deti universidade de aveiro departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Bruno Bastos [93302], Daniel Gomes[93015], Leandro Silva[93446], Mário Silva[93430]
v2021-05-23

1 Introduction

1.1 Overview of the project

The recent pandemic showed how fragile the transportation system is. Many people needed to be confined and public transports started failing. Without the means of transport, these people had to give up on some important products, some of which are medicines. It is important that everyone has access to medicines in order to prolong their life expectancy, but with the current system this priority is not easily accessed by everyone. The product that we want to create, MedEx, tries to provide equal access to medicines to everyone in need without letting anyone out.

1.2 Limitations

The algorithm responsible for the assignment of couriers to a delivery is simple and does not take into consideration the distance to where the products need to be retrieved.
Couriers can not refuse some delivery that is assigned to them.
There is no authentication for hosts, meaning that any host can use the platform.

2 Product concept

2.1 Vision statement

The application will be used for medicine distribution using couriers. Clients can order medicines from local suppliers, usually pharmacies, and their request will be assigned to a nearby courier that will deliver the package to the client. This application tries to provide easier access to medicine to the clients, which most of the times have difficulties finding a transport to a nearby pharmacy or have physical problems that do not allow them to go there in person. Making use of the application the clients can get their medicines without needing to leave their place, reducing the client's risk and worries.

2.2 Personas

Henrique Silva - Henrique is a 45 year old man, residing in Covilhã, Portugal. Recently, Henrique's health has been affected a lot, resulting in some medical conditions that makes him needing some to prescribe medicines from the local pharmacies. In order to prescribe the medicines, he has to walk a considerable distance, which is not practical since Henrique has some difficulties walking. Therefore, it would be nice if there was a way to automate and deliver his medicines without having to walk to acquire them.

Clara Almeida - Clara is a 32 year old woman that manages her own pharmacy on Covilhã. Due to the current pandemic and, consequently, the restrictions that it imposed, Clara's local pharmacy revenue has affected a lot, since people do not go to the pharmacy as much as they used to. Her current goals consist in expanding her business in the sense that the medicines could be delivered to anyone's houses.

Tiago Correia - Tiago is a 24 year old man from Castelo Branco that unfortunately has lost his Job due to the economic downfall that resulted from the COVID-19 pandemic. Now, as a way to earn some money while finding a new Job, he would like to make deliveries with his motorcycle. However, from his residence area, there are not any delivery services as in big cities like Porto or Lisbon, and, therefore, it would be good for him if any new local delivery Service arised.

Leandro Caetano - Leandro is a 35 year old entrepreneur that found the need of a delivery system in his hometown, Covilhã. Therefore, he went forward and created one himself, and now he is responsible for managing and controlling the system.

2.3 Main scenarios

In this section, it will be mentioned and described some scenarios representing core functionalities of our system. It is also important to mention that these scenarios are based on each person's points of view and needs, and obviously, do not approach all details of the system.

- **Scenario 1** : Henrique has run out of the medicines he has to take, and he recently heard from his son that there is a new application, MedEx, that delivers medicines. Therefore, inside the application, he is able to search for the name of it, and consequently check the list of pharmacies that have the product in stock. He realises that the pharmacy he used to go to his residence area is on that list, and selects it. After selecting, he inputs the quantity that wants to purchase as well as the payment method. Finally, after introducing his home address, Henrique confirms the order, and checks on the system, the estimated amount of time until the delivery, and also the rider in charge of the delivery. Once the delivery has arrived, Herique confirms the delivery on the application.
- **Scenario 2**: Clara revenue has been decreasing due to the pandemic. Some of her longtime customers mentioned how hard it is for them to get medicines during the confinement. One of her customers suggested the use of ExDelivery, a new app that allows companies to integrate the delivery system they offer, on any companies' services. Therefore, she decided to create an application, MedEx, that allows ordering medicines from her pharmacies that are delivered by couriers of the ExDelivery system. She posts on MedEx the available medicines, their stock and the location of the supplier. Once someone makes an order, she is notified through the app, which makes it easier to start packaging the order to be delivered by a courier.
- **Scenario 3:** Tiago wants to make a greater use of his motorcycle and earn money by working in a delivery system as a courier. After seeing a local advertisement of the ExDelivery Application, he rushed into downloading it, and registered himself on the platform as a Courier. Tiago went to the main dashboard for couriers, changed his status to "Available" and was suggested on the platform with some deliveries that he could do near his residence area. After checking the available ones, he chooses one and receives the location of the destination. After arriving at the final destination, he confirms on the platform the delivery, and, after the end-user, confirms too, a commission will be put on his account.
- **Scenario 4**: Leandro would like to manage his delivery system in a simple and practical way, therefore, it would be interesting to check on an Administrator Platform, the state and details of each delivery, as well as the user who made the delivering request and the courier assigned. This simple visualization could help in checking whether the delivering system is working correctly. Besides this, he would also be responsible for managing couriers, such as accepting their registrations and firing them.

2.4 Project epics and priorities

In order to implement an incremental solution and prepare the functionalities for each sprint iteration, we organized the following user stories in epics, as demonstrated below.

Client - MedEx
User Story: As Henrique I want to register on the pharmacy platform as a client.
User Story: As Henrique I want to browse through all Products.
User Story: As Henrique I want to be able to order medicine.
User Story: As Henrique I want to be able to see the status of my current order.
User Story: As Henrique I want to make a review of the delivery once I get my order.

Pharmacy Owner - MedEx
User Story: As Clara I want to register my pharmacy.
User Story: As Clara I want to be able to add a Product to the platform.
User Story: As Clara I want to be able to update a Product.
User Story: As Clara I want to be able to be notified when a Product is ordered.

Courier - ExDelivery

User Story: As Tiago I want to be able to register as a courier.
User Story: As Tiago I want to be able to see all my orders.

Delivery System Admin - ExDelivery
User Story: As Leandro I want to be able to fire couriers.
User Story: As Leandro I want to be able to see the list of all orders.

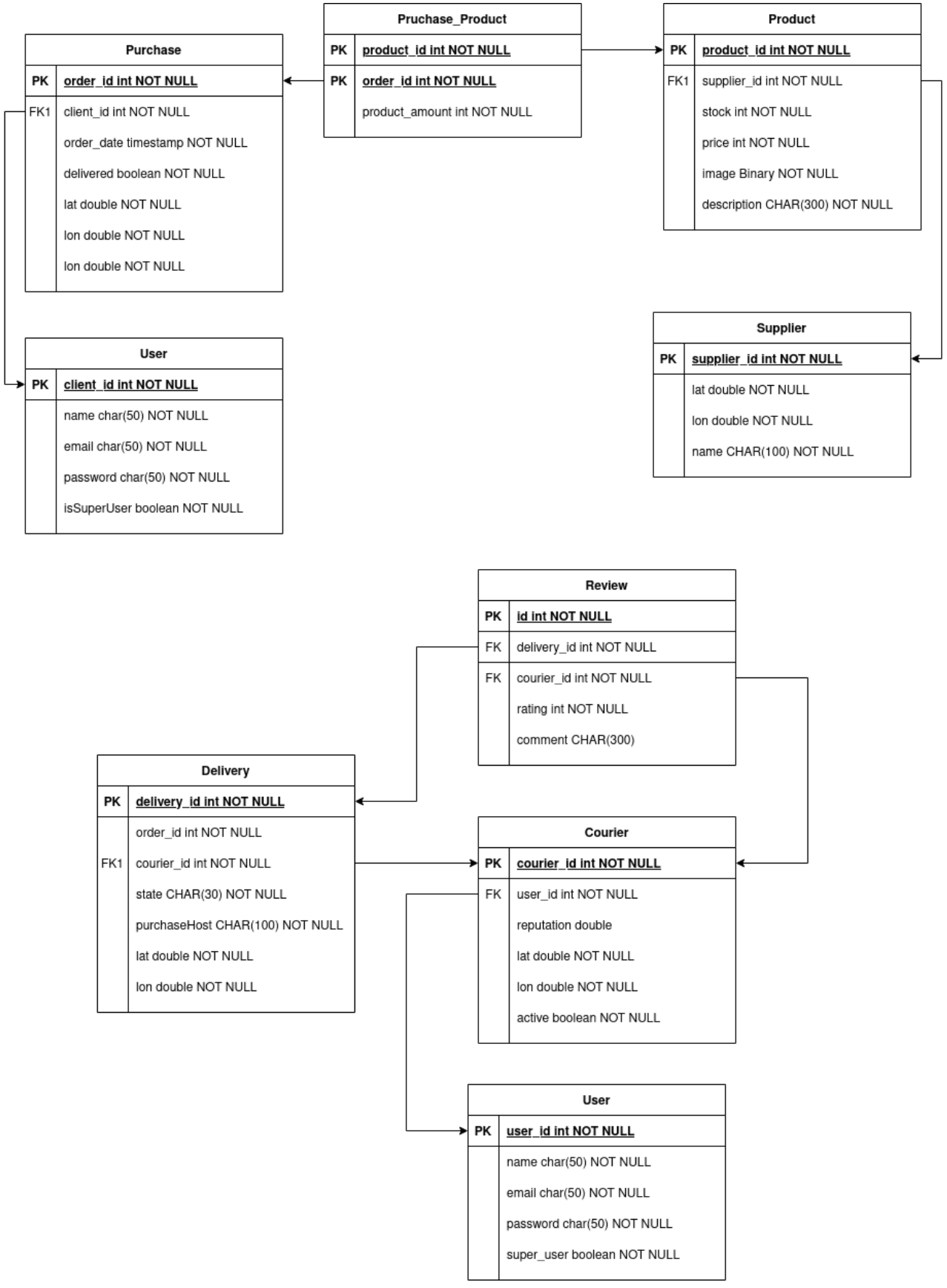
Organization in epics:

Epic 1 - Authentication and Product Management
User Story: As Henrique I want to register on the pharmacy platform as a client.
User Story: As Henrique I want to be able to order medicine.
User Story: As Henrique I want to browse through all Products.
User Story: As Clara I want to register my pharmacy.
User Story: As Clara I want to be able to add a Product to the platform.
User Story: As Clara I want to be able to update a Product.
User Story: As Tiago I want to be able to register as a courier.

Epic 2 - Order Management and Assignment
User Story: As Henrique I want to be able to see the status of my current order.
User Story: As Tiago I want to be able to see all my orders.
User Story: As Leandro I want to be able to see the list of all orders.
User Story: As Clara I want to be able to be notified when a Product is ordered.

Epic 3 - Courier rating system
User Story: As Henrique I want to make a review of the delivery once I get my order.
User Story: As Leandro I want to be able to fire couriers.

3 Domain model



The above diagram shows the database implementation for both services. At top we have the database model for Medex, the medicine provider. The model below was used in ExDelivery, the product delivery platform.

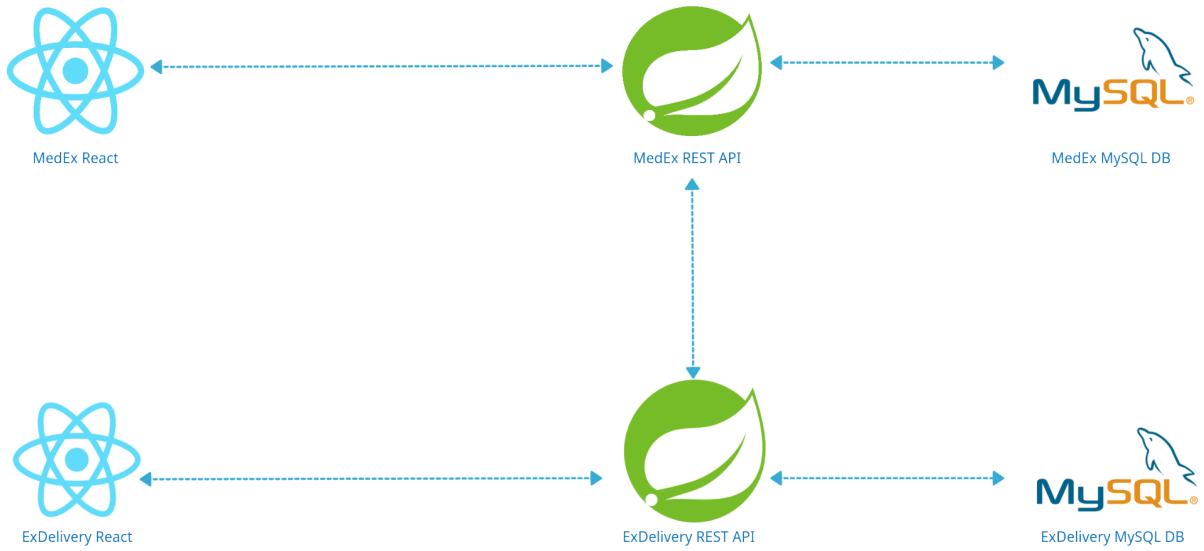
4 Architecture notebook

4.1 Key requirements and constraints

We want our application to be able as a mobile application and as a website. Our frontend technology should be easily converted to both platforms so that we can reuse the code, thus reducing the development time.
The backend of the application should provide us with the testing tools necessary to ensure the quality of our services. It should be capable of hiding and protecting information from users that are not authenticated, providing some privacy to the end users of the services.
As for the database, the technology used must be able to persist great amounts of data, allowing a fast retrieve of that data. The DBMS should support a Relational schema since it is the most fitted for our needs and it is the one we are more familiarized with.

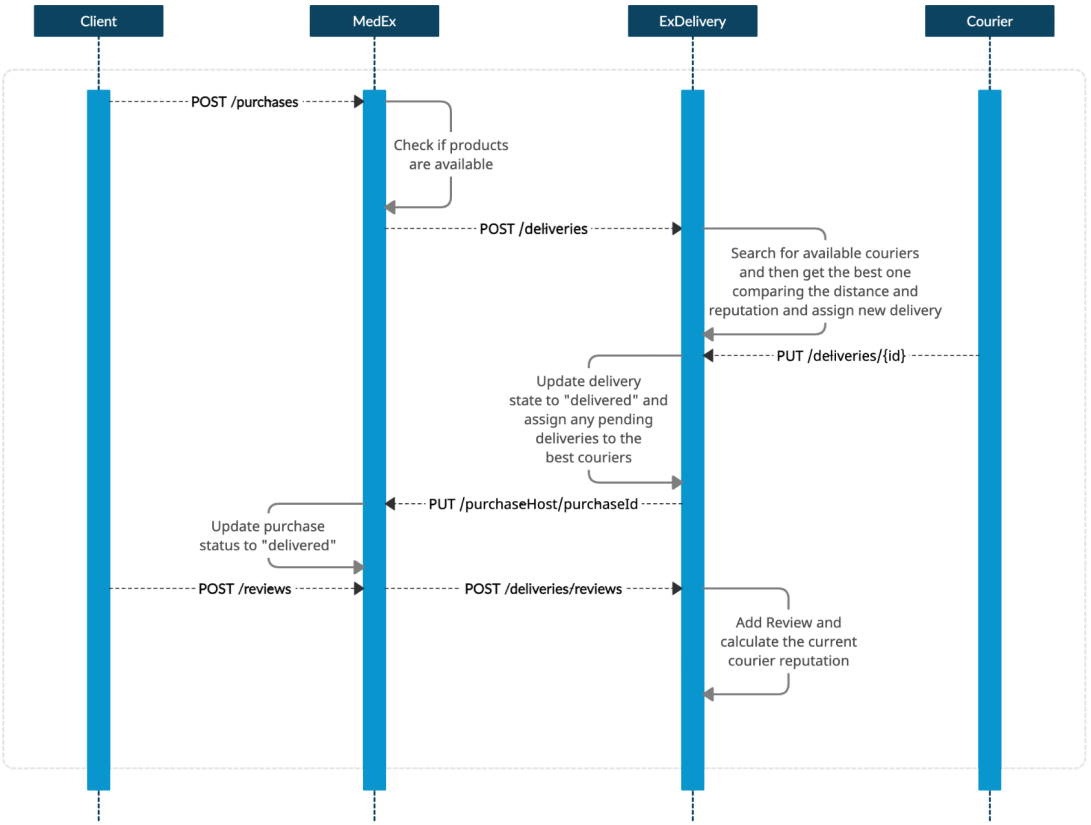
4.2 Architectural view

In this section, it will be presented the architecture that we defined and constructed for our solution, which is visible on the following picture.



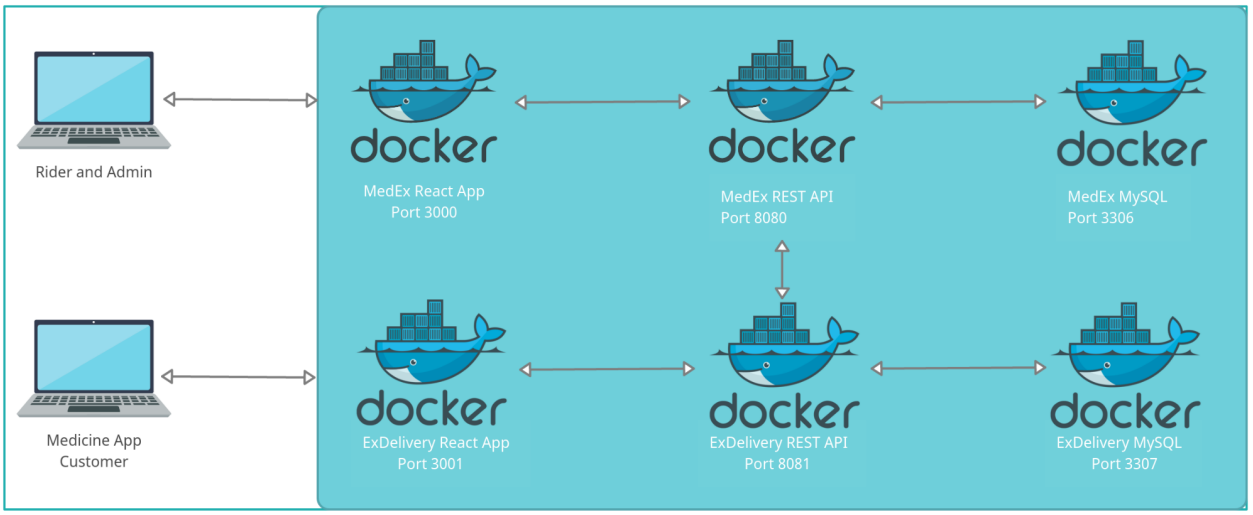
In this architecture there are three main modules in which our system is divided: Frontend, Backend and Storage. For our Frontend, which will be responsible for rendering the Web Pages and providing the Web Interaction, we decided to use *React*, as the Javascript and TypeScript Framework, since we, as a team, have considerable experience using this technology. Besides this, React speeds up the process of building Web Pages, and provides a dynamic rendering of the DOM elements on the Web App. Going over the Backend, the framework used was SpringBoot as it is heavily recommended by our Professor for us to use, which is fundamental to provide our REST API, and develop the business Logic. The Backend communicates with the Storage module, which consists of a MySQL database. MySQL has the advantages of having better performance on reading considerable amounts of Data,

and besides this, it happens to be one of the most suitable databases for us, since it is a Relational One.
The idea is to have a client make a purchase through medex, purchasing any product from any supplier registered on MedEx, then, if the products are available MedEx will use the ExDelivery RestAPI for creating a delivery for the client's purchase. ExDelivery will then try to find a courier that is best suitable for the delivery and will assign him to make the delivery. Once the courier completes the delivery he confirms through the mobile or web app that he has delivered. ExDelivery will then update the delivery state to "delivered" and notify the host of that purchase that it has been delivered, in this case, MedEx. The client will now be able to make a review of the delivery service, MedEx will once again use the ExDelivery RestAPI and send the review made to the courier's delivery, which will then be calculated and saved the new courier's reputation.



4.3 Deployment architecture

In order to deploy our product in our production environment, we went forward to package each of the services in Docker containers, as it is a technology that we are really familiar with and allows virtualization at OS level. To be more specific, we recurred to *docker-compose,* which allows the deployment of multiple services on docker containers, "*on the fly*". Therefore, as it is possible to see from the following image, we have 6 docker containers from which the end user will most times communicate with the services, through the Web Apps.

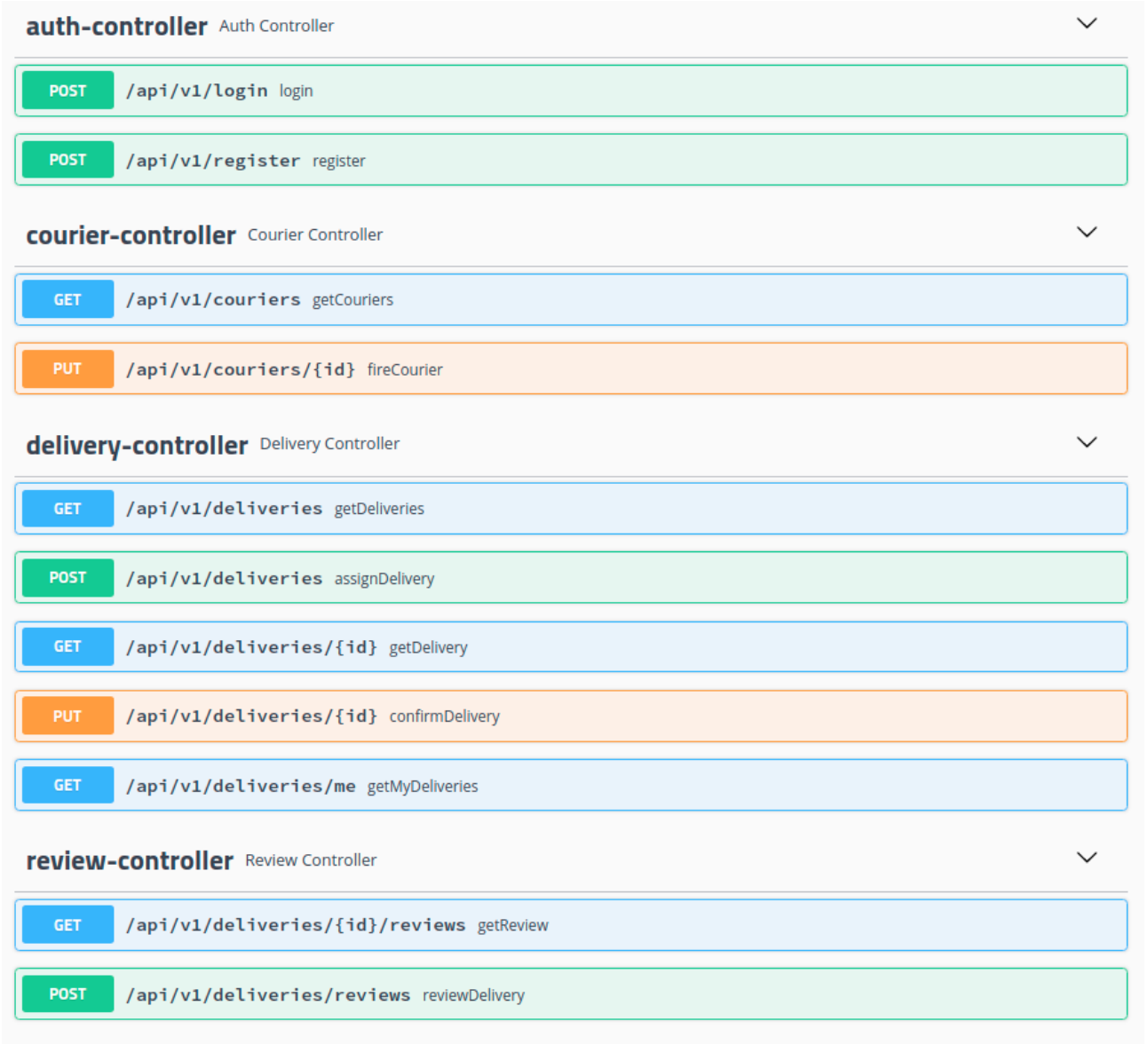

Rider and Admin
Medicine App
Customer
docker
MedEx React App
Port 3000
docker
MedEx REST API
Port 8080
docker
MedEx MySQL
Port 3306
docker
ExDelivery React App
Port 3001
docker
ExDelivery REST API
Port 8081
docker
ExDelivery MySQL
Port 3307

5 API for developers

The APIs documentation were made with Swagger, and can be consulted on the following links:

- ExDelivery: http://192.168.160.231:8081/swagger-ui.html
- MedEx: http://192.168.160.231:8080/swagger-ui.html

5.1 ExDelivery API



The auth controller handles the authentication of the users, providing them with a token to make requests to the remaining endpoints, and designates the users role on the platform, that is, if they are administrators or couriers.

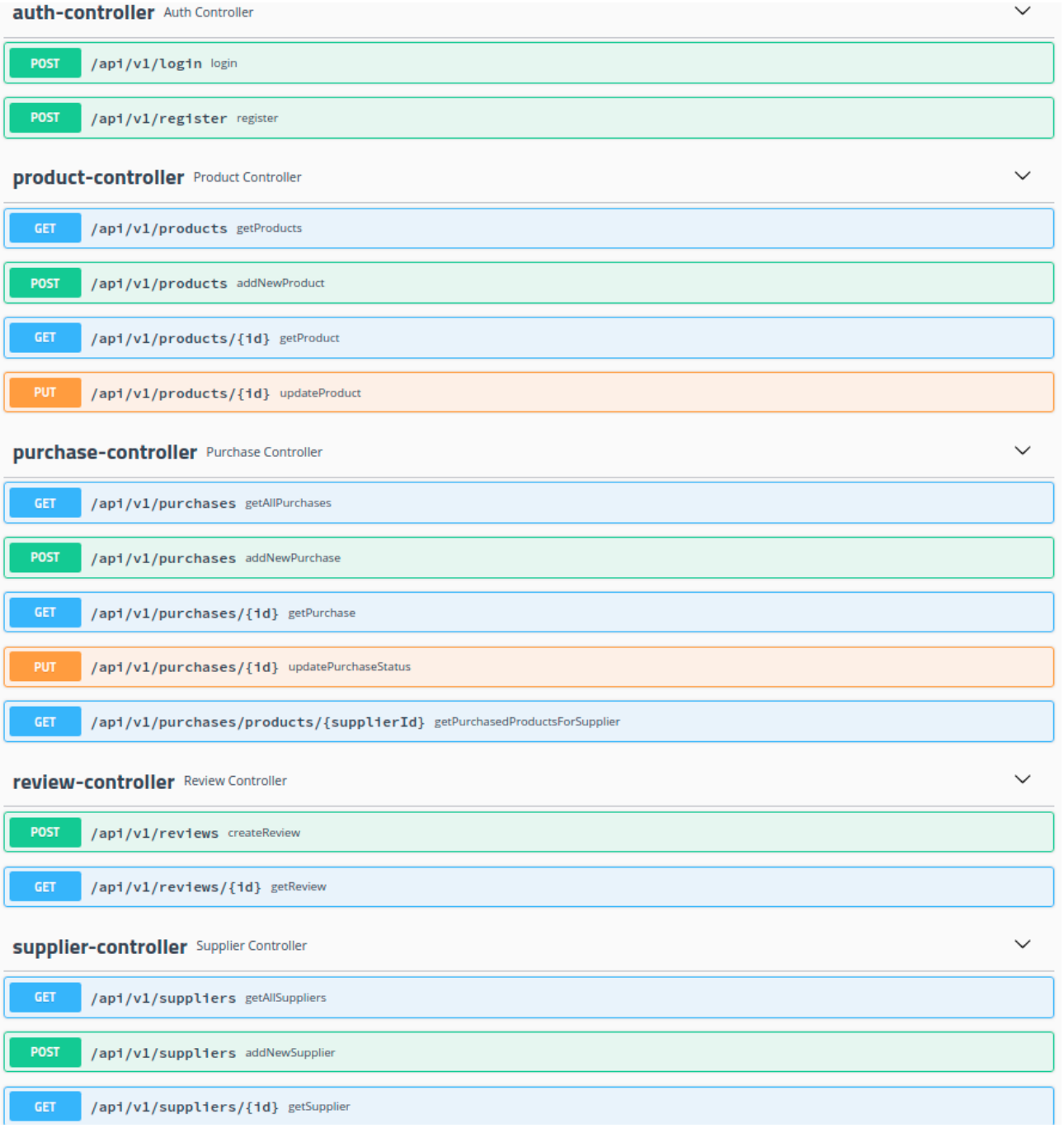
The courier controller shows all couriers registered on the platform. There is also a method to update the status of one courier, which is only used to fire couriers by the administrators.

The delivery controller is the main controller of this API. It shows all the deliveries, a specific delivery, and all deliveries assigned to one courier. It also handles the assignment of one delivery to a courier and the confirmation of the delivery.

The review controller handles the reviews made to the deliveries which are mandatorily delivered. These reviews have a rating which will update the couriers' reputation.

The methods that should be used by other hosts which desire to use the ExDelivery services are the **assignDelivery**, **reviewDelivery** and probably the **getReview**.

5.1 MedEx API



The MedEx API has a similar structure as the one from ExDelivery, in which the purchase related to the delivery on ExDelivery.

It has in addition the product controller, which handles all the products which are in storage and able to be purchased, and the supplier controller, which handles the suppliers from which the products belong.

The methods that are dependent on the ExDelivery services are the **addNewPurchase**, which calls the **assignDelivery**, and the **createReview** and **getReview**, which call the **reviewDelivery** and **getReview**, respectively.