TQS: Product specification report

David Amorim [112610], Francisca Silva [112841], Guilherme Amaral [113207], Henrique Freitas [114990]
v2025-06-09

1 Introduction

1.1 Overview of the project

Our project, developed in the scope of TQS, aims to develop a full stack web application while applying good testing and quality assurance guidelines in a CI/CD environment. Our application, named Multi Stations for Electric Vehicles (MSEV), aims to integrate several charging stations networks into one, to make it easier for drivers to find, book and charge their vehicles.

1.2 Known limitations

Although all minimal required features were implemented, our system could use a more enriched and detailed monitorization view in both charging sessions and stations by implementing more followable stats such as estimated time remaining for full charge, peak usage hours and most profitable stations. We could also implement web sockets instead of api calls for faster feedback of monitorization stats.

1.3 References and resources

- Stripe (payment sandbox integration): https://docs.stripe.com/sandboxes

2 Product concept and requirements

2.1 Vision statement

The application we proposed to develop, Multi Stations for Electric Vehicles, comes as a solution for improving the electric vehicle ecosystem through a platform that allows drivers to search, book and pay charging services across several integrated charging networks. At the same time, charging stations owners will be able to register their stations and manage their availability, prices and maintenance.

2.2 Personas and scenarios

Personas:

John Boyle

Age: 43 years old

Role: Insurance Inspector

City: New York

Background: John is a wealthy insurance inspector that is always on the move from city to city due to his job. He is single and barely has time to rest at home, since most inspections are scheduled without warning. He recently bought an electric vehicle in hopes of leading a more sustainable lifestyle.

Problem: Due to the low availability of electric vehicles charging stations, John has been having difficulties finding close stations to charge while on his trips.

Needs: A platform that helps him locate available charging stations close to the cities he visits and book slots in advance if necessary.

Scenario: John has just received a last-minute assignment in Boston and starts the drive from Philadelphia with a limited battery. Knowing he won't have time to scout for a charger on arrival, John opens the MSEV app. Inserting the desired location in Boston, he quickly receives all compatible EV charging stations available along the way, including their estimated availability at arrival time. He then books a slot at a station just 10 minutes from the inspection site. With peace of mind, John continues his journey, confident that his vehicle will be charged and ready when needed. The app also stores his booking in case plans change, allowing him to cancel or reschedule without penalty.

Ron Swanson

Age: 35 years old

Role: Charging station Operator

City: Ohio

Background: Ron is a married men with one son, he recently invested in the electric car business by buying and setting up charging stations across his city and now works managing all his stations and expanding his business. Since he loves spending time with his family, he values not having to leave home for extended periods a lot.

Problem: Since his investment bloomed due to an increased demand for electric vehicles, he started having trouble managing all stations inside his network, having to personally visit each station weekly to check if everything is ok.

Needs: An app that helps him keep track of each station, allowing him to disable or even register new stations.

Scenario: Ron just finished dinner with his family when he received a notification that one of his charging stations is reporting inconsistent power levels. Instead of leaving home, Ron can open the MSEV Manager app. From the dashboard, he selects the affected station, checks its status logs, and notices a voltage fluctuation issue. With a single click, he remotely disables the station for safety, allowing him to go check the problem the next day.

2.3 Project epics and priorities

2.3.1 Epic 1: Station Discovery & Navigation

Enable EV drivers like John Boyle to discover charging stations easily and view their availability.

User Stories:

1.As an EV driver, I want to search for available charging stations near a location so that I can plan my trips.

Acceptance Criteria:

- The User should be able to search for charging stations by introducing an address
- The User should be able to search for charging stations by dropping a pin on the map
- Charging stations should be displayed on the map as pins
- When a User conducts a search, the result should be a list ordered by proximity of stations
- Only stations in a 5km radius to where the search is conducted should be loaded
- After searching for closest stations, it should not take longer than 2 seconds to load all stations
- Each station list should show its name, address and distance to the search location

Tests:

1) **Given** that I am an authenticated user on the MSEV app
2) **When** I enter an address or drop a pin on the map
3) **Then** the system should display a list of charging stations within a radius
4) **And** the list should include basic station information such as name, address, and distance from the search location

2.As an EV driver, I want to see the real-time availability status of chargers so that I avoid wasting time.

Acceptance Criteria:

- Upon search, the result should also list all chargers available for each station
- Each charger's availability should be shown together with its other details
- The availability should be "Available", "In Use", "Booked" or "Out of Order"
- Users should be able to see what time each charger is booked for the next five days.

Tests:

1) **Given** that I have performed a station search or selected a station on the map
2) **When** the station details are displayed
3) **Then** I should see a real-time indicator of each charger's status such as "Available", "In Use", "Booked" or "Out of Order"
4) **And** the status should list if the charger is booked in the near future.

3.As an EV driver, I want to view charger types (e.g., fast/slow) and prices so I can choose the most suitable option.

Acceptance Criteria:

- Each charger details should show their charger types

- Along with the charger types, their prices should appear

Tests:

1) **Given** that I have made a station search
2) **When** I open the station details view
3) **Then** I should see a list of its chargers
4) **And** each charger should display its type
5) **And** each charger should display its pricing

2.3.2 Epic 2: Station Reservation & Trip Planning

Allow drivers to reserve charging slots and see their consumption.

User Stories:

4.As an EV driver, I want to reserve a charging slot in advance, so I don't have to wait in line.

Acceptance Criteria:

- When seeing each charger's availability for the next days, a User should be able to book free slots
- When a charger is booked, it should appear booked for other users
- A booking should appear on the User bookings page
- A user should be able to cancel a reservation by accessing his bookings page
- To access a booked charger, the User should open the booking page containing the desired reservation during the scheduled time and press "Start Charging"

Tests:

1) **Given** that I am viewing a charging station with available booking slots
2) **When** I select a date and time and confirm the reservation
3) **Then** the system should lock the selected time slot for my user account
4) **And** my booking should appear in my bookings page

5. As an EV driver, I want to be able to monitor my energy consumption while my vehicle is charging.

Acceptance Criteria:

- While a user's vehicle is charging, the User should be able to watch its energy consumption on the charging page
- While a user's vehicle is charging, the User should be able to watch the estimated time remaining on the charging page

- While a user's vehicle is charging, the User should have the station details in its charging page

Tests:

1) **Given** that my vehicle is actively charging at a supported MSEV station
2) **When** I open the charging session screen in the app
3) **Then** I should see real-time data such as current kWh consumption and charging speed

2.3.3 Epic 3: Operator Dashboard & Remote Management

Allow station operators like Ron Swanson to manage their charging network remotely.

User Stories:

6. As a station operator, I want to monitor the status of all stations in real-time so I can detect issues quickly.

Acceptance Criteria:

- There should be an Operator Dashboard page for station operators
- The Operator Dashboard page should list all stations and their details, such a list of chargers
- Stations with problems or "Out of Order" should be highlighted
- Chargers with problems should also be highlighted

Tests:

1) **Given** that I am logged into the MSEV operator dashboard
2) **When** I access the station monitoring section
3) **Then** I should see a list of all my stations with status indicators such as "Enabled", or "Disabled"
4) **And** any critical alerts should be clearly highlighted

7. As a station operator, I want to disable a malfunctioning station remotely so that users are not impacted.

Acceptance Criteria:

- For each displayed station and associated chargers in the Operator Dashboard, a "Disable"/" Activate" button should be available
- All "Disabled" chargers should appear as "Temporarily Disabled" for Users and Station Operators
- "Temporarily Disabled" chargers should not be allowed to charge vehicles
- If a Station is disabled, all its chargers should appear "Out of Order"

Tests:

1) **Given** that I am viewing the details of a malfunctioning station
2) **When** I select the "Disable Station" option and confirm the action
3) **Then** the system should mark the station as "Disabled"
4) **And** prevent new charging sessions to any charger in it

8. As a station operator, I want to register new stations via the app so I can expand my network easily.

Acceptance Criteria:

- In the operator dashboard, an “Add station” button should be available
- When adding a station, an operator should add address and name, along with any number of chargers with their corresponding types and prices.
- After finishing adding a station, a QR code to connect the station to the app should appear.
- Added stations should appear to both users and station operators

Tests:

1) **Given** that I am logged into the MSEV operator dashboard
2) **When** I access the “Add Station” form and enter required details (location, charger types, pricing, capacity)
3) **Then** the system should validate the input
4) **And** the new station should be added to the network and appear in the monitoring dashboard

2.3.4 Epic 4: Payments and Monitoring

Keep operators informed through resourceful monitorization and allow users to pay their charging.

User Stories:

9. As an EV driver, I want to be able to pay my charging fees directly through my app.

Acceptance Criteria:

- While charging at a station, the charging page of a User should have a “Finish charging” button
- Once a User finishes charging, a payment page should open where the user inserts its payment information
- After the payment concludes, the charger should be liberated and become available if no booked slot is next.

Tests:

1) **Given** that I have my charging panel open.
2) **When** I choose to end the charging and pay.
3) **Then** the system should inquire me about my payment method
4) **And** my car should be released once I finish the process

10. As an operator, I want to be able to check each station's performance and consumption so I can adjust their prices accordingly.

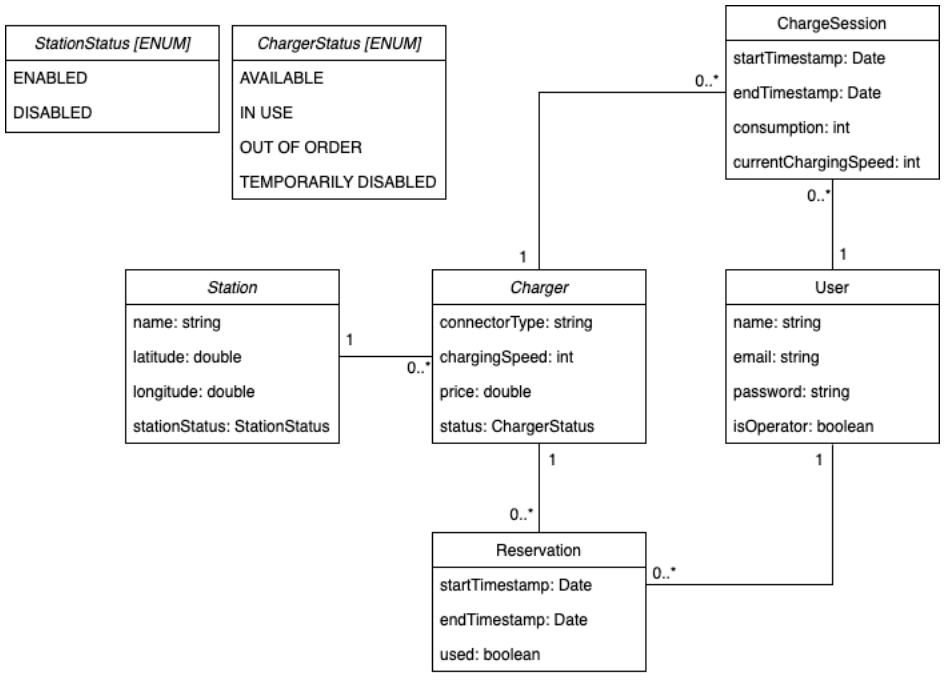
Acceptance Criteria:

- Operators should be able to access each station's historical performance and energy consumption
- On the performance page of each station a button to update each charger type price should be available to allow changing prices based on efficiency and inflation
- Peak Usage hours and total sessions at each station/charger should be available

Tests:

1) **Given** that I am logged into the operator analytics panel
2) **When** I select a specific station and click "Monitoring"
3) **Then** I should see aggregated data such as total sessions, average session time, total kWh delivered, and peak usage hours

3 Domain model



4 Architecture notebook

4.1 Key requirements and constraints

Although there can be a more complex way to approach the problem at hands since there is a real-time aspect to it, for simplicity and time constraints we opted for a more centralized, layered based architecture.
The architecture could very well expanded to a micro-services one, since there is a possibility of monitoring different stations at different locations with different availability profiles, but in our case, we choose to eliminate that complexity through the introduction of a charger and station Api's.

About the implementation of other platforms, we believe that by providing a more general view of the functionalities through a website, both for the client and station operator, we can easily expand the frontend, for future work, to a PWA providing a similar experience, primarily, to client in which it would be able to make reservations anywhere.

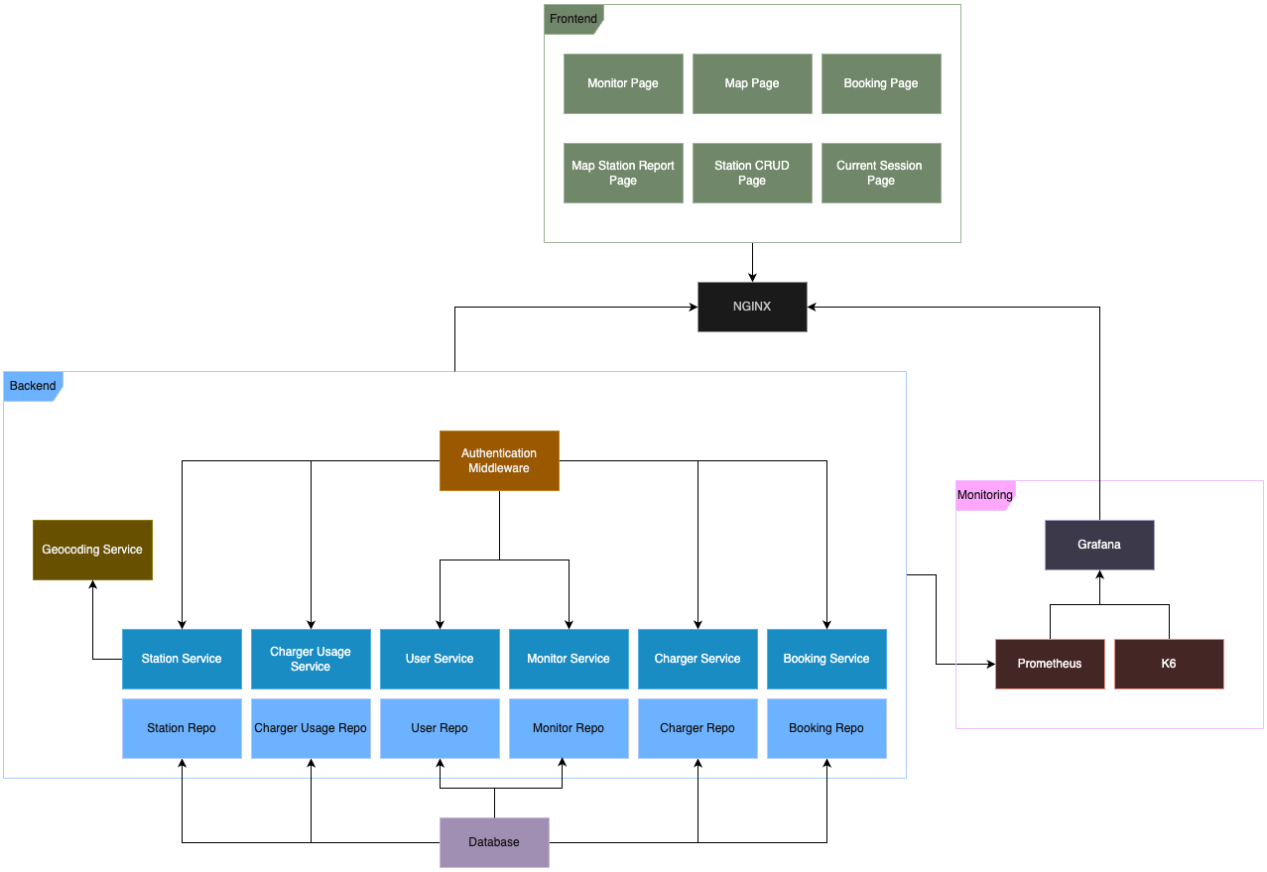
Requirements:

- A station operator must be able to keep the values of the reservation and user's intact and immutable, for monitoring.
- Booking a station must be atomic, to prevent scheduling time conflicts.
- A user can't access the operator's actions; Different routes must be protected.

Non-Functional Requirements:

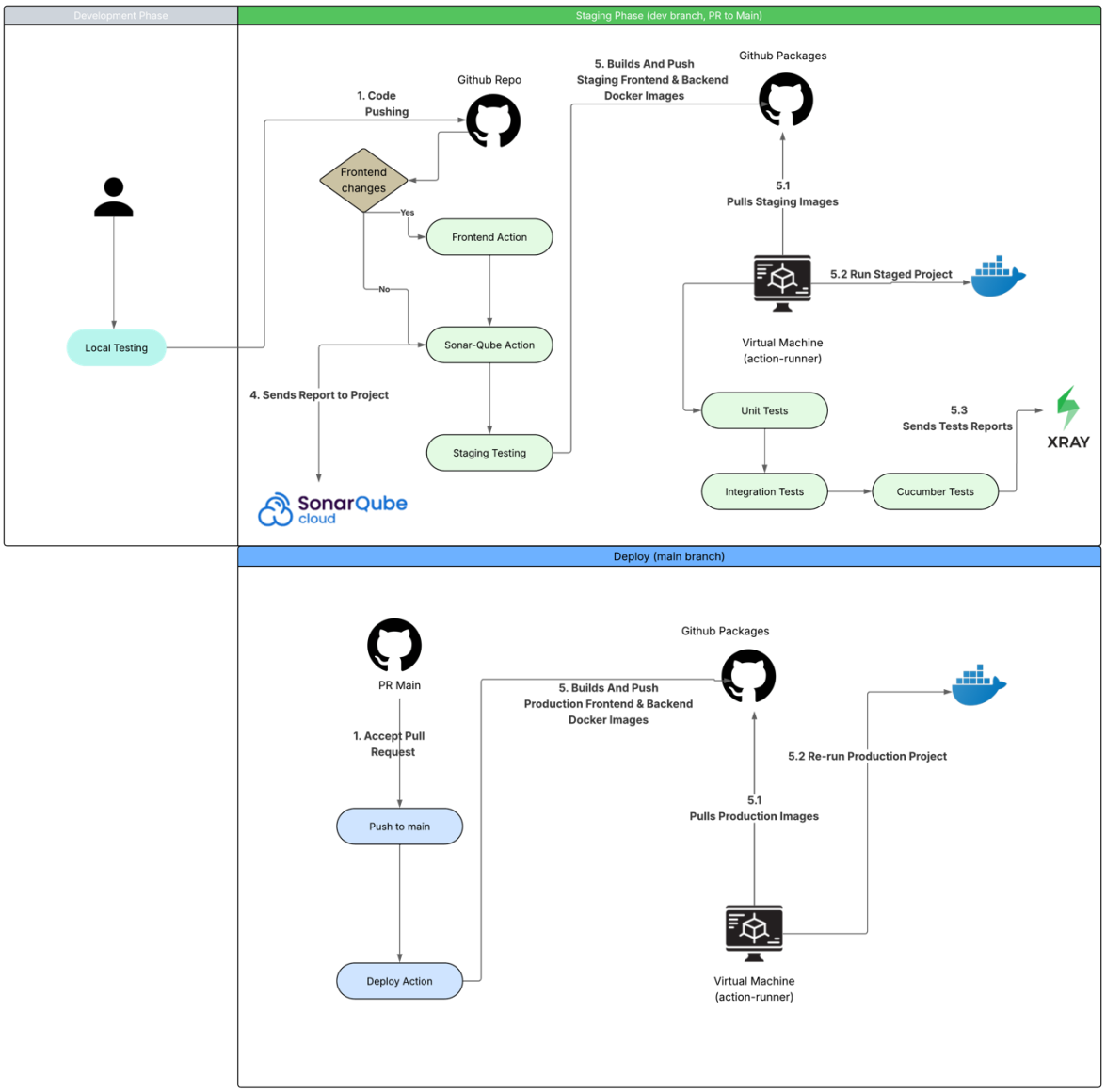
- Requests to the backend can't take longer than 2 seconds

4.2 Architecture View



4.3 Deployment View

The deployment process outlined in the diagram begins when a pull request is approved and merged into the main branch, marking the transition from staging to production. This action triggers a GitHub workflow that initiates the production deployment pipeline. As part of this process, both the frontend and backend components of the application are built and packaged into Docker images. These production-ready images are then pushed to GitHub Packages, serving as the central registry. A virtual machine, acting as a GitHub action-runner, pulls the latest production images from this registry and uses them to re-run the application in a live environment. This fully automated workflow ensures a smooth, consistent, and traceable release process, minimizing human error and promoting reliability in production deployments.

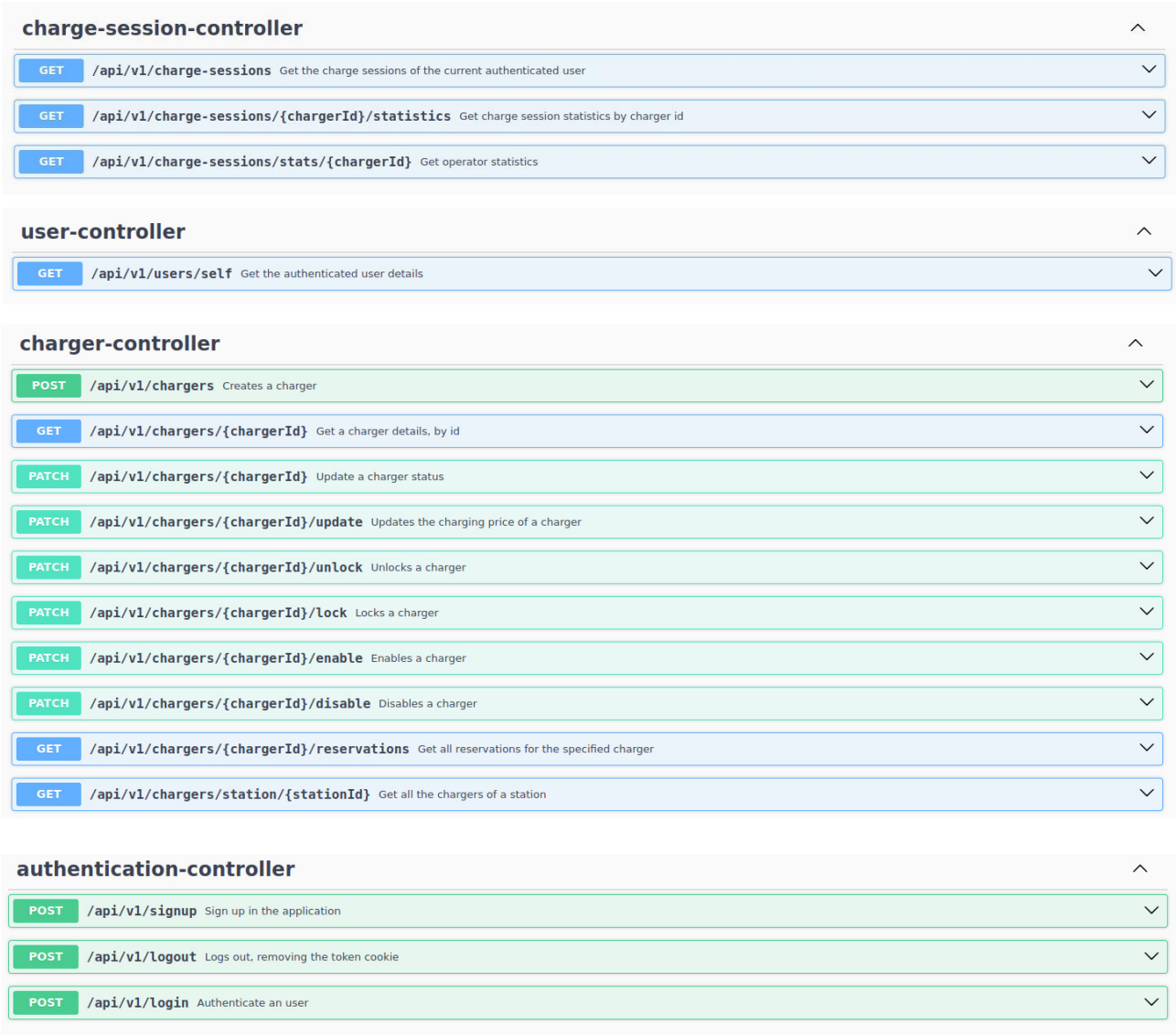


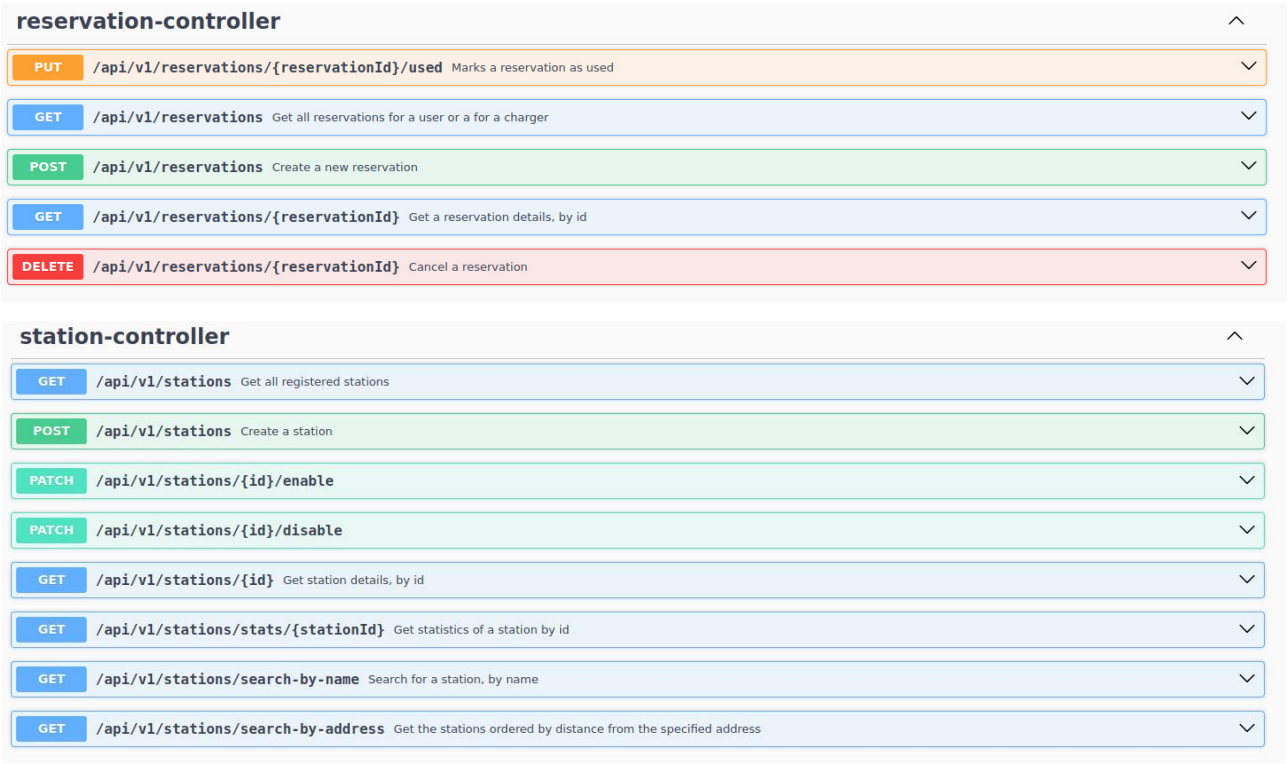
5 API for developers

For our application we created an API with six main collections, which are correctly documented using Swagger:

- **charge-session-controller:** Deals with charging sessions, allowing to retrieve both user directed and operator direct sessions statistics
- **user-controller:** To retrieve authentication details
- **charger-controller:** To deal with all charger logic such as creation, status updates and usage
- **authentication-controller:** For authentication purposes
- **station-controller:** To manage all station logic such as creation, search and status updates
- **reservation-controller:** To manage reservations

Below we present the different endpoints for each controller.




reservation-controller
PUT /api/v1/reservations/{reservationId}/used Marks a reservation as used
GET /api/v1/reservations Get all reservations for a user or a for a charger
POST /api/v1/reservations Create a new reservation
GET /api/v1/reservations/{reservationId} Get a reservation details, by id
DELETE /api/v1/reservations/{reservationId} Cancel a reservation
station-controller
GET /api/v1/stations Get all registered stations
POST /api/v1/stations Create a station
PATCH /api/v1/stations/{id}/enable
PATCH /api/v1/stations/{id}/disable
GET /api/v1/stations/{id} Get station details, by id
GET /api/v1/stations/stats/{stationId} Get statistics of a station by id
GET /api/v1/stations/search-by-name Search for a station, by name
GET /api/v1/stations/search-by-address Get the stations ordered by distance from the specified address