TQS: Product specification report

Hugo Ribeiro [113402], Eduardo Lopes [103070], Rodrigo Abreu [113626], Joao Neto [113482]
v2025-05-08

1 Introduction

1.1 Overview of the project

This project was developed in the context of the Test and Quality Software (TQS) course, which emphasizes building robust, testable, and maintainable software systems. Throughout the project, we applied principles of agile development, user-centered design, and quality assurance.

Our product, ChargeHub, is a digital platform designed to improve the experience of electric vehicle (EV) drivers and facilitate efficient management for station operators. It aims to simplify the process of discovering, reserving, and managing EV charging sessions, while also providing essential administrative tools. Bellow lies a comprehensive list of our core functionalities

Station Discovery
- Search for nearby charging stations based on location.
- Filter stations by charger type, availability, and cost.

Slot Booking & Scheduling
- Reserve a charging slot in advance.
- Receive notifications and booking confirmations.

Charging Session Management
- Start/stop charging sessions using a reservation token.
- View real-time session data including time, cost, and energy used.

Payment Integration
- Review and pay for charging sessions within the app.
- Station operators can access financial reports and track payments.

User Profiles & Charging History
- Track past charging sessions and total energy consumption.

Back Office Operations for Station Operators
- Update station data such as pricing and availability.
- Report user misconduct.
- Admin features include account creation and station assignment.

ChargeHub reflects the TQS course objectives by combining well-structured software design with comprehensive feature testing and quality control measures.

1.2 Known limitations

In the **Station Discovery** module, features like the map view and real-time route tracing to charging stations were excluded from the MVP. Although visualizing stations on a map and guiding users to their destination would have significantly enhanced usability, the development effort required to integrate mapping APIs and manage location data in real time was beyond the scope of the initial release. Similarly, the ability to plan long-distance trips with intermediate charging stops was deferred due to its high complexity and limited necessity for everyday use cases.

Within **Slot Booking & Scheduling**, the system does not currently support operator-defined custom schedules or advanced notification flows. While users receive immediate booking confirmation, reminders such as those scheduled 15 minutes before a reserved slot or alerts about charger disruptions were not fully implemented. These features were deprioritized in favor of ensuring reliable booking and confirmation workflows.

In the **Charging Management** area, several enhancements were considered but ultimately left out. For instance, users are not yet able to monitor real-time metrics like battery level, cost accumulation, or energy consumption during a session. Additionally, the option to restart a recently stopped session was not included, as it involves maintaining complex session state and reconnection logic. A live dashboard for station operators to monitor charging sessions was also scoped out for future development, as it required significant backend support.

From a **sustainability and user history** standpoint, the platform does not currently track estimated CO_2 emissions saved—an optional feature designed to give users a sense of environmental impact. Similarly, administrative dashboards to analyze platform-wide trends and KPIs were planned but not pursued in this version, as the focus remained on delivering a strong user-facing experience.

Lastly, in the **account and backoffice management** domain, full administrative control over user accounts and a user-led charger feedback mechanism were not implemented. While operators can manage station data and create new accounts, broader moderation and analytics capabilities were reserved for future iterations. Additionally, the ability for users to permanently delete their accounts was not completed in time for the initial release.

In summary, the omitted features reflect a careful balancing of ambition and feasibility. While not part of the current release, they represent valuable directions for future development, ensuring that **ChargeHub** remains a scalable and feature-rich platform as it evolves beyond the scope of the TQS course.

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

Personas

EV Driver

Sofia is a thirty-nine-year-old Marketing Consultant based in Lisbon, Portugal. She recently made the switch from a diesel car to an electric vehicle. Though confident in her decision, she wants to ensure it makes long-term financial sense, especially given how often she drives for work.

She frequently travels across city and suburban areas to meet clients and attend events. Despite her marketing background, navigating EV-related logistics is new to her. She often finds herself concerned about where and when she can charge her vehicle and how charging costs compare to the fuel expenses of her previous car.

Sofia is motivated to make smarter decisions about EV ownership. She would benefit from a user-friendly app that shows available charging stations along her route, provides real-time charger availability, and offers expense comparisons. This would give her peace of mind during daily commutes and work trips.

Motivation: Sofia wants to improve her EV experience and be sure her switch from diesel is both eco-friendly and economically viable.

Station Operator

Miguel is a forty-seven-year-old station manager who oversees two Galp service stations in Aveiro, Portugal. He is married and has worked in the fueling industry for more than two decades. Recently, Miguel installed electric chargers at both locations to modernize his business and meet the rising demand for EV infrastructure.

Despite this upgrade, Miguel struggles with making the most of the chargers. He needs real-time insights into charger usage, availability, and performance. His goal is to keep the stations visible and attractive to EV drivers and avoid chargers being offline or underused.

Miguel is looking for a smart management system that lets him track usage patterns, schedule maintenance, and update station status instantly. He also wants a tool to advertise his chargers' availability to drivers nearby.

Motivation: Miguel wants to maximize charger usage, attract more EV drivers, and run his stations efficiently.

Admin

José is a twenty-seven-year-old System Administrator working in Aveiro, Portugal. He is married and plays a key role in managing the user accounts and access permissions for an EV charging platform.

José's work revolves around ensuring that the right people have access to the right tools. He is responsible for creating and managing accounts for station operators, assigning stations to them as needed, and updating platform access to reflect organizational changes. He also oversees EV driver accounts and ensures that user data stays clean, current, and compliant with platform rules.

Although he doesn't deal directly with technical maintenance or infrastructure, José's work is essential for keeping operations organized and secure. He values clarity and simplicity in admin tools, and needs efficient interfaces to carry out tasks quickly and accurately.

Motivation: José wants to manage users and station assignments effectively and ensure the platform reflects the latest organizational setup at all times.

Scenarios

Sofia searches for a charger nearby

- Sofia notices her EV battery is getting critically low while driving to an important client meeting. She opens the app and is greeted by a clean interface showing nearby charging points. The app detects her current location and displays available fast chargers within reach. She quickly filters by availability and connector type, selects the nearest compatible charger, and the app provides real-time navigation to the station.

Sofia checks for chargers along her planned route

- Sofia is planning a work trip from Lisbon to Porto and wants to ensure charging won't disrupt her schedule. She opens the trip planner in the app and enters her destination. The system calculates the best route and overlays charging stations along the way. She taps on a few stations to compare details and estimated costs, helping her plan a smooth and stress-free trip.

Sofia reserves a charger in advance

- Knowing she'll arrive in a busy downtown area during peak hours, Sofia wants to avoid the risk of waiting in line. She searches the app for chargers in the area and sees that only a few are available. She selects a station, reviews the terms, and reserves a slot. Upon arrival, she checks in via the app and begins charging immediately.

José suspends an EV driver account due to policy violation

- José receives an automated alert about repeated misuse of charging sessions by an EV driver, such as overstaying after sessions end. He reviews the user's charging history and confirms multiple violations. Through the admin panel, he temporarily suspends the driver's account.

José adds a new station operator and assigns them their stations

- A new charging station operator joins the platform. José logs into the admin system and creates the operator's account. He selects the appropriate role and access permissions, then assigns the relevant stations. After final review, he activates the profile. The new operator now has access to manage pricing, monitor charger status, and view usage analytics for their assigned stations.

Miguel configures pricing and details for his assigned charging stations

- Miguel has recently been assigned a new set of EV chargers at his Galp stations. He logs into the management portal for the first time to complete the initial setup. He selects each station and inputs key details, such as the type of connectors available, maximum charging power and pricing. Once the configuration is complete, the stations become visible to EV drivers through the platform's app.

2.3 Project epics and priorities

EPIC 1: Station Discovery

User Stories:

- **As an EV Driver**, I want to search for charging stations near my location so that I can access stations within a short distance from me.
    - **Priority/Estimate:** High, 2
    - **Acceptance Criteria:**
        - Given I have opened the charging station app,
        - When I allow location access
        - Then I see a list of charging stations ordered by proximity to my current location

- **As an EV Driver**, I want to filter stations by charger type, availability, and cost so that I can find stations that match my technical needs and budget.
    - **Priority/Estimate:** Medium, 2
    - **Acceptance Criteria:**
        - Given I am viewing the list of charging stations,
        - When I apply filters for charger type, availability or cost,
        - Then only the charging stations that meet the selected criteria are displayed

- **As an EV Driver**, I want to see charging stations on a map so I can easily locate them.
    - **Priority/Estimate:** Low, 3
    - **Acceptance Criteria:**
        - Given I have granted location access,
        - When I open the map view,
        - Then I see markers for all nearby charging stations

- **As an EV Driver**, I want to trace a route to a nearby charging station so that I can visualize the path to it.
    - **Priority/Estimate:** Optional, 4
    - **Acceptance Criteria:**
        - Given I select a charging station,
        - When I select “Get Directions”,
        - Then a route from my current location to the station is shown on the map

- **As an EV Driver**, I want to see the closest charging stations when planning a trip, so that I can see where I should stop to charge my vehicle.
    - **Priority/Estimate:** Optional, 5
    - **Acceptance Criteria:**
        - Given I input my trip’s start and end points,
        - When I begin trip planning,
        - Then I see a list of charging stations closest to the path between those two points,
        - And each station includes the distance from the departure and estimated time deviation from the route.

EPIC 2: Slot Booking & Scheduling

User Stories:

- **As an EV Driver**, I want to book a charging slot in advance so that I can ensure availability.
    - **Priority/Estimate:** High, 3
    - **Acceptance Criteria:**
        - Given I have selected a charging station and charger type,
        - When I choose a time slot and confirm the booking,
        - Then the system reserves that slot for me and prevents other users from booking it.
        - And I receive confirmation with the booking details, including the charger type, time and a token.

- **As a Station Operator**, I want to set custom schedules for my charging stations, so that I can control when each station is available for use.
    - **Priority/Estimate:** Optional, 4
    - **Acceptance Criteria:**
        - **Given** I am logged in as a station operator,
        - **When** I open the schedule management interface for a station,
        - **Then** I can define available and unavailable time slots for each charger.

- **As an EV Driver**, I want to receive notifications 15 minutes before my booked time starts and when it starts, so that I am reminded to arrive at the charging station on time.
    - **Priority/Estimate:** Optional, 2
    - **Acceptance Criteria:**
        - Given I have an active charging slot booking,
        - When my scheduled time is approaching,
        - Then I receive a notification 15 minutes before the booking starts.
        - And I receive another notification at the exact start time of the booking.

- **As an EV Driver**, I want to receive a notification if a charger I booked becomes disabled before my scheduled time, so that I can make alternative arrangements in advance.
    - **Priority/Estimate:** Low, 2
    - **Acceptance Criteria:**
        - Given I have an active charging slot booking,
        - When the assigned charger becomes unavailable before my scheduled time,
        - Then I receive an immediate notification informing me of the issue,
        - And I am prompted to either rebook at another station or cancel the booking.

EPIC 3: Charging Management

User Stories:

- **As an EV Driver**, I want to use my reservation token in the app to unlock the charger when I arrive, so that I can start my charging session securely and only during my reserved slot.
    - **Priority/Estimate:** High, 3
    - **Acceptance Criteria:**
        - Given I have an active reservation and have arrived at the charging station,
        - When I open the app and input my reservation token,
        - Then the corresponding charger is remotely unlocked,

        - And my charging session begins.

- **As an EV Driver**,I want to see real-time charging status including battery percentage, current cost, and energy consumed, so that I can monitor and manage my charging session according to my needs and budget.
    - **Priority/Estimate:** Optional, 4
    - **Acceptance Criteria:**
        - Given I have an active charging session,
        - When I open the session details in the app,
        - Then I see live updates of the battery level, cost so far, and energy (kWh) delivered.

- **As a Station Operator**, I want to monitor all active and historical charging sessions, so that I can ensure operational visibility, detect issues, and analyze usage patterns
    - **Priority/Estimate:** Optional, 4
    - **Acceptance Criteria:**
        - Given I am logged as station operator,
        - When I open the charging session panel,
        - Then I see a list of all current sessions with user info, charger status, duration, and energy consumed.
        - And I can also access a history of past sessions with filtering options.

- **As an EV Driver**, I want to stop the charging session from the app at any time, so that I can charge only the amount I need and control the cost.
    - **Priority/Estimate:** High, 2
    - **Acceptance Criteria:**
        - Given I am in an active charging session,
        - When I select “Stop Charging” in the app,
        - Then the charger stops delivering power and the session ends,
        - And I see a summary of total cost, time, and energy charged.

- **As an EV Driver**, I want to restart the charging session from the app if I stop it accidently, so that I can continue charging without needing to make a new reservation.
    - **Priority/Estimate:** Optional, 2
    - **Acceptance Criteria:**
        - Given I have recently stopped a charging session,
        - When I select “Restart Charging” in the app,
        - Then the charger resumes delivering power using the same reservation

---

EPIC 4: Payment Integration

User Stories:

- **As an EV Driver**, I want to receive the payment summary in the app when my charging session ends, so that I can easily review and pay for the charging cost directly through the app.
    - **Priority/Estimate:** High, 4
    - **Acceptance Criteria:**
        - Given my charging session has ended,
        - When I open the app,
        - Then I see a payment summary with the total amount due, energy consumed, and charging duration,
        - And I can choose a payment method and complete the transaction within the app.

- **As a Station Operator**, I want to access financial reports per charging session and verify their payment status, so that I can monitor revenue and ensure that all sessions have been correctly billed and paid.
    - **Priority/Estimate:** Medium, 4
    - **Acceptance Criteria:**
        - Given I am logged as station operator,
        - When I open the financial reporting section,
        - Then I can view individual charging sessions along with payment status, amounts charged, and payment timestamps

---

EPIC 5: User Profiles & Charging History

User Stories:

- **As an EV Driver**, I want to view my charging history and total energy consumption over time, so that I can track my usage patterns and manage my charging behavior more efficiently.
    - **Priority/Estimate:** Medium, 4
    - **Acceptance Criteria:**
        - Given I am logged into my account,
        - When I open the "Charging History" section,
        - Then I see a list of past charging sessions with dates, locations, energy consumed (kWh), duration, and cost.

- **As an EV Driver**,I want to track my estimated CO_2 savings based on my charging activity, so that I can understand the environmental impact of using electric vehicles instead of fuel-based alternatives.
    - **Priority/Estimate:** Low, 4
    - **Acceptance Criteria:**
        - **Given** I have completed one or more charging sessions,
        - **When** I visit the sustainability section in my profile,
        - **Then** I see an estimate of CO_2 emissions avoided (in kg or tons) based on energy charged and average EV efficiency.

- **As an Admin**, I want to view aggregated user statistics and charging behavior trends, so that I can analyze platform usage and optimize the service for users.
    - **Priority/Estimate:** Optional, 5
    - **Acceptance Criteria:**
        - Given I am logged into the admin dashboard,
        - When I open the analytics or reporting section,
        - Then I see data such as total users, active users, average energy per session, peak usage times, and CO_2 saved across the platform.

EPIC 6: Backoffice Operations

User Stories:

- **As a Station Operator**, I want to update my station's information such as availability and pricing, so that users always see accurate data when booking a charging slot.
    - **Priority/Estimate:** Medium, 3
    - **Acceptance Criteria:**
        - Given I'm logged into the operator dashboard,
        - When I access a station's details and make changes to availability, pricing, or charger info,
        - Then the updated information is saved and reflected in the user-facing app immediately.

- **As an Admin,** I want to create station operator accounts, so that new employees can manage charging stations assigned to them.
    - **Priority/Estimate:** High, 3
    - **Acceptance Criteria:**
        - Given I am logged into the admin dashboard,
        - When I fill in the required details to create a new operator account,
        - Then the operator receives credentials and can log into the platform with their permissions.

- **As an Admin,** I want to assign or remove stations from specific station operators, so that I can manage responsibilities across the network effectively.
    - **Priority/Estimate:** High, 3
    - **Acceptance Criteria:**
        - Given I am on an operator's profile in the admin dashboard,
        - When I select stations to assign or remove,
        - Then the operator's access is updated accordingly.

- **As an Admin**, I want to manage all user accounts, including EV drivers and station operators, so that I can enforce platform rules and support operations effectively.
    - **Priority/Estimate:** Low, 4
    - **Acceptance Criteria:**

        - Given I am in the admin dashboard,
        - When I view the list of users (EV Drivers and Operators),
        - Then I can search, edit, deactivate, or delete any account as needed.

- **As an Admin**, I want to view system-wide reports and statistics, so that I can monitor platform performance and usage trends.
    - **Priority/Estimate:** Optional, 4
    - **Acceptance Criteria:**
        - Given I am logged into the admin dashboard,
        - When I access the analytics section,
        - Then I see KPIs like total sessions, revenue, energy delivered, user growth, etc.

- **As a Station Operator**, I want to report users for misconduct such as not paying or damaging equipment, so that the platform can take appropriate action.
    - **Priority/Estimate:** Medium, 3
    - **Acceptance Criteria:**
        - Given I'm on the operator dashboard and have identified a problematic session,
        - When I click "Report User" and provide a reason,
        - Then the report is submitted to the admin team for review.

- **As an EV Driver**, I want to report chargers I use, so that other drivers can benefit from my feedback.
    - **Priority/Estimate:** Optional, 2
    - **Acceptance Criteria:**
        - Given I've completed a charging session,
        - When I visit the charger's detail page,
        - Then I can submit a report (comment)

EPIC 7: Authentication & Account Management

User Stories:

- **As an EV Driver,** I want to update my account information, so that my profile stays accurate and secure.
    - **Priority/Estimate:** Medium, 3
    - **Acceptance Criteria:**
        - Given I am logged into my account,
        - When I navigate to the profile settings,
        - Then I can edit and save my personal details.

- **As an EV Driver,** I want to permanently delete my account, so that I can stop using the service and remove my personal data from the platform.
    - **Priority/Estimate:** Optional, 2
    - **Acceptance Criteria:**
        - Given I am logged into my account,
        - When I choose to delete my account and confirm the action,

- Then my account and associated data are removed and I am logged out.

3 Domain model

For our domain model, we tried to keep it as simple as possible without compromising any functionalities or requirements of our product. There are two main entities: **Client** (normal users of our application) and **Staff**. Regarding the Staff, there are two types of roles: **Staff.OPERATOR** (responsible for a station) and **Staff.ADMIN** (responsible for adding the new employees to the database with the corresponding roles, and adding new stations).
Which **Charger** is associated with a **Station**, and as a set of attributes necessary to assure a good use of them.
It is possible to book slots (**Booking**) and start a **ChargingSession** using the associated token to unlock the charger. The **Payment** entity helps the station operator validate if the client actually paid for the charging session, and allows the admin to create statistics relative to the usage of the chargers and the preferred paying methods.
The User can add a **Vehicle** to their account to facilitate the charger's choosing process (based on voltage, connector type, and charging type). There is also a **Notification** entity used to remind/notify the user when their charging session is about to start.
The domain model is presented in the image below:

4 Architecture notebook

4.1 Key requirements and constraints

Several critical concerns related to interoperability, responsiveness, platform diversity, and real-time operations shape this system's design. The following architectural characteristics were identified as essential to guide the software architecture and ensure it supports users' needs, operational requirements, and future extensibility.

The system supports multiple user-interfacing platforms, including:

- A web-based frontend built with ReactJS and LeafletJS for interactive map rendering.
- Potential integration with mobile platforms or lightweight embedded displays in the future.

To support this, the architecture is logically separated into a frontend and backend through standard HTTP REST APIs, ensuring loose coupling between presentation and business logic. This allows UI interfaces to evolve independently of backend logic, while reusing the same API services across platforms.

Some system components, such as booking confirmation or charging status, require real-time updates. This is achieved through backend services that track and manage data such as notifications and session progress. The Notification Module, composed of a controller, service, and repository, is responsible for delivering system messages and alerts and supporting real-time feedback mechanisms.

The system interfaces with physical infrastructure, particularly charging stations and possibly booking kiosks. These are abstracted through backend services like the Charge Service and Station Service, which isolate hardware-specific interactions from the rest of the business logic. This approach allows the software to remain hardware-agnostic and easily adaptable to changes in device models or communication protocols.

Given the involvement of multiple user roles (clients, staff, and admin), the architecture incorporates strict authentication and authorization mechanisms. These controls ensure that users can only access actions and data appropriate to their role. For example:

- Staff can update station statuses and charging sessions.
- Clients can book stations and view their vehicle data.

Security is embedded at both the API endpoint level and within business logic, minimizing the risk of unauthorized access or misuse.

Operating in domains such as mobility infrastructure or healthcare, the system must perform reliably even under heavy usage. The modular backend, coupled with efficient database interactions via Spring Data JPA and a reactive frontend, is designed to scale both vertically and horizontally as needed. Critical paths such as booking, session tracking, and notifications are optimized for low latency and high throughput, ensuring responsiveness even during peak load conditions.

To ensure that the system can be deployed consistently across different environments, the entire application (backend, frontend, database) is containerized using Docker. Docker containers encapsulate the environment, dependencies, and runtime configuration, allowing the system to run identically on local developer machines, staging servers, or production infrastructure. This significantly reduces the risk of deployment issues and simplifies environment management.

This architecture provides a solid foundation for delivering a robust, scalable, and extensible system. It balances user-facing interactivity, backend performance, security, and flexibility for future growth and platform expansion.

4.2 Architecture view

The system uses a client-server architecture, composed of a modular backend, a rich client-side frontend, and integrations with external services. A frontend interface communicates with backend services over HTTP, while the backend handles all business logic and data access responsibilities. An external routing and map provider supports geolocation and mapping needs. A payment gateway handles transactions securely. All components are logically decoupled and follow a clear separation of concerns.

The backend consists of several logical modules, each responsible for a specific functional domain. All modules follow a controller-service-repository pattern and communicate through internal calls. The main modules are:

- **Booking Module** - The Booking Module is responsible for managing all operations related to user bookings of chargers on a specific station. This module ensures that users can securely and efficiently reserve services within the system.
- **Staff Module** - The Staff Module handles operations related to system staff or administrative users. This module ensures that administrative functions are isolated and governed by strict access rules.
- **Client Module** - The Client Module manages all functionality related to application users (i.e., clients who interact with the system to book and charge vehicles). This module plays a central role in enabling end-users to access and utilize the system effectively.
- **Station Module** - The Station Module governs the behavior and status of charging stations. This module integrates closely with the map services to support location-based queries and decisions.
- **Notification Module** - The Notification Module is dedicated to sending real-time alerts, status updates, or confirmations to clients and staff. This module is essential for maintaining user awareness and timely communication within the system.
- **Charging Module** - The Charging Module handles the full lifecycle of charging sessions. This module integrates tightly with both the station and client modules to ensure secure, efficient, and trackable energy usage.

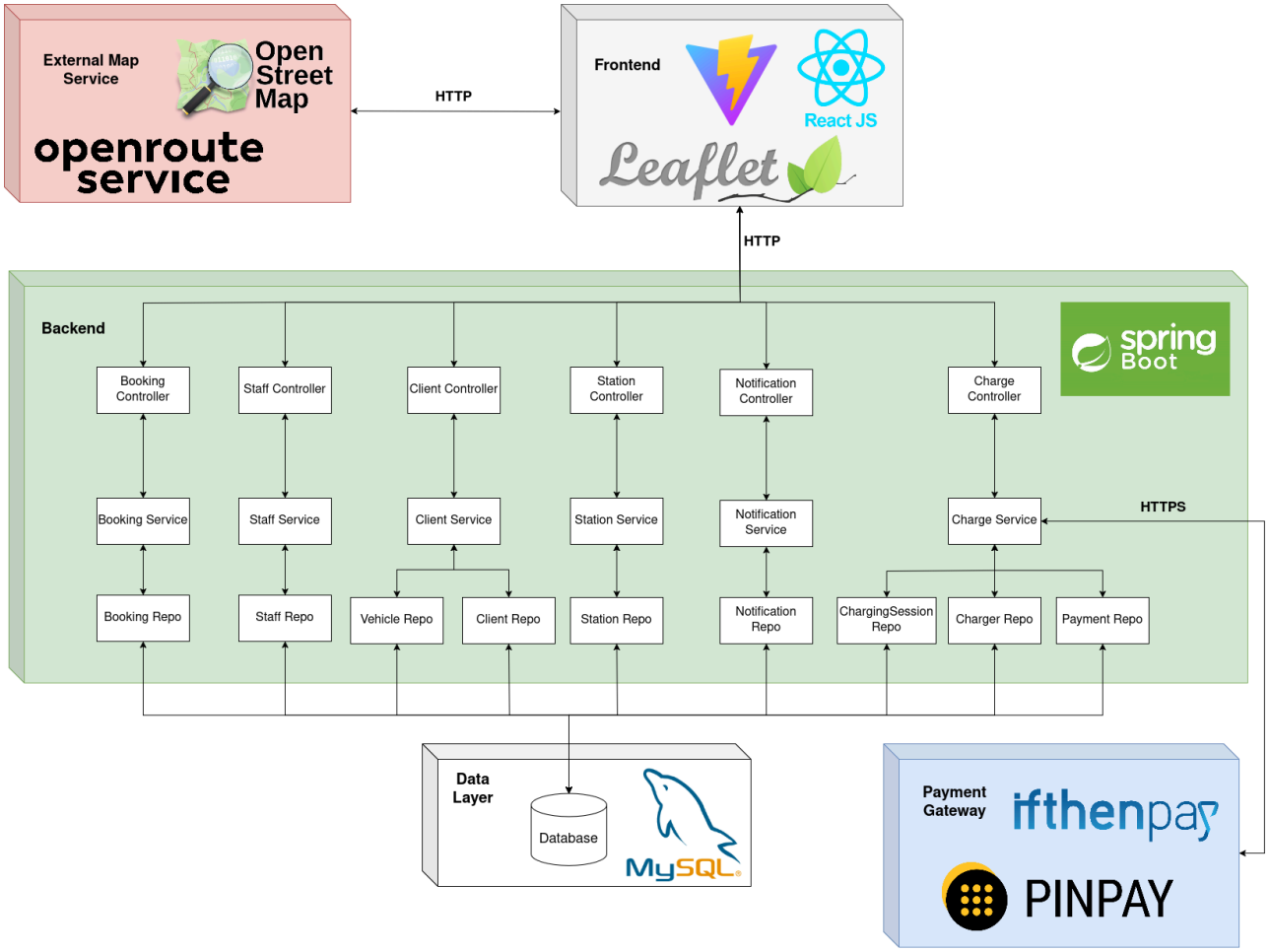
All backend modules interact with a central relational database via repository interfaces. The data model includes clients, vehicles, bookings, station information, staff, notifications, and charge sessions. This centralized data layer ensures consistency and referential integrity across components (all data is persisted in this single instance).

The frontend is a web-based interface that supports different types of users. The interface integrates with **Leaflet** for interactive maps, fetching real-time geospatial data from an external map provider.

Regarding external services, our app uses two:

- **Map Service**: Integrates with OpenRouteService and OpenStreetMap to offer real-time route data, distance calculations, and map tiles (HTTP APIs fetch map data and routes from OpenRouteService).

- **Payment Gateway**: Supports online payments via secure HTTPS communication with PINPAY gateways from IfThenPay (HTTPS APIs send secure payment transactions to the payment gateway).



4.3 Deployment view

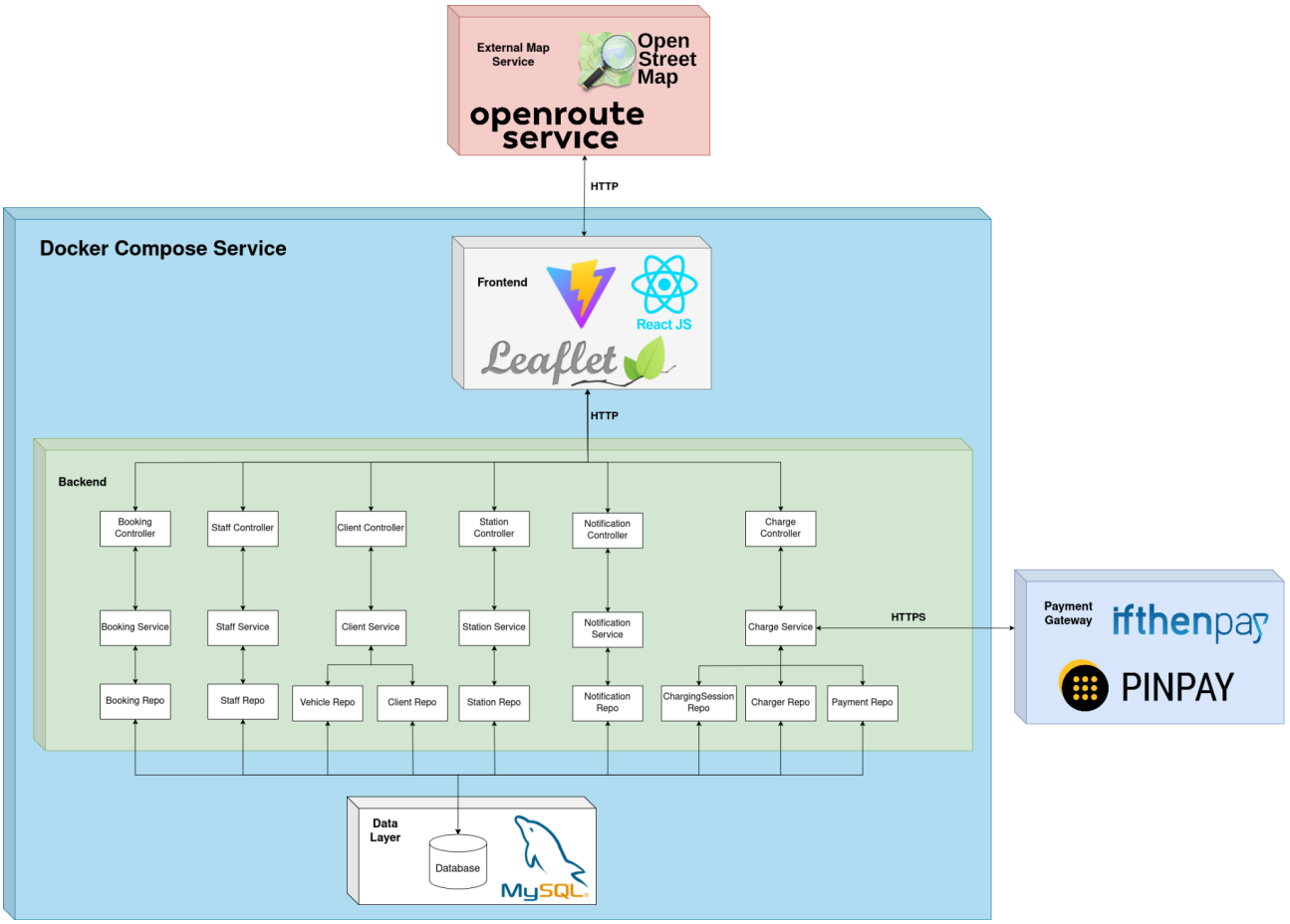
To simplify deployment and ensure consistency across development and production environments, the system is fully containerized using Docker. Each functional component of the system is deployed in its own isolated container, coordinated via Docker Compose, allowing for scalable, reproducible, and environment-agnostic deployments.

The system is divided into multiple Docker containers, each responsible for a specific subsystem:

- **Backend Container** - Runs the Spring Boot backend application, exposing its RESTful API over HTTPS. It handles business logic, database interaction, and external system integration.
- **Frontend Container** - Hosts the React.js web frontend (built using Vite), which provides the user interface for clients and staff to interact with the system.
- **Database Container** - A MySQL container (as shown in the diagram) acts as the persistent data layer for all application modules. It is accessed internally by the backend via a secure network.

- **Map Services** - While not deployed internally, the frontend makes HTTP calls to OpenStreetMap and OpenRouteService, which are considered external dependencies and accessed over the public internet.
- **Payment Gateway** - The system integrates with external payment providers, namely PINPAY, which are not deployed internally but are securely accessed by the backend over HTTPS. These gateways handle payment authorization, transaction processing, and status feedback. The Charge Service within the backend communicates with these gateways during charging sessions to initiate and validate payments, ensuring a seamless and secure payment experience for end users.

This containerized structure ensures the system can be easily deployed to any infrastructure, whether cloud-based, on-premise, or hybrid. The separation of roles, clean API boundaries, and containerized deployments also prepare the system for CI/CD automation, load balancing, and high availability configurations in the future.



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)