deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS: Product Specification Report

Alexandre Serras [97505], Gonçalo Leal [98008], João Reis [98474], Pedro Duarte [97673]

v2022-06-22

Index

1 Introduction

1.1 Overview of the project

This project aims to create an online shop and delivery system for Coviran. To add some abstraction we will create two Spring Boot APIs: a generic one and another one Coviran specific. In order to achieve a quality result, we will write tests for our code (implementing Test Driven Deployment) and implement a CI/CD flow.

With our solution, we want to offer a system capable of allowing users to order their groceries online and have them delivered at the location of their choice. Our system should be a good task assignment platform, assigning couriers to an order efficiently.

1.2 Limitations

Not every planned feature was developed due to time limitations but there are many examples which we would still like to mention.

Asynchrounous Deliveries' Notification

In the Couriers' platform, the page currently requests deliveries to a REST API. While this approach allowed us to reduce complexity of our implementation, we originally intended to provide an websocket endpoint so that a Courier's page would update automatically whenever a new delivery is put into queue.

This is also the case in the online shop, where a client should be able to track its delivery state in real time, and in the Businesses' platform, where a shop employee should see new orders as soon as possible so as to prepare it for being picked by the Courier.

Shops' Admin/Staff Features

Currently, the data in the shop needs to be inserted either by the Flyway migrations or the REST API. A dashboard should be created so as to facilitate the management of the shop.

Better Support For Mobile Devices

While our webpages are usable on small screens, fully integrated options should be thought so as to provide better usability.

Our webapps already take a step into that world, being single-page applications developed in Angular, which already gives the user a more fluent

experience. These pages could be easily transformed into an application using the React Native's Webview.

Assignment of Deliveries to Couriers

In our implementation a Courier is able to accept any available Delivery. While this may work for demonstration, in the real world, if the service grew in active users, it would be nearly unusable for the Couriers, as deliveries would be constantly generated and accepted by other couriers.

This could be solved by limiting the amount of deliveries that a Courier is notified about, but would definitely be a complex task as a Courier or Delivery could easily be excluded from the algorithm.

2 Product concept

2.1 Vision statement

With the exponential advance of technology, one of the problems of super/hypermarkets today is the need to expand their business into the digital market. One of these companies is *Coviran*, a Spanish supermarket chain, also spread throughout Portugal, that doesn't have this digital side in their business, yet.

Therefore, the goal of this product is to innovate the digital market of *Coviran* supermarkets, providing them with a service to deliver their products to their customers' homes.

The super/hypermarkets companies already have applications with similar goals, for example, the portuguese hypermarkets chain, *Pingo Doce*, provides a mobile application with the goal of listing all products and promotions/discounts available in their markets. The existing application that most closely resembles the purpose of this product is *Continente Online*, another big Portuguese hypermarkets chain, where it is possible, not only to list all the products available and on sale, but also the delivery service is available.

2.2 Personas

2.2.1 Lucília Santos

Lucília is a 74 year old retired woman with reduced mobility from Aveiro, in Portugal. She is a widow, has 1 adult son and uses a cane to help her walk.

She finds it hard enough to go to the supermarket due to her reduced mobility. So, her son usually does her shopping every two weeks at Coviran, the nearest supermarket. She always has the same list of products.

MOTIVATION: Lucilia doesn't want to keep asking her son to go shopping for her. She wants to be independent in that aspect and not bother her son anymore.

2.2.2 André Vaz

André is a 39 year old manager of the Coviran's supermarket in Aveiro, in Portugal. He is married and has one son, one daughter and one dog.

He's a very active and focused person not only in his job, but also in his personal life. He never stops! If he isn't working, he is doing sports or spending quality time with his family.

MOTIVATION: As manager, André wants to control all delivery's employees, check if everything is working fine and analyze some statistics.

2.2.3 Duarte Henriques

Duarte is a 19 year old student from Faro. Last year, he finished High School, and now he is studying Public Administration at Universidade de Coimbra. He is a person very independent and focused on his future. He has a small van and a rented room in Coimbra.

As an independent student, he wants to start saving money to help his parents pay for university, but he knows that it is not easy to combine studies and work. So, he is looking for a simple job that does not require a lot of resources, only his van and his computer/phone.

MOTIVATION: Duarte wants a temporary job that doesn't need much resources.

2.3 Main scenarios

- Lucília needs some food for her lunch tomorrow, but her son is working. As the list of products is always the same, she wants to save it to order every week.

    - opens the app, logs in, creates a new product list, searches for the products he needs, chooses the quantity of each one, adds them to the list, chooses the list that she wants to order, selects the time we want to receive the order, and finally orders her products.
- Duarte is working for Coviran right now. He is a courier.
    - opens the app, logs in as a courier, receive a delivery order, accept or deny it:
        - If he accepts it, he will receive the list of products and the client's address. In the end, he marks that order as done, and he is available again for another delivery.
        - If he denies it, he waits for another delivery order.
- At the end of each working day, André will check the delivery statistics of his store.
    - opens the app, logs in as a manager, selects his supermarket and checks the delivery statistics on that day.
- Lucília is finishing her lunch and realizes that the dishwashing liquid is empty.
    - chooses the category of the wanted product (in this case, personal hygiene), chooses the product, adds it to the cart, and orders the product.

2.4 Project epics and priorities

Week 1

1 - Configure Specific Backend App DataModel

2 - Configure Generic Backend App DataModel

3 - Create Prototype of Web Apps

4 - WebApplication Docker Setup

Week 2

1 - (Epic) All three web applications can create an account and log in with JWT

2 - (Epic) In the store application, a client can add products to the cart

3 - Setup Foreseen Pom Dependencies (such as HTTP Client, JSON Utils)

Week 3

1 - (Epic) The web application client has access to a functional shop page.

2 - (Epic) The business can access the deliveries made to their shops.

3 - (Epic) Riders can see the list of queued deliveries.

Week 4

1 - (Epic) Allow Couriers to apply to a business and refuse work from it.

2 - (Epic) Allow Businesses to accept a courier as well as block them.

Week 5

1- (Epic) Integration of the generic service in the shop (create and cancel order and track its state).

3 - (Epic) The web application for couriers is fully functional.

4 - (Epic) The web application for businesses is fully functional.

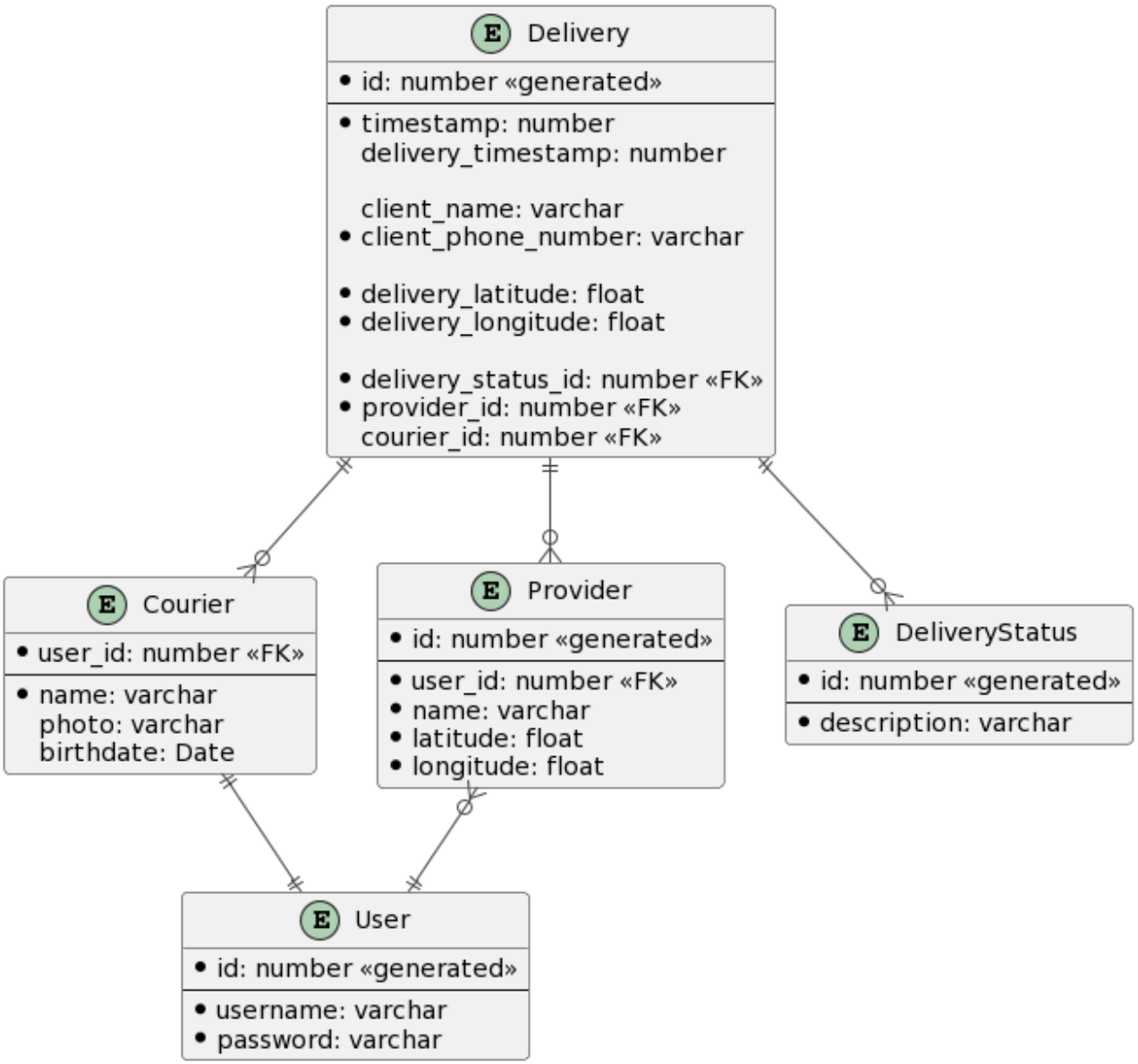
5 - (Epic) CI/CD pipeline is fully functional.

3 Domain model

As can be seen from the DB ER of delivery system image, the delivery system will have its database as generic as possible and where it is only interested in having what is extremely essential for a delivery service, that is, the order, who made the order, the status of the order and the courier responsible for this order, which is why there are only 4 tables, namely: Delivery, DeliveryStatus, Provider and Courier.

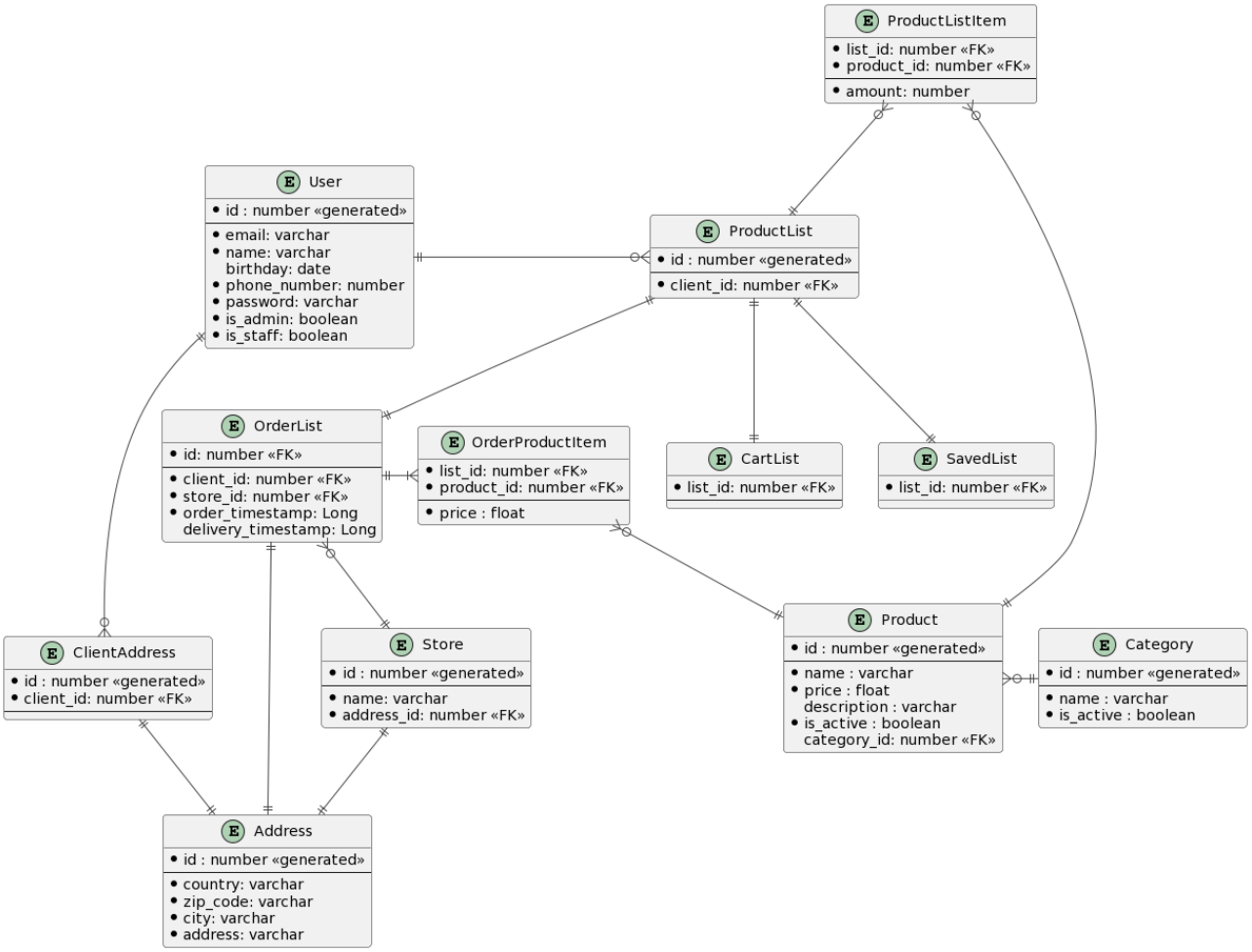
Regarding particular order information, none of this will be maintained here because it will differ from shop to store, and the fundamental concept is that this system may be used by other retailers, therefore the goal is to have the most generic delivery system possible.

For the database, it has been agreed that MySQL would be utilized.



DB ER of delivery system

Looking at figure Store DB ER you can see the data model and how everything interacts between each table selected for our shop, i.e. the individual app. A total of 12 tables were built to supply everything our business needs to work properly. MySQL will also be utilized for the database on this side.



Store DB ER

4 Architecture notebook

4.1 Key requirements and constraints

Delivery System requirements and restrictions:

- The delivery system is intended to be possible to integrate with any store, not only with the one that will be developed by us.
- To be a courier, you need to register in the application.
- As personal data are being handled, authentication of both couriers and administrators will be mandatory
- Platform for couriers to accept services, and where you will inform them of what they will have to transport and where
- Platform for admins to see system statistics
- As the API will be public, confidential system information cannot be passed in requests to the respective API

Store requirements and restrictions:

- Mandatory to use the Delivery System API to order deliveries
- Authentication to make purchases in the Store

4.2 Architectural view

As can be observed from the Generic App and the Specific App, the software solution built for the problem stated is separated into two categories.

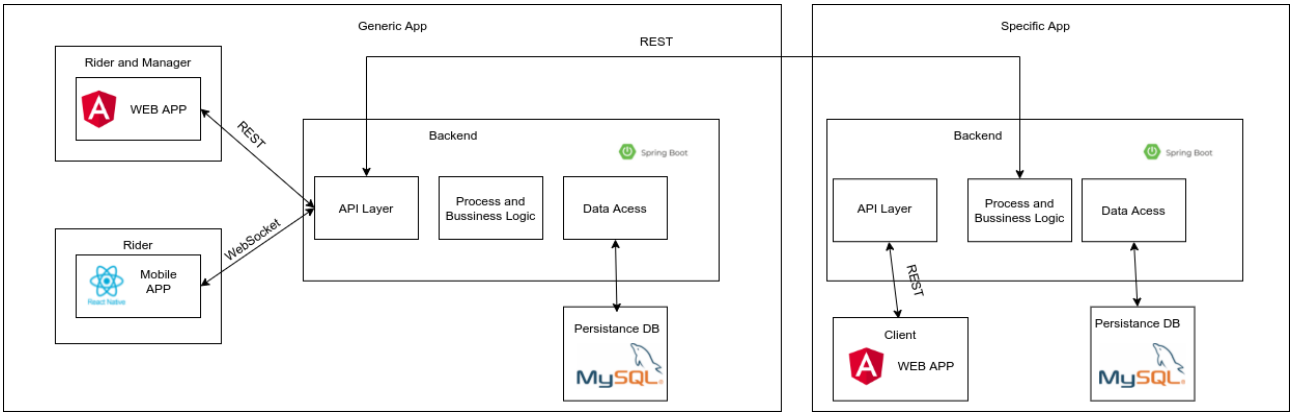
MySQL will always be used for databases, spring boot was the sole mandatory technology need for the backend, Angular will be used for Web Applications, and Angular was picked for the Messenger app.

All technologies were freely picked by the group; the choice is mostly due to the fact that the group members have a greater understanding of them.

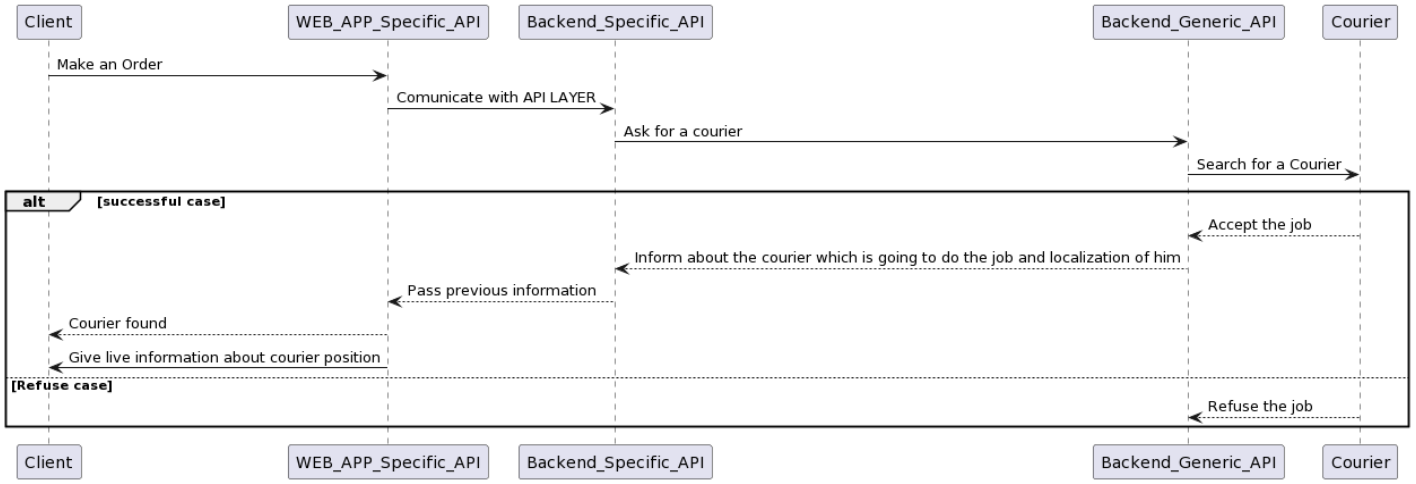
The processes that the system will require will be easy to comprehend thanks to the sequence diagrams; nonetheless, the primary issue will be connecting the specific shop to the general one, as the generic app is designed to be used by any retailer.

The backend will connect with the databases via the data access provided by this first layer, and the frontend will always interface with the backend via the API layer. Finally, the two apps will communicate via the backends of one and the other. another, and it appears that we are able to communicate effectively amongst the many components of the system in this manner.

The sequence diagram depicts the typical flow of how the system operates.



System architecture diagram



Sequence Diagram

4.3 Deployment architecture

Our code is managed through github, where the code is divided in different repositories which belong to the same organization. Using github we are able to implement our CI/CD flow, namely through github actions and github agents which allow running tests and validations and deploy our solution.

We have a development deployment to have the opportunity to validate if the result is the expected. This deployment is made in the same way as the production one.

The code will be then deployed through several docker containers.

5 API for developers

Both APIs used in the work are described in Swagger, making them easily accessible; it is crucial to emphasize that REST API best practices were followed.
API for a delivery platform (generic):

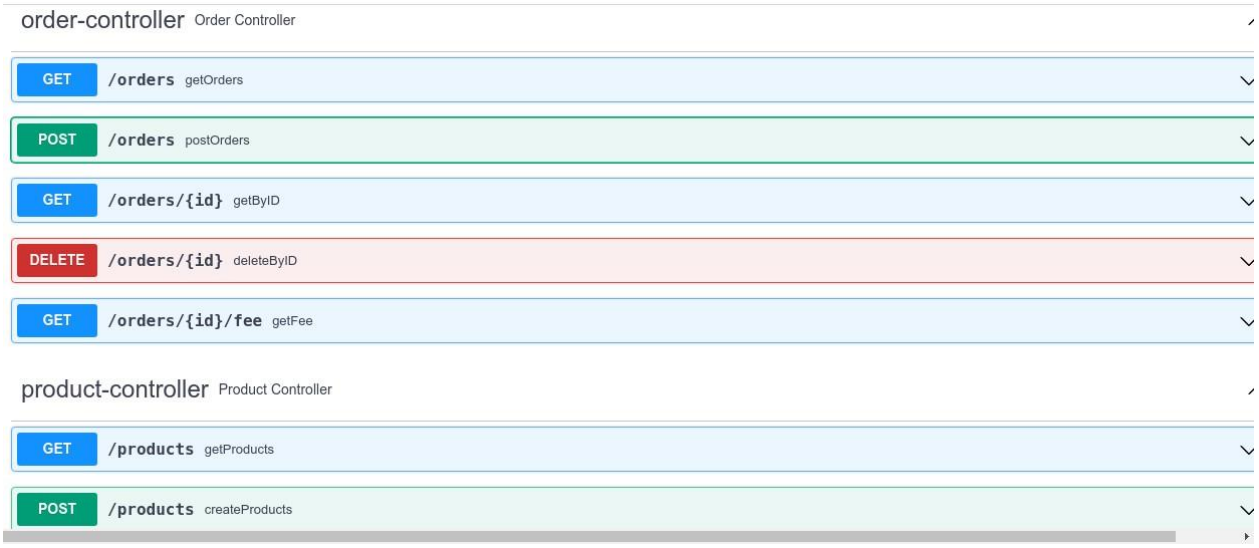
https://app.swaggerhub.com/apis/tqs-deliveries/tqs-deliveries-core/3.0.0#

This API includes delivery system functionality, allowing you to carry out any actions linked to a courier, deliveries, and business.
Any other company that wishes to incorporate our delivery system into their website may do so fast and easily; simply join up and begin sending delivery requests.

Small excerpt from the swagger for our Delivery platform API

Store API (specific) :

https://app.swaggerhub.com/apis/tqs-deliveries/tqs-deliveries-specific/2.0.0



6 References and resources

- https://www.baeldung.com/
- https://www.docker.com/
- https://www.section.io/engineering-education/containerizing-an-angular-app-featuring-nginx-web-server-using-docker/
- https://uapt33090-my.sharepoint.com/personal/ico_ua_pt/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fico%5Fua%5Fpt%2FDocuments%2FUA%2DDETI%2FClasses%2FTQS%2D202122%2D2S%2FTQS22%5FMaterials%2FeLearning%2DTQS22&ga=1