# Software Architecture Design Document

## Donuts on Demand (DoD) Project

Software Architecture Design Class 2021

*Team members*:

Titouan Le Mao - st122014

Cedric Le Mercier - st122004

# Table of contents

# 1. Introduction

The purpose of this document is to give an overview of the software architecture design of our Donuts on Demand (DoD) project. Our system will focus on four main quality attributes which are: usability, availability, maintainability and testability. We chose those aspects in order of what we think will be important for our application.

Hence, this document will talk about the background of our project, and the functional requirements of our architecture design. We will illustrate the attributes mentioned above through a set of scenarios or use cases our users will go through, then introduce the big picture of our system architecture, followed by UML diagrams and a couple views into the system. Finally, we will address acknowledgements, references and appendices.

The goal of this software architecture design document is to provide the development team with a high-level overview of our system architecture. Additionally, it is for the project sponsors to conclude on the proposed architecture before going into a more detailed design.

# 2. Background

Doughnuts or donuts are made from fried dough confectionery, either in homes or purchase from outlets, and it is popular among kids and youngsters. In 2016, the market of doughnuts reached 8 billion USD in SouthEast Asia which makes the second largest region after North America [1]. There are different varieties of doughnuts around the world such as glazed, chocolate, fashioned, croissant, jelly, etc. But what could be more infuriating than to go to a store to buy something, and then have to leave empty-handed due to a lack of stock?

The purpose of this project is to set up an innovative service for our customers, at the crossroads of the e- Shop and the drive-in. This service, called "Donuts on Demand » (DoD), will allow customers to make their selection from our range of donuts, place their order and specify the date and time they will pick them up. The same services already exists for custom-made cupcakes, and in this project, we aim to produce something similar for doughnuts.

Ordering on Donuts on Demand ensures that you always get your favorite cookie recipe, right out of the oven, on time. No more queues in the store, no more unpleasant surprises such as chocolate being out of stock. Donuts are always fresh, available in less than two hours and recovered while still hot for even more gourmet family moments.

# 3. Stakeholders

The main stakeholders of Donut On Demand are employees of the company and customers. The application will henceforth support 4 main types of users :

**Customer**

The customer is our consumer and who will be mainly browsing a variety of doughnuts on our website, placing doughnuts orders and picking them up.

**Manager**

The manager is the one who will create, manage and sell donuts. This person can be the donut shop owner, or a doughnuts enthusiast who will be creating recipes and making handmade donuts. He / she will be tracking sales statistics, set store opening hours, etc. Eventually, it will be the person or the person representing the company who we will sell the application to.

**Order Supervisor**

The order supervisor refers to anyone who will be managing orders and the pick up of orders. This person can be the manager of a single store outlet for larger companies.

**Cashier**

The cashier is the person facing customers and who validates orders withdrawals, updating sales and such. The requirements for this stakeholder are the lowest.

# 4. Functional Requirements

It is possible for anyone to create an order, choose their recovery store and appointment date and then pay the amount due by credit card. An order consists of a set of "donuts" and the desired quantity, among the pre-existing recipes for which the establishment is famous (for example, the "Sooo Chocolate", the "Dark Temptation", the "Krispy Kreme", ...).

Each donut recipe is defined by its type of dough, an optional flavor, up to three toppings, a type of mix, and finally a type of cooking.

- Dough: Plain, Chocolate, Peanut butter, Oatmeal
- Flavour: Vanilla, Cinnamon, Chili
- Topping: White chocolate, Milk chocolate, M&M'sTM, Reese's buttercup
- Mix: Mixed or Topped
- Cooking: Crunchy or Chewy

Customers must present the order form (each recipe has a given price) obtained from the system to retrieve their order in the store. A new recipe can be added at any time, and some recipes may disappear because of lack of buyers.

Store managers can define their opening hours. They can also add new employees in their shop. The system should allow the display of statistics on CoD usage (recovery time). The store must verify that it is able to ensure an order, even a personalized one.

# 5. Quality Attributes

## 5.1. Usability (H)

One of our highest quality attributes is usability. In the case of our Donuts on Demand project, this refers to how intuitive the interface is, how easy it is to learn and to use. This is especially important for our customer's first use of the application, as they need to figure out how it works and eventually convert first time visitors into loyal customers.

- ➢ The app should be compatible with the most popular browsers (Chrome, Firefox, Safari and Internet Explorer)
- ➢ The app should be responsive to any device
- ➢ The app has to be easy to use and has to only require a few seconds to understand what it does, especially for customers

## 5.2. Performance (H)

To support our usability requirements, our app has to satisfy performance requirements as well. Doughnuts are quick snacks that customers can have on the go and so our application performance has to follow the same logic with blazing fast performance.

- ➢ The process from registration to completing an order should take less than 5 steps (or screens)
- ➢ The application should be able to handle a high traffic load
- ➢ Response time for any action should be less than 2s

## 5.3. Security (M)

Our architecture will incorporate a login function which requires personal information to be stored securely, as well as a way to differentiate between user types, their permissions and what they have access to. At the moment, our system will not incorporate payment since payments will be done in-store and because of lack of time available.

- ➢ Customers personal information should be stored securely

## 5.4. Availability (M)

As mentioned before, doughnuts should be available to order anytime on the go, especially during peak times such as late afternoons when kids finish school, after classes or after work. It should be ready to handle spikes of traffic during peak hours and not fail.

## 5.5. Scalability (L)

Our architecture needs to be able to expand to further doughnuts customisations beyond what we currently have (Topping, Dough, Flavour, Mix, Cooking) if needed. The architecture needs to be able to incorporate a payment system at a further stage, for customers who wish to pay with a credit card and to ensure orders can’t be cancelled after the products have been made.

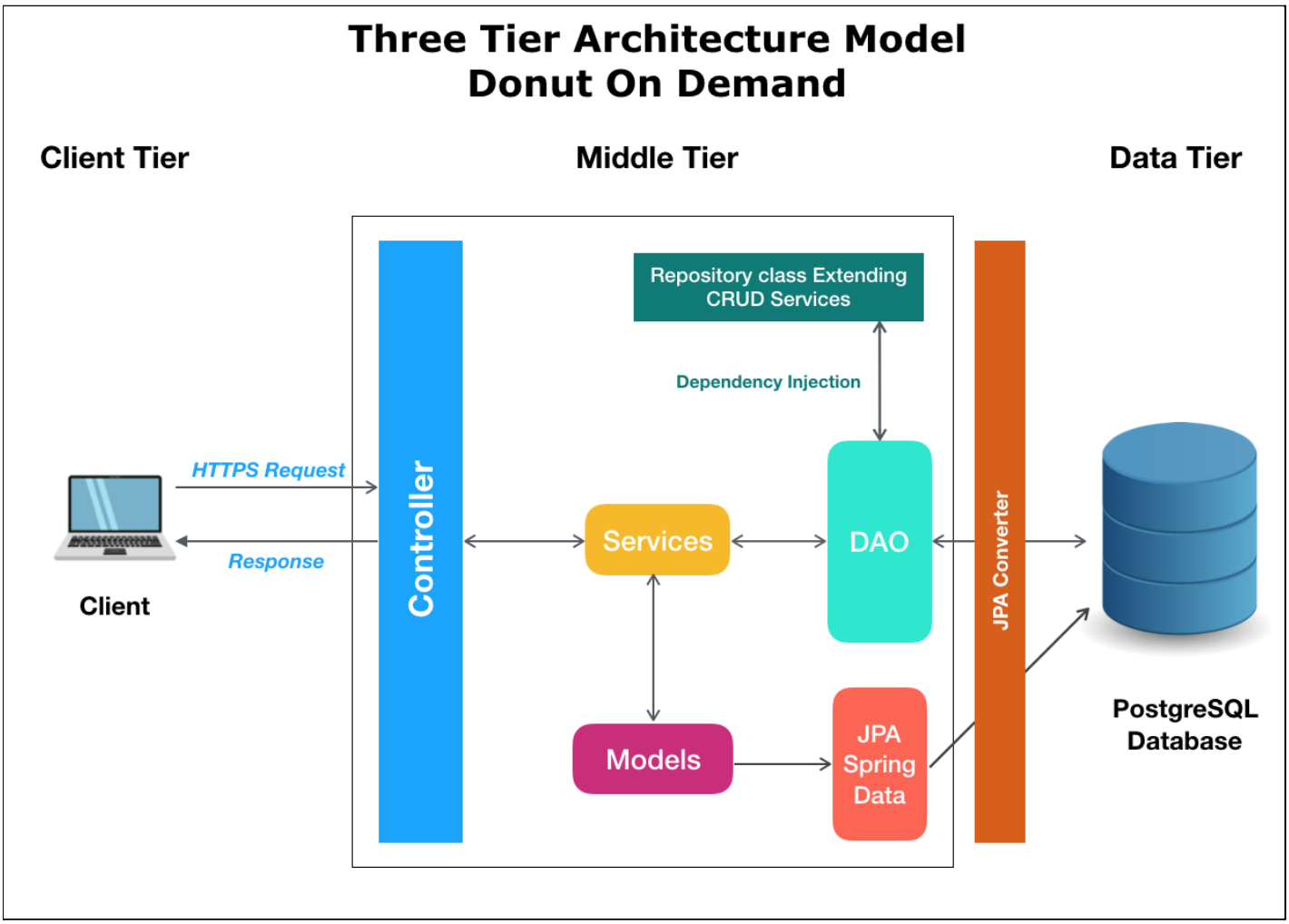
# 6. Data

For our application, we need to store the data in a persistent and structured way in order to have access to it even if the server stops. This is why we chose to use PostgreSQL which is a relational and object database management system. We chose it because we had already used this tool before and it met our needs perfectly.

We will keep all the data indefinitely because the different orders made by the customers will allow us to make statistics and analysis to know for example which donut is the most successful.

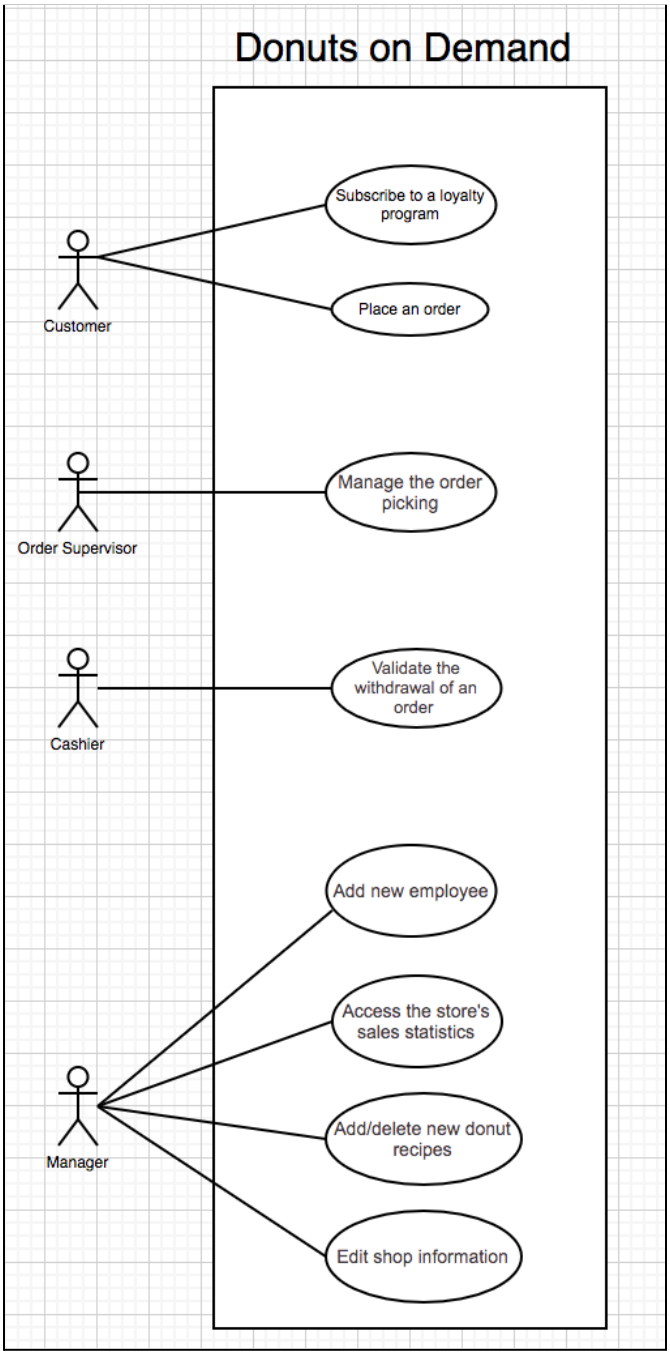
# 7. Architecture Overview

## 7.1. Big Picture

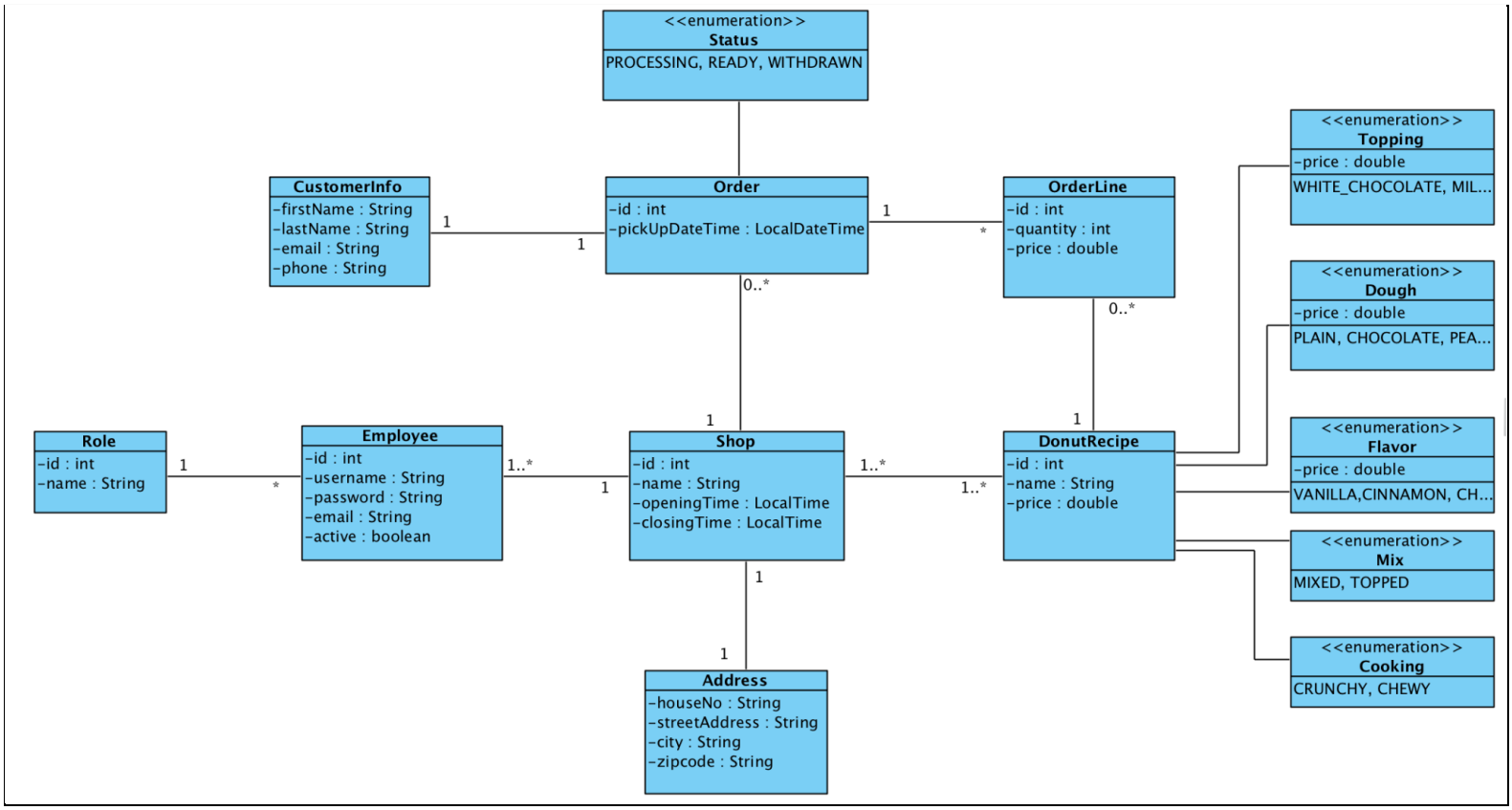


## 7.2. Diagrams

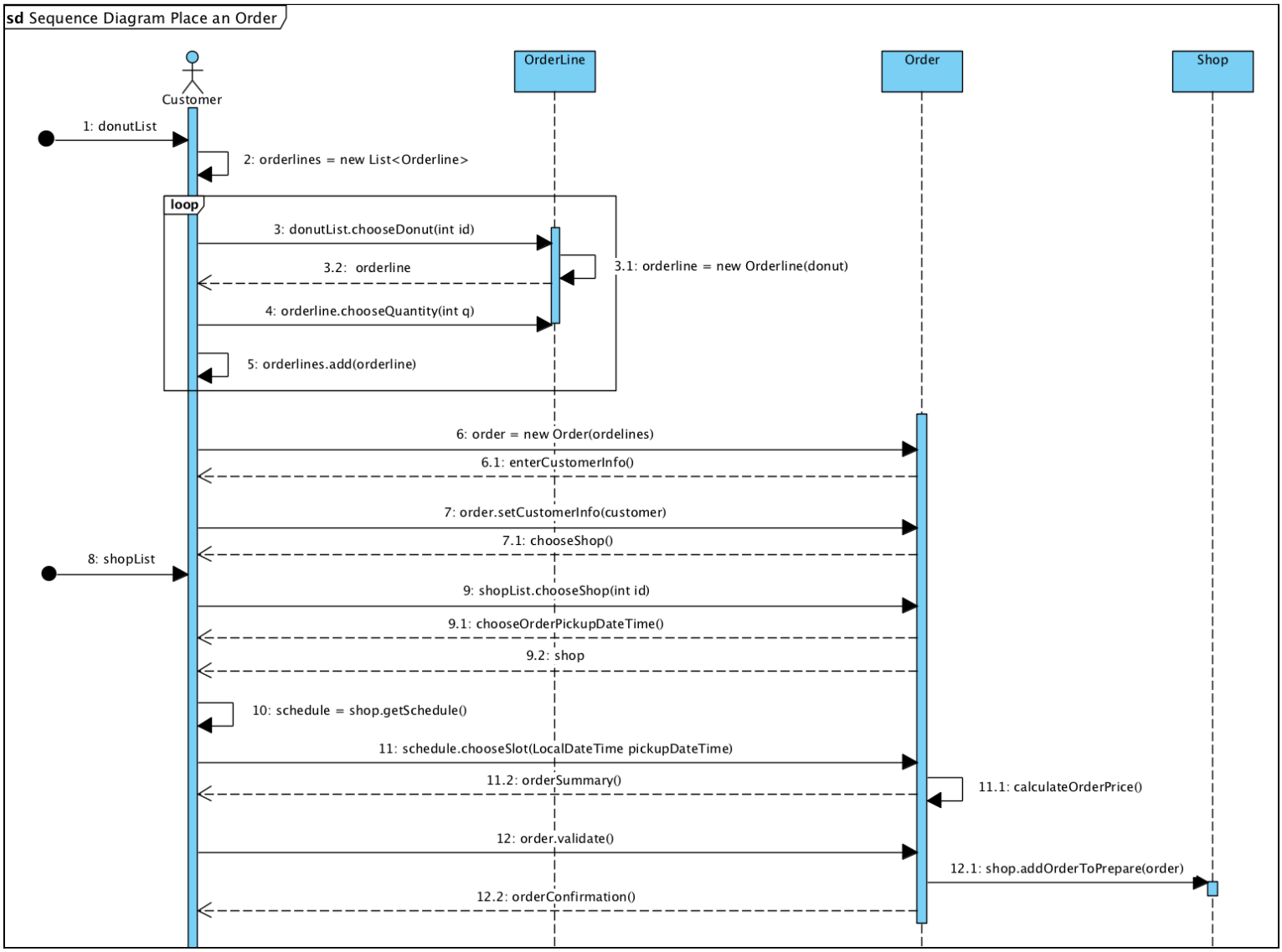
### Use Case Diagram



Class Diagram



## Sequence Diagram



# 8. Risk and Mitigation Plan

| Risk # | Cause | Effect | Likelihood | Consequences | Risk Ranking | Mitigations strategies | Owner |
|---|---|---|---|---|---|---|---|
| 1 | Delays in work due to competing work for the dev team | Delays in overall progress in development | Likely | Major | Very High | 1/ Devise appropriate schedule for work<br>2/ Make proper breakdown of the tasks and assign equally among dev team | Developpement team |
| 2 | Underestimating the time and difficulty of a task | Delays in overall progress in development | Possible | Major | High | 1/ Break down tasks as finely as possible in order to focus on one feature<br>2/ Use tools to estimate the difficulty and usefulness of the feature | Developpement team |
| 3 | Little experience in programming with Spring boot | Delays in overall progress in development | Certain | Major | High | 1/ Training with the different labs<br>2/ Search for solutions on the internet (Stackoverflow) | Developpement team |
| 4 | Announcing the achievement of too many features in a too short time | The project will not be finished at the end | Possible | Crucial | Very High | 1/ Define the time required for each feature<br>2/ Good distribution of tasks in the development team<br>3/ Spread the realization of the features over several 2-week sprints | Developpement team |

# 9. What will not be implemented

Due to the short time we have to develop such an application, we decided to focus mainly on developing a clean and efficient architecture for our application.

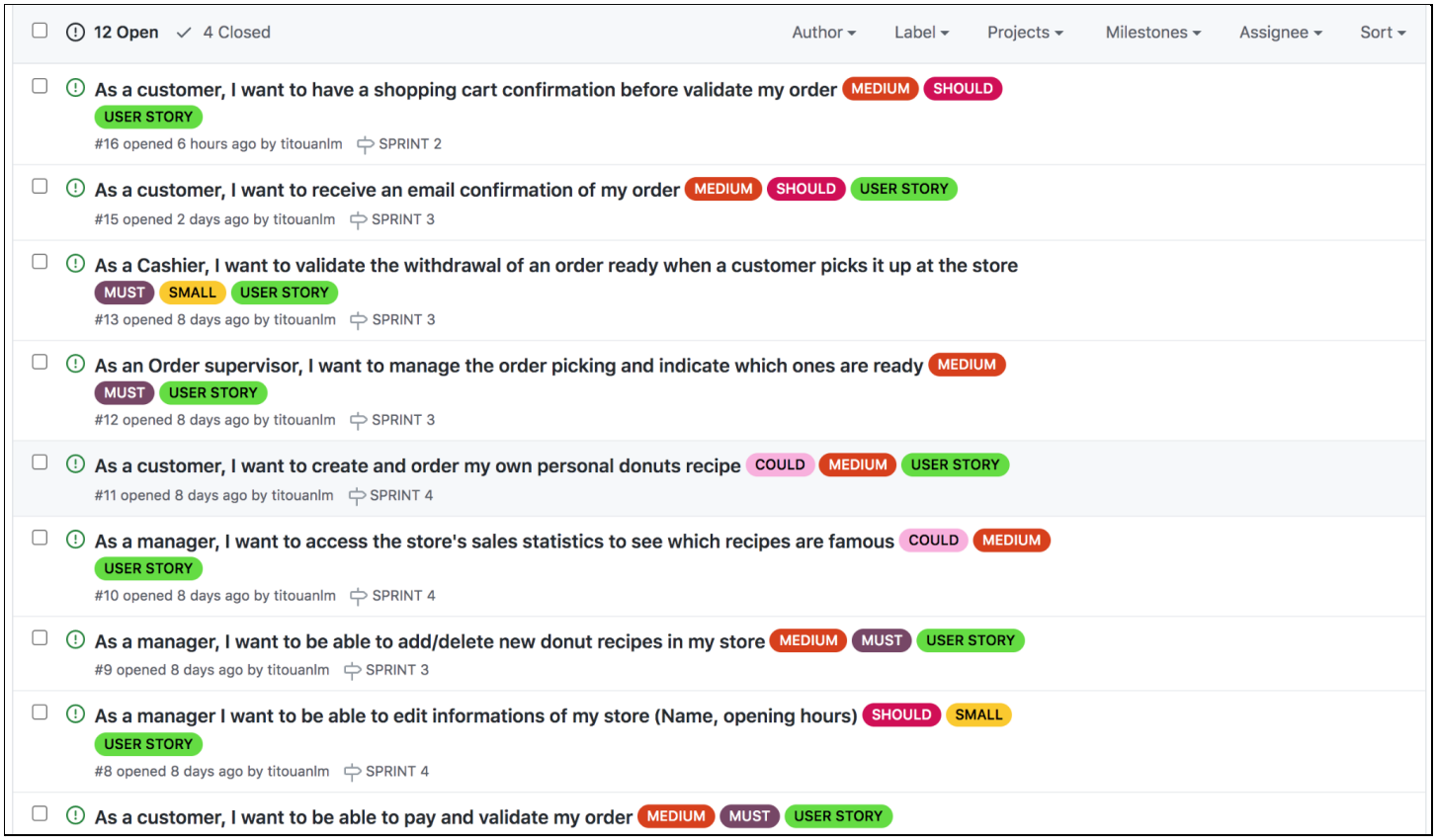
Therefore, we will not develop an ergonomic and beautiful user interface.

Furthermore, regarding the management of credit card payments, we unfortunately have no experience in developing such a feature and we believe that learning this will take too long to be finished in time. Thus, we concluded that this part of the application would not be implemented.

# 10. Tools

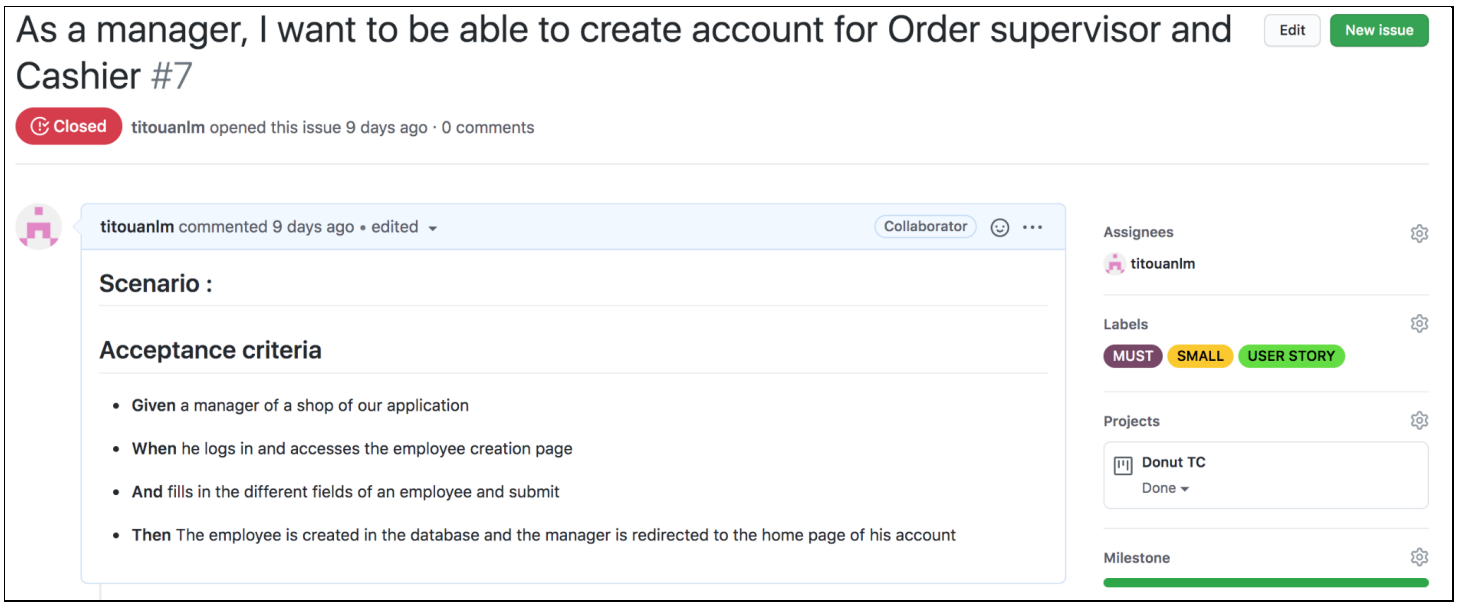
## Github Issues and Labels

In order to better organize and visualize the different functionalities we have to realize, we used the tool Issues in github. This one allowed us to define more precisely each requirement. Moreover, the labels available on github allowed us to define several information about each feature. This allows us to have a quick overview of the feature because we defined for each issue, their priority (thanks to the MoSCoW method) as well as their difficulty of realization (using a tee shirt sizing technic) S is equivalent to SMALL, M is MEDIUM and L is LARGE, which means that the feature is consistent and quite difficult to realize.



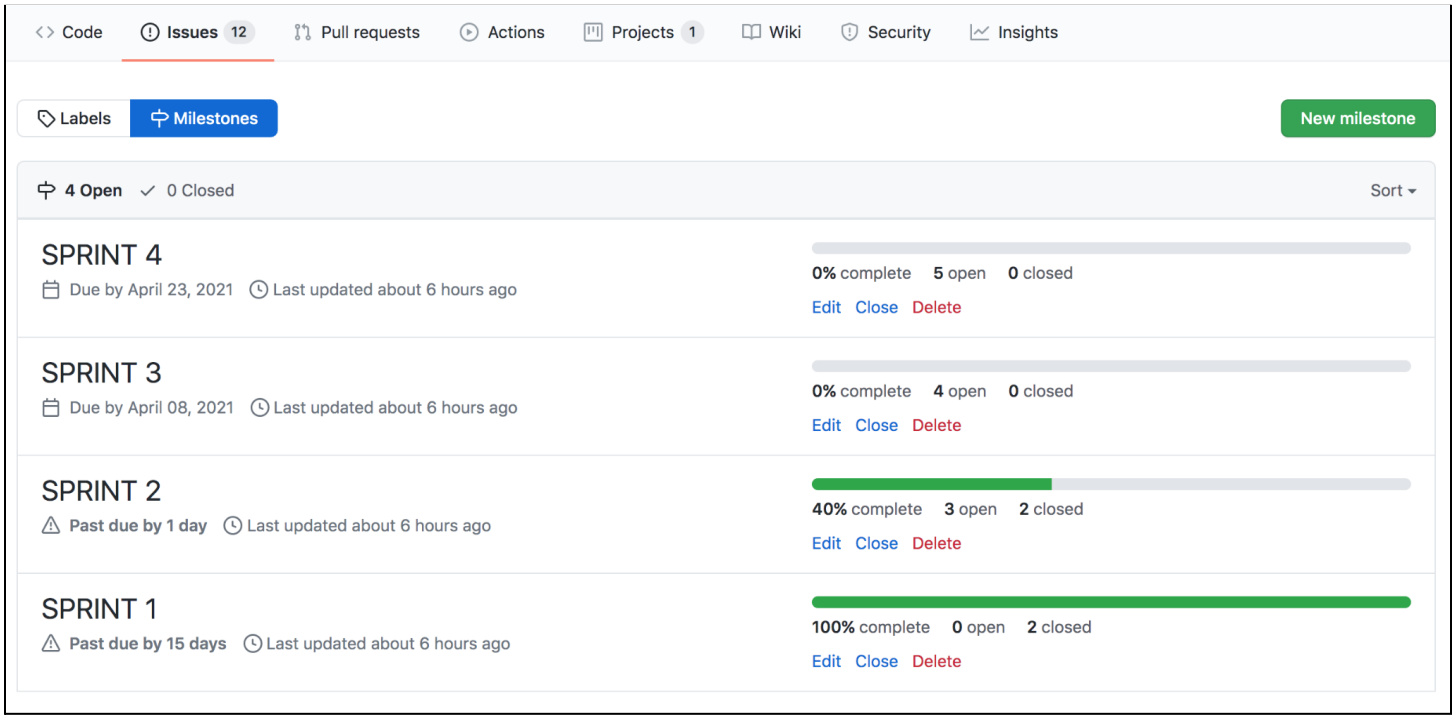
## Acceptance criteria

Each issue may also contain a description and other information. Thus, we used it to define the criteria that allow us to affirm that the functionality is correctly realized.



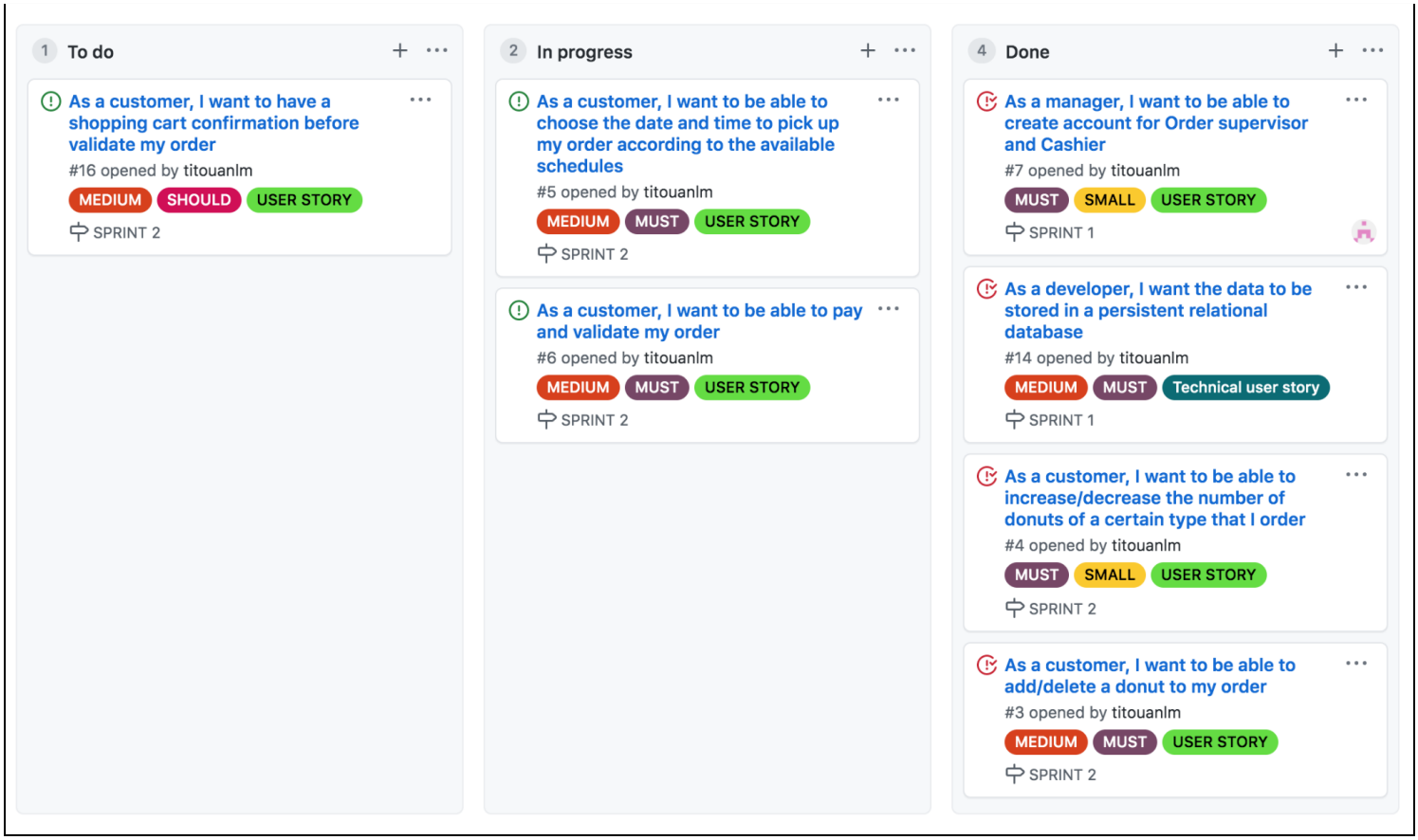
## Milestones

Github also allows you to group several issues into a single milestone and indicate the deadline. So we used this feature to define our different sprint and to constantly see the progress of them. Indeed, each time we validate the implementation of a feature, we close the issue in order to see only the one that remains to be done



## Kanban project

Finally, in order to manage our project and to organize ourselves in an optimal way in the distribution of the tasks, we used the Project functionality of github which allows to define a Kanban project with the various informed issues. It allows us to see clearly who does what, what is done and what is left to do.



# 11. References

[1] Goldstein Market Intelligence, 2020, Global Doughnuts Market Report 2030, *On the basis of products, On the basis of types, By Region With COVID-19 impact | Forecast Period 2017-2030.*