E-MALL

e-mobility for all

Design Document

Haotian Zhang

Jiaheng Xiong

Chenyu Zhao

Supervisor: Prof. Matteo Camilli

07.01.2023

Contents

1. Introduction

1.1 Purpose

The continued increase in carbon emissions will cause global warming and sea level rise, producing climate anomalies, increased desertification area, and increased pests and diseases. Therefore, the reduction of room temperature gas emissions and the reduction of "carbon footprint" should be the consensus of human beings to achieve sustainable development. In 2022, the United Nations Conference on Environment and Sustainable Development will be held in Stockholm with the theme of a healthy planet for the prosperity of all --our responsibility, our opportunity. However, compared with fuel vehicles, electric vehicles still require longer charging time, while the construction of infrastructure such as charging piles is still imperfect. Therefore, reasonable distribution of charging pile resources and intelligent aggregation of charging pile information can greatly facilitate electric vehicle travel and contribute to the popularity of electric vehicles, thus achieving the goal of reducing carbon footprint.

Our purpose is to develop a new system eMall -- e-Mobility for All that (i) provides endusers with charging station aggregation query services, preferential information, and charging payment and management functions, as well as intelligent charging planning functions, and (ii) provides Charging Point Operator (CPO) with management and interaction with Distribution System Operator (DSO). The service can optimize the operation process of charging service providers and improve the charging experience of end users.

1.2 Scope

The scope of this design document is to outline the overall design and architecture of the system, including the key components that make up the system, the user interface design for interacting with the system and a plan for the implementation, testing and integration of the system. The document will also describe the key challenges and constraints that the eMall must address and will outline the design principles and technical standards that will guide the development of the system.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Definition	Description
End User ID	This is a unique ID generated by eMSPs for the end user.
CPO ID	This is a unique ID generated by CPMS for the CPO.
Charge Station ID	This is a unique ID generated by CPMS for the charge station.

1.3.2 Abbreviations

Abbreviation	Description

RASD	Requirements Analysis and Specification Document
API	Application Programming Interface
RX	Requirement number X
eMall	e-Mobility for All eMSPs
eMSPs	e-Mobility Service Providers
CPMS	Charge Point Management System
CPO	Charging Point Operators
DSOs	Distribution System Operators
DBMS	Database Management System
DB	Database

1.4 Revision history

1.5 Reference Documents

• The specification document "Assignment RDD AY 2022-2023_v3.pdf"

• The RASD document

1.6 Document Structure

This document contains seven sections, detailed below.

- The first section provides an overview of the eMall, including its purpose, scope, and target audience.
- The second section focuses on describing and motivating the architectural design of the system to be. 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
- The third section presents design mockups of the user interface.
- The fourth section contain 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.
- The fifth section describes the suggested implementation and test plan for the system to be.
- The sixth section contains the effort spent on this report by the authors.
- The seventh section contains the references used.

2. Architecture Design

2.1 Overview: High-level components and their interaction

The project chooses the **Microservice Architecture**, which is a cloud-native solution and involves breaking down a larger system into smaller, independent components that can be developed, deployed, and scaled individually.

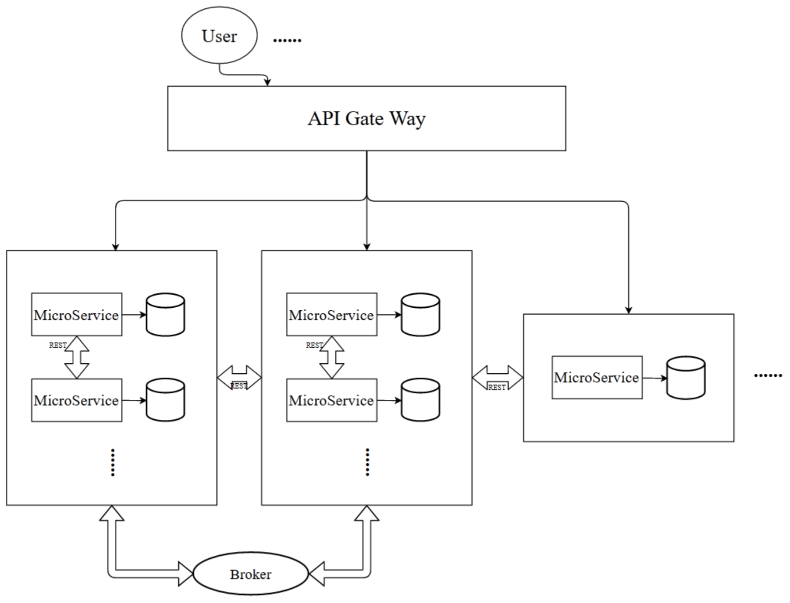


Figure 2-1-1 Microservice architecture diagram

In the context of eMSP and CPMS, the microservices pattern could be used to create individual microservices for tasks. Each of these microservices could be developed and maintained independently.

The users access these microservices through the **API Gate Way,** its role includes:

- Provide a unified service portal to make microservices transparent to the frontend
- Aggregate the services in the background to save traffic and improve performance
- Provide security, filtering, flow control and other API management functions

All microservices are independent Java processes running on separate virtual machines, and the microservices communicate with each other through **IPC (inter process communication).** We will use REST for synchronous communication and introduce Brokers for asynchronous communication.

2.2 Component view

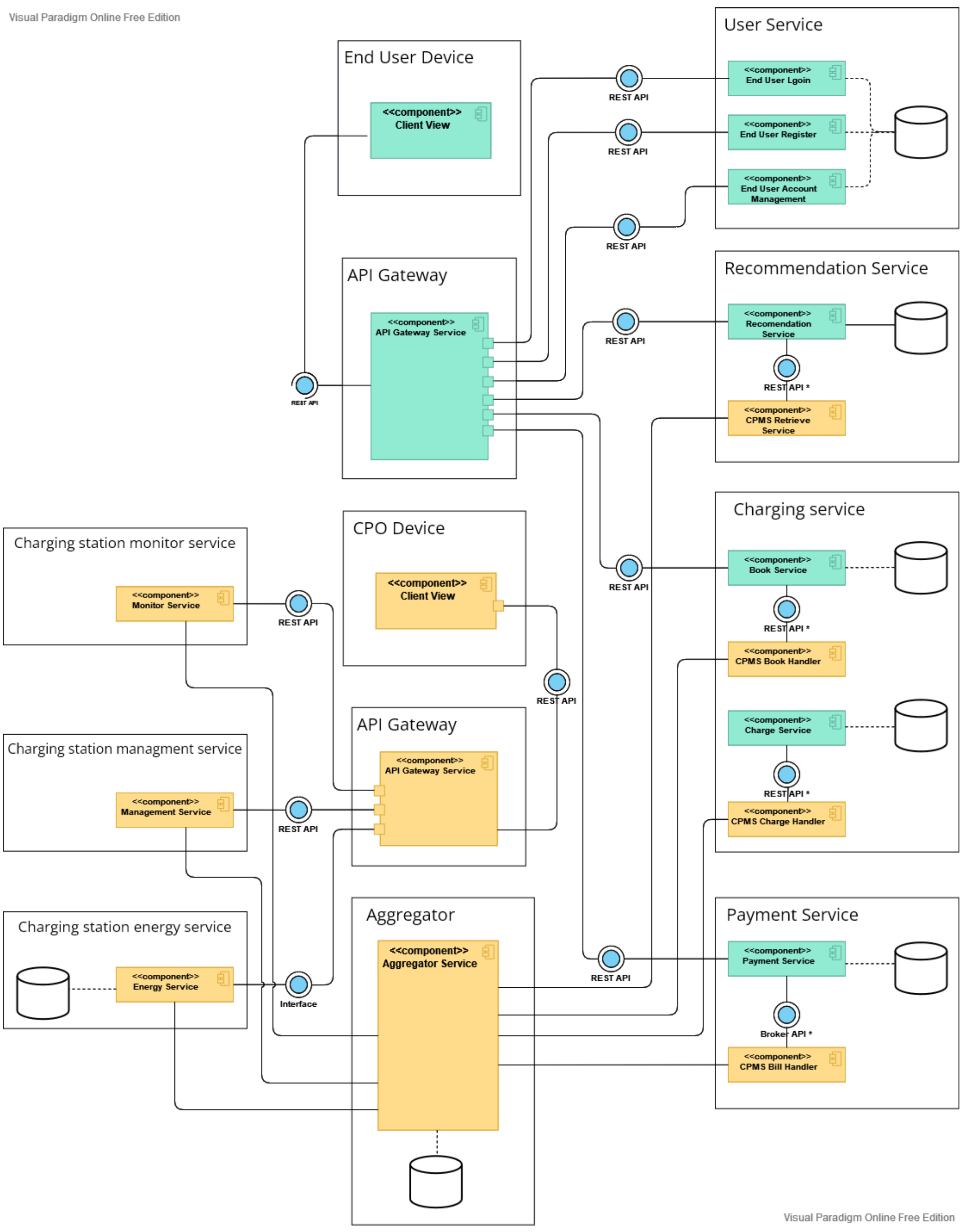

Visual Paradigm Online Free Edition
End User Device
<<component>> Client View
User Service
<<component>> End User Lgoin
<<component>> End User Register
<<component>> End User Account Management
REST API
REST API
REST API
API Gateway
<<component>> API Gateway Service
REST API
Recommendation Service
<<component>> Recomendation Service
REST API
REST API *
<<component>> CPMS Retrieve Service
Charging station monitor service
<<component>> Monitor Service
REST API
CPO Device
<<component>> Client View
Charging service
<<component>> Book Service
REST API
REST API *
<<component>> CPMS Book Handler
<<component>> Charge Service
REST API *
<<component>> CPMS Charge Handler
REST API
Charging station managment service
<<component>> Management Service
REST API
API Gateway
<<component>> API Gateway Service
Charging station energy service
<<component>> Energy Service
Interface
Aggregator
<<component>> Aggregator Service
REST API
Payment Service
<<component>> Payment Service
Broker API *
<<component>> CPMS Bill Handler
Visual Paradigm Online Free Edition

Figure 2-2-1 Component view

In this diagram, eMSP and CPMS are referred to as a combination of a series of Microservices, and there is no clear boundary between the Microservices of the two systems. However, it is worth noting that eMSP can communicate with a third-party CPMS through APIs, and this type of interface will be marked with an * symbol in the diagram.

eMSP subsystem in a microservices architecture:

- **User service:** This service will handle user authentication and authorization, as well as manage user profiles and preferences.
- **Charging service:** This service will also handle the booking and charging of charging slots. This service will handle the sending of notifications to users, such as alerts when charging is complete.
- **Payment service:** This service will handle payment processing for the charging service.
- **Recommendation service:** This service will proactively suggest to users that they go and charge their vehicle based on various factors, such as the status of the battery, the user's schedule, and special offers from CPOs.

CPMS subsystem in a microservices architecture:

- **Charging Station Monitor service:** This service will monitor the charging process at the charging stations, including starting and stopping charging, monitoring the charging process, and inferring when the battery is full.
- **Charging Station Energy service:** This service will handle the acquisition of energy from DSOs and will be responsible for making decisions about which DSO to acquire energy from and how much energy to acquire. And the service will also manage the energy plan of the charging stations.
- **Charging Station Management service:** This service will manage the charging station, such as adjusting the cost of the charging sockets, turning off certain sockets, or provides special offers.
- **Aggregator:** The aggregator is responsible for handling database query and modification requests from each microservice via the REST API.

2.3 Deployment view

We use Azure Kubernetes Service to architect the microservice system.

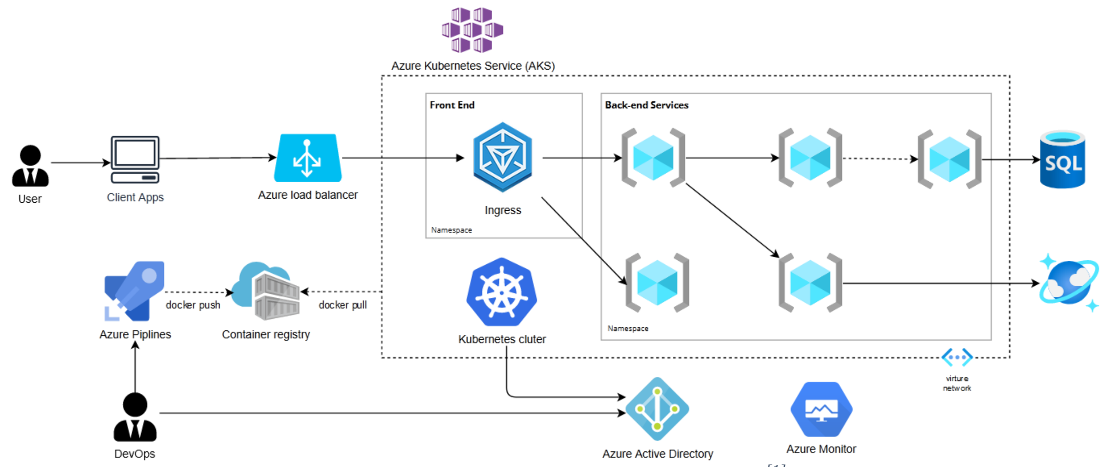


Figure 2-3-1 Microservice on Azure Kubernetes Service [1]

The architecture consists of the following components [1]:

Azure Kubernetes Service (AKS). AKS is a managed Kubernetes cluster hosted in the Azure cloud. Azure manages the Kubernetes API service, and you only need to manage the agent nodes.

Virtual network. By default, AKS creates a virtual network into which agent nodes are connected. You can create the virtual network first for more advanced scenarios, which lets you control things like subnet configuration, on-premises connectivity, and IP addressing.

Ingress. An ingress server exposes HTTP(S) routes to services inside the cluster, which works like an API gateway.

Azure Load Balancer. After creating an AKS cluster, the cluster is ready to use the load balancer. Then, once the NGINX service is deployed, the load balancer will be configured with a new public IP that will front your ingress controller. This way, the load balancer routes internet traffic to the ingress.

Azure Active Directory. AKS uses an Azure Active Directory (Azure AD) identity to create and manage other Azure resources such as Azure load balancers. Azure AD is also recommended for user authentication in client applications.

Azure Container Registry. Use Container Registry to store private Docker images, which are deployed to the cluster. AKS can authenticate with Container Registry using its Azure AD identity. AKS does not require Azure Container Registry. You can use other container registries, such as Docker Hub. Just ensure your container registry matches or exceeds the service level agreement (SLA) for your workload.

Azure Pipelines. Azure Pipelines are part of the Azure DevOps Services and run automated builds, tests, and deployments. You can also use third-party CI/CD solutions such as Jenkins.

Helm. Helm is a package manager for Kubernetes, a way to bundle and generalize Kubernetes objects into a single unit that can be published, deployed, versioned, and updated.

Azure Monitor. Azure Monitor collects and stores metrics and logs, application telemetry, and platform metrics for the Azure services. Use this data to monitor the application, set up alerts, dashboards, and perform root cause analysis of failures. Azure Monitor integrates with AKS to collect metrics from controllers, nodes, and containers.

2.4 Runtime view

2.4.1 User service

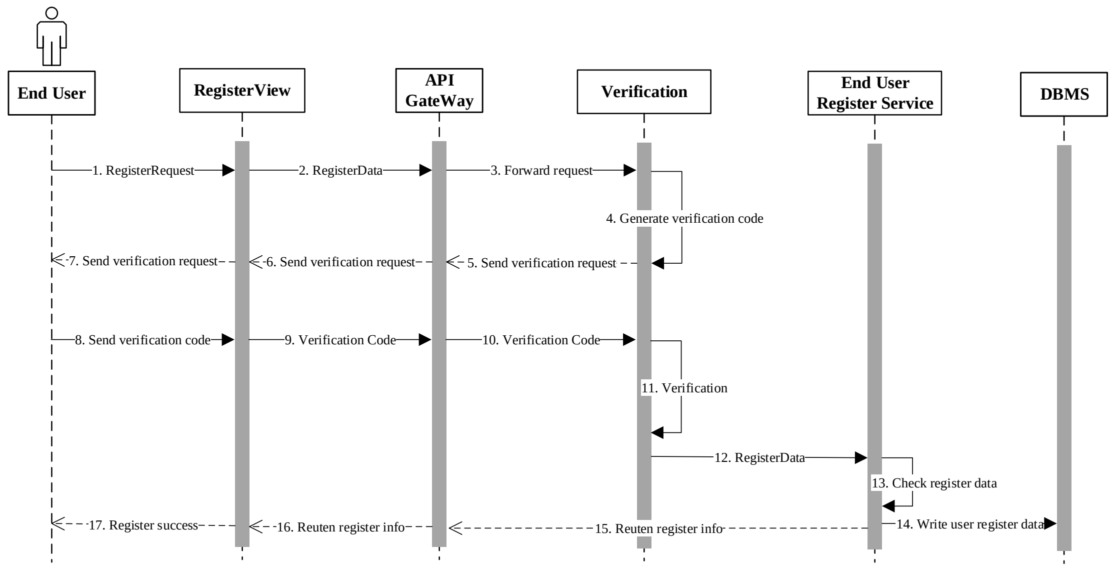


Figure 2-4-1 Sequence diagrams of "Register"

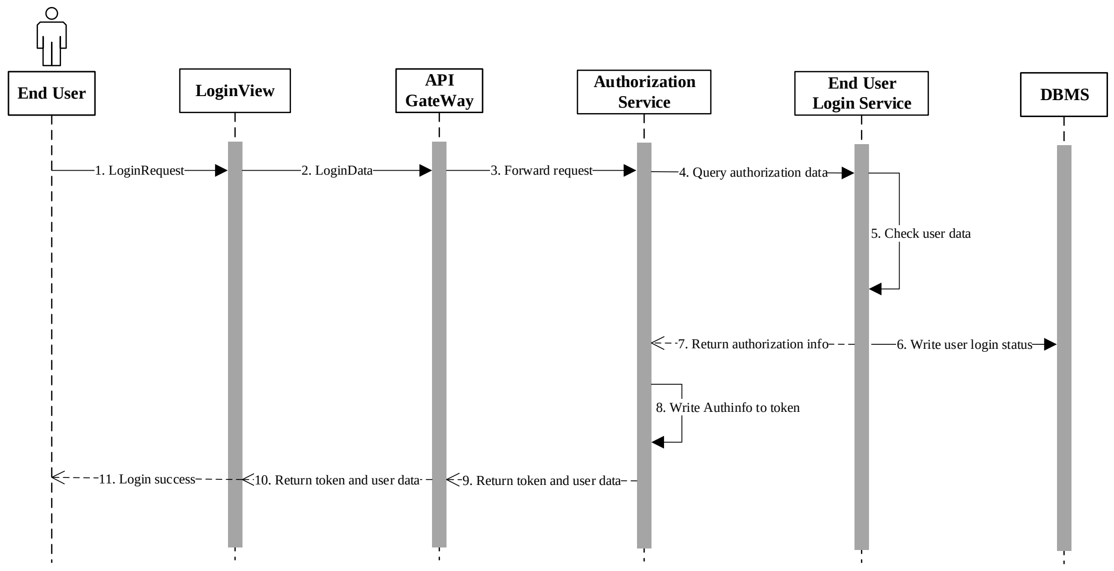

End User
LoginView
API GateWay
Authorization Service
End User Login Service
DBMS
1. LoginRequest
2. LoginData
3. Forward request
4. Query authorization data
5. Check user data
6. Write user login status
7. Return authorization info
8. Write Authinfo to token
9. Return token and user data
10. Return token and user data
11. Login success

Figure 2-4-2 Sequence diagrams of "Login"

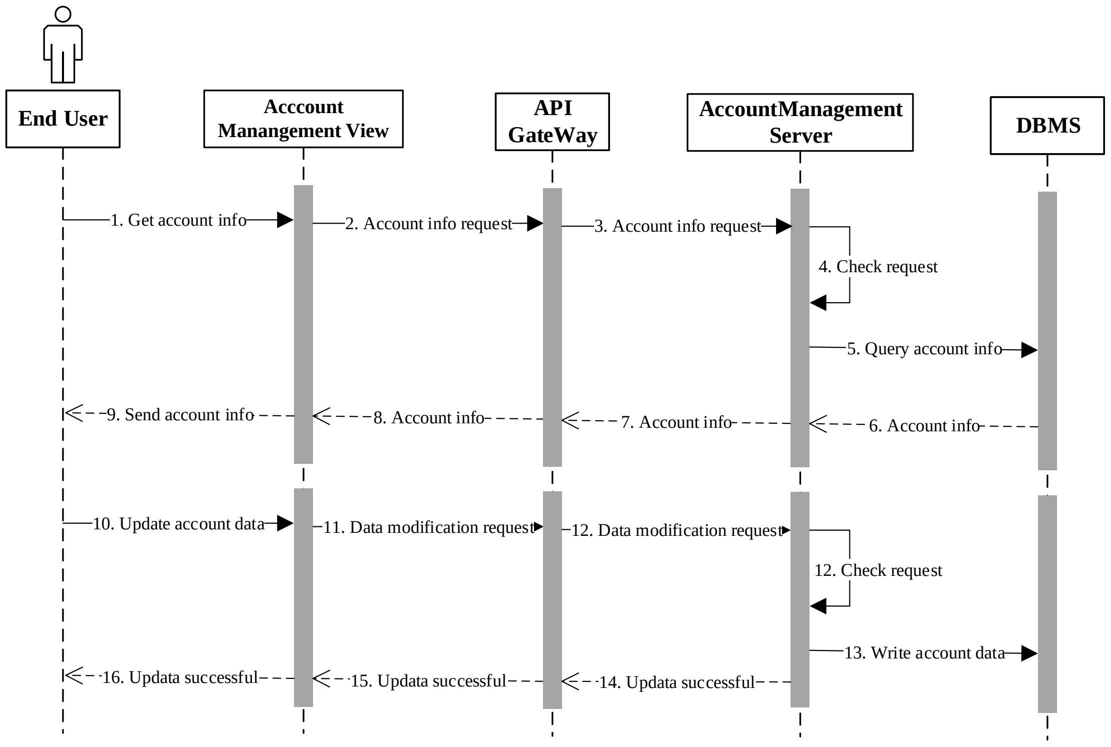


Figure 2-4-3 Sequence diagrams of "Account information management"

2.4.2 Recommendation service

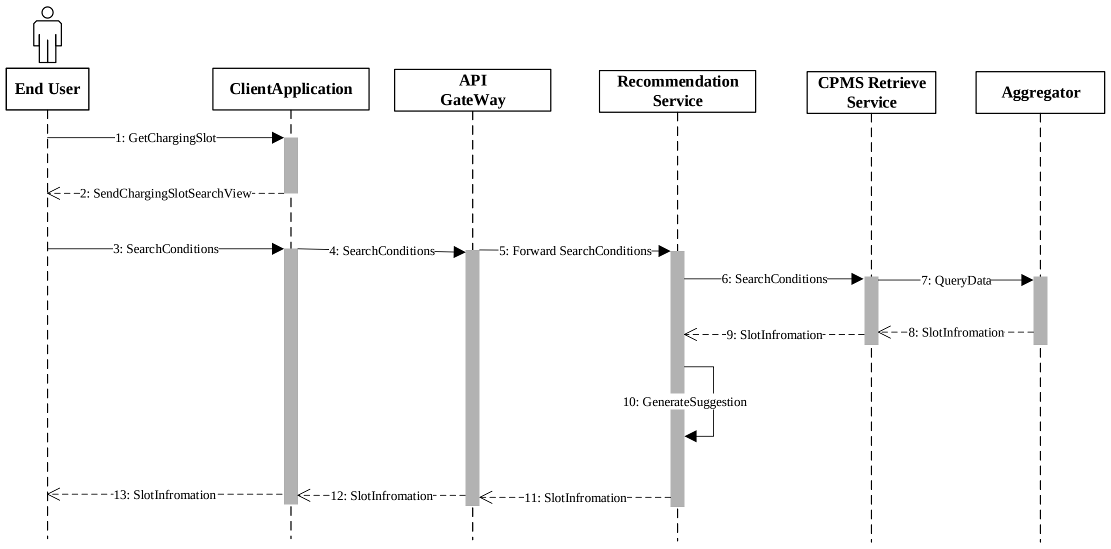


Figure 2-4-4 Sequence diagrams of "Search or get suggestion for charging slots"

2.4.3 Payment service

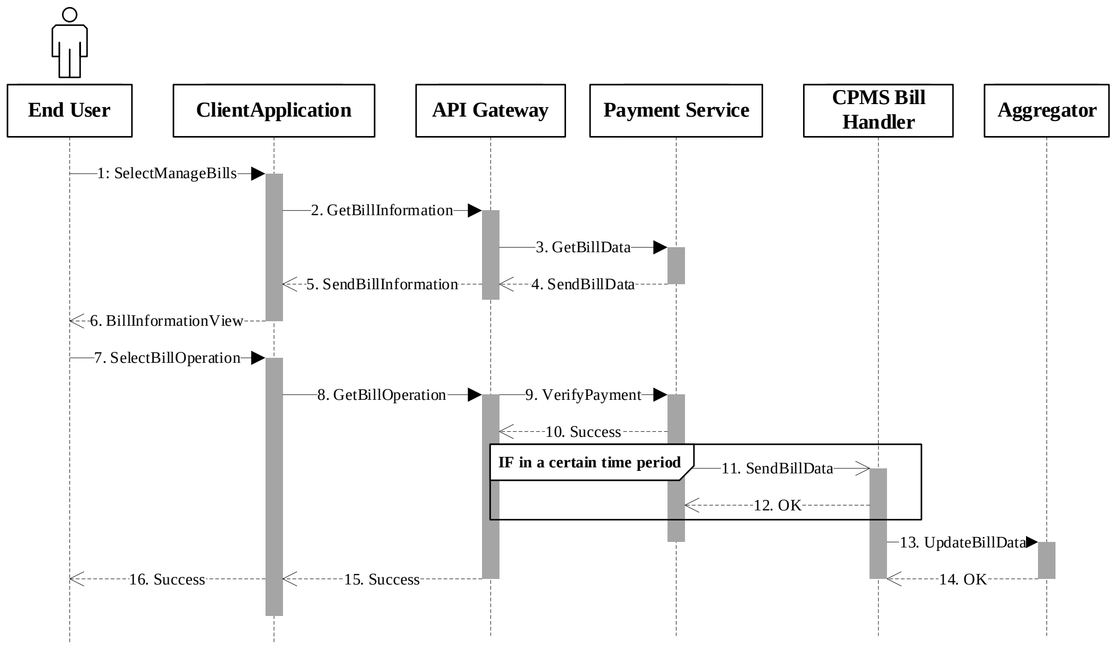


Figure 2-4-5 Sequence diagrams of "Bills management"

2.4.4 Charging service

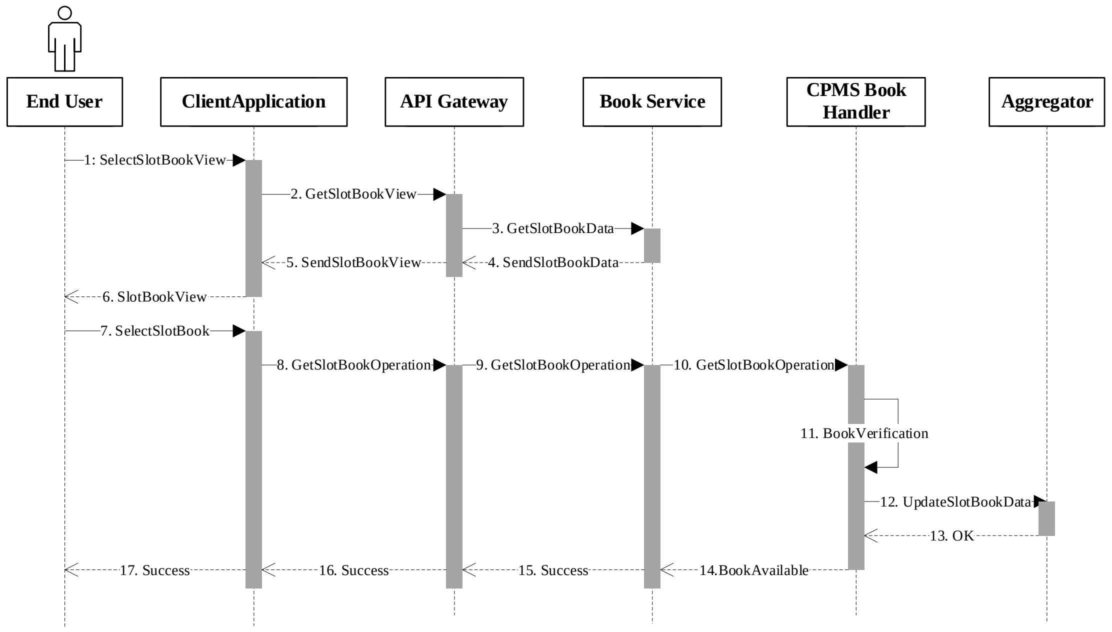


Figure 2-4-6 Sequence diagrams of "Book a charge slot"

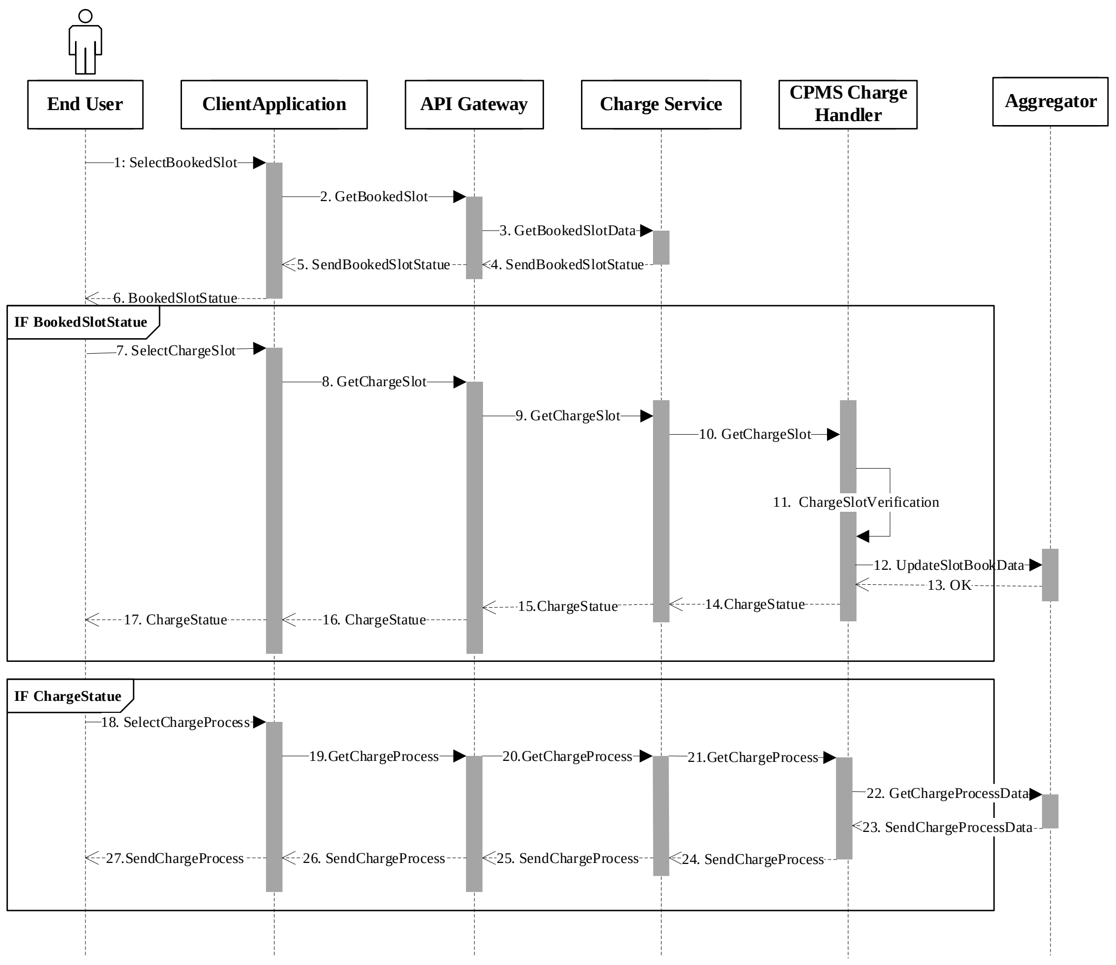
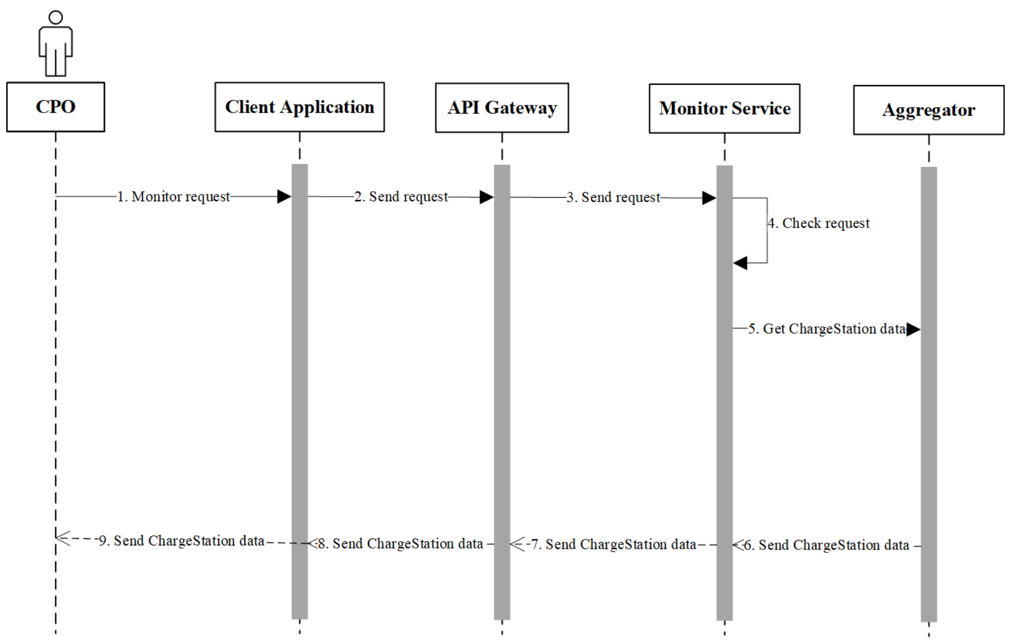


Figure 2-4-7 Sequence diagrams of “Charge vehicles and get to know charging process”

2.4.5 Charging station monitor service（CPO）



CPO
Client Application
API Gateway
Monitor Service
Aggregator
1. Monitor request
2. Send request
3. Send request
4. Check request
5. Get ChargeStation data
6. Send ChargeStation data
7. Send ChargeStation data
8. Send ChargeStation data
9. Send ChargeStation data

Figure 2-4-8 Sequence diagrams of “CPO monitor charging stations”

2.4.6 Charging station management service（CPO）

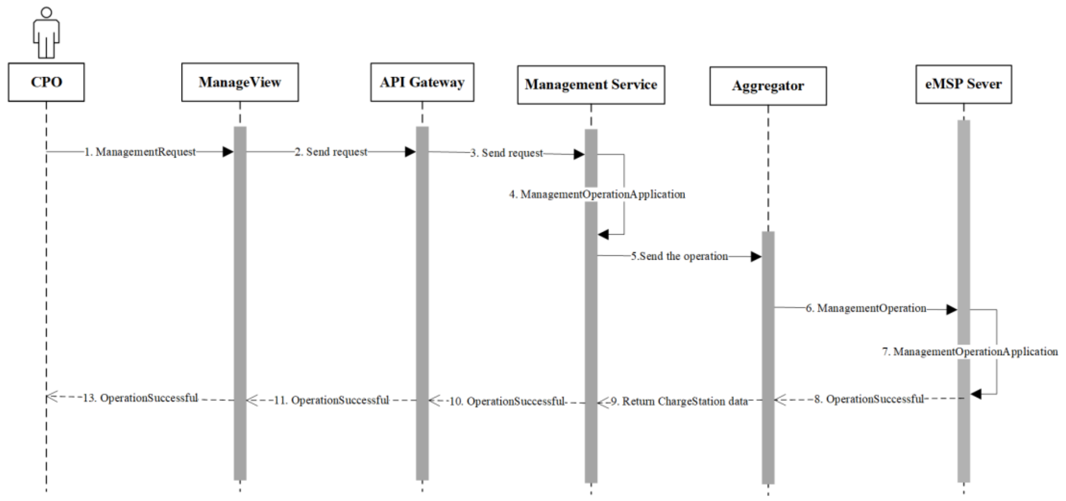


Figure 2-4-9 Sequence diagrams of "CPO manages charging stations."

2.4.7 Charging station energy service（CPO）

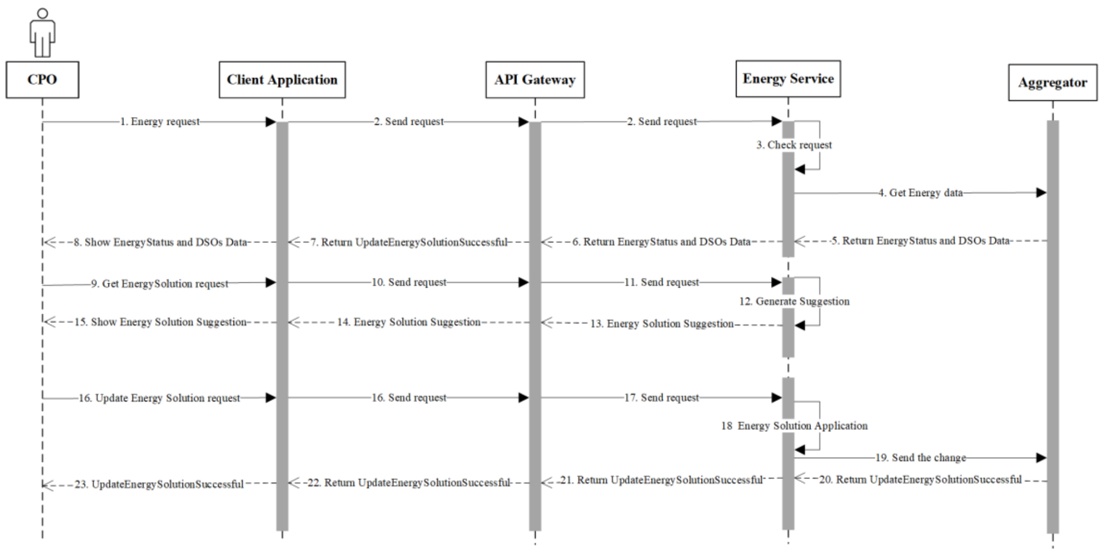

CPO
Client Application
API Gateway
Energy Service
Aggregator
1. Energy request
2. Send request
2. Send request
3. Check request
4. Get Energy data
5. Return EnergyStatus and DSOs Data
6. Return EnergyStatus and DSOs Data
7. Return UpdateEnergySolutionSuccessful
8. Show EnergyStatus and DSOs Data
9. Get EnergySolution request
10. Send request
11. Send request
12. Generate Suggestion
13. Energy Solution Suggestion
14. Energy Solution Suggestion
15. Show Energy Solution Suggestion
16. Update Energy Solution request
16. Send request
17. Send request
18 Energy Solution Application
19. Send the change
20. Return UpdateEnergySolutionSuccessful
21. Return UpdateEnergySolutionSuccessful
22. Return UpdateEnergySolutionSuccessful
23. UpdateEnergySolutionSuccessful

Figure 2-4-10 Sequence diagrams of "CPO chooses an energy solution for charging"

2.5 Component interfaces

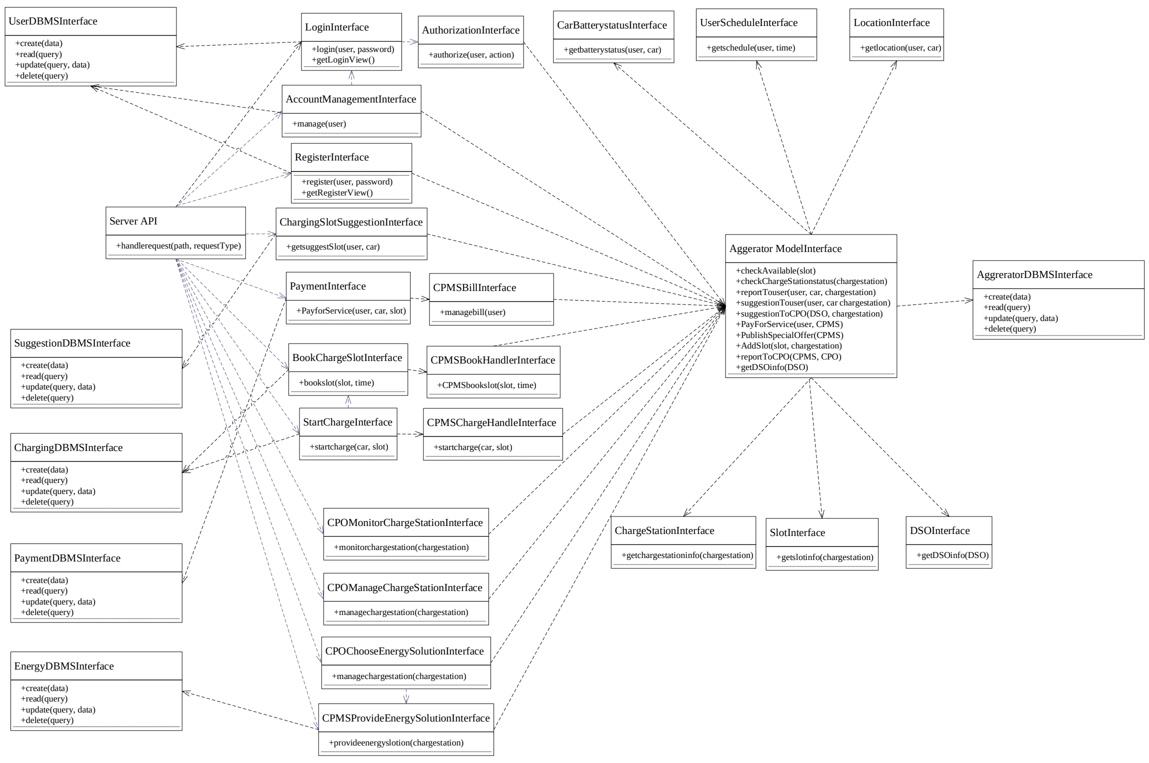


Figure 2-5-1 Diagram describing the component interfaces, and their dependencies

2.6 Selected architectural styles and patterns:

We use **Microservices Architecture** for eMSP and CPMS subsystems. In a microservices architecture, a software system is decomposed into a set of independent, self-contained services that can be developed, deployed, and scaled independently. Each service is responsible for a specific business capability and communicates with other services through well-defined interfaces, typically using lightweight, web-based APIs such as REST.

The microservices architecture for the eMSP and CPMS subsystems follows common design patterns below:

- **Service-oriented architecture (SOA) [2]**: The microservices architecture is a form of service-oriented architecture (SOA), in which the system is composed of a set of services that communicate with each other using well-defined interfaces. SOA promotes loose coupling between services, allowing them to evolve independently and be composed and reused in different contexts.
- **RESTful API design [3]**: The microservices in the system communicate with each other using RESTful APIs, which is a common architectural style for building web-based APIs. REST stands for Representational State Transfer and is based on the HTTP protocol, which makes it simple to use and widely supported. RESTful APIs use HTTP methods (e.g. GET, POST, PUT, DELETE)

to indicate the desired action and HTTP status codes to indicate the outcome of the request.

- **Domain-driven design (DDD) [4]**: The microservices in the system are organized around specific business capabilities, such as user management, charging, payment, and recommendation. This aligns with the principles of domain-driven design (DDD), which advocates for aligning the design of the system with the business domain and using a common language to communicate between domain experts and technical teams.

We will use Azure Kubernetes Service (AKS) [1] to develop and deploy the microservice system. AKS is a managed Kubernetes service that makes it easy to deploy, scale, and manage containerized applications. Based on the functionality of AKS, the development and deployment process can be briefly described as:

1. Containerize the services: Containerize the services using Docker. This involves packaging each service and its dependencies into a container image, which can then be run as a container. And then we can use a continuous integration/continuous deployment (CI/CD) pipeline to automate the build, test, and deployment of your container images. Or using Azure's PaaS services, we can also develop microservices based on Azure Function App.
2. Set up an AKS cluster: An AKS cluster will be set, which is a group of virtual machines that run the Kubernetes runtime and host your containerized services.
3. Deploy the services to the AKS cluster.
4. Expose your services to the external world: We will use load balancer and an ingress controller to make the services accessible to external clients. And Azure Active Directory can manage user access rights.
5. Monitor and manage your services: Azure Monitor can monitor the health and performance of the services.

2.7 Other design decisions

The make the eMSP smarter, smarter, we should also concern:

- Personalization: To make the recommendations more relevant to the user, we will consider incorporating personalization into your recommendation algorithms. This could involve using user profile data, such as the user's preferred charging locations or charging habits, to tailor the recommendations to the individual user.
- User feedback: To improve the accuracy and effectiveness of the recommendation algorithms over time, we will consider incorporating user feedback into the process. This could involve gathering explicit feedback from users (e.g. through ratings or reviews), or using implicit feedback signals (e.g. whether the user follows the recommendation) to refine the algorithms.

3. User Interface Design

3.1 User service

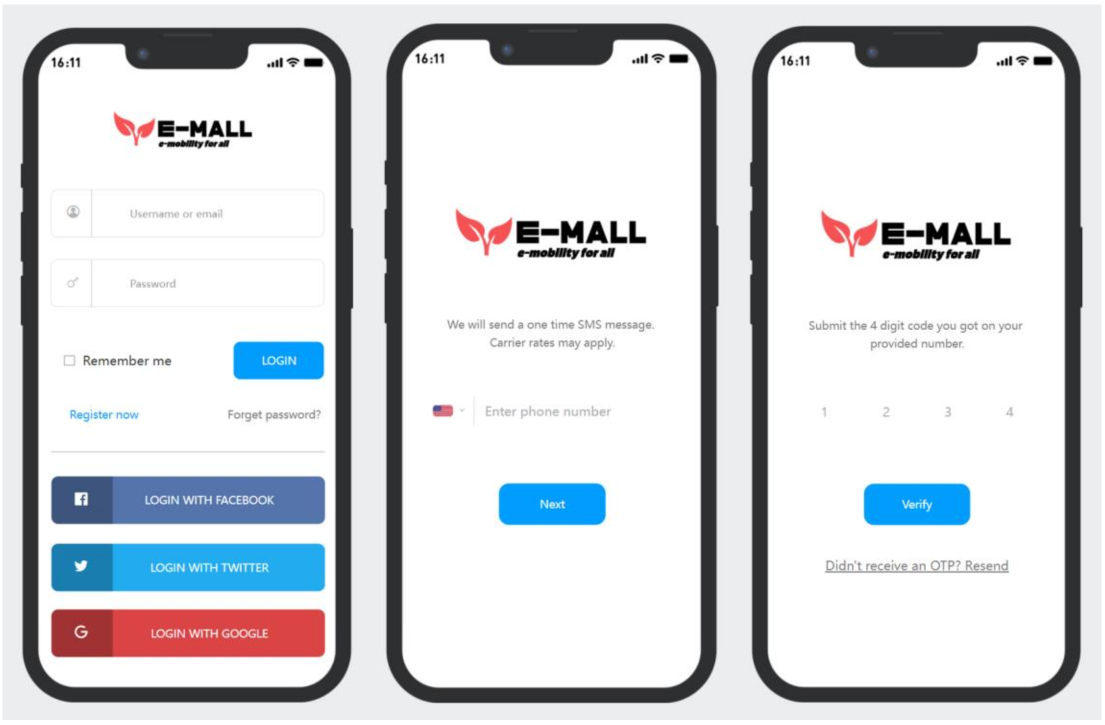


Figure 3-1-1~3 End users can log in or register an account using their cell phone number

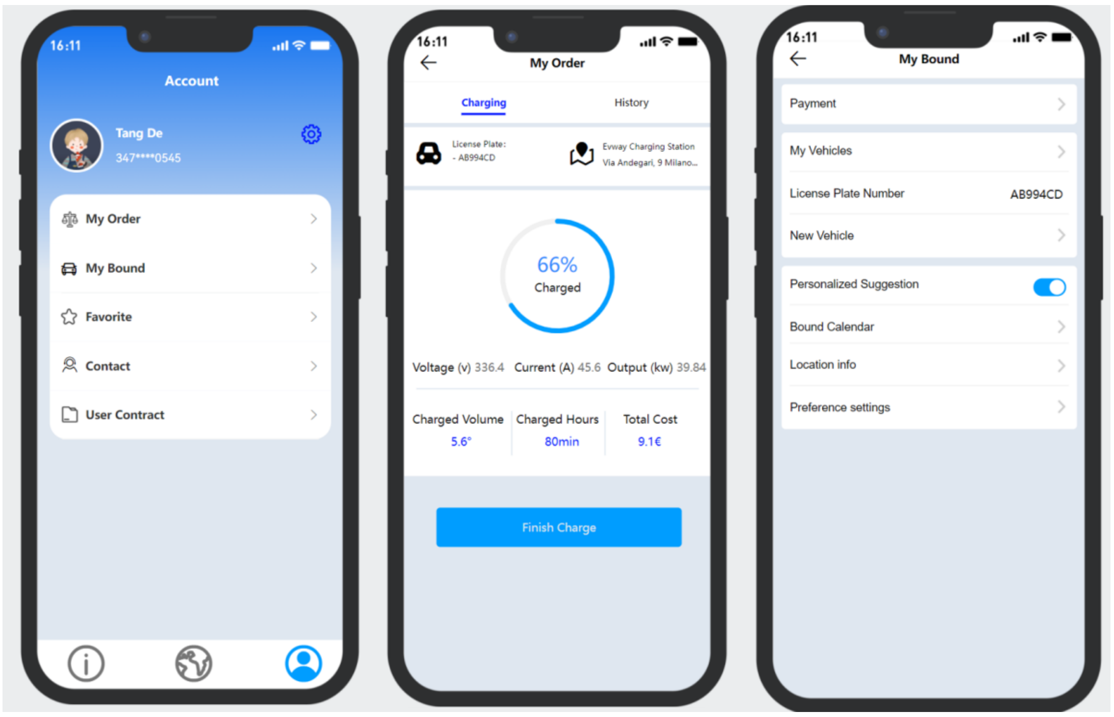
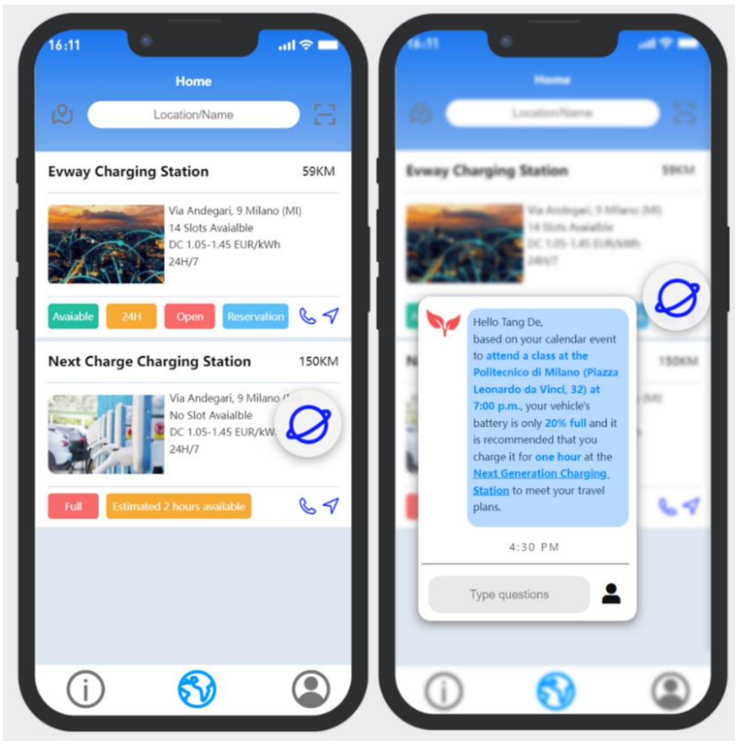


Figure 3-1-4~6 "Account" page, End users can manage their account. In my order subpage, end users can check the current charging process, and history orders. In My Bound subpage, end users can manage their payment and vehicles, and personalized suggestion authorization.

3.2 Recommendation service


16:11

Home

Location/Name

Evway Charging Station 59KM

Via Andegari, 9 Milano (MI)
14 Slots Avaialble
DC 1.05-1.45 EUR/kWh
24H/7

Next Charge Charging Station 150KM

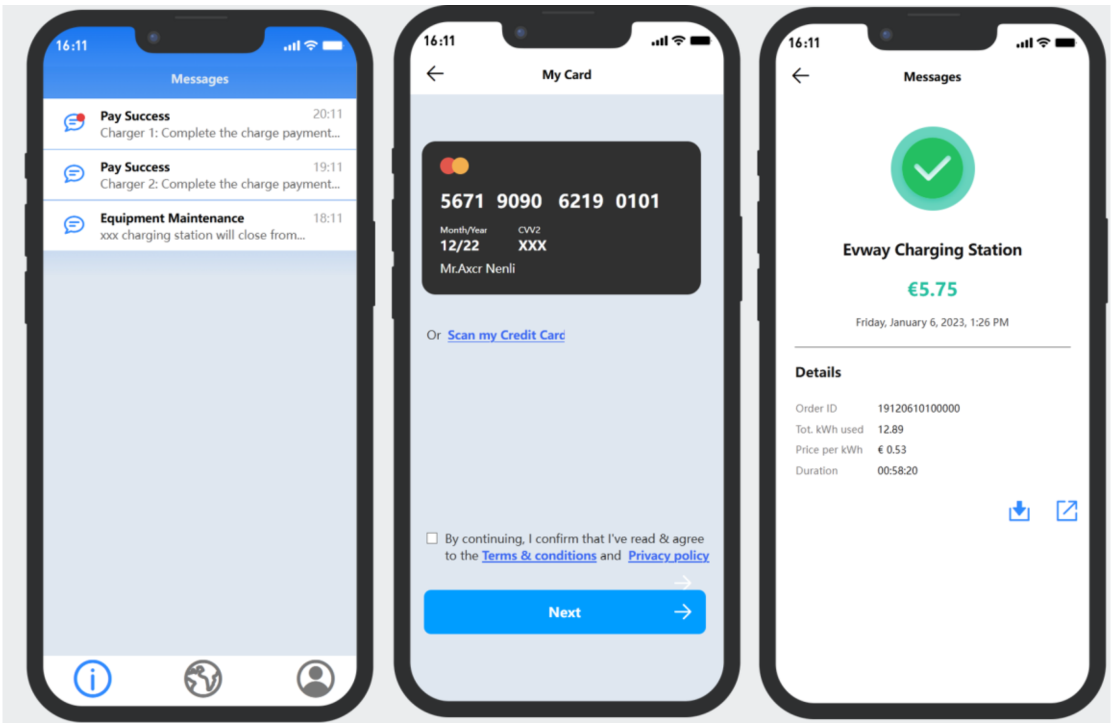
Via Andegari, 9 Milano
No Slot Avaialble
DC 1.05-1.45 EUR/kWh
24H/7

Hello Tang De,
based on your calendar event to attend a class at the Politecnico di Milano (Piazza Leonardo da Vinci, 32) at 7:00 p.m., your vehicle's battery is only 20% full and it is recommended that you charge it for one hour at the Next Generation Charging Station to meet your travel plans.

4:30 PM

Figure 3-2-1~2 "Home" page, end users can search for charging stations and browse basic information. And by clicking the globe icon on the page, e -Mall will intelligently recommend charging stations to users based on the information they have bound.

3.3 Payment service


16:11

Messages

Pay Success Charger 1: Complete the charge payment...	20:11
Pay Success Charger 2: Complete the charge payment...	19:11
Equipment Maintenance xxx charging station will close from...	18:11

My Card

5671 9090 6219 0101
Month/Year 12/22
CVV2 XXX
Mr.Axcr Nenli

Or Scan my Credit Card

☐ By continuing, I confirm that I've read & agree to the Terms & conditions and Privacy policy

Messages

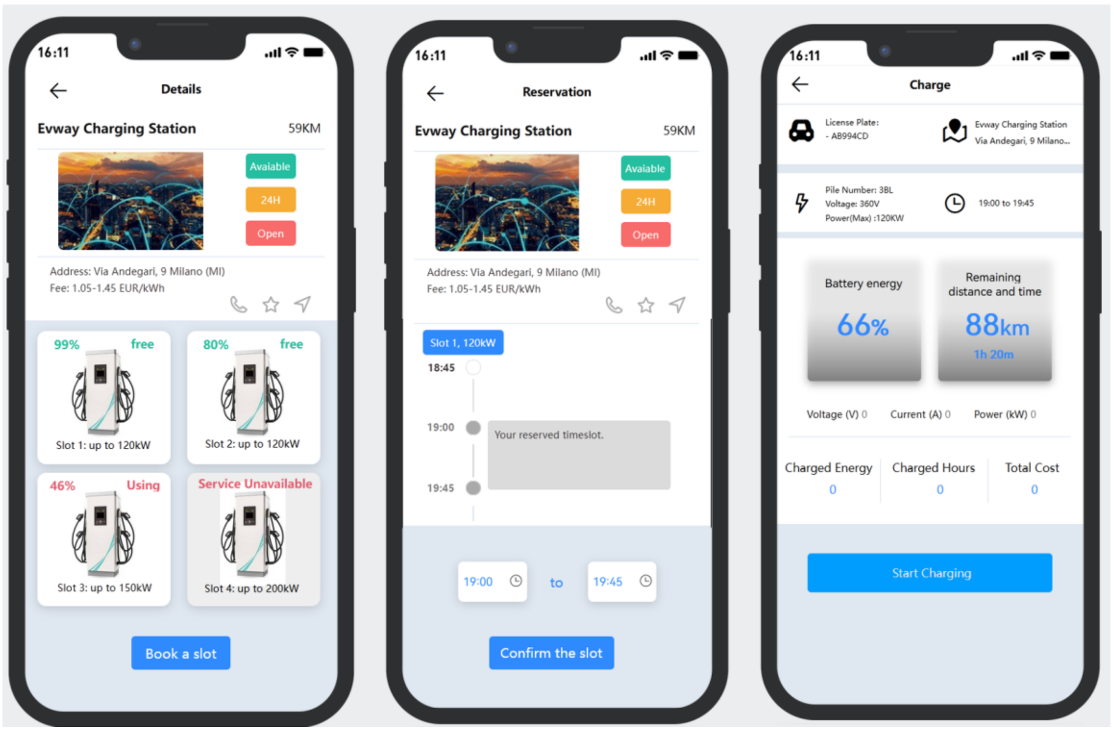
Evway Charging Station
€5.75
Friday, January 6, 2023, 1:26 PM

Details

Order ID	19120610100000
Tot. kWh used	12.89
Price per kWh	€ 0.53
Duration	00:58:20

Figure 3-3-1~2 "Messages" page, notify the end users of the payment status, and click to view the details of the bill.

3.4 Charging service


16:11

Details

Evway Charging Station 59KM

Avalable

24H

Open

Address: Via Andegari, 9 Milano (MI)
Fee: 1.05-1.45 EUR/kWh

99% free
Slot 1: up to 120kW

80% free
Slot 2: up to 120kW

46% Using
Slot 3: up to 150kW

Service Unavailable
Slot 4: up to 200kW

Book a slot

16:11

Reservation

Evway Charging Station 59KM

Avalable

24H

Open

Address: Via Andegari, 9 Milano (MI)
Fee: 1.05-1.45 EUR/kWh

Slot 1, 120kW

18:45

19:00

Your reserved timeslot.

19:45

19:00 to 19:45

Confirm the slot

16:11

Charge

License Plate:
- AB994CD

Evway Charging Station
Via Andegari, 9 Milano...

Pile Number: 38L
Voltage: 360V
Power(Max) :120KW

19:00 to 19:45

Battery energy
66%

Remaining distance and time
88km
1h 20m

Voltage (V) 0 Current (A) 0 Power (kW) 0

Charged Energy 0

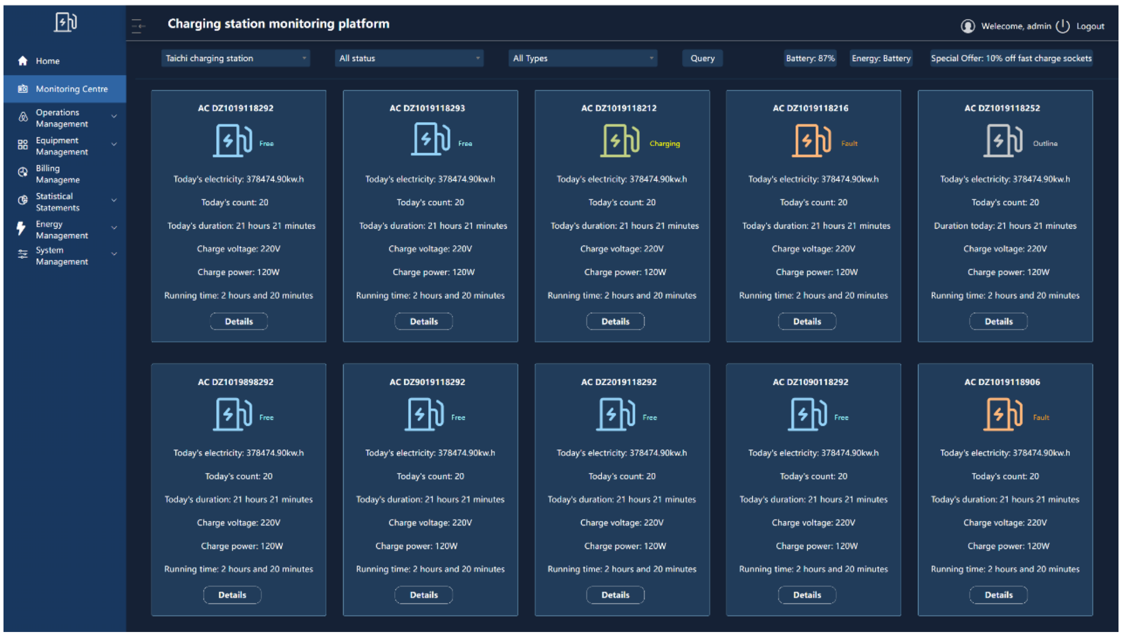
Charged Hours 0

Total Cost 0

Start Charging

Figure 3-4-1~3 In "Home" page, click on the available charging stations, select a suitable charging slot, and make an appointment at the desired time, and start charging the vehicle.

3.5 Charging station monitor service


Charging station monitoring platform

Welecome, admin Logout

Home

Monitoring Centre

Operations Management

Equipment Management

Billing Manageme

Statistical Statements

Energy Management

System Management

Taichi charging station

All status

All Types

Query

Battery: 87%

Energy: Battery

Special Offer: 10% off fast charge sockets

AC DZ1019118292
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019118293
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019118212
Charging
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019118216
Fault
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019118252
Outline
Today's electricity: 378474.90kw.h
Today's count: 20
Duration today: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019898292
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

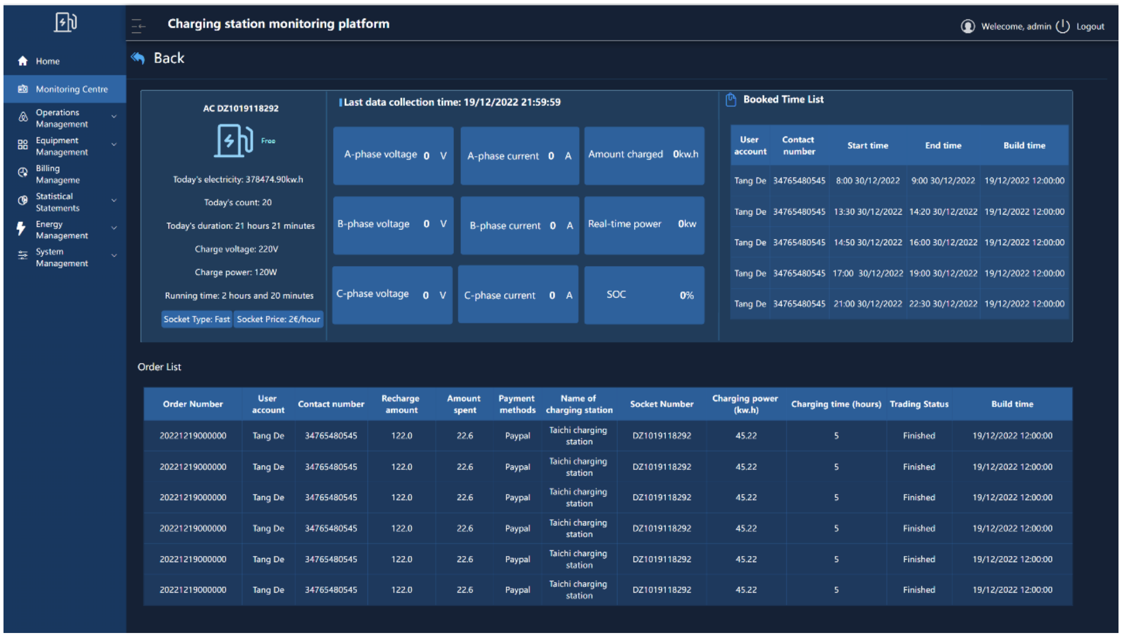
AC DZ9019118292
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ2019118292
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1090118292
Free
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

AC DZ1019118906
Fault
Today's electricity: 378474.90kw.h
Today's count: 20
Today's duration: 21 hours 21 minutes
Charge voltage: 220V
Charge power: 120W
Running time: 2 hours and 20 minutes
Details

Figure 3-5-1In "Monitoring Centre" page, the administrator can view information on selected charging stations, including energy type, special offer and information on each socket. And by selecting the current status of the outlet and the type of outlet in the drop down box he can find the eligible sockets.



AC DZ1019118292

Today's electricity: 378474.90kw.h

Today's count: 20

Today's duration: 21 hours 21 minutes

Charge voltage: 220V

Charge power: 120W

Running time: 2 hours and 20 minutes

Socket Type: Fast | Socket Price: 2€/hour

Last data collection time: 19/12/2022 21:59:59

A-phase voltage 0 V	A-phase current 0 A	Amount charged 0kw.h
B-phase voltage 0 V	B-phase current 0 A	Real-time power 0kw
C-phase voltage 0 V	C-phase current 0 A	SOC 0%

Booked Time List

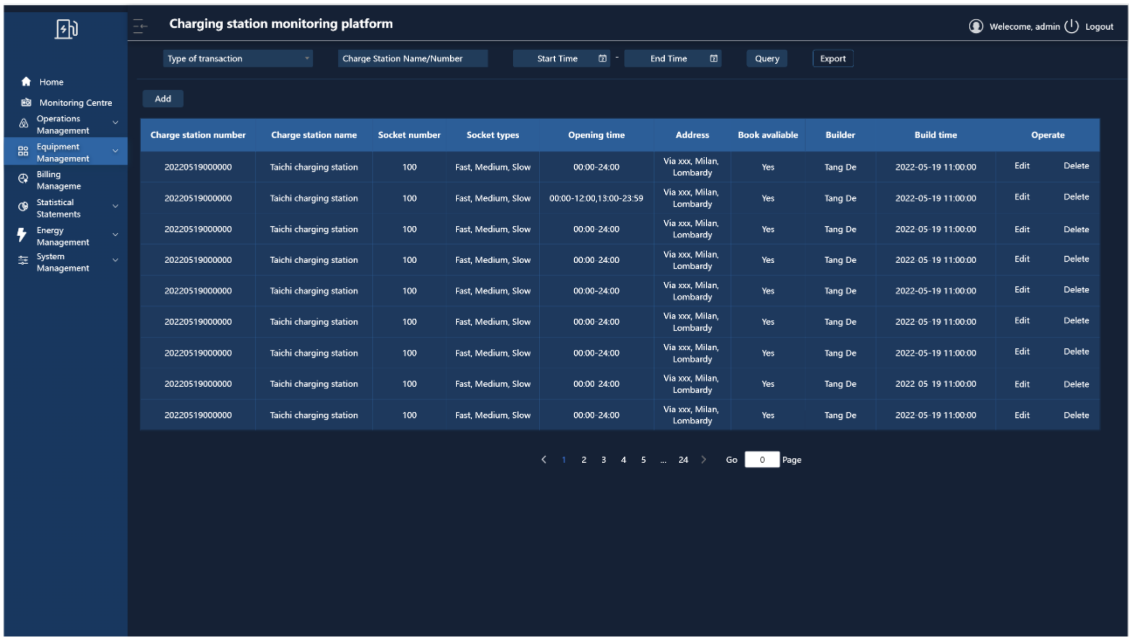
User account	Contact number	Start time	End time	Build time
Tang De	34765480545	8:00 30/12/2022	9:00 30/12/2022	19/12/2022 12:00:00
Tang De	34765480545	13:30 30/12/2022	14:20 30/12/2022	19/12/2022 12:00:00
Tang De	34765480545	14:50 30/12/2022	16:00 30/12/2022	19/12/2022 12:00:00
Tang De	34765480545	17:00 30/12/2022	19:00 30/12/2022	19/12/2022 12:00:00
Tang De	34765480545	21:00 30/12/2022	22:30 30/12/2022	19/12/2022 12:00:00

Order List

Order Number	User account	Contact number	Recharge amount	Amount spent	Payment methods	Name of charging station	Socket Number	Charging power (kw.h)	Charging time (hours)	Trading Status	Build time
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00
20221219000000	Tang De	34765480545	122.0	22.6	Paypal	Taichi charging station	DZ1019118292	45.22	5	Finished	19/12/2022 12:00:00

Figure 3-5-2 By clicking on the Details button below each socket in Figure 3-5-1, the administrator can view more detailed information about the socket, including voltage phase, current phase, socket type, socket price, booked list and its order list.

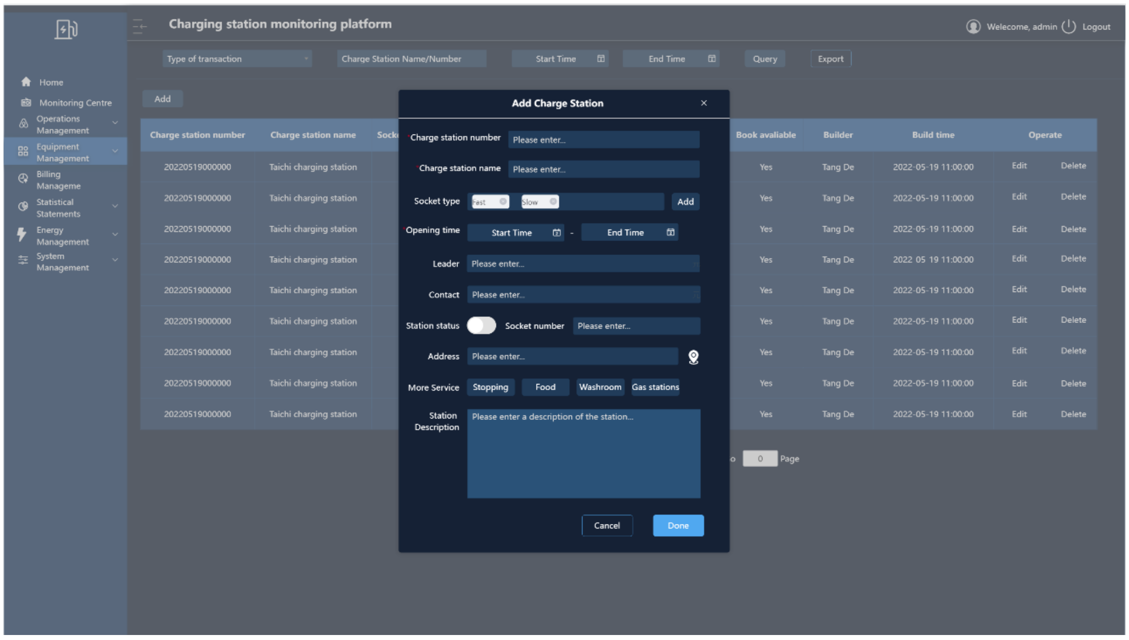
3.6 Charging station manage service



Charge station number	Charge station name	Socket number	Socket types	Opening time	Address	Book avaliable	Builder	Build time	Operate	
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-12:00,13:00-23:59	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit	Delete

< 1 2 3 4 5 ... 24 > Go 0 Page

Figure 3-6-1 On the device management page, the administrator can manage each charging station. He can use the drop-down box at the top of the page to set the search criteria, including the type of charging station transaction (whether it supports reservations), the name of the station and its opening hours to find the stations that match the criteria. He can also add and delete stations by clicking on the Add button and Delete button.



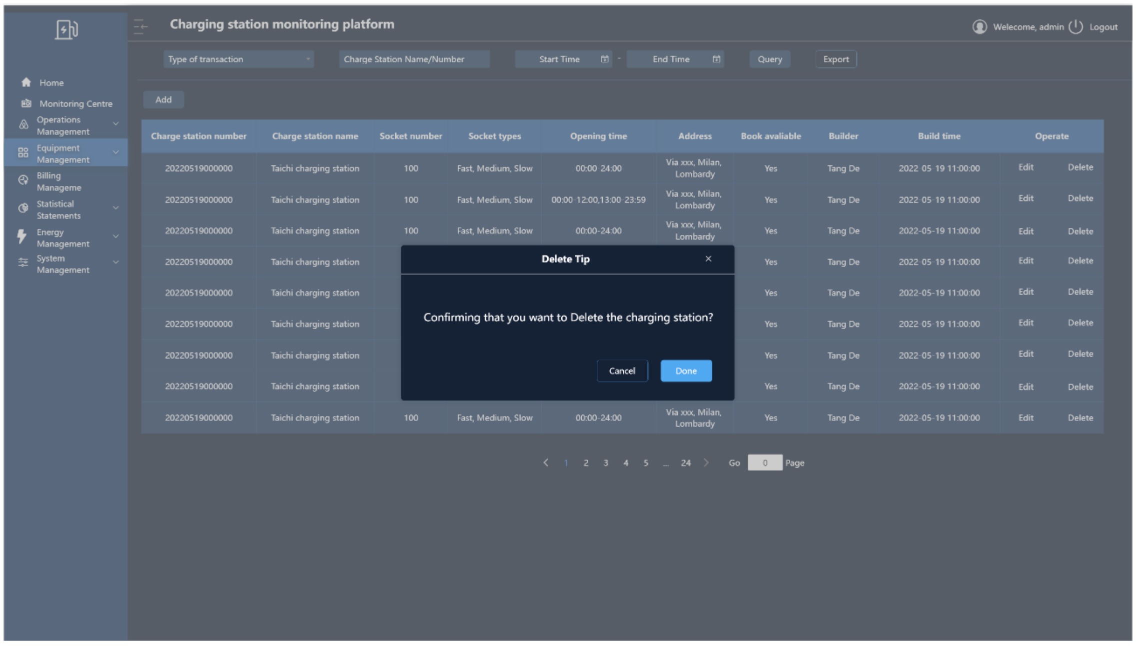
Add Charge Station

*Charge station number	Please enter...
*Charge station name	Please enter...
Socket type	Fast, Slow — Add
*Opening time	Start Time - End Time
Leader	Please enter...
Contact	Please enter...
Station status	Socket number: Please enter...
Address	Please enter...
More Service	Stopping, Food, Washroom, Gas stations
Station Description	Please enter a description of the station...

Cancel Done

Charge station number	Charge station name	Sock	Book avaliable	Builder	Build time	Operate
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022 05 19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station		Yes	Tang De	2022-05-19 11:00:00	Edit Delete

Figure 3-6-2 By clicking on the Add button in Figure 3-6-1, the administrator can add a new charging station, which must specify the station number, the name of the charging station and its opening hours. In addition to this, the administrator can also add the type of socket the station has and add its services.



Charge station number	Charge station name	Socket number	Socket types	Opening time	Address	Book avaliable	Builder	Build time	Operate
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00 24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022 05 19 11:00:00	Edit Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00 12:00,13:00 23:59	Via xxx, Milan, Lombardy	Yes	Tang De	2022 05 19 11:00:00	Edit Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station					Yes	Tang De	2022 05 19 11:00:00	Edit Delete
20220519000000	Taichi charging station					Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station					Yes	Tang De	2022 05 19 11:00:00	Edit Delete
20220519000000	Taichi charging station					Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station					Yes	Tang De	2022-05-19 11:00:00	Edit Delete
20220519000000	Taichi charging station	100	Fast, Medium, Slow	00:00-24:00	Via xxx, Milan, Lombardy	Yes	Tang De	2022-05-19 11:00:00	Edit Delete

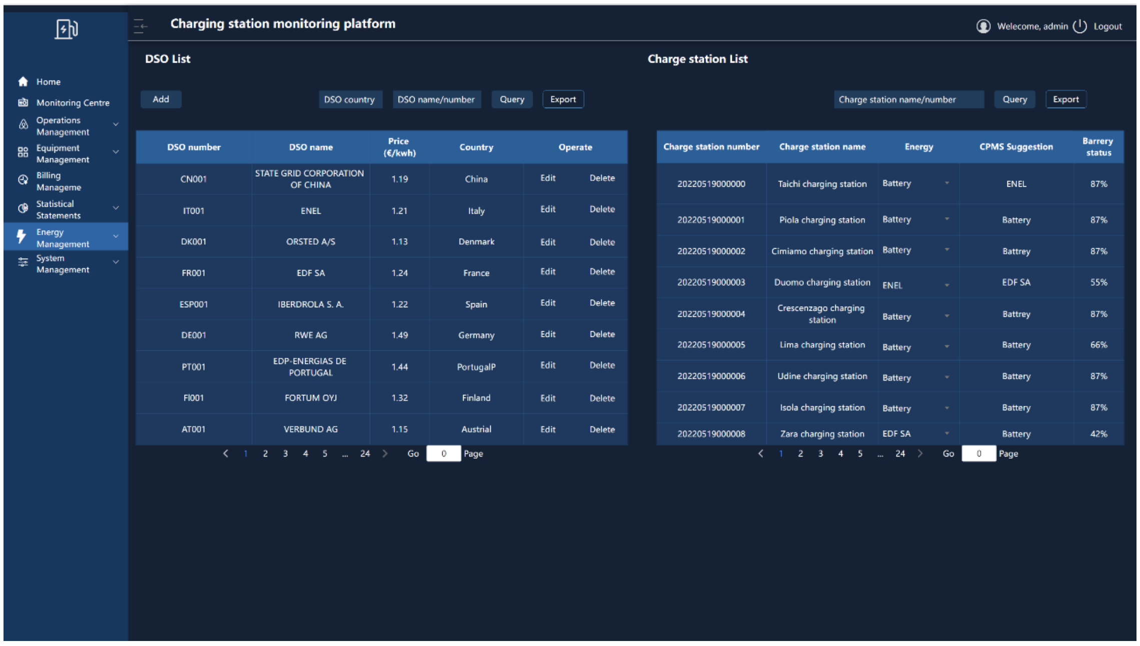
Delete Tip

Confirming that you want to Delete the charging station?

Cancel Done

Figure 3-6-3 By clicking the Delete button in Figure 3-6-1, the administrator can delete the charging station and a delete prompt will pop up.

3.7 Charging station energy service



Charging station monitoring platform

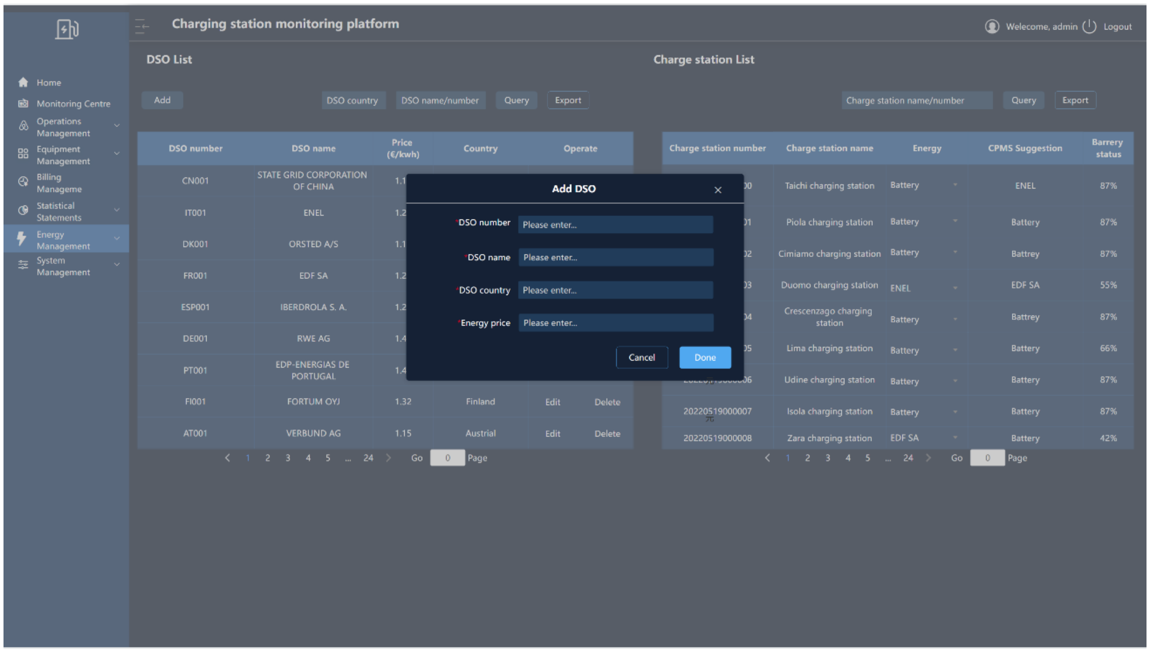
DSO List

DSO number	DSO name	Price (€/kwh)	Country	Operate	
CN001	STATE GRID CORPORATION OF CHINA	1.19	China	Edit	Delete
IT001	ENEL	1.21	Italy	Edit	Delete
DK001	ORSTED A/S	1.13	Denmark	Edit	Delete
FR001	EDF SA	1.24	France	Edit	Delete
ESP001	IBERDROLA S. A.	1.22	Spain	Edit	Delete
DE001	RWE AG	1.49	Germany	Edit	Delete
PT001	EDP-ENERGIAS DE PORTUGAL	1.44	PortugalP	Edit	Delete
FI001	FORTUM OYJ	1.32	Finland	Edit	Delete
AT001	VERBUND AG	1.15	Austrial	Edit	Delete

Charge station List

Charge station number	Charge station name	Energy	CPMS Suggestion	Barrery status
20220519000000	Taichi charging station	Battery	ENEL	87%
20220519000001	Piola charging station	Battery	Battery	87%
20220519000002	Cimiamo charging station	Battery	Battrey	87%
20220519000003	Duomo charging station	ENEL	EDF SA	55%
20220519000004	Crescenzago charging station	Battery	Battrey	87%
20220519000005	Lima charging station	Battery	Battery	66%
20220519000006	Udine charging station	Battery	Battery	87%
20220519000007	Isola charging station	Battery	Battery	87%
20220519000008	Zara charging station	EDF SA	Battery	42%

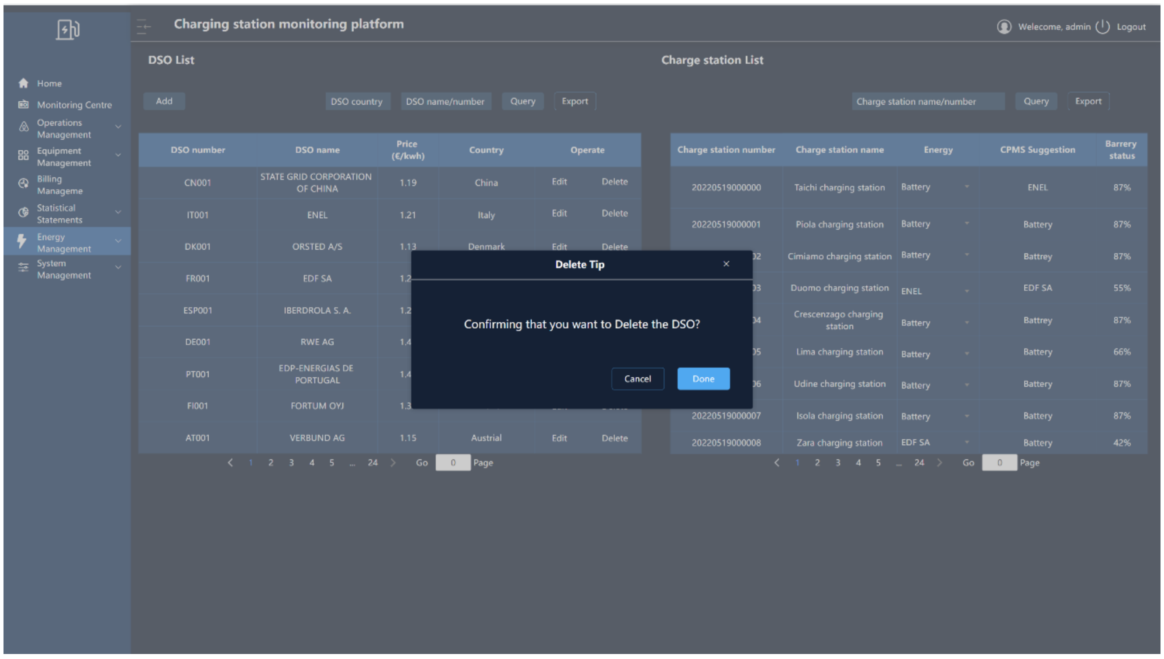
Figure 3-7-1 On the energy management page, the administrator can view the energy list and the list of charging stations. The energy list provides the DSOs available for that CPO and their prices, and the drop-down box above allows you to set up a conditional search. The charging station list contains all charging stations and the type of energy they are using, including the recommended energy type for the CPMS. The drop down box above enables conditional queries.



Add DSO

*DSO number	Please enter...
*DSO name	Please enter...
*DSO country	Please enter...
*Energy price	Please enter...

Figure 3-7-2 By clicking on the Add button above the DSO list in Figure 3-7-1, the administrator can add a DSO.



Confirming that you want to Delete the DSO?

DSO number	DSO name	Price (€/kwh)	Country	Operate	
CN001	STATE GRID CORPORATION OF CHINA	1.19	China	Edit	Delete
IT001	ENEL	1.21	Italy	Edit	Delete
DK001	ORSTED A/S	1.13	Denmark	Edit	Delete
FR001	EDF SA	1.2			
ESP001	IBERDROLA S. A.	1.2			
DE001	RWE AG	1.4			
PT001	EDP-ENERGIAS DE PORTUGAL	1.4			
FI001	FORTUM OYJ	1.3			
AT001	VERBUND AG	1.15	Austrial	Edit	Delete

Charge station number	Charge station name	Energy	CPMS Suggestion	Barrery status
20220519000000	Taichi charging station	Battery	ENEL	87%
20220519000001	Piola charging station	Battery	Battery	87%
02	Cimiamo charging station	Battery	Battrey	87%
03	Duomo charging station	ENEL	EDF SA	55%
04	Crescenzago charging station	Battery	Battrey	87%
05	Lima charging station	Battery	Battery	66%
06	Udine charging station	Battery	Battery	87%
20220519000007	Isola charging station	Battery	Battery	87%
20220519000008	Zara charging station	EDF SA	Battery	42%

Figure 3-7-2 By clicking on the Delete button in the DSO list in Figure 3-7-1, the administrator can dalete a DSO.

4. Requirements Traceability

4.1 Requirements and the design elements

Requirements traceability is the process of tracking the relationships between requirements and the design elements that fulfill them. In this section, requirements traceability would involve linking the requirements defined in the RASD to the corresponding components in the design document. To save some space, the components have been given abbreviations, see list below.

- CV - Client View
- API - API Gate Way
- RGS - Registration Service
- ACS – Authentication Service
- AZS - Authorization Service
- AMS - Account Management Service
- RS - Recommendation Service
- BS - Book Service
- CS - Charge Service
- PS - Payment Service
- MS - Monitor Service
- SMS - Station Management Service
- ES - Energy Service
- RH - CPMS Retrieve Handler
- BH - CPMS Book Handler
- CH - CPMS Charge Handler
- PH - CPMS Payment Handler

- AS - Aggregator Service
- DB – Database

Additionally, the requirements defined in RASD are as follows.

Requirements	**Description**
R1	The system should allow an unregistered end user to register an account.
R2	The system should allow a registered end user to insert data about personal, vehicle, or cell phone and email binding information.
R3	The system should allow a registered end user to authorize eMSP to bind their own schedule, location information and vehicle battery status.
R4	The system should allow a registered end user to search charging slots.
R5	The system should be able to suggest a personalized charging slot.
R6	The system should allow a registered end user to book a charging slot in advance.
R7	The system should allow a registered end user to confirm the booking and initiate the charging process.
R8	The system should be able to control and monitor the charging of electric vehicles at the charge points.
R9	The system should be able to generate a bill for the charging service.
R10	The system should allow a registered end user to pay the bill.
R11	The system should allow a registered CPO to monitor charging stations.
R12	The system should allow a registered CPO to manage charging stations.
R13	The system should allow a registered CPO to acquire the current energy prices from DSOs.
R14	The system should be able to suggest a range of energy solutions.
R15	The system should allow a registered CPO to choose to select one of the suggested solutions, or manually input their own energy solution.
R16	The system must allow registered end users to login.
R17	The system must allow registered CPO to login.
R18	The system must be able to send notifications about the status and availability of the charge points, as well as any issues or updates related to the system.
R19	The system must be able to collect, store, and analyze data on the use of the charge points.
R20	The system must be able to ensure the security and integrity of the system.
R21	When a user registers, the system must authenticate that the user is associated to the specified electric vehicle

4.2 Mapping on Components

Requirements	**Components**
R1	CV, API, RGS, DB
R2	CV, API, ACS, AMS, DB
R3	CV, API, ACS, AZC, AMS, DB
R4	CV, API, ACS, RH, DB, AS
R5	CV, API, ACS, RS, RH, DB, AS
R6	CV, API, ACS, BS, BH, DB
R7	CV, API, ACS, BS, BH, DB, AS
R8	CV, API, ACS, CS, CH, DB, AS
R9	CV, API, ACS, PS, DB, AS
R10	CV, API, ACS, PS, PH, DB, AS
R11	CV, API, ACS, MS, DB, AS
R12	CV, API, ACS, SMS, DB, AS
R13	CV, API, ACS, ES, DB
R14	CV, API, ACS, ES, DB, AS
R15	CV, API, ACS, ES, DB, AS
R16	CV, API, ACS, DB
R17	CV, API, ACS, DB
R18	CV, API, AS
R19	CV, API, DB, AS
R20	CV, API, DB, AS
R21	CV, API, ACS, AZC, DB

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. This division is strongly based on the component diagram, see Figure 2-2-1. The system is to be developed according to the four subparts listed below, with the aim of following a from the outside in path of development (considering the component diagram). The intention with this is to have the data requirements (i.e every service's DBMS and Aggregator component) dictate the rest of the system structure.

5.1.1 Microservice & Database

This subpart involves implementing the microservices and their associated databases.

5.1.2 API Gateway

This subpart involves implementing the API Gateway component, which is responsible for routing API requests from the eMSP and CPO to the appropriate microservices and aggregating the responses.

5.1.3 Aggregator & Database

This subpart involves implementing the Aggregator component, which is responsible for collecting data from the various charging point management systems (CPMSs) and presenting it to the e-mobility service providers (eMSPs) in a consistent format. The database tables with the specified rows will be created, in tandem with the Aggretor component.

5.1.4 Client Application

This is the view part of the system, where all the user interactivity will be developed. The modular way in which this system is structured means that this component only has to be responsible for the graphical user interface and nothing else.

5.2 Integration & Test plan

The sequence in which we integrate components will follow the implementation order described in the previous section. The outside in order, as seen from the component diagram, corresponds to an outside in testing strategy which should work well considering the quite simple hierarchical structure of the system. With this approach drivers need to be implemented for every component that is integration tested.

The choice of an outside strategy was natural and with it incrementally testing throughout the development of the system should be the same. Before testing to integrate a component, two conditions should be fulfilled. Firstly, the component's interface should offer all functionality specified in the component interface diagram, see Figure 2-5-1. Secondly, the component should pass all unit tests.

Of course, when replacing a driver with the implementation, the integration tests need to be checked again.

To further clarify, below the order in which to run the integration tests is presented with explanatory graphics. The graphics only show one of the microservice, though the rest of them follow in the same way as they are on the same level in the hierarchy.

5.2.1 Microservice & DBMS

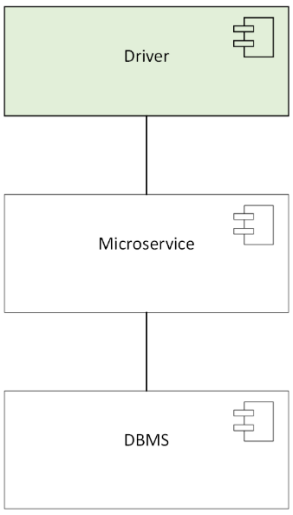


Figure 5-2-1 Integration test of Microservice & DBMS

5.2.2 External API

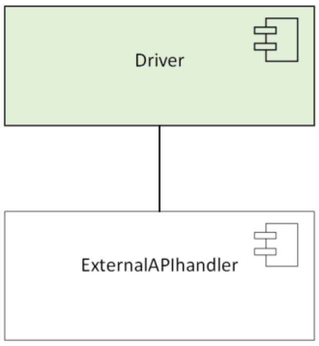


Figure 5-2-2 The integration of the external API component

5.2.3 Performance Calculator

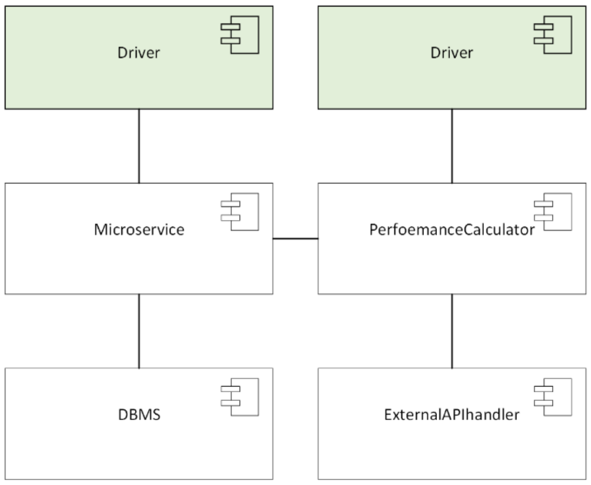


Figure 5-2-3 Integrating the performance calculator

5.2.4 Gateway API

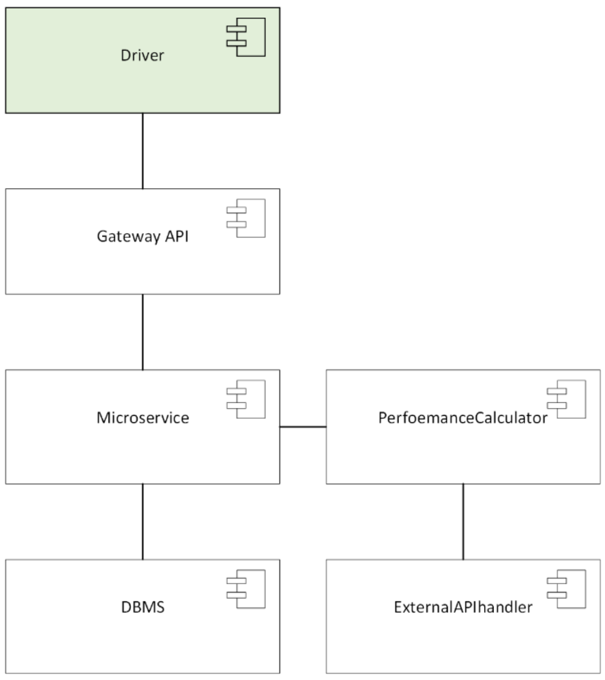


Figure 5-2-4 Integrating the Gateway API

5.2.5 Aggregator & DBMS

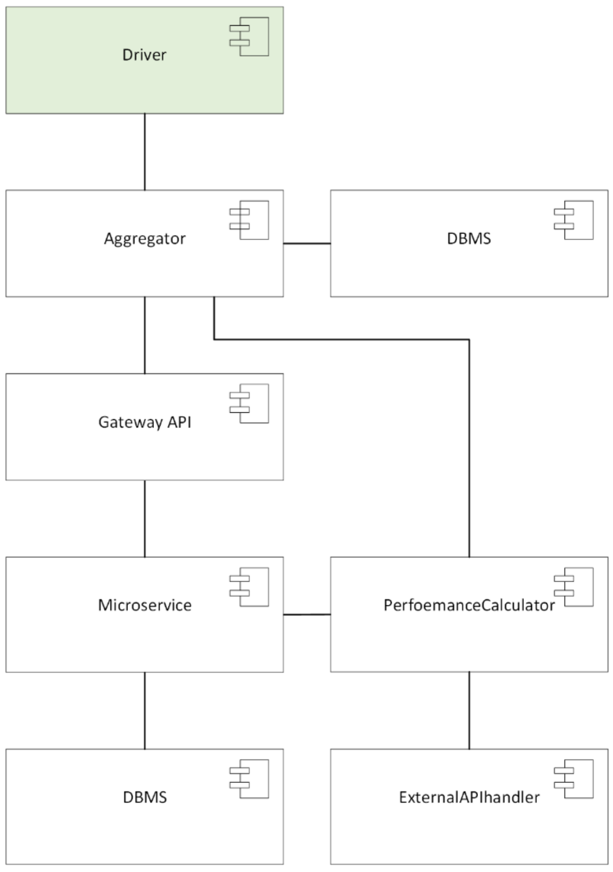


Figure 5-2-5 Integrating the Aggregator & DBMS

5.2.6 ClientAPP

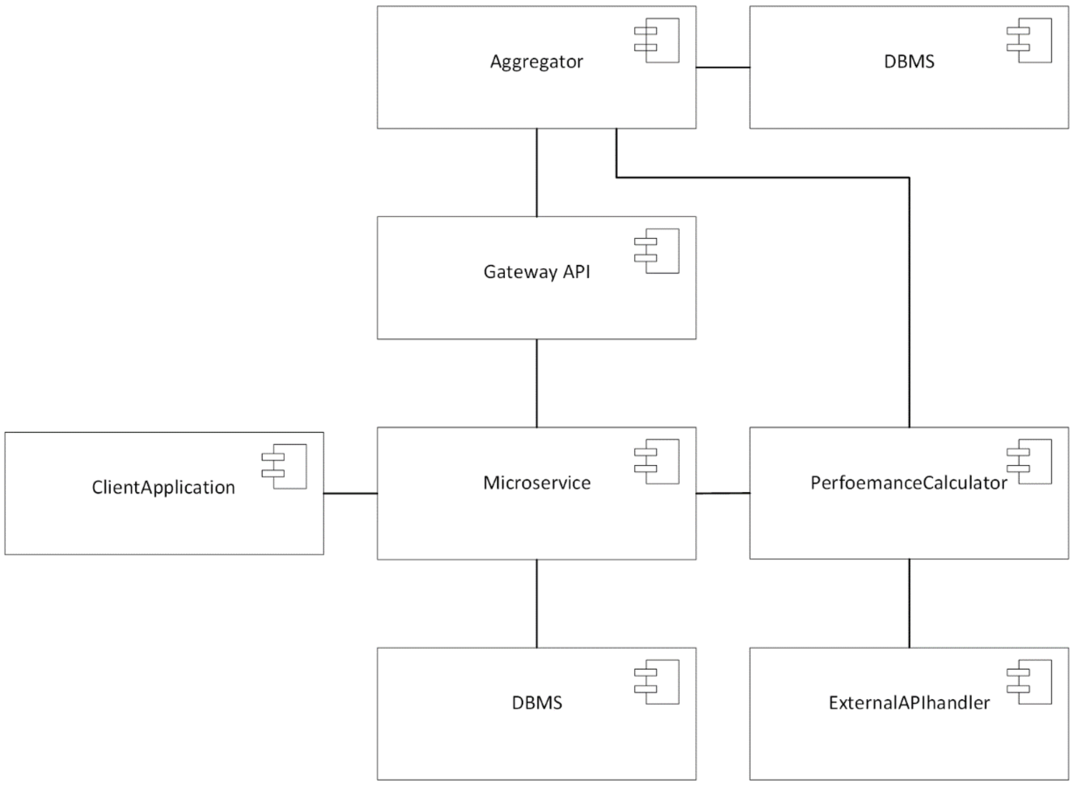


Figure 5-2-6 Integrating the Client Application

As soon as the whole system has been integrated, system testing can begin in order to assess the fulfilment of functional and n on-functional requirements. If possible, some form of usability testing should be conducted after the system tests pass. This, among other things, to evaluate the design on its user friendliness.

6. Effort Spent

6.1 Haotian Zhang's effort

Task	**Time spent**
Introduction	8 h
Architectural design	22 h
User Interface Design	18 h
Requirements Traceability	5 h
Implementation, Integration and Test Plan	2 h
Total	**55 h**

6.2 Jiaheng Xiong's effort

Task	**Time spent**
Introduction	8 h
Architectural design	10 h
User Interface Design	20 h

Requirements Traceability	2 h
Implementation, Integration and Test Plan	10 h
Total	**55 h**

6.3 Chenyu Zhao’s effort

Task	**Time spent**
Introduction	5 h
Architectural design	15 h
User Interface Design	20 h
Requirements Traceability	10 h
Implementation, Integration and Test Plan	5 h
Total	**55 h**

7. References

[1] Martinekuan. “Microservices with AKS and Azure DevOps - Azure Solution Ideas | Microsoft Learn.” Microsoft Learn: Build Skills That Open Doors in Your Career, https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/microservices-with-aks. Accessed 7 Jan. 2023.
[2] Nishanil. “Service-Oriented Architecture | Microsoft Learn.” Microsoft Learn: Build Skills That Open Doors in Your Career, https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/service-oriented-architecture. Accessed 7 Jan. 2023.
[3] martinekuan. “Web API Design Best Practices - Azure Architecture Center | Microsoft Learn.” Microsoft Learn: Build Skills That Open Doors in Your Career, https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design. Accessed 7 Jan. 2023.
[4] kexugit. “Best Practice - An Introduction To Domain-Driven Design | Microsoft Learn.” Microsoft Learn: Build Skills That Open Doors in Your Career, https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/best-practice-an-introduction-to-domain-driven-design. Accessed 7 Jan. 2023.