceability:

R	UA	UV	UAT	UR	PM	SS	CR	SC	IS	SM	MC	Model	EAH	DMBS
R1	✔			✔								✔		✔
R2	✔	✔	✔	✔		✔						✔		✔
R3	✔	✔	✔	✔		✔						✔		✔
R4	✔	✔	✔	✔			✔					✔		✔
R5	✔	✔	✔	✔			✔					✔		✔
R6	✔	✔	✔	✔				✔				✔		✔
R7	✔	✔	✔	✔								✔	✔	✔
R8	✔	✔	✔	✔								✔	✔	✔
R9	✔	✔	✔	✔	✔							✔	✔	✔
R10	✔	✔	✔	✔						✔		✔	✔	✔
R11	✔	✔	✔	✔								✔	✔	✔
R12	✔	✔	✔	✔								✔	✔	✔
R13	✔	✔	✔	✔						✔	✔	✔	✔	✔
R14	✔	✔	✔	✔					✔	✔		✔		✔
R15	✔	✔	✔	✔								✔		✔
R16	✔	✔	✔	✔	✔							✔		✔

In brief, every controller has its independent function contribute to at least one requirement. The model, DBMS, and the components responsible for registration contribute to most of the requirements.

5 Implementation, integration and test plan

In this section the plans for the implementation and testing of the system will be described.

5.1 Implementation

First of all, in order to have a clear development path, all the components will be divided into serval groups according to the component diagrams. Secondly, the main strategy of the implementation is that the system should be developed based on the database. Although the three-tier architecture allows each layer of the system to be developed independently, the shared data tier obviously is the most important part of our system. After the development of the data layer is completed, both systems can also be developed separately based on the data layer. At the same time, this strategy can be helpful to simplify the test plan.
In details, the system will be developed in the following sequence of four subparts. Both eMSP and

CPMS will use the same implementation strategy.

1. **DBMS & Model**

This part contains the main database of the system and the most basic component which is responsible for communicating with the data tier.

2. **ExternalAPIHandler**

This past contains the components which supply the controllers with the rest of the data they need to function correctly.

3. **Controllers**

This part contains the controllers like “SearchStation”, “ChargeReservation”, etc. All of them have their own independent function so that they can be developed in parallel.

4. **User/CPO Application**

This part contains the components which are responsible for the interaction with users/CPOs.

5.2 Integration & Test plan

The order in which we integrate components will be consistent with the system development order mentioned above. With the help of the simple system hierarchy, each integration tested component can be implemented by a driver to test.

In order to present more clearly, some explanatory graphics will be presented below.

1. **Integration test of DBMS & Model**

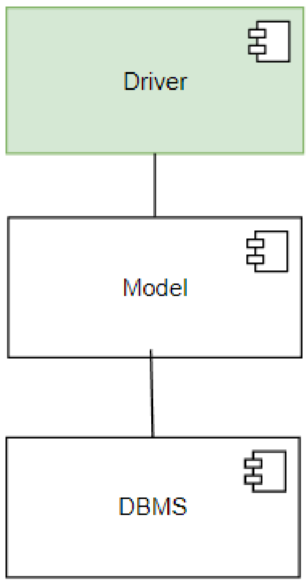


Figure 22: Integration test of Model and DBMS

2. **The integration test of the external API handler**

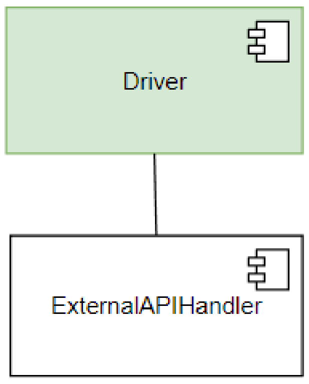


Figure 23: Integration test of the external API component

3, The integration test of the controllers

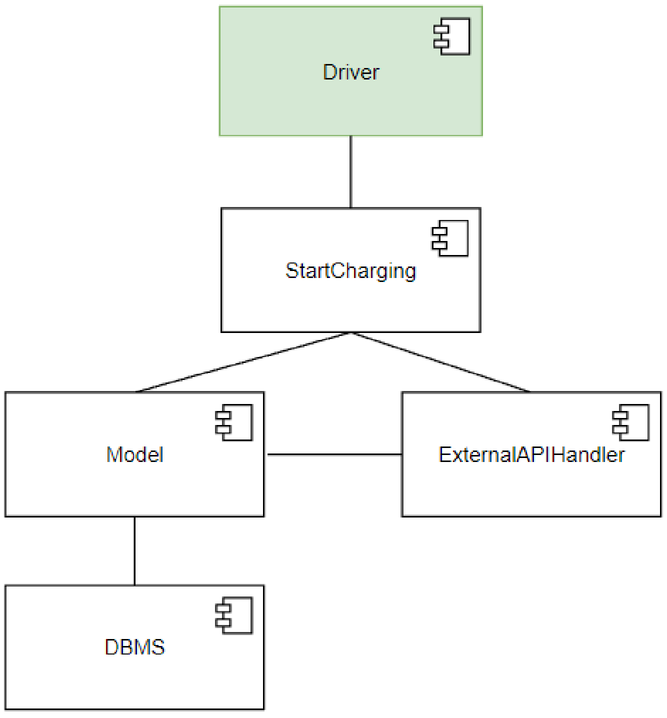


Figure 24: Integration test of controllers

In this case, we choose "StartCharging" as an example, the integration test of other controllers are similar or less complicated (for those controllers don't need ExternalAPIHandler).

4, The integration test of the User/CPO application

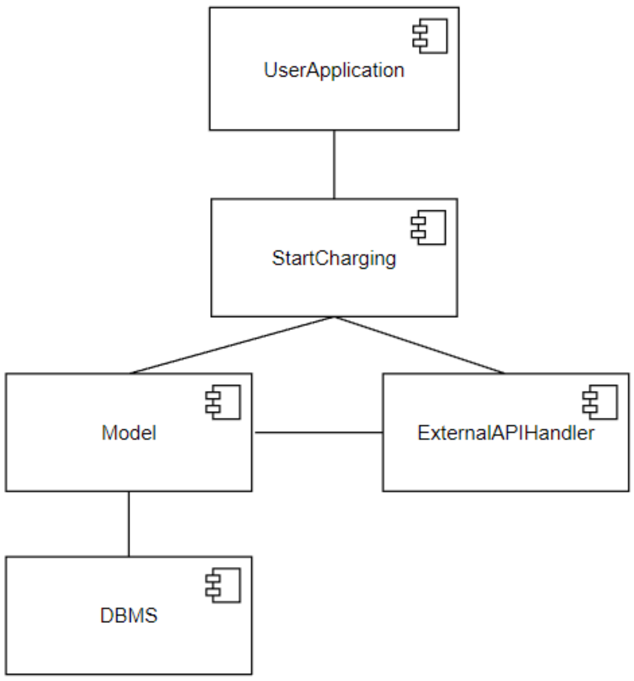


Figure 25: The final integration to complete the system, UserApplication

As soon as the whole system has been integrated, system testing can begin in order to assess the fulfilment of functional and non-functional requirements. At the same time, try to be more user-friendly during system testing.

6 Effort spent

Task	**Time spent**
Introduction	7h
Architectural design	35h
User Interface Design	15h
Requirements Traceability	7h
Implementation, Integration and Test Plan	10h
Total	74h

7 References

- Serverless Multi-tier Architecture on AWS: https://cloudtweaks.com/2019/08/serverless-multi-tier-architecture-aws/
- MVC Frame work:

https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

- Pattern: Shared database: https://microservices.io/patterns/data/shared-database.html