TQS: Product specification report

Diogo Costa[112714],Bruno Tavares[113372],Francisco Pinto[113763],André Alves[113962]
v2025-XX-XX

1 Introduction

1.1 Overview of the project

We will develop a web application with a Spring Boot backend to manage electric vehicle charging. The project will follow DevOps and Software Quality Assurance practices, including CI/CD, automated testing, and code analysis. It will feature functionalities such as charger search, booking, and payment.

1.2 Known limitations

<explain the known limitations, especially the features that were planned/expected but not implemented (and why…)

To be reviewed and completed by the end of the project >

1.3 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>

2 Product concept and requirements

2.1 Vision statement

<functional (black-box) description of the application: Which is the high-level/business problem being solved by your system? Which are the key features you promise to address it?>
<if needed, clarify what was planned/expected to be included but was changed to a different approach/concept >
<optional: how is your system different or similar to other well-known products?>
<optional: additional details on the process for the requirements gathering and selection (how did we developed the concept? Who helped us with the requirements? etc)>

2.2 Personas and scenarios

Persona: Intermittent Charger User (Zezinho,24 years)

Description: Zezinho is an IT Support Technician living in the Aveiro suburbs. Commutes daily to work (approx. 30 km round trip) in his Peugeot e-208, makes weekend trips occasionally, values time efficiency and planning. Comfortable with apps and digital services, uses Waze and MBway regularly.
Goals: Zezinho wants to find available and nearby chargers quickly, especially during commutes or when low on battery. Track charging status and history to manage costs and battery healthBook charging slots in advance to avoid waiting times..

Needs: He requires a user-friendly app interface with a clear map and filtering options, Integrated payment system, and reliable information about charging speed, and estimated waiting time. Support and contact in case of charger malfunction or issues during charging

Persona: Contínuos Charger User (Zezinho 2,24 years)

Persona: Station Operator (Chicão, 35 years)

Description: Chicão is a Charging Station Operator & Technician living in the Porto suburbs, he manages several charging points across urban and semi-urban locations, is also responsible for ensuring uptime, reporting to a municipal or private energy provider.

Goals: Chicão wants to register and configure new charging stations quickly and accurately, Monitor the operational status of all assigned stations in real-time,Get alerts or reports about failures, unusual behavior, or excessive idle time

Needs: He requires a centralized dashboard to manage all stations (status, bookings, usage), tools to register new chargers, including metadata like location, type, power output, and availability schedule. Access to Logs and reports for audits, performance reviews, and environmental tracking (e.g., CO_2 savings).

Scenarios:TODO

2.3 User Stories

Epic : User account management
User Story 1 : User registration
Priority: High
Cost: 2

As a Charger user,
I should be able to create a new account in a system,
so that I can use the system more effectively.

Acceptance Criteria:

The user must be able to access the registration page.
He should be able to input information needed for the registration.
The system must validate the information .
If some information is incorrect , the system should display an appropriate error message.

Epic : User account management
User Story 2: User Login
Priority: High
Cost: 2

As a Charger user,
I should be able to login into the system,
so that I can access essential information.

Acceptance Criteria:

The user must be able to access the login page.
He should be able to enter their register email and password.
The system must validate the information .
If some information is incorrect , the system should display an appropriate error message.

Epic : User analytics
User Story 3: See personal information

Priority: High

Cost: 5

As a Charger user,

I must be able to see my information,

so that I can manage my account.

Acceptance Criteria:

The user must be able to go to their personal page.

The system must show information about the account .

The user can alter account information .

He can see his consumptions, locations of charge, charge duration, money spent and CO2 savings and personal vehicles .

Epic : Account management

User Story 4: Add personal vehicle

Priority: High

Cost: 4

As a Charger user,

I must be able to add a vehicle to the account,

so I can charge the car.

Acceptance Criteria:

In the personal page the user must press the add car button.

He has to fill a form with information about the car.

The user should be able to delete the car.

The user should be able to update information about the car.

Epic : Station information

User Story 5: See station location

Priority: High

Cost: 4

As a Charger user,
I must be able to locate a charging station,
so I can go there and charge the car.

Acceptance Criteria:

The user must access the map in the website.
He can click on a station on the map.
It should show the distance and approximate time of travel.

Epic : Station information
User Story 6: See station Information
Priority: High
Cost: 8

As a Charger user,
I must be able to obtain information about the charging station,
so I can make a decision.

Acceptance Criteria:

The user must be able to access the station page .
It should show information about the charging station like total chargers, occupied chargers and overall statistics.
The system must show this in the current time and future time.

Epic : Booking Options
User Story 7: Book a slot in a station
Priority: High
Cost: 8

As a Charger user,
I must be able to reserve a slot in a charger,
so I can guarantee that he can charge his car.

Acceptance Criteria:

The user must be able to access the station page .

He should be able to press the add reservation.

He can choose a time from a minimum of 2 hours and a maximum of a month.

He should be able to pay.

Epic : Payment system

User Story 8: Define a method of payment

Priority: High

Cost: 4

As a Charger user,

I must be able to choose my method of payment,

so I can best suit my situation.

Acceptance Criteria:

The user must be able to access his personal page .

In the payment options he should be able to choose between pay per use or subscription options.

If he chooses pay per use, every time he uses a charging station he must pay .

If he chooses a subscription, he has various options of wattage per month with some discount .

Epic : System management

User Story 9: Station management

Priority: High

Cost: 4

As a Station Operator,

I must be able to see station statistics,

so I can make decisions on what to do.

Acceptance Criteria:

The Station Operator must be able to see all stations .

He must see information about the stations.
He must be able to disable a station or enable it.
If he disables a station, the system must send a message to every user that has a booking in it to inform them.
He can also choose to disable the station for a determined time

Epic : System management
User Story 10: Add new Station
Priority: High
Cost: 4

As a Station Operator,
I must be able to add a new station to the system,
so the clients have more options.

Acceptance Criteria:

The Station Operator must go to the add station page.
He must fill a form with station information.

2.4 Project epics and priorities

Project Epics:

User account management
User analytics
Station information
Booking Options
Payment system
System management

Iterations:

Iteration 0:

Define the product concept: Personas, main scenarios, Epics and User Stories.
Team resources setup: code repository, documents space.

Begin backlog usage.

Iteration 1:

Define system architecture.
Define the SQA tools and practices.
Create CI Pipeline.
Start working on the product specification report.
Create a prototype of the UI.

Iteration 2:

Develop the main User stories.
Define testing strategy.
Create the QA manual.
Start working on the API

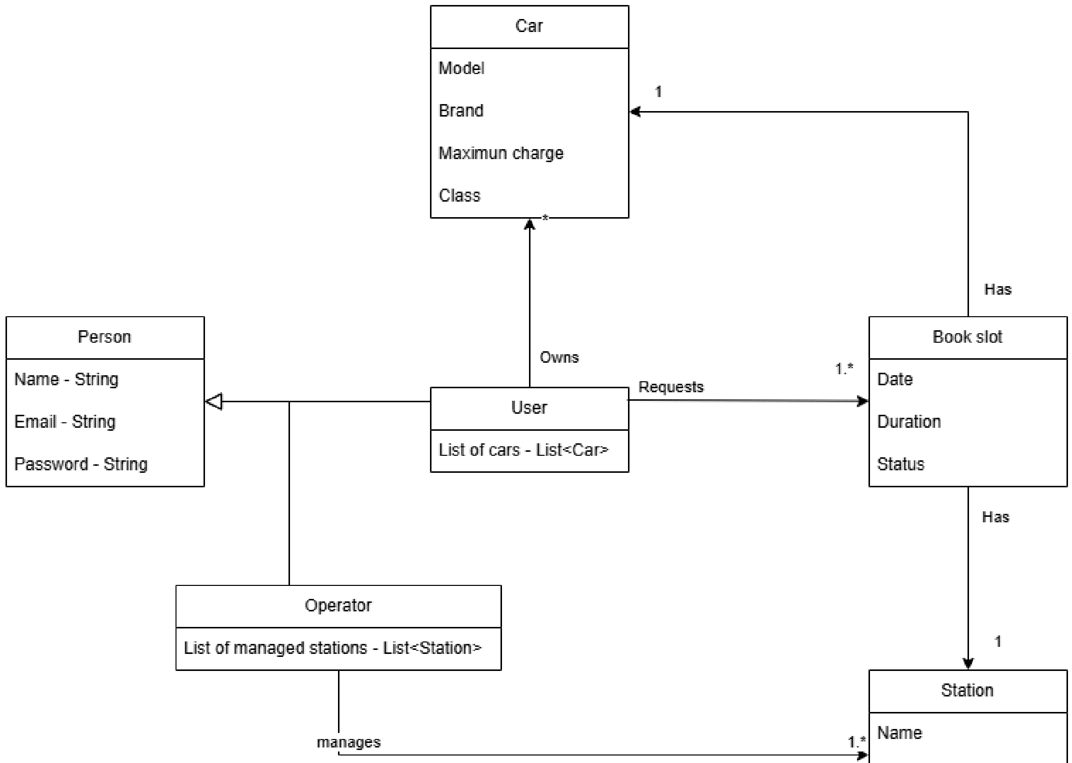
Iteration 3:

Set up the CD pipeline.
Continue the development of the user stories.
Continue various tests implementation.

Iteration 4:

Complete user stories implementation.
Complete QA manual.
Final deployment.

3 Domain model

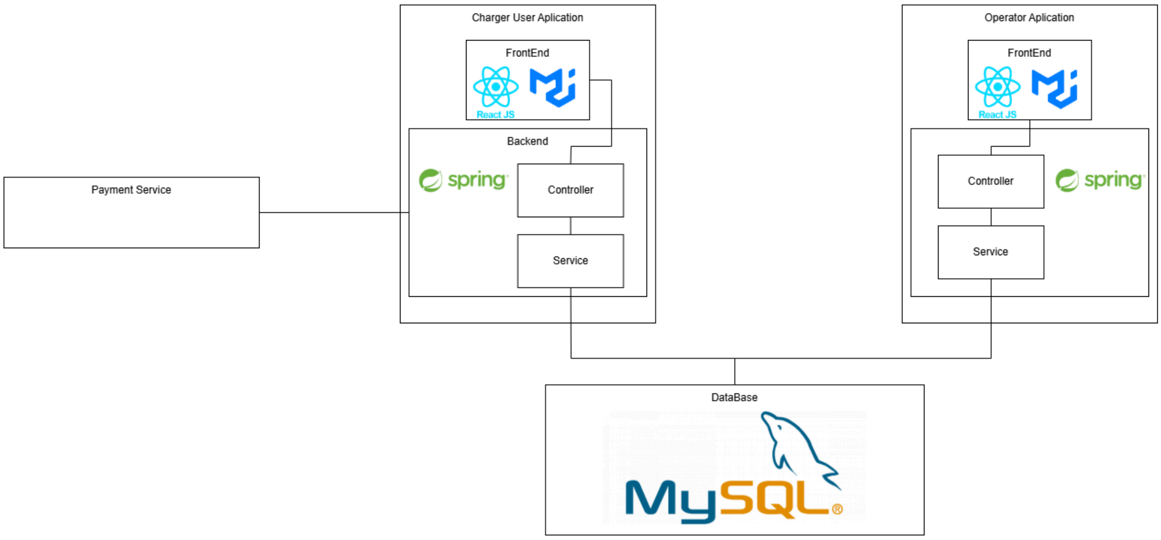


4 Architecture notebook

4.1 Key requirements and constraints

- The system must be capable of dealing with several users at once.
- The system must have an authentication system to distinguish user types.
- The different users must have a different platform.
- The system must be user-friendly.
- The system be integrated with a payment system.

4.2 Architecture view



EXPLAIN ARQUITECTURE

4.3 Deployment view

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Anotar, no diagrama, as tecnologias de implementação, e.g.: colocar o simbolo do PostgreSQL na Base de dados,…]. Indicar a existência de containers (Docker), endereços IP e portos,... Esta parte será completada quando houver efetivamente deployments

5 API for developers

[Explicar genericamente a organização da API e coleções principaus. Os detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o Swagger, Postman documentation, ou incluída no próprio desenvolvimento (e.g.: maven site)

 Be sure to use best practices for REST Api design. Keep minda REST API applies a resource-oriented design (APIs should be designed around resources, which are the key entities your application exposes, not actions)