TQS: Product specification report

Patrícia Dias [98546], Pedro Lopes [97827], Raquel Ferreira [98323], Sophie Pousinho [97814]

v2022-06-23

1 Introduction

1.1 Overview of the project

This project's objectives were to develop a viable software product with the specification and enforcement of Software Quality Assurance.

Our project includes two applications: Delivera and Frostini.

The Delivera application consists of a platform that aims to provide delivery services and means of easy access to users, companies and riders. It should make it easy for all the types of clients to interact with our service, allowing them to create and receive orders, create and register riders as well as rate them, and register companies and manage their products.

Frostini, a local ice cream shop, craves innovation, therefore, is trying to find a way of distributing its self-made products across a wider area of impact. They ask for a large company to make them a service which they can make use of to distribute their products and keep track of incoming orders.

1.2 Limitations

When it comes to limitations, we were not able to implement all of the statistics that the Delivera administrator was supposed to check, including the statistics related to the stores and the orders. We also were not able to implement RabbitMQ to update the state of the orders. The features of the manager of Frostini are also very incomplete, most of it is only static.

2 Product concept

2.1 Vision statement

This system can be used for the distribution of orders placed by customers of a service to various riders, being those services, partners of the delivery company (Delivera). The products that can be ordered are provided by the stores that register as partners of Delivera, being Frostini an example of that.

This delivery system allows the stores that subscribe to the delivery service a way to have more freedom to manage their products from the original service, whereas many services nowadays, like the famous UberEats, provide one platform that contains all the stores and they can't personalize it as much.

In order to collect and select requirements, our group would meet to brainstorm about the ideas we had in mind.

2.2 Personas

Ice Cream Courier

João Almeida is an eighteen-year-old, who recently graduated high school and intends to continue his studies at university, but he also wants to start earning money by working in the summer break, without neglecting his time with his friends. Since he has his driving license and his dad bought him a motorcycle for his birthday. He applied to work at Delivera and is now a rider.

Manager

Maria Conceição is thirty-one years old and she started her own ice cream shop Frostini 5 years ago, but she thinks it's time for innovation since many customers suggested they would like a more accessible way to get the shop's products.

Client

Xavier Gonçalves is a thirty-two-year-old busy doctor and father. After getting his medical degree he's been working in the hospital of Aveiro and his job is very time-consuming, as a result, he values the times he has with his family usually going to the park near their house or going to the beach. Xavier and his family really like the ice cream shop and would like to order some of their products, to enjoy while at the park/beach.

Delivera Manager (admin)

Carlos Oliveira is twenty-two-year-old and is the owner of Delivera. As he is administrating the company, he feels the need to check how the system is behaving in order that the stores that subscribe to the service are pleased.

2.3 Main scenarios

Scenario 1: João Almeida sets his status as available and receives delivery requests

After spending time with his friends the night before, João wants to work, so he logs in the application, goes to the profile page and sets his status as available. João will get the nearest orders on the home page and will accept the one he prefers. Afterward, he goes to the Ice Cream Shop to get the order and delivers it to the client.

Scenario 2: Maria Conceição registers her store in the Delivera application and manages its products

Since Maria wants to have an online ice cream shop, she registers her shop in the Delivera application and adds the available products online.

Scenario 3: Xavier Gonçalves places an order

After lunch, the Gonçalves family wants to enjoy an ice cream at the shop while dipping their feet in the water at the beach, so Xavier registers in the application and places the order. When it gets accepted, he will be able to track his order in real-time.

Scenario 4: Carlos Oliveira checks statistics about Delivera

Carlos Oliveira wants to ban the profits of this month, so he logs in the Delivera, and checks statistics about the profits of the deliveries.

2.4 Project epics and priorities

In order to have an incremental implementation we prepared our functionalities for each sprint iteration organizing them in epics, and therefore, user stories.

Ice Cream Courier - Delivera

User Story 1: As a rider, I want to register in the application so that I can log in.

User Story 2: As a rider, I want to share my location so that I get delivery requests close to my location.

User Story 3: As a rider, I want to accept a certain delivery request so that I start working on it.

User Story 4: As a rider, I want to set my status as unavailable so that I don't get delivery requests.

User Story 5: As a rider, I want to finish my task by setting my status as available so that I get more delivery requests.

User Story 6: As a rider, I want to login so that I can get delivery requests.

Manager - Frostini

User Story 1: As the manager, I want to register my store in Delivera so that my store sells products online.

User Story 2: As the manager, I want to manage the products available to buy so that products are up to date with the available products.

User Story 3: As the manager, I want to check the statistics of the purchases so that I can check the profits of using Delivera

Client - Frostini

User Story 1: As a client, I want to register so that I can log in.

User Story 2: As a client, I want to log in so that I can order ice creams.

User Story 3: As a client, I want to search for a product so I can add it to my order.

User Story 4: As a client, I want to order so that I can get the products.

User Story 5: As a client, I want to track my order so that I can visibly know where the driver is.

User Story 6: As a client, I want to rate my driver so that I can give feedback on my experience.

Administrator - Delivera

User Story 1: As an admin, I want to check statistics about the deliveries, including the number of orders made to a specific store and the number of stores, and manage the stores in the system.

User Story 2: As an admin, I want to check on statistics about the riders' productivity, including the total number of riders, the average time to deliver an order and the rider's average rating.

User Story 3: As an admin, I want to check statistics about the users, including the number of orders by the user.

User Story 4: As an admin, I want to log in so that I can check Delivera statistics

(Epic 1) The user places an order

- ➔ User Story: As a client, I want to log in so that I can order ice creams.
- ➔ User Story: As a client, I want to register so that I can log in.
- ➔ User Story: As a client, I want to search for a product so I can add it to my order.
- ➔ User Story: As a client, I want to order so that I can get the products.

(Epic 2) Riders sign up and log in

- ➔ User Story: As a rider, I want to register in the application so that I can get delivery requests.
- ➔ User Story: As a rider, I want to login so that I can get delivery requests.

(Epic 3): The admin (Delivera) wants to check statistics

- ➔ User Story: As an admin, I want to check statistics about the deliveries, including the number of orders made to a specific store and the number of stores and manage the stores in the system.
- ➔ User Story: As an admin, I want to check on statistics about the riders' productivity, including the total number of riders, the average time to deliver an order and the rider's average rating.
- ➔ User Story: As an admin, I want to check statistics about the users, including the number of orders made by the user.

- ➔ User Story: As an admin, I want to check statistics about the users, including the number of orders by the user.
- ➔ User Story: As an admin, I want to log in so that I can check Delivera statistics

(Epic 4): The manager (Frostini) wants to manage

- ➔ User Story: As the manager, I want to register my store in Delivera so that my store sells products online.
- ➔ User Story: As the manager, I want to manage the products available to buy so that products are up to date with the available products.
- ➔ User Story: As the manager, I want to check the statistics of the purchases so that I can check the profits of using Delivera

(Epic 5): User tracks their order

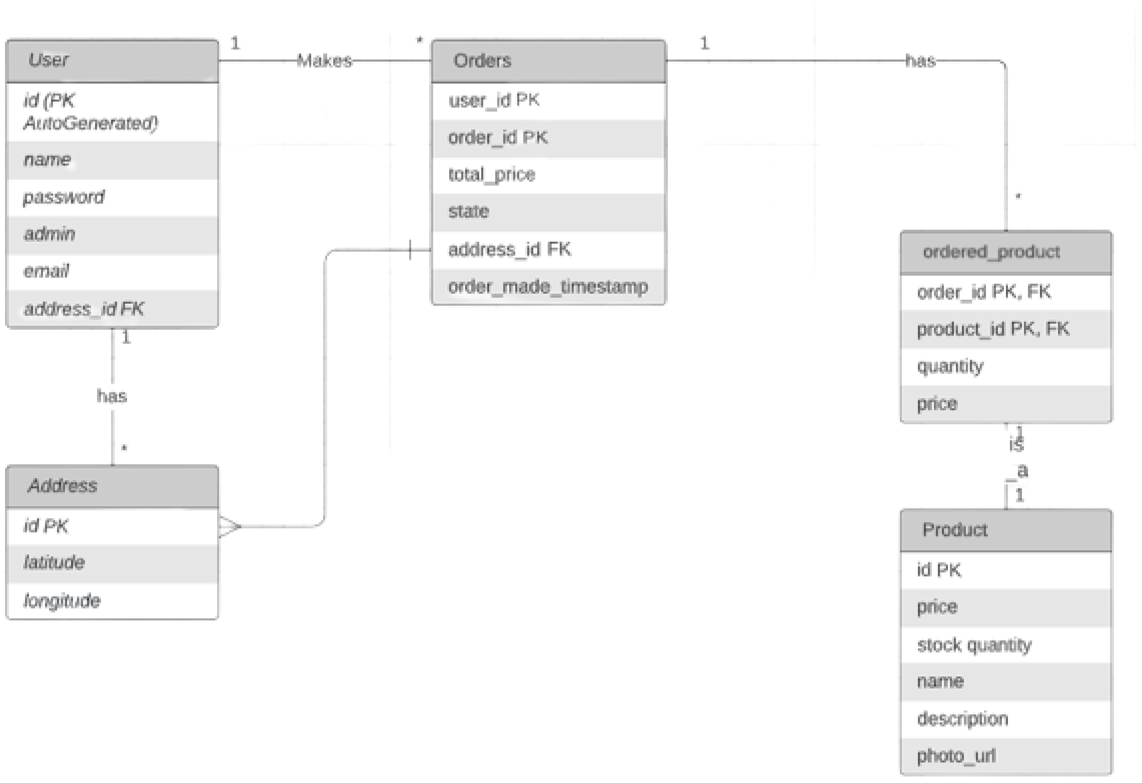
- ➔ User Story: As a client, I want to track my order so that I can visibly know where the driver is.
- ➔ User Story: As a client, I want to rate my driver so that I can give feedback on my experience.

(Epic 6): Riders make deliveries

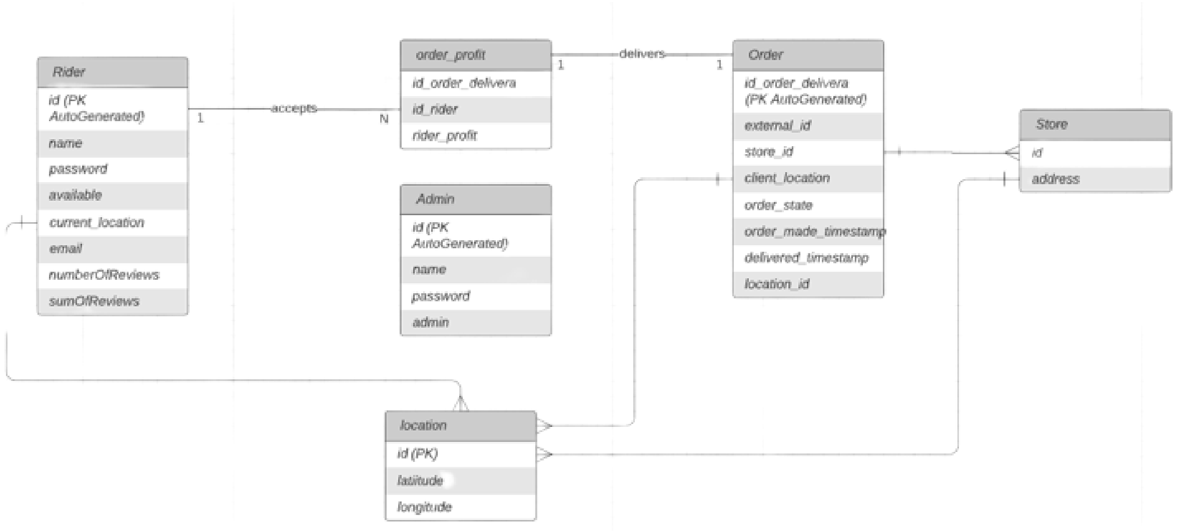
- ➔ User Story: As a rider, I want to set my status as unavailable so that I don't get delivery requests.
- ➔ User Story: As a rider, I want to share my location so that I get delivery requests close to my location.
- ➔ User Story: As a rider, I want to finish my task by setting my status as available so that I get more delivery requests.
- ➔ User Story: As a rider, I want to accept a certain delivery request so that I start working on it.

3 Domain model

Frostini domain model



Delivera domain model



4 Architecture notebook

4.1 Key requirements and constraints

This system must compliant to some directives:

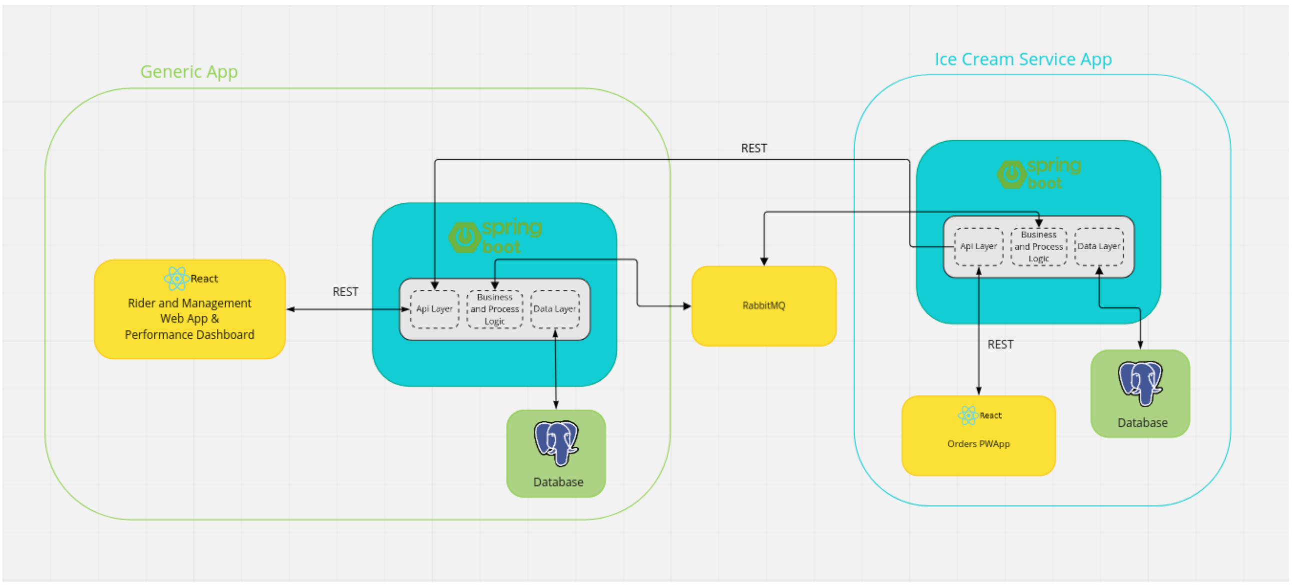
- Capable of dealing with large amounts of data arriving nearly in real time
- Deal with multiple simultaneous requests
- Be available nearly everyday everytime
- Be hosted by a provider that can assure any network problems, such as connectivity
- It should be built in way that it can be maintained in the long run
- It must be capable of showing relevant information in the multiple views we must provide, admin level, user lever, manager and rider level
- Provide useful, easy to use and intuitive endpoints/API

Since we will also provide a mobile application (PWA) we need to take in considerations both different operational systems as well as their different versions.

4.2 Architectural view

Our project intends to develop a fully pledged system capable of being used everywhere, for this reason, it is important to choose software solutions that are adequate to our use case. Since we also intend to ensure that our system is correctly tested we made the decision to use Spring Boot as our backend for both, the generic application and the specific one, this decision was made on top of the knowledge of the team as well as their overall experience with this framework.

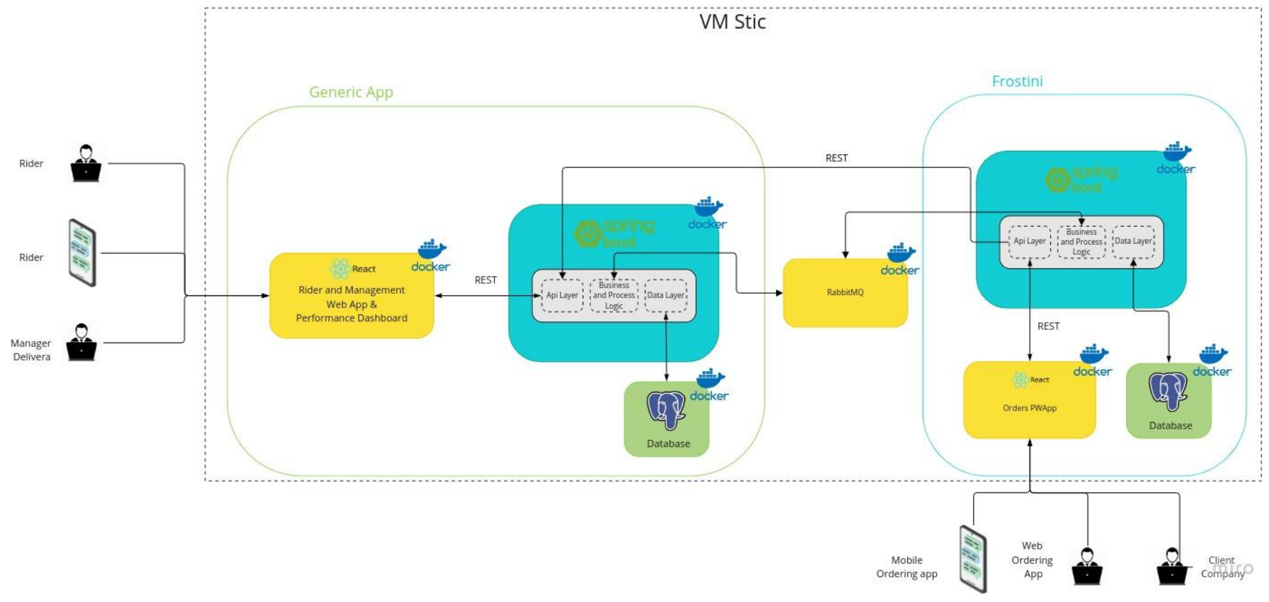
As of the mobile and web components, we opted by using React and React Native, these two technologies ensure that our services can be used independently from what the user is relying on to access our services, meaning that a user with an iOS device can interact with them in the same way an Android user does.



4.3 Deployment architecture

In order to deploy our system we used dockers. All our applications are stored in a container registry that can be consulted and pulled through this link https://github.com/sophjane?tab=packages&repo_name=tqs-delivery-p1g2.

Our application is currently deployed here http://deti-tqs-02.ua.pt in the following ports, 3000, 3001, 8083, 8080.



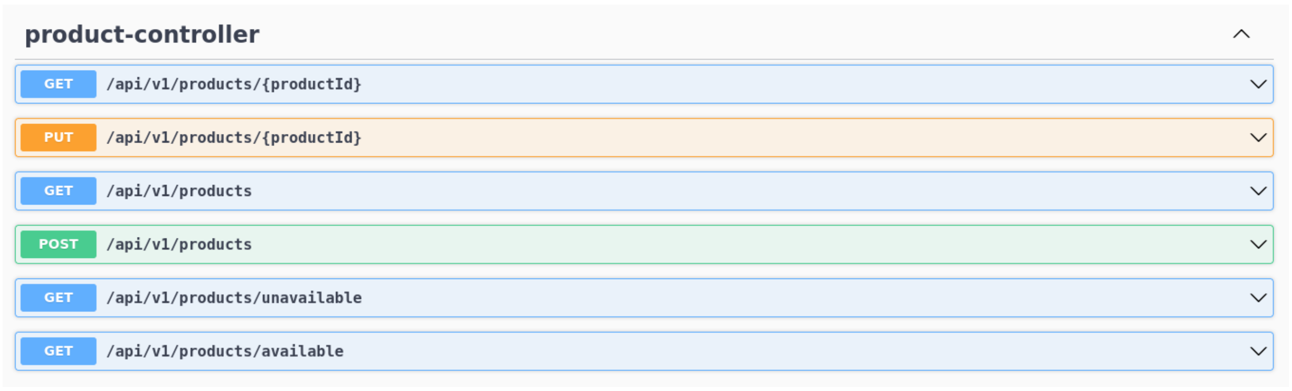
5 API for developers

To build both API's documentation we included a spring dependency that automatically creates documentation based on annotations held in the methods of the controllers.

This helps both the developer and the programmer since there is no need to further document or write information asynchronously about their created endpoints.

The documentation can be seen here for the Frostini API and here for the Delivera AP.

Frostini



product-controller

GET /api/v1/products/{productId}

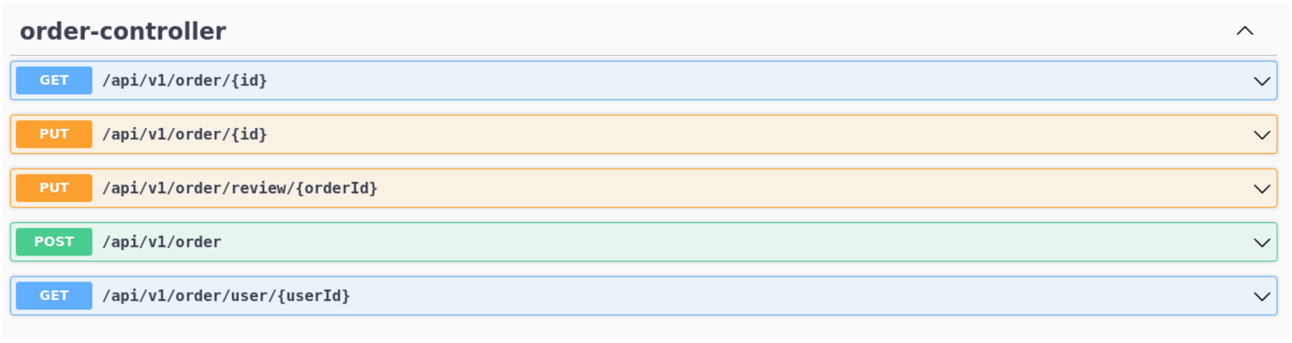
PUT /api/v1/products/{productId}

GET /api/v1/products

POST /api/v1/products

GET /api/v1/products/unavailable

GET /api/v1/products/available



order-controller

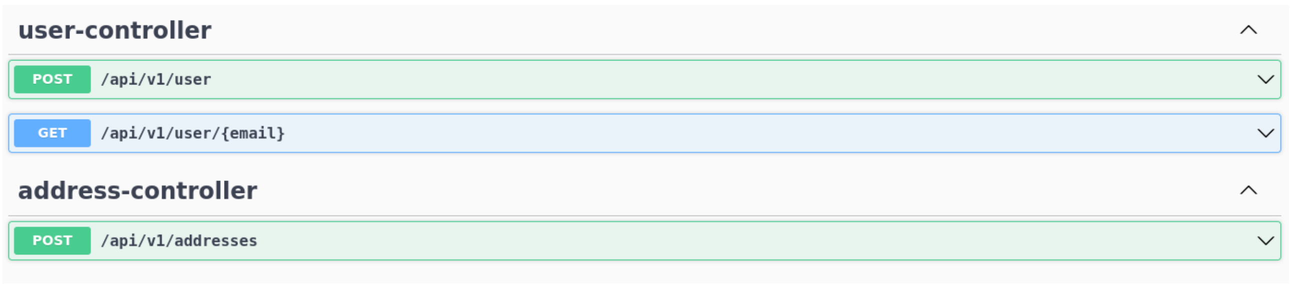
GET /api/v1/order/{id}

PUT /api/v1/order/{id}

PUT /api/v1/order/review/{orderId}

POST /api/v1/order

GET /api/v1/order/user/{userId}



user-controller

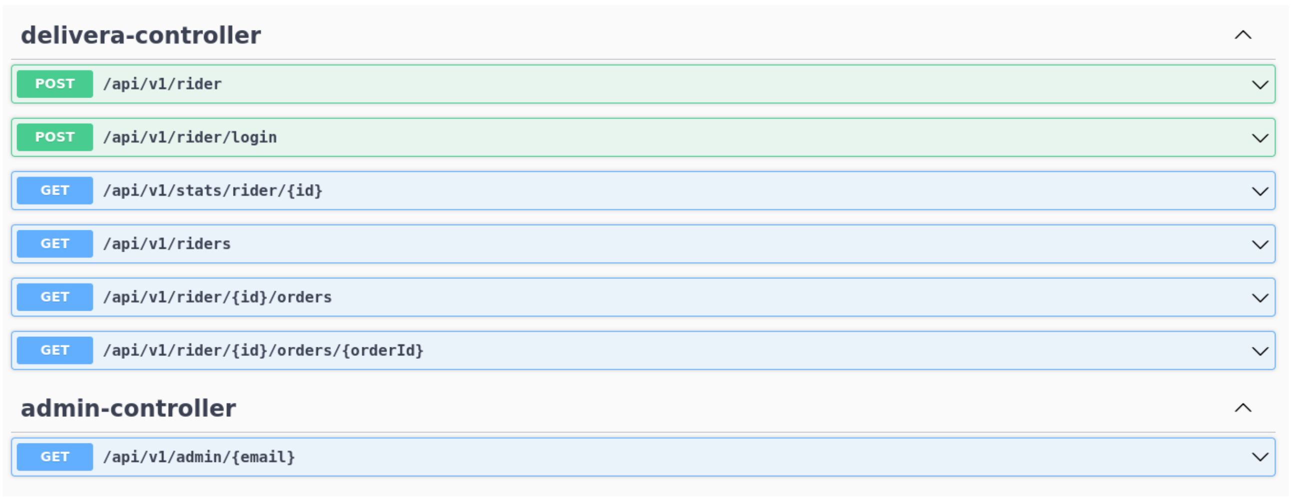
POST /api/v1/user

GET /api/v1/user/{email}

address-controller

POST /api/v1/addresses

Delivera



delivera-controller

POST /api/v1/rider

POST /api/v1/rider/login

GET /api/v1/stats/rider/{id}

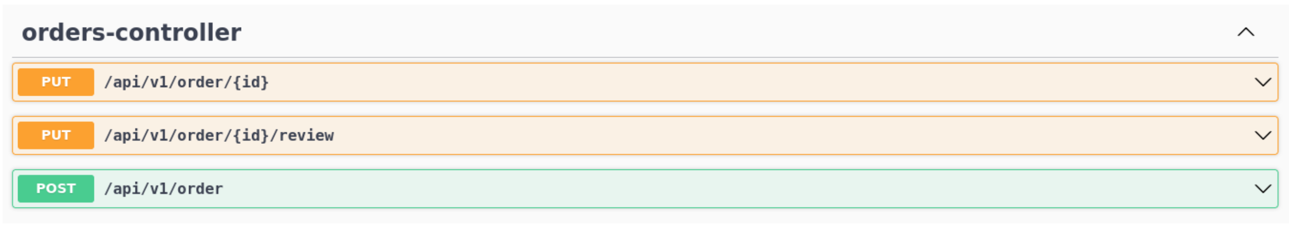
GET /api/v1/riders

GET /api/v1/rider/{id}/orders

GET /api/v1/rider/{id}/orders/{orderId}

admin-controller

GET /api/v1/admin/{email}


orders-controller

PUT /api/v1/order/{id}

PUT /api/v1/order/{id}/review

POST /api/v1/order

6 References and resources

https://mui.com/

https://www.creative-tim.com/product/material-dashboard-react

https://codepen.io/rares-lungescu/pen/KLbMvo