POLITECNICO
MILANO 1863

Politecnico di Milano

Software Engineering 2 Project
a.y. 2020-21

Customers Line-up
Requirements Analysis and Specifications Document

Version 0.0

Banfi Stefano Alessandro, 853195
Bresciani Matteo, 944638

Referent professor: Di Nitto Elisabetta

November 1, 2020

Contents

Chapter 1

Introduction

1.1 Purpose

The main scope of this document is to define requirements for the application development, in order to make a correct project planification. To do this, we will analyze:

- System;
- Functional and unfunctional requirements;
- Constraints;
- Relationships between stakeholders;
- Possible scenarios and tests;

These will be shown using different types of languages, starting from the natural language to the structured languages such as Alloy and UML. Below, we will define the context in which our application will be developed. During Sars-Cov-2 emergency, several countries imposed the lockdown in order to hinter the virus diffusion. People had to change their habits, in fact they could go out only for necessary needs, such as going to the supermarket or pharmacy. A lot of rules were introduced: not only using the masks or clean your hands but also keeping a social distance. For instance people must pay attention while they're entering in the market due to the long queue, which could increase the possibility of virus diffusion. This fact obliged people to stay for a long time standing up and waiting their turn losing a lots of time.

1.2 Scope

The aim of the project is to develop an application which, thanks to an intuitive interface, will avoid a long waiting outside the market. The application will provide a QR code as a virtual ticket on your own smartphone and to wait easily at home or everywhere instead of doing the physical queue for minutes or hours. In addition it will provide the position in queue and the time estimation of your turn. The system will generate a QR code for the authentication at the entry, in order to verify the position in the queue. Moreover, the application gives the possibility to "book a visit" for who indicates the approximate expected duration of the visit and how many products they'll buy.

Although the reservation is made entirely with your smartphone, the market gives anyway the possibilities to provide the ticket locally on site, acting as proxies for the customers. This is mostly for aged people who are technologically backward or has no smartphone. Once this procedure is completed, a SMS will be sent to the customers for noticing their turn. [todo]

In order to achieve the goal, we define the World, Machine and Shared Phenomena, which are really crucial to describe the system. They're characterized by fact that the first and the second ones are not visible among them; so we need a link between them which i represented by the Shared phenomena.

1.2.1 World

It's the portion of system to be developed. The main *World Phenomena* are:

- Punctuality;
- Loss of connection;
- Respecting social distance;
- Shopping duration;
- Checking temperature;
- Respecting social distance;

1.2.2 Machine

It represents the portion of system to be developed. The main *Machine Phenomena* are:

- Average of shopping time;

1.2.3 Shared Phenomena

They're phenomena shared between the World and the Machine which are controlled or observed by the system. In this scenario mainly are:

- Book/delete a visit in the supermarket;
- Get a virtual ticket;
- Send a notice;
- Send a SMS;

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- *User*: individual who plan to shop in the market. He's registered in the market system with his personal data and he'll have its QRCode to enter in the market on his turn;
- *User non registered*: individual who plan to shop in the market. He's not registered in the market system, so he is not able to get his QRCode to stay virtually in the queue. So the store deals with managing his turn [TODO]
- *Booking*: it occurs when a User do the shopping. In addition Users can indicate the range of his purchases between *small, medium, large*, in order to allows the system having a better estimation of residence time;
- *Visit*: it occurs when a User do the shopping but, instead of the (simple) Visit, specifying more informations. In particular they have to specify the categories (or better the items) that are going to buy. Through this information the system can estimate the user's path due to (improving accuracy,) maximize the number of people allowed in the store;
- *Alternative booking*: reservations made by the users who are not able to book the visit with "the app". In this case [...];
- *Ticket Call*: it happens when the system allows n users entering in the market according to the order in the queue by calling their numbers;
- *Valid ticket*: the ticket is valid if and only if the system allows the user entering in the market;
- *Delayed ticket*: a ticket is delayed when, after the its ticket call, the user is not submit the QRCode in time (threshold of 1-2 minutes);
- *Cancelled ticket*: a ticket is cancelled when it's already delayed and passed too much time (threshold of 15-20 minutes);

- *BookingID*: string of n alphanumerical characters that is represented by the QRCode. If the user is not registered [...];
- *Mobilephone*: Elettronic device without CLup App;
- *Smartphone*: Elettronic devide with Clup App;

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 Revision history

1.5 Reference Documents

1.6 Document Structure

G.1 Avoid to generate a long queue (Max 10 people) G.2 Grant the possibility of respecting social distance

Chapter 2

Overall Description

2.1 Product perspective

The aim is to build a system that manage the users'reservations and grants the possibility of booking the place in queue without wasting time while they're waiting their own turn outside the market.

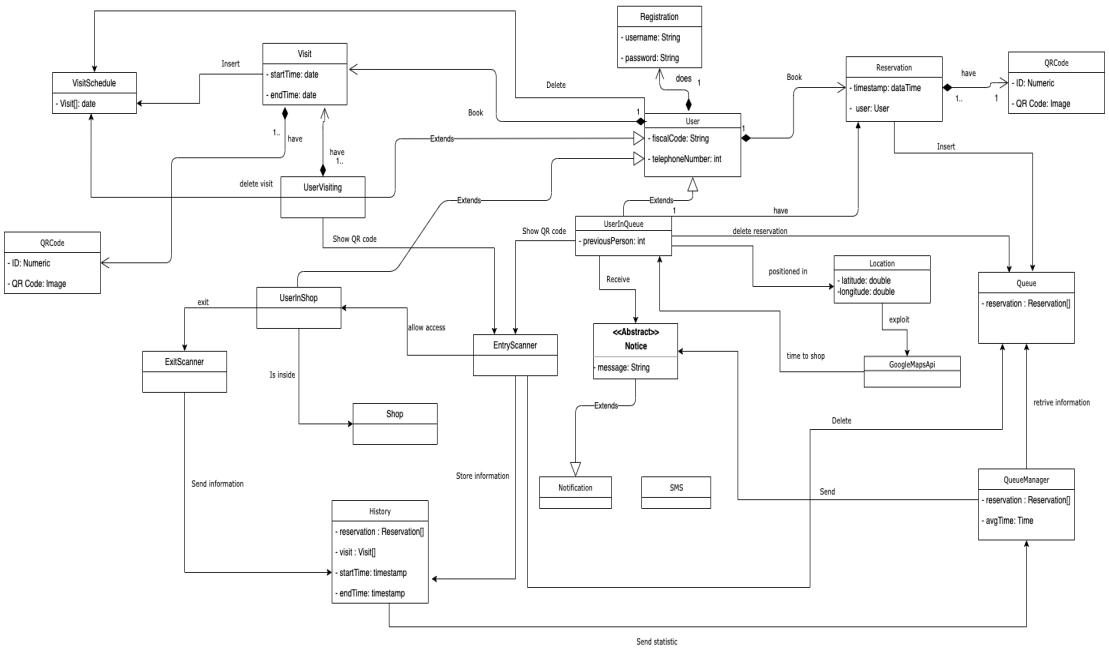
–The system have to provide User infomation about queue's dynamic in real time (i.e the number of people ahead and the time estimation before his own turn). – To do this, the application should try, in the best way, to estimate the time that the customers will spend in the shop, in order to notify in time users who are waiting their turn.
The application analyses the customers' statistics and computes the average shopping time. The calculation considers the time range between the moments in which the User goes in and out.

In this way, the customers in queue will be notified in time and will be able to use GoogleMaps APIs to compute the path for arriving in time to market. The choice of using Google Maps APIs derives from the easy usability, frequency update and very large documentation.

Moreover, it will be possible to book a visit choosing the date and time in advance (of some days?). In this case, users will also have to indicate an approximation of the shopping time.
The time will be splitted into 30 minutes slots, and customers will choose which they want among the free ones.

In the UML diagram below will be list the main classes in order to understand how the whole system works.

Figure 2.1: Class diagram with UML



As we can see from the class diagram in figure 2.1, the user can book once or a visit or a reservation. In both cases will be provide him a QRcode, which will be submit to enter in the market. If the the User decided to undo his reservation in the queue could do it by making a cancellation from the application.

Moreover, if the User allows to share his position, the system will notice him in advance due to his punctuality.
In addition the system will notice the User through an SMS or a notice when it's almost his turn (10-15 minutes before).
Now we will analyze the interaction between the User and the system, in order to understand possible criticities.

In the Figure 2.2 first state diagram it can be osserved how a generic User can make a reservation thorugh the system. It's sufficient making this action to be added in queue to enter in the market.
If something goes wrong the User will be redirect to the home screen (initial state).
Usually the system reject the request if the market is next to closure (or for logistic problem).

The Figure 2.3 explains instead how to book a visit in the market. Once the

Figure 2.2: State diagram of the reservation in the queue

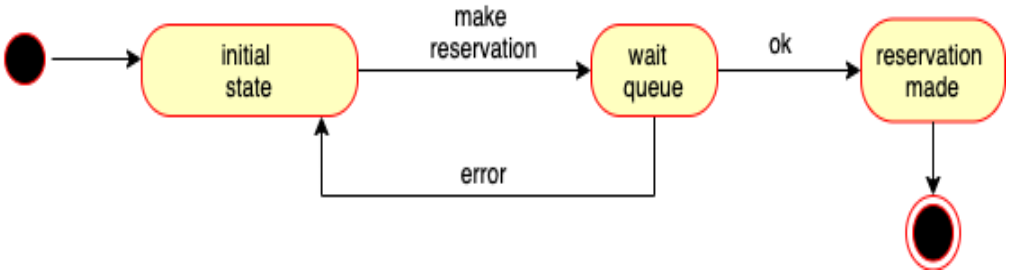
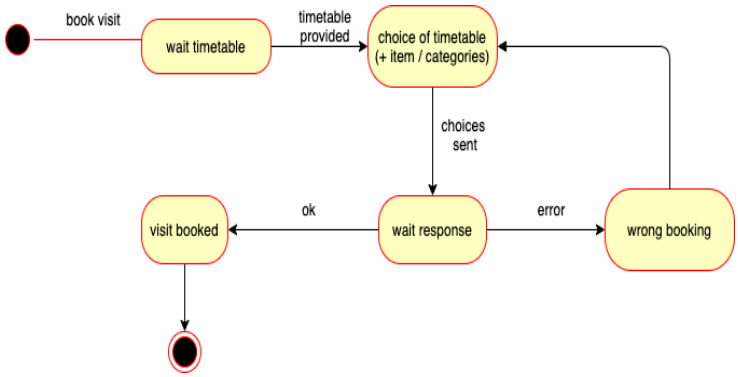


Figure 2.3: State diagram of booking a visit



timetable is provided, the User have to select the range time of the visit and the items that he wants to purchase.

If the choice made by the user is wrong (i.e timetable's slot full), the system notify him; the User will try again until his choice is correct.

2.2 Product functions

2.2.1 Queue Manager

The most important function is the *Queue Manager* because it must avoid the users waiting too much time their turn due to a wrong use of notifications. [TODO] In fact, it have to foresee, through statistics from users' information, the correct time in which the users will enter in the market once arrived. This can be done thanks to the notifications sent to the user It will also have to understand when accept and when refuse reservations, according to shop closing time and the number of people in queue.

2.2.2 Data Collection

The *Data Collection* is essential for the correct behavior of the Queue Manager described previously, because it will have to provide precise dates according to client's information. Therefore, the system will have to ask clients precise questions according to keep useful informations for estimating the shopping time into the supermarket, without violating users' privacy.
In order to achieve this goal, it needs to oblige the user to register himself in the system and to check the item to allow the processing of his personal data, necessary to reserve virtually the seat in the queue.
One of the possible information asked could be the dimension of the expenses, which can be estimate by the number of item that are going to be purchased. This will be used, with the entry time, to track the number of user inside the market who is finishing. Then will be possibile to notify in advance users in queue about the closeness of their turn.

2.3 User characteristics

We distinguish the actors into our application based on actions and interactions with the external world:

- *User*: he's a client who has signed in the system and he can book a visit or take his queue number.
- *UserInQueue*: he's a User who has taken his own turn in queue and he's waiting for the system notification
- *UserVisiting*: he's a User who has booked a visit and he's still waiting for entering into the supermarket.
- *UserInShop*: he's a client who has taken his own queue ticket or he has booked a visit. After that he is arrived at the supermarket, he has scanned the QR code and he has entered into the shop.

2.4 Assumptions, dependencies and constraints

Chapter 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.4.1 Standards compliance

3.4.2 Hardware limitations

3.4.3 Any other constraint

3.5 Software System Attributes

3.5.1 Reliability

3.5.2 Availabilitys

3.5.3 Security

3.5.4 Maintainability

3.5.5 Portability

Chapter 4

Formal Analysis Using Alloy

Chapter 5

Effort Spent

Chapter 6

References