universidade de aveiro
deti departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Francisco Jesus [89084], João Marques [89234], Miguel Matos [89124], Tomás Costa [89016]
v2020-05-13

1 Introduction

1.1 Overview of the project

This objective of this project is creating an online marketplace. Since this project is for the TQS course it will require that we use what we learned on this course, so it will require that we focus our attention on creating tests for our application, so that we can end with a functional application that passes all the tests created.

Our application is called Renti. It is an online marketplace where users can rent products that other users put for renting.

1.2 Limitations

We were planning on doing a price system, where a product could have different prices based on the number of days of the rent. But considering some difficulties in doing it on the frontend and comparing our app to other similar app, we decided to use a contact system to decide the price when the number of days is bigger than 3.

Other feature was the use of notifications when a rent was made or when the status of a product changed. But due to some difficulties doing it was not implemented.

2 Product concept

2.1 Vision statement

Renti is an application made for users that want to rent products made available by other users. In this case vendors put a product for renting and other users can get that product for a limited amount of time. The price of the rent is based on the time that the client will use the product. The time is decided before the product is rented.

This app distinguishes from other similar apps, because of its renting nature. It is an app based only on renting products. So, with this app we intend to give users the opportunity to put their unused product for renting so that other users can give some use to them. And with this, clients can get products at a cheaper price and vendors can get larger profits from unused products at long-term.

2.2 Personas

Vendor: Jack Payne – Twitch Streamer
Jack is a twenty-eight-year-old twitch streamer, that plays video games on different consoles and streams those playthroughs.

Jack started is career as a streamer while he was studying Psychology in the University of Oxford. And during this time, he started getting more and more subscribers that would pay to watch his content. This combined with is growing disinterest for his course made him drop out of the university and start working full-time as a streamer.

During his career he bought various video game consoles, but some of them started to get outdated, which would get less use, and others just were not used much. So, Jack started thinking what could he do with those consoles that he would not use and were just occupying space in his house to get some money out of them.

MOTIVATION: Jack would like to have a way to use his unused consoles to get money, so that he can use that money to improve the quality of his streams.

Client: Elise Johnson – Internal Medicine Physician
Elise is a twenty-seven-year-old internal medicine physician. Besides her work, Elise likes to play video-games and watch movies at the cinema with her friends or family.

She is married and has a young daughter named Mary.

However, combining the time she needs to dedicate to her family and to her job, which requires a lot of her time, Elise does not have a lot of free time to dedicate to the things she likes to do, especially playing video-games, which was one of those things she could dedicate amore time when she was younger.

Elise has been considering buying a PlayStation 4, but considering the problem with her free time it could just end up being a waste of money.

MOTIVATION: Elise would like to have a way to get the products she needs to play video games without having to use to much money and without worrying that those products can end up be unused for a long time.

2.3 Main scenarios

Jack puts his PS4 for renting – Jack opens the application and clicks on the button for renting. After that, he sees a page with various fields for describing the product he wishes to publish. So, he starts taking various photos of his PS4 and puts them on the correspondent field. Next, he fills the remaining fields (name, description, price …) and finally publishes the PS4 for renting.

Elise wants to keep track of Jack's PS4 – Elise opens the application and sees a list of products. But since she knows that she will have some free time after two weeks, she decides that she wants to rent a PS4, but it is something that she is not completely sure. So she searches for PS4s in the search box and sees a list with PS4s with different prices and locations, In the end she chooses Jack's PS4. So she clicks this product and sees a page with more information and after being sure that this is the product that she wants, she adds it to her favourites.

Elise wants to rent Jack's PS4 – Elise after confirming that she will have some free time, she decides to rent Jack's PS4. So, she opens the application and checks that she did not receive any notifications from that product being rented by someone else. She goes to her favourites page and clicks on Jack's PS4 and clicks the button to rent the product and gives the necessary information for the delivery and the time of the renting and finally pays for the product.

2.4 Project epics and priorities

Epic: Base functionalities for client side

- See list of products
- Search products by name
- Purchase product

Epic: Base Vendor functionalities

- Put a product for renting
- List of products the vendor put for renting

Epic: Advanced User functionalities

- Search for product using different parameters
- Order product list based on some parameters
- Select amount of time for rent and get price actualized

Epic: Other lists and advanced functionality for vendor

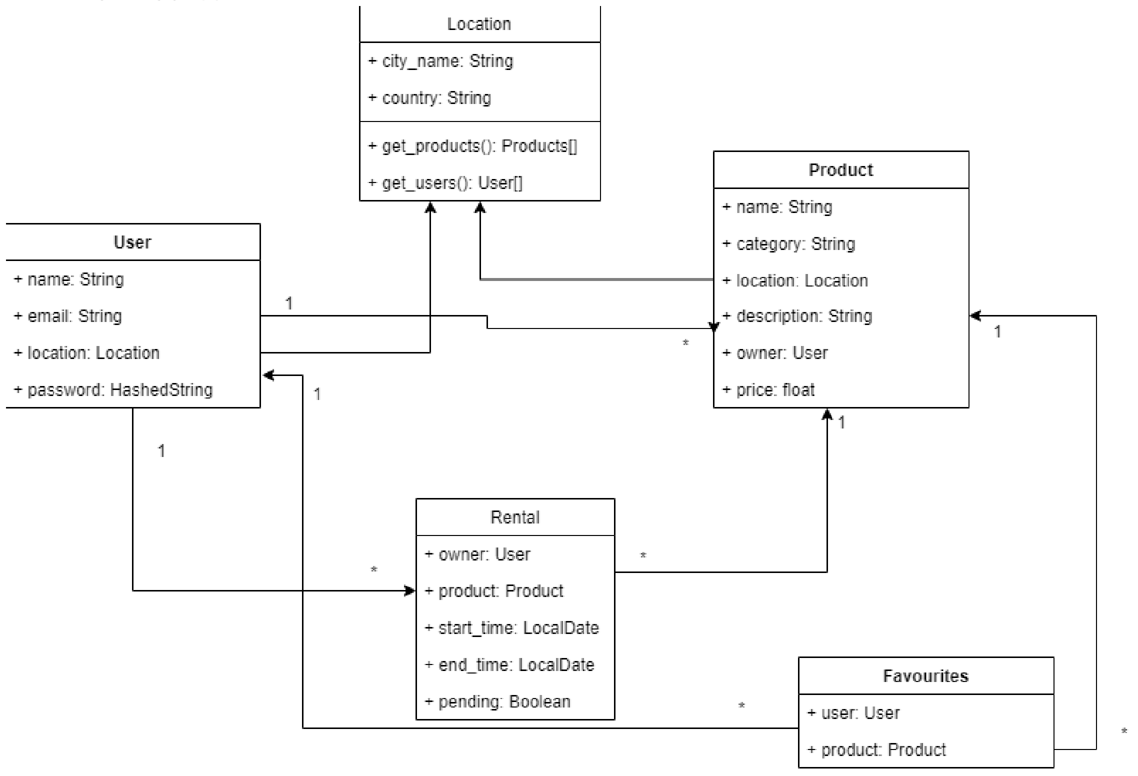
- User favourites list
- Vendor products for renting list

Epic: Notifications (not implemented)

- User favourites update notifications
- Vendor products rent notification
- User/Vendor list of notifications

3 Domain model

<which information concepts will be managed in this domain? How are they related?>
<use a logical model (UML classes) to explain the concepts of the domain and their attributes>



4 Architecture notebook

4.1 Key requirements and constrains

Given our project concept, we will need to establish an architecture that assures high availability and response, as well as preserving an ease of maintenance and support.

We also need to implement a dynamically scalable system, as we expect different intensities of use through time.

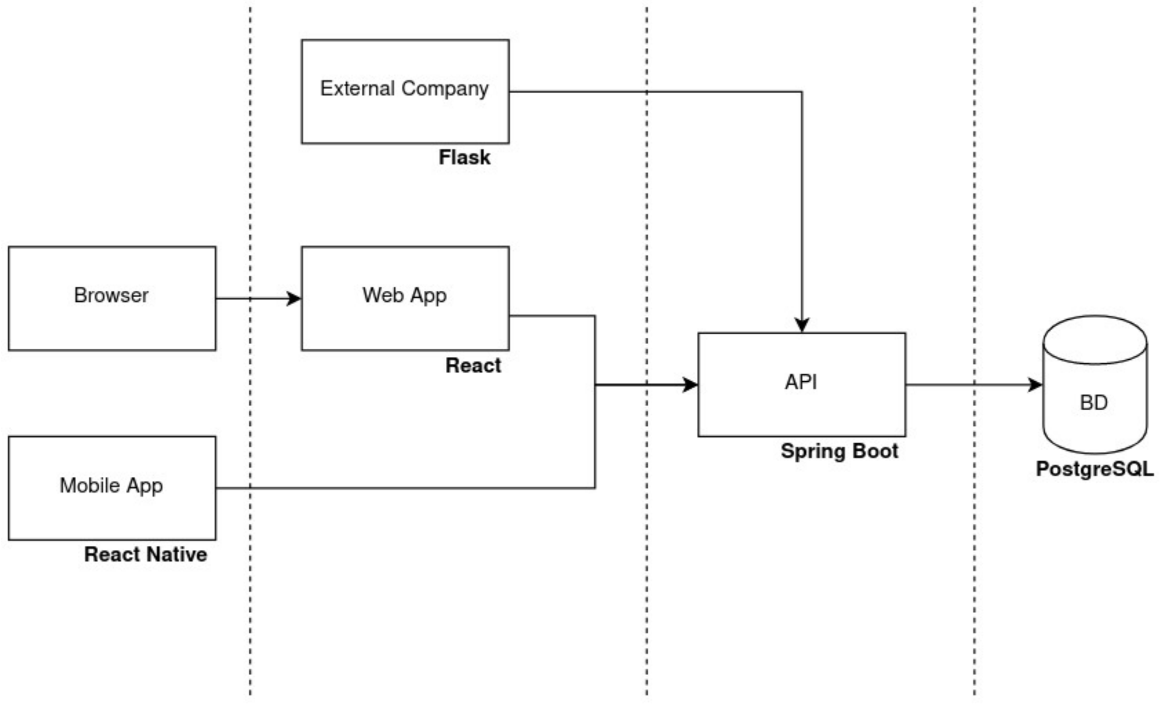
On the other hand, we also need to ensure security, privacy and reliability.

4.2 Architetural view

Addressing the needs explained before, we opted by the use of docker as our main system packaging and management. Besides presenting major advantages during development, this will allow us to have an easily scalable solution, as well as a secure system.

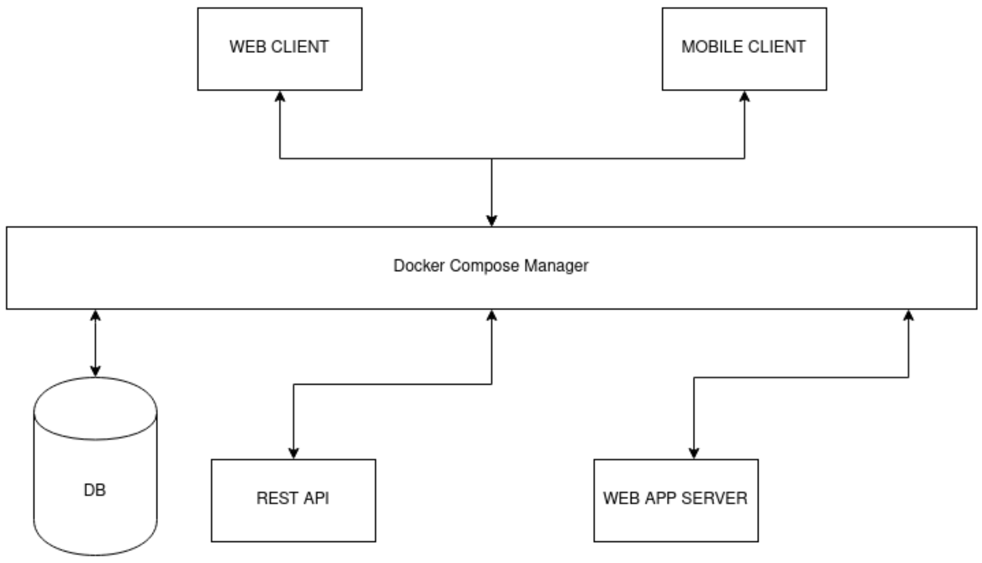
We plan to use **docker-compose** and **docker swarm**, should we need to scale and replicate our system.

The following diagram presents our architectural ideas.



4.3 Deployment architecture

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Modelar num diagrama de *deployment*]



This deployment diagram allows us to see our plan for deploying the solution. During the first iteration, we are planning on simply using **docker-compose** as a local, single-machine manager on our deployment server. In this case, docker-compose will manage the connections from the clients, and all the requests will be focused on the **web reverse proxy**. Should we need more, we will implement and deploy using **docker swarm**.

5 API for developers

The documentation for the API can be accessed from 192.168.160.62:8080/swagger-ui.html

From the API is possible to obtain the list of products, that can be filtered by name, category, location, price range and user; can be ordered by price and name. There are also operation for adding, deleting and updating a product.

There is also the option to obtain a list of users, a specific user and operation of PUT, DELETE AND POST.

There are endpoints used to get information about the rentals of a user, product and by approved state.

And finally, a endpoint to get the products a user favorited and the operations of POST and DELETE.