Product specification (Draft)

Version	Date	Author	Description
0.1	23 May 2021	Rodrigo Lima	Draft Document - placeholders and Qourier specifications

Contents

1. Introduction

This document has the goal to capture the main project requirements and considerations. It tries to go into detail about every aspect of the project.

Every change to the document must be consented by the consensus of the working project group.

2. Project Objectives

The project is composed of two sub projects. The first sub project is called Quorier and its being built with the goal of bridging the independent every day courier and big companies who need something to be delivered on a small scale.

The project will have four key deliverables:

1. Prototype project, focused on UX and story driven (26/5)
2. Core stories implementation and CI integration to backend (02/6)
3. API final implementation, CD pipeline integration and QA Manual final version (09/6)
4. Stable MVP, deployment of project on the cloud, final product specifications report (17/6)

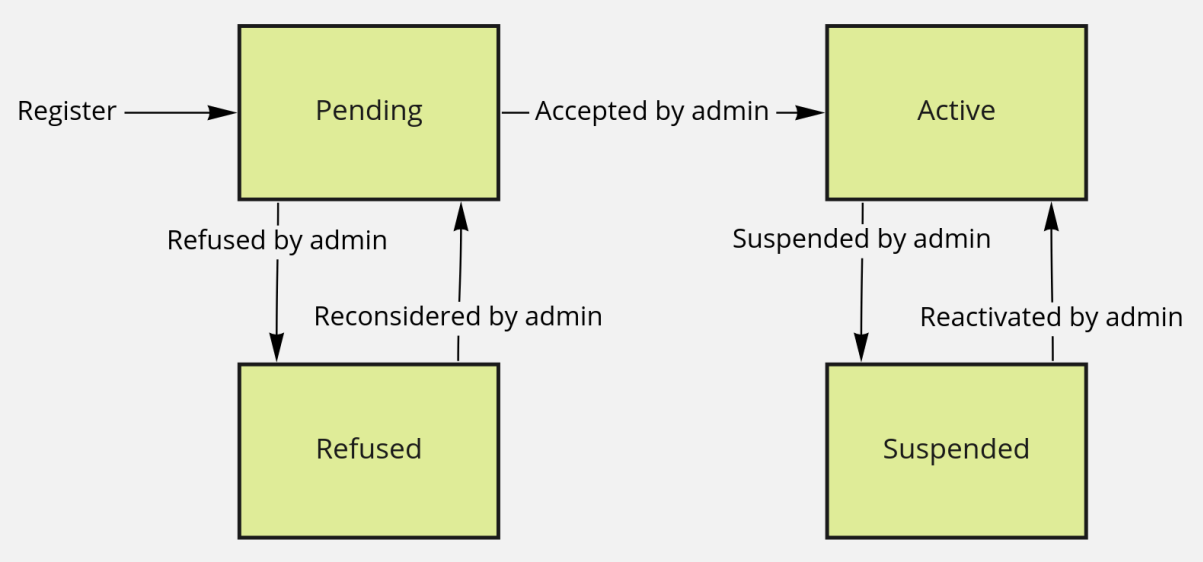
3. Success Criteria

TO DO

4. Functional Specification

The sub project Quorier will have the following functionalities:

- Register delivery jobs and manage status
- Check delivery progress
- Register account as rider or customer
- Log in into the account and be redirected to the right page, this will be done by checking the account role and status on the database
- Manage Riders and Customers account status:



- Check account details
- Check platform performance and API metrics
- Generate scores for riders with efficiency and customer rating into account
- Biding system will decide on who will deliver the package by first seing who is closer or if the position isn't given or in case of a stale mate it will be decided by the highest ranking, in case of the same ranking it will be random.

5. Technical Specification

The sub project Quorier will have the following technologies:

- Spring for website copntroller;
- Thymeleaf for website templating;
- Grafana for graph information;
- Rabbitmq for pub sub status information;
- MySQL for the database;
- Swagger for API documentation;
- Prometheus for metrics broker;
- Android application built in kotlin for the riders application;

6. Wireframes/storyboards

The sub project Quorier will have the following personas:

- Alberto (administrator)
- Diego (rider)
- Cristina (concrete delivery service)

Diego user stories:

- Bid a delivery job
  - Visual feedback with "activated" theming
  - Registered the Rider's bid in the database
- Accept a delivery job
  - Bid operations restricted
  - Visual feedback with "accepted" message
  - Registered as the Rider for the job in the database
  - Delivery job not up for bidding
- Confirm a delivery job
  - Registered the delivery job as done
- Check profile and stats
  - Is in profile page, with the details inputted at registration
  - Contains a section with statistics about, at least, the number of deliveries done and average time spent

Cristina user stories:

- Register delivery
  - Delivery up for bidding
- Check progress of deliveries
  - Contains a table with all deliveries requested, in progress and completed
  - Deliveries in progress contain a progress bar

- Check profile and stats
  - Is in profile page, with the details inputted at registration
  - Contains a section with statistics about, at least, the number of deliveries requested and time taken until delivery completion
  - Contains the API key for the operations, which is hidden by default

Alberto user stories:

- Check Riders' progress
  - Has a table with all riders that accepted a delivery and their progress, with a progress bar
- Check each Rider's performance
  - The Rider has a section with statistics, such as the number of deliveries done and average time spent
- Check each Rider's profile
  - The Rider has a profile with all details inputted in their registration
- Check each Customer's stats
  - The Customer has a section with statistics, such as the number of deliveries requested and time taken until delivery completion
- Check each Customer's profile
  - The Customer has a profile with all details inputted in their registration
- Accept/Refuse Rider/Customer applications
  - There is an Applications page with two tables, one for Riders and another for Customers, where each row is a concrete application
  - Clicking on an application summons a modal with details of the application, with two buttons from which the application can be accepted/refused
  - Contains per-table filtering options for, at least, Accepted or Refused applications. Only accepted applications are shown by default
- Suspend/Reactivate accounts of Riders/Customers
  - Within the Rider/Customer profile, there should be a button to suspend/activate their account

    - Suspending an account disables all operations from the account
    - Activating an account enables all operations from the account
- Check platform performance and number of API requests
    - Check platform performance metrics in graphs
    - Check the rate of API requests in graphs

7. Testing

TODO

8. Updates and Maintenance

TODO