L.M. Geoinformatic Engineering

AA 2020/2021

RASD

Requirement Analysis Specification Document

Version: First release

Date: 17/12/2020

Authors: Marianna Alghisi
Gabriele D’Ascoli
Martina Pasturensi

Table of Contents

1. Introduction

1.1. Purpose

The following documentation contains, describes and motivates all the decisions taken through the Requirement Engineering process, which is the process of discovering the purpose for which the software system was intended by the involved stakeholders.

The RASD - Requirements Analysis and Specification Document has the objective of summing up in a detailed way the process of Requirement Analysis in Software Developing, that consist in identifying the goals, the domain and the requirements of the software system and documenting these in a form that is amenable to analysis, communication, and subsequent implementation.

1.2. Scope

The required application, named Costumer Line-Up – CLup, is aimed to help people in facing the sanitary emergency due to Covid-19. In fact, the goal of this project is to develop an easy-to-use application that, on the one side, allows store managers to regulate the influx of people in the building and, on the other side, helps the costumers in handling and saving time spent for going out of home for essential needs, avoiding crowds and long lines out of the shop. In particular, the application focuses on grocery shopping and supermarkets, by the way it could be extended to all kind of shops. More precisely:

- ***Basic Function***: the costumer should be able to access the application and line-up for a selected supermarket among the five closest to him (according to his position) and then wait until it is his turn. The costumer should be able to check through the application the number of people currently in line for the five closest supermarkets. If the costumer is lining for a certain supermarket, the application should inform the in-line costumer about the number of people in line before him and the approximate time of his turn. When the costumer asks to be put in-line for the selected supermarket receives through the application a QR code. Once his turn has arrived, before entering the supermarket his QR code will be scanned by an operator through an appropriate device, that update the status of the queue for the entering the supermarket.

- ***Advanced Function 1***: The costumer should be able to reserve a visit to a selected supermarket, by choosing the day and the time slot, through the available ones, in which he is going to visit the shop and indicate the approximate duration of the visit. The costumer will receive a specific QR code that allow the costumer to enter in the supermarket only in the selected day and time.

- ***Advanced Function 2***: The application provides to the shop a statistical analysis of the per hour affluences, in such a way that the shop can better organize the employees shifts.

Basic Function and Advanced Function 1 are based on the use of a specific tool: the QR code, which is a black and white matrix used to memorize specific information, that can be read by a specific application. In this case the QR code, given to the user, works as an entrance ticket to the supermarket. It will contain information about the user, the selected supermarket and the entry time. Therefore, it has an expiration time: the code expires after 15 minutes since the entry time.

In order to evaluate the approximate duration of the visit, the client, during the reservation (both in the basic function and advanced function 1), must indicate in a form, during the reservation, the categories of items that he intends to buy among 10 main categories. Each category corresponds to a specific sector of the supermarket. More information about this aspect are given in the follow.

In addition, it will be also possible for costumer to physically line up to the supermarket, without the using the application. Costumers will have to go to the supermarket and there they will receive their entry ticket: an employee, with his device, will print them the QR code, containing also information about the number of people in queue and the mean entry time.

1.2.1. Application Domain

The application has been thought to face the problems related to Covid-19 sanitary emergency. Possible users of the application are costumers, that need to go out of home for their essential needs, and supermarket owners, that want to experiment a new innovative way to handle the influx of people inside and outside the shop.

The application can be use both in big metropolitan cities, where we have many supermarkets and a high number of habitants per square kilometre, but also in the small provincial towns, where there are less habitants, but a supermarket can be shared between different villages.

1.2.2. Relevant Phenomena

Once having clarified the project's main goals, is important to focus on the phenomena that occurs in our horizon. We have decided to analyse the phenomena by using "The World and the Machine Approach", by M. Jackson and P. Zave.

With “World” we refer to the real and physical environment in which the software is going to be used by clients. The World’s Phenomena refer to all the phenomena occurring in the real world that cannot be controlled by the Machine.

Machine’s Phenomena refers to the phenomena occurring inside the software system, at software and hardware level.

The aim of this chapter is to distinct the phenomena that occur in the world, in the machine and the phenomena that are shared between them. In particular, the shared phenomena must be distinguished between phenomena that are controlled by the world and observed by the machine and phenomena that are controlled by the machine and observed by the world.

World Phenomena:

WP1	User has the necessity of going to the supermarket
WP2	User has a smartphone or a laptop
WP3	User goes to the supermarket in time for his turn

Machine Phenomena:

MP1	Application retrieves the five closest supermarkets, according to user position
MP2	Application generate the QR code
MP3	Statistical processing on the data of the affluences
MP4	Update of the line of each supermarket after every user request

Shared Phenomena:

SP1	Application retrieves User position
SP2	Costumer selects to visit a supermarket in the list proposed by the application
SP3	Costumer asks to be put in-line for the selected supermarket
SP4	Costumer books a visit to a selected supermarket
SP5	Supermarket Administrator access to the statistical analysis of affluences
SP6	Costumer decides to physically line up to the supermarket
SP7	Costumer decides to modify or delete a booked visit to the supermarket
SP8	Costumer is allowed to enter in the supermarket
SP9	Costumer is allowed to exit from the supermarket
SP10	Queue status is updated

1.2.3. Goals

The following paragraph will sum up all the goals to fulfil of the software system, detected during the Requirement Analysis process.

G1	Allow user to check for the number of people in line for the five closest supermarkets
G2	Allow user to know the waiting time for accessing the supermarket
G3	Allow user to line up for a selected supermarket
G4	Allow user to book a visit to the supermarket in the following days
G5	Allow user to delete a visit to the supermarket
G6	Allow user to modify a visit to a supermarket
G7	Inform the supermarket about the number of people currently inside the shop
G8	Inform the supermarket about the number of people in line
G9	Menage the entries in such a way that the number of people currently inside the supermarket doesn’t exceed the legal safety limit
G10	Perform statistical analysis on the information collected in the database
G11	Allow the supermarket to visualize the statistical analysis of collected data

1.3. Acronyms and Abbreviations

1.3.1. Acronyms

RASD	Requirement Analysis Specification Document
GPS	Global Positioning System
UML	Unified Modeling Language

1.3.2. Abbreviations

WPx	World Phenomena
MPx	Machine Phenomena
SPx	Shared Phenomena
Gx	Goal
Rx	Requirement
UCx	Use Case

1.4. Revision History

Date	Modifications
17/12/2020	First Version

1.5. Reference Documents

The production of this Requirement Analysis has been obtained with the support of the following documents:

- Specification Document: “CLup Assignment”
- Slides of the lectures

1.6. Document Structure

Chapter 1 – Introduction
The first chapter gives an introduction of the purpose and the scope of the software to develop, it highlights the main goals of the stakeholders involved in the project, the domain assumptions and the phenomena occurring in the environment. In addition, it contains specifications about the definitions, acronyms and abbreviations used for the drafting of the document, information about the revision history and the reference documents.

Chapter 2 – Overall Description
The second chapter gives the overall description of the software system from a high-level point of view, by using a UML class diagram. It contains the description of all the main functions and their state charts. In the user characteristic paragraph are explained the different types of actors that can access the application. In conclusion, the chapter lists all the domain assumptions.

Chapter 3 – Specific Requirements
The chapter start with the definition of the interface requirements, focusing on the interfaces of the user, the hardware, the software and the communication. Then, it focuses on the functional requirements and the mapping of them, which consist in the definition of the relationship between requirements, goals and domain assumption.
Functional requirements are submitted with a list of scenarios and use cases with their corresponding sequence diagrams. The last part of the chapter focuses on performance and non-functional requirements.

Chapter 4 – Alloy
Includes the alloy code and the corresponding metamodels generated from it, with a brief introduction about the main purpose of the alloy model.

Chapter 5 – Effort Spent
This chapter shows the effort spent by each member of the group.

Chapter 6 – References
The last chapter contains the reference documents used to produce the RASD.

2. Overall Description

2.1. Product Perspective

2.1.1. UML Description

The UML below describes from a high-level point of view the model of the system to be developed, it does not include every class that will be necessary for the definition of the complete architecture of the system.

CLup offers different functions based on the type of user. In fact, the user registers to the application providing all necessary information (information required are different as if he register as Costumer or Supermarket User). Here we can identify the main aspects related to CLup:

- Costumer User can access the application and retrieve the five closest supermarkets (according to his GPS position) and retrieve information about the number of people currently in line to get in the supermarket and the mean waiting time.
- Costumer Users can "line up" from their home, they will get a QR code that works as an entry ticket. They will be able to wait at home until comes their turn to get in the store. The application works as a digital counterpart to the common situation where people, who are in line for a service, retrieve a number that gives their position in the queue.
- Users can "book a visit" to the supermarket, specifying the day and the time of the visit, and indicating the categories of items that they intend to buy, in such a way that the system is able to compute the mean visit duration.
- Since there are not available slots for the day and time requested by the user, the application provides a suggestion of alternative slots for visiting the store, to balance out the number of people in the store.
- All the data about requests, queue and booked visits are stored in the database.
- Supermarket User (for instance a store manager) can access the application to retrieve the statistical analysis of the affluences to the store.

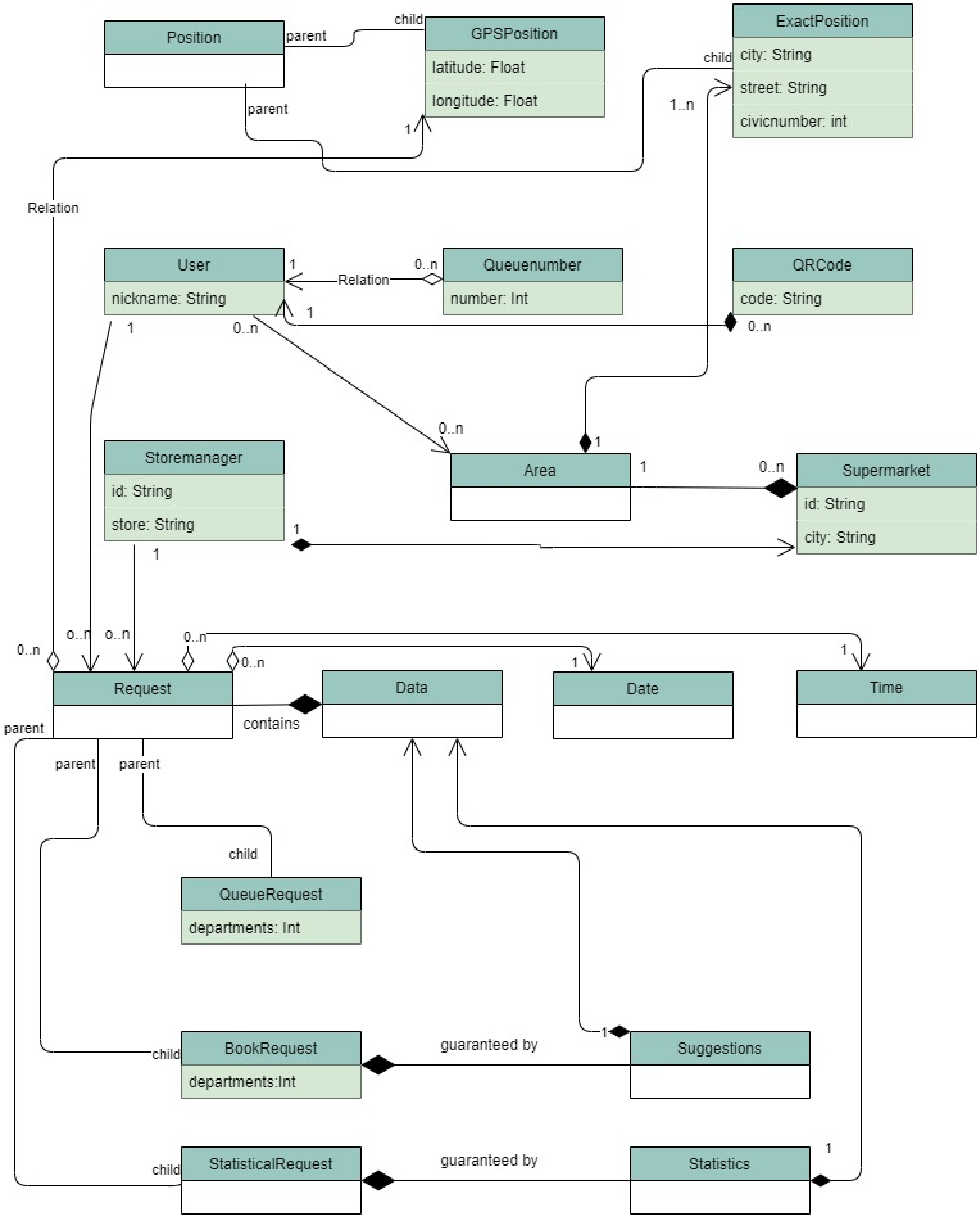


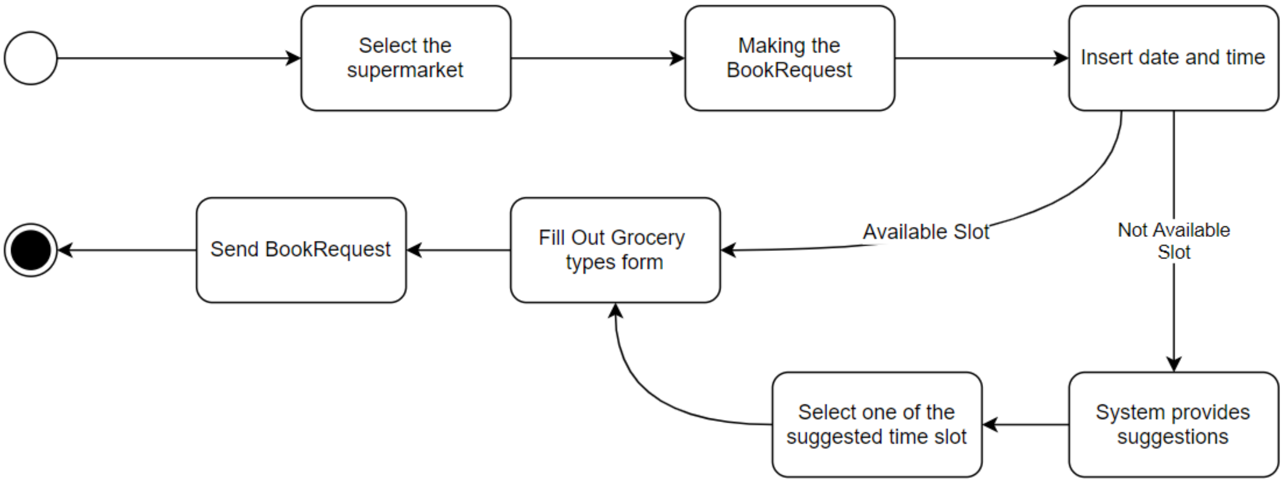
Figure 1. Class Diagram Using UML

2.1.2. State Charts

The following paragraph focuses on the analysis of some critical aspects of the application, modelling their behaviour and showing their state evolution over time through the appropriate state diagrams reported below.

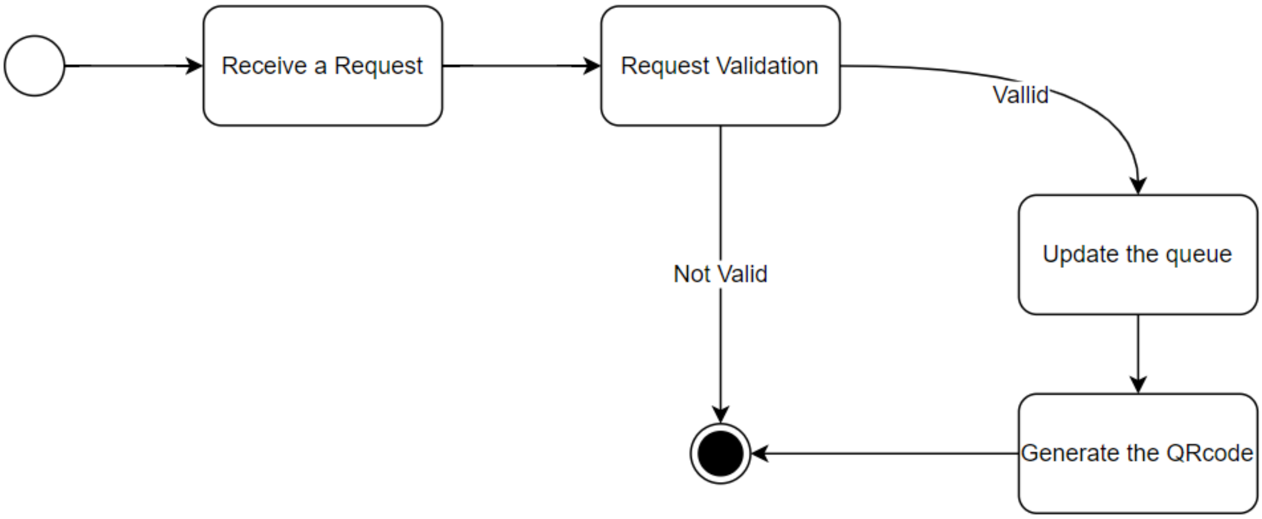
1. Creation of a Book Request

The following state diagram shows the creation of a new book request by the costumer user, showing the main action he must perform. If the time selected is not available the system provides "suggestions", which are alternative time slots available for the selected day.



2. QR code generation

Since a request is valid and no errors occurs, the system generates the QR code, encoding information related to the user and its appointment. The same procedure is valid for both BookRequest and QueueRequest.



2.2. Product Functions

The following paragraph focuses on the description of some of the main product functions of the software system, that performs the most important requirements.

- Get in Line

  The main function of the application is to let the costumer users “line up” from their home, sending a request for the generation of a number that is inserted in a virtual queue. Since the visit to the supermarket is possible at that time (the number of requests guarantee the social distancing) the number is generated: the user is put in the virtual queue and he will retrieve a QR code that let him access the shop only when is his turn.

  If the number of requests is excessive the application can suggest to “book a visit” to the store, to balance out the number of people, or it suggests checking the queue status in the other stores close to the user.

- Book a visit

  The application gives the possibility to the costumer users to book a visit to the store. The user will select the day and the time of the future visit, the system, since there are available slots, will ask the user to declare the categories of goods he is intended to buy. This would allow the application to plan visits in a finer way, for example allowing more people in the store, if it knows that they are going to buy different things, hence they will occupy different spaces in the store when they visit (thus respecting the requirement that people keep enough distance between them).

  The costumer user that books a visit will receive a QR code valid for the selected day and time.

- Monitor and Control

  Supermarket user can use the application to know the exact number of people that are in line to get in the shop and the number of people currently inside the shop.

  Costumer users are required to scan their QR code to entry and exit from the shop, in this way the system can compute the number of people inside the shop and control that it doesn’t exceed the maxim legal value.

- Statistical Analysis

  Supermarket user can access the application to retrieve a statistical analysis of the affluences.

2.3. User Characteristics

The possible users of the application are:

- Costumers: civilian, forced by the restriction due to the global pandemic to stay home. He is a potential customer; he downloads the application on his device in order to optimize his time spent in going out for grocery shopping. He can virtually line up for the store or book a visit.
- Supermarket: every grocery store that download and register to the application, in order to menage the influx of people inside and outside the shop, respect the constraints on the safe distance between people, get statistical analysis about costumers and also promote himself. In fact, the application suggests to every costumer the five closest supermarkets, this is a good chance for a new shop to be known in the neighbourhood.

2.4. Assumptions, Dependancies and Constraints

2.4.1. Domain Assumptions

D1	Costumer respects queue order
D2	Costumer approaches to the store in time when is his turn
D3	Costumer goes to the supermarket alone
D4	Costumer respects the booking of a visit
D5	Costumer respects the duration of the visit indicated
D6	Costumer buys only the items stated in the grocery types list
D7	Costumer that physically go to the store respects the queue order, also considering the virtual queue
D8	Internet connection works properly
D9	Each device has a GPS sensor integrated
D10	GPS is active while the app is running
D11	GPS provides the exact position with an error of 5 meters at most
D12	An employee is always present at the entrance of the store to scan the QR codes and make sure that the procedures are followed by the customers
D13	Queued customers do not leave their shift unless they cancel their booking from the app
D14	The user buys in one of the five nearest stores
D15	Users always buy something when they visit the store
D16	Each supermarket has an authentication code that proves he is a shop for registering to the application

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

When the user opens the application, he will be asked to declare if he intends to use the application as a Costumer User or as a Supermarket User. Then, after having choose his user identity the application will return a page where there are available two options:

- "Log In" function gives the user the possibility to log in if he is already registered.
- "Register" function gives the user the possibility to sign up to the application as a new user.

Once user logs in the application, the system will return different home pages depending on whether he is a customer or a supermarket employee.

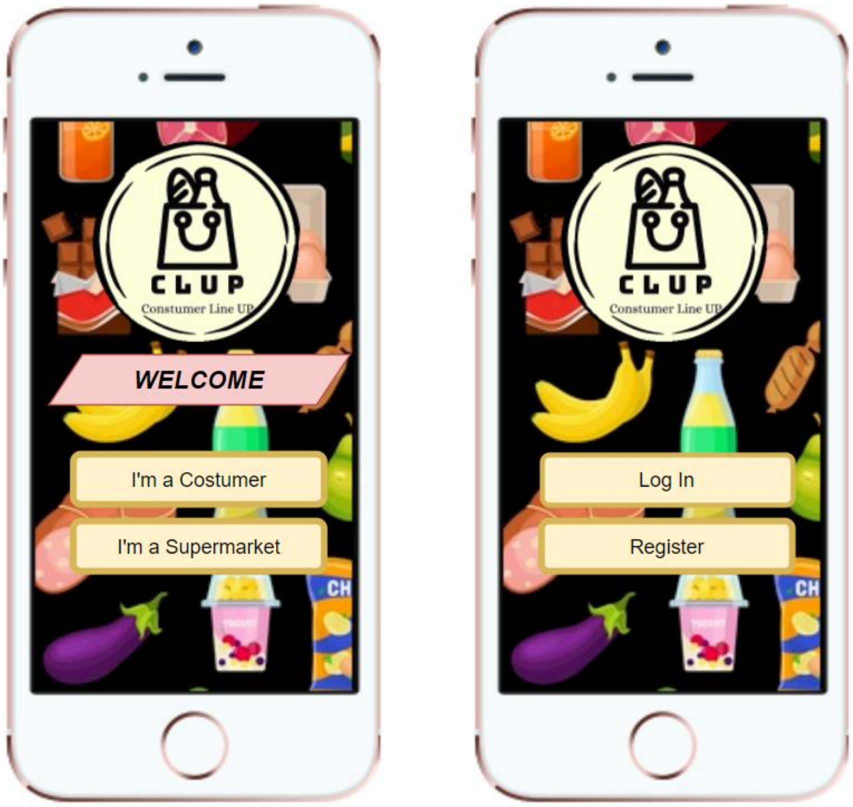


FIGURE 2 - REGISTRATION AND LOGIN MOCKUP

Since the user logs as a customer, he will visualize the home page where he can choose between two options:

- "Supermarket Around" that check for the five closest supermarkets, in order to check the current waiting time and getting in line for the store or book a visit.
- "Booked Visits" that check for his booked visits, in order to modify or delate already scheduled reservations.

Since the user select the option "Supermarket Around", the application will redirect him to a page containing a map showing his current positions and the five closest supermarkets. The page also contains a list where, for each supermarket is specified the current number of people in queue and the medium waiting time to get in.

After consulting the information related to each supermarket, the costumer can select one the supermarket. The system will then ask him if he intend to line up to the selected supermarket or if he wants to book a visit in order to plan and reserve an appointment for going to the supermarket.

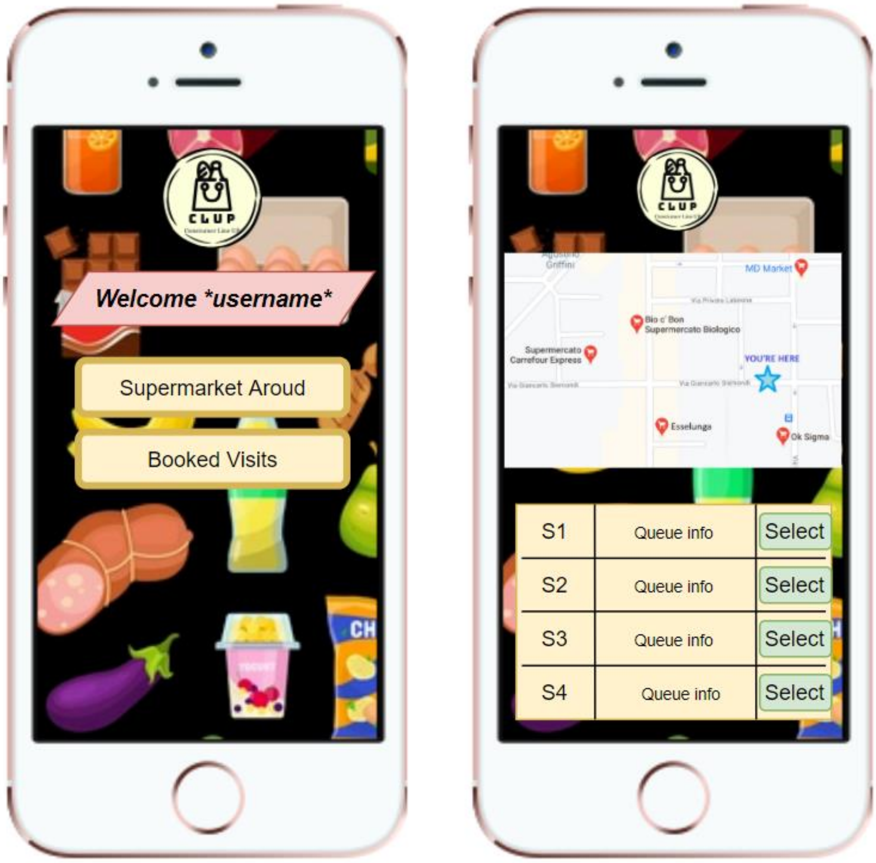


FIGURE 3 - COSTUMER HOME PAGE AND SUPERMARKET AROUND

If the user logs as a supermarket, for instance he is and employee or a store manager, he will visualize a home page in which he can choose between the following options:

- "Scan QR code" option allows the supermarket to scan the costumer's QR code, to know if he is allowed to get in.
- "Print QR code" allows to generate a QR code for the costumers that physically line up for the supermarket.
- "Statistics" option allows the supermarket to consult statistical analysis of affluences in the past.

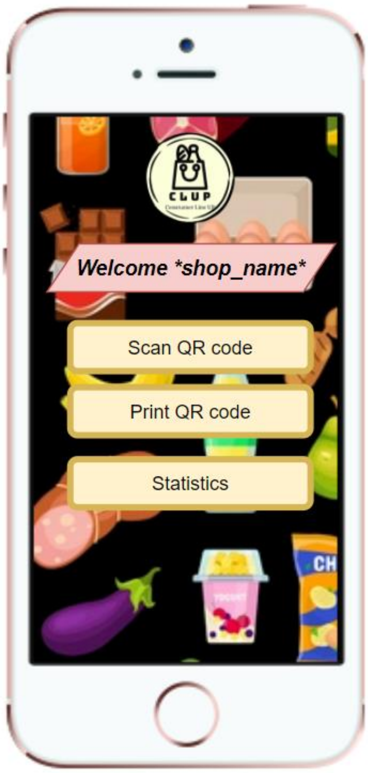


FIGURE 4 - SUPERMARKET HOME PAGE

3.1.2. Hardware Interfaces

Since the software system performs different functions and involves two types of users, also the types of hardware interfaces required change.

- Costumer user, to perform BF and AF1 must have a device with internet connection: the application is thought to be used via smartphone or laptop.
  If the costumer lines up or books a visit for the supermarket, he will receive back a QR code as an entrance ticket. Since he uses a smartphone, he can use the device both for booking and scanning the QR code at the supermarket's entry. Instead, if he decides to access the WebApp through the laptop, he will have to print the QR code at home with a personal printer.

- Supermarket user needs specific hardware to perform different functions:
    1. Supermarket user can perform AF2 by accessing the application through a device with internet connection (smartphone or laptop).
    2. At the entry, the employee logged as supermarket user, to scan the QR code, needs a mobile device with camera and connected to the internet (tablet or smartphone).
    3. A printer machine is needed at the entrance, to print the QR code for the costumers that what to physically line up and don't have the application. This is important to merge the physical and the virtual queue.
    4. Every shop needs a digital turnstile that can scan the QR-code at the exit. The turnstile must be connected to the internet, in order to update the number of people currently in the supermarket.

3.1.3. Software Interfaces

The functioning of the whole software system is also based on external software that develop specific functions:

- City Map
  It's a pubblic API required to visualize the position of the user in the city. Accordingly, it is essential for:
    1. Calculating the shorter path between user position and selected places of interest (in this case the supermarkets).
    2. Selecting the nearest five supermarkets, according to the calculated distance from the user
- QR code reader
  Application that allows on devices with imaging hardware to scan QR codes and retrieve the data encoded.
- QR code generator
  Application that allows to encode data in QR codes.

3.1.4. Communication Interfaces

The device uses internet connection to connect to CLup.

3.2. Functional Requirements

The following paragraph contains the list of all the functional requirements of the system, their relationship with respect to the goals and the domain assumptions. Moreover, in order to better clarify the requirements, common application scenarios are presented in the following. The last part contains the list of the possible Use Cases.

3.2.1. Requirement List

R1	The system access user's GPS position
R2	Users register to the application by filling a form with mandatory fields
R3	Supermarkets are certified with an authentication
R4	Costumer, to get their entry ticket to the supermarket, must select the types of grocery goods they're going to buy
R5	System allows customer line up for the shop by putting the costumer in the last place of the queue
R6	Customer selects the date and the time for booking the visit to the supermarket
R7	System retrieves the free time slots according to the date selected by the user, if the chosen time slot is not available (suggestions)
R8	System generates the booked visit according to the date and time selected by the user
R9	Customer users can modify the time slot of previously booked visits
R10	Customer users can delete their booked visit
R11	The system sends notifications to the users (both costumers and supermarket)
R12	The system evaluates the five closest supermarkets, according to user's position
R13	The system generates the QR code
R14	The system reads the QR code
R15	User can retrieve the QR code on the user interface
R16	The system computes the average time of visit according to the grocery types selected by the user
R17	The system computes the exact number of people currently in the shop, according to the number of entries and exits
R18	The system allows user to select one from the five nearest supermarkets
R19	The system updates the queue automatically
R20	The system computes the exact number of people queuing for a supermarket
R21	The system evaluates the mean waiting time to access the supermarket, according to the number of people waiting in line and the mean visit time of those that are inside the supermarket
R22	Supermarket users are allowed to access to statistical analysis

R23	The system builds statistics according to the data contained in the database
R24	The system asks to the supermarket user to access to the camera in order to scan the QR code
R25	The user can retrieve the list of his future booked visits
R26	Customers can access the supermarket only with a valid QR code
R27	Costumers must scan their QR code to exit from the shop
R28	Supermarket declares the maximum number of people that can access the shop
R29	The system stores information about entries, booked visits, affluences and duration of each visit in a database

3.2.2. Mapping

The following table highlights the relationship between goals, domain assumptions and functional requirements. In order to achieve a goal, we have to specify which hypotheses are taken to be always true (domain assumptions D) and which requirements (R) our system must have.

G1	**Allow user to check for the number of people in line for the five closest supermarkets**	
	D1 – D2 – D3 – D4 – D7 – D8 – D12 – D13	R1 – R2 – R12 – R19 – R20
G2	**Allow user to know the waiting time for accessing the supermarket**	
	D1 – D2 – D3 – D4 – D5 – D6 – D7 – D8 – D12 – D13	R1 – R2 – R4 – R20 – R21
G3	**Allow user to line up for a selected supermarket**	
	D8 – D9 – D10 – D11 – D14	R1 – R2 – R4 – R5 – R11 – R12 – R13 – R15 – R18
G4	**Allow user to book a visit to the supermarket in the following days**	
	D8 – D9 – D10 – D11 – D14	R1 – R2 – R4 – R6 – R7 – R8 – R11 – R12 – R13 – R15 – R16 – R18
G5	**Allow user to delete a visit to the supermarket**	
	D8	R1 – R2 – R10 – R11 – R25
G6	**Allow user to modify a visit to a supermarket**	
	D8	R1 – R2 – R7 - R8 - R9 - R11 - R13 - R15 - R25
G7	**Inform the supermarket about the number of people currently inside the shop**	
	D8 – D16	R2 - R3 - R14 - R17 – R24 - R26 - R27
G8	**Inform the supermarket about the number of people in line**	
	D8 – D16	R2 – R3 – R14 – R19 – R20 – R24 – R26

G9	**Menage the entries in such a way that the number of people currently inside the supermarket does not exceed the legal safety limit**	
	D1 – D2 – D3 – D4 D5 – D6 – D7 – D8 -D12 – D13 – D16	R2 – R3 – R14 – R17 – R24 – R26 – R27 – R28
G10	**Perform statistical analysis on the information collected in the database**	
	D8	R23 – R29
G11	**Allow the supermarket to visualize the statistical analysis of collected data**	
	D8 – D16	R2 - R3 - R22 – R23

G1	**Allow user to check for the number of people in line for the five closest supermarkets**
D1	Costumer respects queue order
D2	Costumer approaches to the store in time when is his turn
D3	Costumer goes to the supermarket alone
D4	Costumer respects the booking of a visit
D7	Costumer that physically go to the store respects the queue order, also considering the virtual queue
D8	Internet connection works properly
D12	An employee is always present at the entrance of the store to scan the QR codes and make sure that the procedures are followed by the customers
D13	Queued customers do not leave their shift unless they cancel their booking from the app
R1	The system access user’s GPS position
R2	Users register to the application by filling a form with mandatory fields
R12	The system evaluates the five closest supermarkets, according to user’s position
R19	The system updates the queue automatically
R20	The system computes the exact number of people queuing for a supermarket
G2	**Allow user to know the waiting time for accessing the supermarket**
D1	Costumer respects queue order
D2	Costumer approaches to the store in time when is his turn
D3	Costumer goes to the supermarket alone
D4	Costumer respects the booking of a visit
D5	Costumer respects the duration of the visit indicated
D6	Costumer buys only the items stated in the grocery types list
D7	Costumer that physically go to the store respects the queue order, also considering the virtual queue
D8	Internet connection works properly

D12	An employee is always present at the entrance of the store to scan the QR codes and make sure that the procedures are followed by the customers
D13	Queued customers do not leave their shift unless they cancel their booking from the app
R1	The system access user’s GPS position
R2	Users register to the application by filling a form with mandatory fields
R4	Costumer, to get their entry ticket to the supermarket, must select the types of grocery goods they’re going to buy
R20	The system computes the exact number of people queuing for a supermarket
R21	The system evaluates the mean waiting time to access the supermarket, according to the number of people waiting in line and the mean visit time of those that are inside the supermarket
G3	**Allow user to line up for a selected supermarket**
D8	Internet connection works properly
D9	Each device has a GPS sensor integrated
D10	GPS is active while the app is running
D11	GPS provides the exact position with an error of 5 meters at most
D14	The user buys in one of the five nearest stores
R1	The system access user’s GPS position
R2	Users register to the application by filling a form with mandatory fields
R4	Costumer, to get their entry ticket to the supermarket, must select the types of grocery goods they’re going to buy
R5	System allows customer line up for the shop by putting the costumer in the last place of the queue
R11	The system sends notifications to the users (both costumers and supermarket)
R12	The system evaluates the five closest supermarkets, according to user’s position
R13	The system generates the QR code
R15	User can retrieve the QR code on the user interface
R18	The system allows user to select one from the five nearest supermarkets
G4	**Allow user to book a visit to the supermarket in the following days**
D8	Internet connection works properly
D9	Each device has a GPS sensor integrated
D10	GPS is active while the app is running
D11	GPS provides the exact position with an error of 5 meters at most
D14	The user buys in one of the five nearest stores
R1	The system access user’s GPS position
R2	Users register to the application by filling a form with mandatory fields

R4	Costumer, to get their entry ticket to the supermarket, must select the types of grocery goods they're going to buy
R6	Customer selects the date and the time for booking the visit to the supermarket
R7	System retrieves the free time slots according to the date selected by the user, if the chosen time slot is not available (suggestions)
R8	System generates the booked visit according to the date and time selected by the user
R11	The system sends notifications to the users (both costumers and supermarket)
R12	The system evaluates the five closest supermarkets, according to user's position
R13	The system generates the QR code
R15	User can retrieve the QR code on the user interface
R16	The system computes the average time of visit according to the grocery types selected by the user
R18	The system allows user to select one from the five nearest supermarkets
G5	**Allow user to delete a visit to the supermarket**
D8	Internet connection works properly
R1	The system access user's GPS position
R2	Users register to the application by filling a form with mandatory fields
R10	Customer users can delete their booked visit
R11	The system sends notifications to the users (both costumers and supermarket)
R25	The user can retrieve the list of his future booked visits
G6	**Allow user to modify a visit to a supermarket**
D8	Internet connection works properly
R1	The system access user's GPS position
R2	Users register to the application by filling a form with mandatory fields
R7	System retrieves the free time slots according to the date selected by the user, if the chosen time slot is not available (suggestions)
R8	System generates the booked visit according to the date and time selected by the user
R9	Customer users can modify the time slot of previously booked visits
R11	The system sends notifications to the users (both costumers and supermarket)
R13	The system generates the QR code
R15	User can retrieve the QR code on the user interface
R25	The user can retrieve the list of his future booked visits

G7	**Inform the supermarket about the number of people currently inside the shop**
D8	Internet connection works properly
D16	Each supermarket has an authentication code that proves he is a shop for registering to the application
R2	Users register to the application by filling a form with mandatory fields
R3	Supermarkets are certified with an authentication
R14	The system reads the QR code
R17	The system computes the exact number of people currently in the shop, according to the number of entries and exits
R24	The system asks to the supermarket user to access to the camera in order to scan the QR code
R26	Customers can access the supermarket only with a valid QR code
R27	Costumers must scan their QR code to exit from the shop

G8	**Inform the supermarket about the number of people in line**
D8	Internet connection works properly
D16	Each supermarket has an authentication code that proves he is a shop for registering to the application
R2	Users register to the application by filling a form with mandatory fields
R3	Supermarkets are certified with an authentication
R14	The system reads the QR code
R19	The system updates the queue automatically
R20	The system computes the exact number of people queuing for a supermarket
R24	The system asks to the supermarket user to access to the camera in order to scan the QR code
R26	Customers can access the supermarket only with a valid QR code

G9	**Menage the entries in such a way that the number of people currently inside the supermarket does not exceed the legal safety limit**
D1	Costumer respects queue order
D2	Costumer approaches to the store in time when is his turn
D3	Costumer goes to the supermarket alone
D4	Costumer respects the booking of a visit
D5	Costumer respects the duration of the visit indicated
D6	Costumer buys only the items stated in the grocery types list
D7	Costumer that physically go to the store respects the queue order, also considering the virtual queue
D8	Internet connection works properly
D12	An employee is always present at the entrance of the store to scan the QR codes and make sure that the procedures are followed by the customers

D13	Queued customers do not leave their shift unless they cancel their booking from the app
D16	Each supermarket has an authentication code that proves he is a shop for registering to the application
R2	Users register to the application by filling a form with mandatory fields
R3	Supermarkets are certified with an authentication
R14	The system reads the QR code
R17	The system computes the exact number of people currently in the shop, according to the number of entries and exits
R24	The system asks to the supermarket user to access to the camera in order to scan the QR code
R26	Customers can access the supermarket only with a valid QR code
R27	Costumers must scan their QR code to exit from the shop
R28	Supermarket declares the maximum number of people that can access the shop
G10	**Perform statistical analysis on the information collected in the database**
D8	Internet connection works properly
R23	The system builds statistics according to the data contained in the database
R29	The system stores information about entries, booked visits, affluences and duration of each visit in a database
G11	**Allow the supermarket to visualize the statistical analysis of collected data**
D8	Internet connection works properly
D16	Each supermarket has an authentication code that proves he is a shop for registering to the application
R2	Users register to the application by filling a form with mandatory fields
R3	Supermarkets are certified with an authentication
R22	Supermarket users are allowed to access to statistical analysis
R23	The system builds statistics according to the data contained in the database

3.2.3. Scenarios

The following chapter will focus on the presentation of general scenarios, which are "a narrative description of what people do and experience as they try to make use of computer systems and applications" (M. Carrol, "Scenario-based Design", Wiley, 1995. Each scenario describes a possible usage of the software by the users in real life.

1. Alice is a young woman that lives in Milan and works for a big software company. As required by the DPCM, to prevent the spread of covid-19, she is currently working from home and she is allowed to exit her home only for essential needs, such as grocery shopping. Despite the lockdown, Alice has a lot of work to do, so she cannot waste time in long lines out of the supermarkets waiting her turn to get in. She decides to download and register to CLup application, to get information about the lines in the nearest supermarkets, "virtually" line-up for the selected supermarket and go to the shop only when it's almost her turn.

2. Barbara works as a nurse at Niguarda Hospital in Milan, since the beginning of the health emergency her shifts have increased and when she comes back home from work she wants to relax. She has the possibility to go out for grocery shopping only during her rest-days, which are scheduled in advance by the working calendar. Barbara decide to download CLup to be able to reserve a visit to the supermarket in advance during the scheduled rest-day, to be sure that she can go grocery-shopping in that specific day.

3. Carlo is a retired man; he does not get along well with technology and he does not have any laptop or smartphone because he prefers to communicate with his family with his landline phone. Anyway, as every human being, he also needs to go out for grocery shopping. He decides to go to the supermarket and "physically" line up out of the store after getting his ticket through a physical machine that indicates the medium waiting time.

4. Daniele is the supermarket manager, due to covid-19 many of his employees are currently sick at home and he needs to organize work shifts of the available employees for the next two weeks. He decides to use CLup and consult the statistics on the data contained inside the database about the mean affluences of the past week, to find out when the supermarket is more crowd during the day and increase the number of employees in the work shift corresponding to that time.

5. Elisa works in politics; she is member of the regional council and she must be available 24/7 to discuss about possible emergencies in the board. Elisa booked through CLup a visit for going to the supermarket on the next Sunday, however some protests has occurred during Saturday night and she must attend in the next day an

emergency meeting in the council to solve the problem of protests. Elisa use CLup to cancel her booked visit to the supermarket.

3.2.4. Use Cases

Once having described some of the possible scenarios around the software system we proceed with the definition of the related Use Cases, which are a generalization of the given scenarios, highlighting the participating actors, entry and exit conditions and the possible exceptions. Every Use Case is followed by the corresponding Sequence Diagram that gives a graphical description of the interactions between the objects participating in a use case.

1) **User resister to the application**

Name:	**User registration**
Participating actors:	Every user
Entry conditions:	User is not registered
Flow of events:	- User accesses the application and declare which type of user is (Costumer or Supermarket) - System redirect the user to the homepage - User asks for the registration form - System opens the registration form and establish a connection with the database - User inserts data - User submits the form to the system - The system verifies the data in the form inserted by the user If the data inserted are corrected, the system send a confirmation message to the user
Exit conditions:	The user has successfully registered, he is able to log in the application
Exceptions:	- User inserts wrong data: InputVerification = false, if this happen the system return an error message and redirect the user to the registration form System fails during the execution of the process
Special Requirements:	System operation must be very fast

The procedure is the same for both costumer and supermarket users, the difference is in the data required by the registration form. In fact, costumer users will be asked to insert username, password, e-mail address and telephone number; supermarket clients will be ask to insert also an authentication that proves that they are a commercial point.

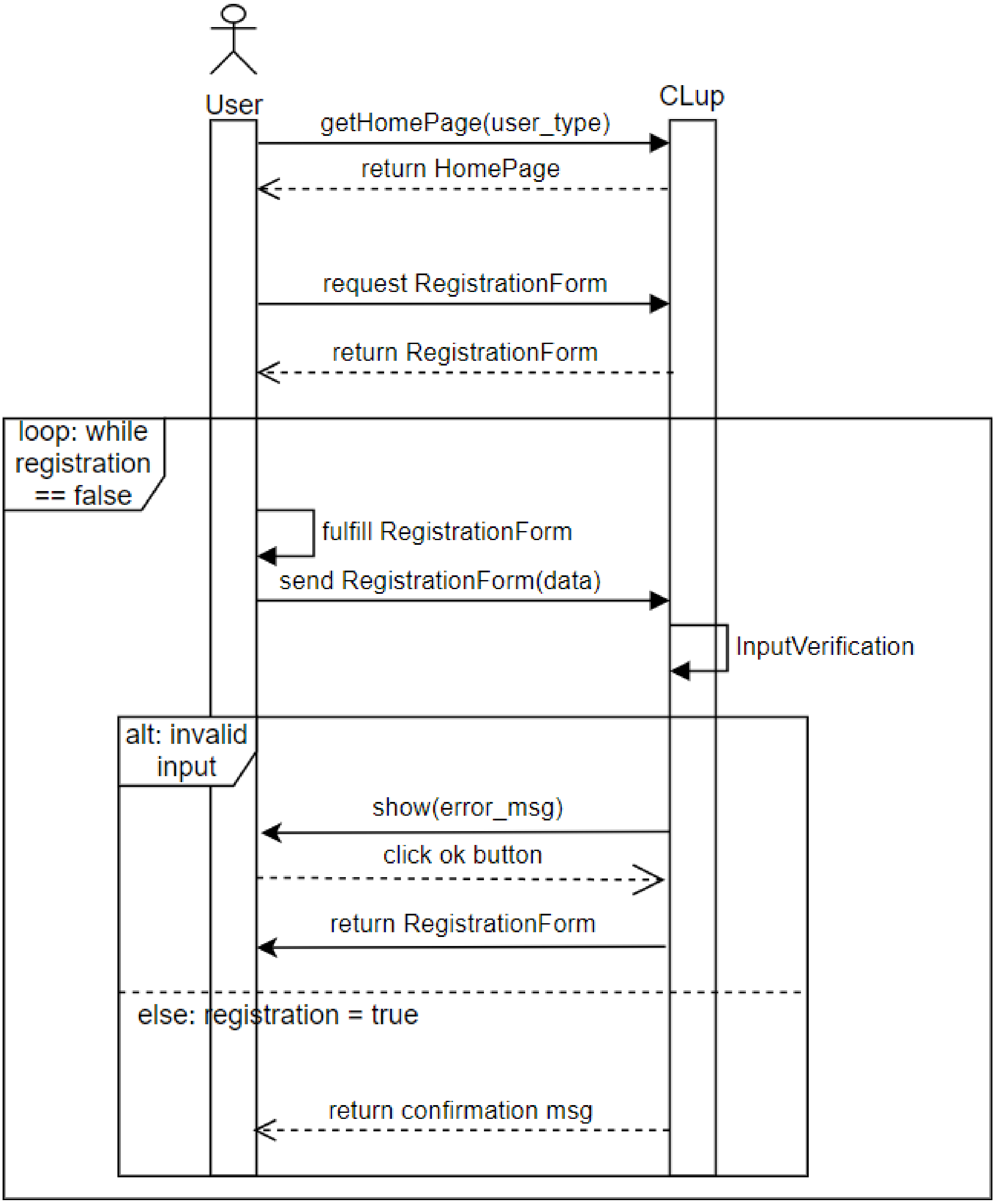


Figure 5 - UC1: user registration

2) User logs in the application

Name:	**User's login**
Participating actors:	Every user
Entry conditions:	The user is already registered
Flow of events:	- User clicks on "Log In" button - System opens the page to enter data - User fulfill the mandatory fields with necessary information - User submits the login form - System informs the user about the end of the operation
Exit conditions:	The system confirm that the user is entered in his private area of the system
Exceptions:	The system failed or the user inserts wrong data

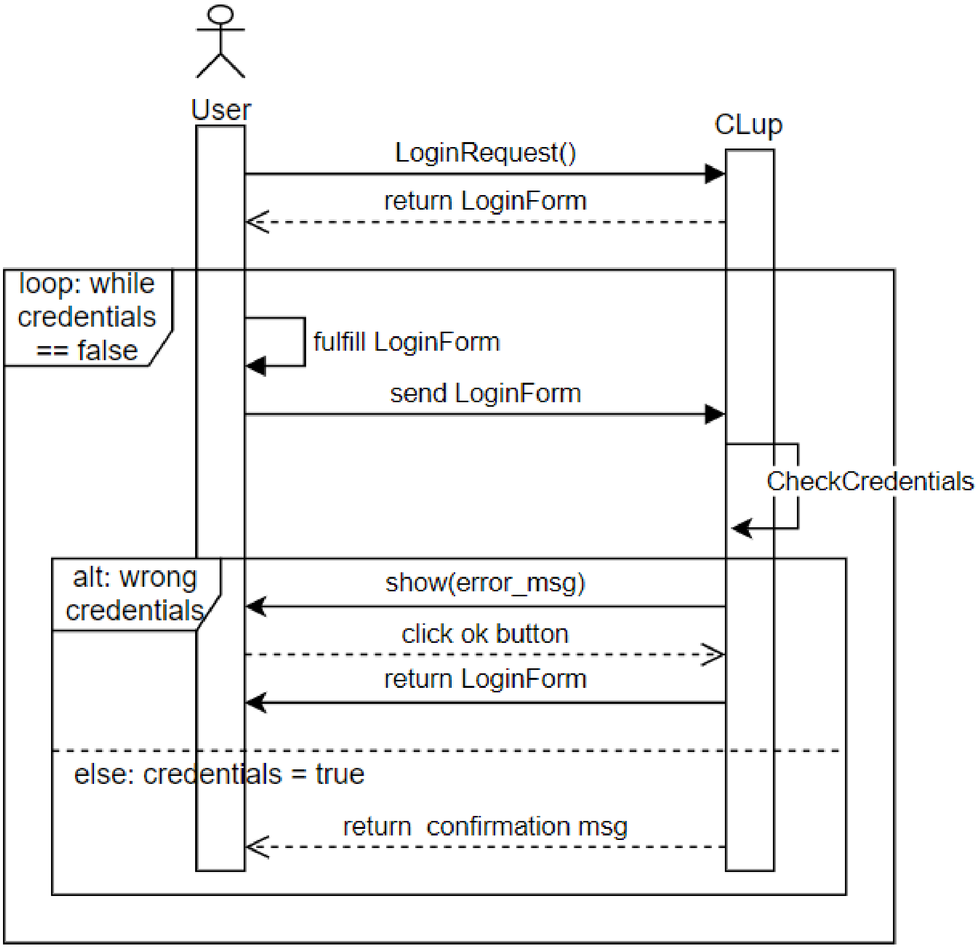


FIGURE 6 - U2: USER LOG IN

3) User logs out

Name:	**User's logout**
Participating actors:	Every user
Entry conditions:	The user is already logged in
Flow of events:	- User access the Home Page - User clicks the logout button - System closes the session
Exit conditions:	The system redirects the user to login page
Exceptions:	The system fails in executing the process

4) **Costumer User accesses the application and retrieves the five closest supermarkets, the number of people in line for each one of them and the mean waiting time**

Name:	**Supermarket around**
Participating actors:	Costumer User
Entry conditions:	The user is already registered and logged in
Flow of events:	- User accesses the Home Page - User selects "Supermarket Around" option - System get the user's position - System elaborates through an external software the five closest supermarkets - System extract from the internal database the number of people in queue and the mean waiting time for entry in each supermarket - System dynamically renders the user interface - System return the page to the client
Exit conditions:	User visits the page and access to the information contained
Exceptions:	- System fails in getting User's position - External software fails - System fails
Special Requirements:	- System should be able to retrieve the user's position - External software is required to elaborate the request

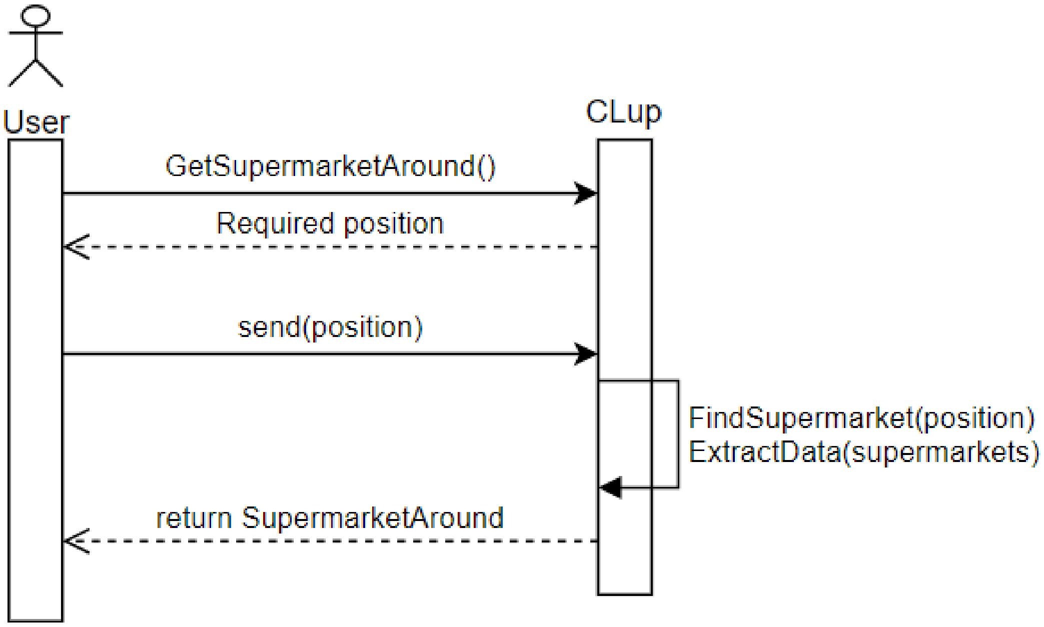


FIGURE 7 - U4: SUPERMARKET AROUND

5) Costumer User lines up for a selected supermarket

Name:	**Line Up**
Participating actors:	Customer User
Entry conditions:	User is logged in and visit the home page
Flow of events:	- User checks the list of the five closest supermarkets - User selects one of the five supermarkets in homepage - System provides the options available for the selected supermarket - User makes a request for getting in line - User inserts the types of articles he wants to buy - User submits his data - System generate the QR code, updates the queue - System returns the QRcode and the number of people in line before the user
Exit conditions:	System returns the countdown until user's shopping
Exceptions:	- User is not logged in - Grocery Types form empty - System fails
Special Requirements:	The system must automatically update the queue and generate the QR code

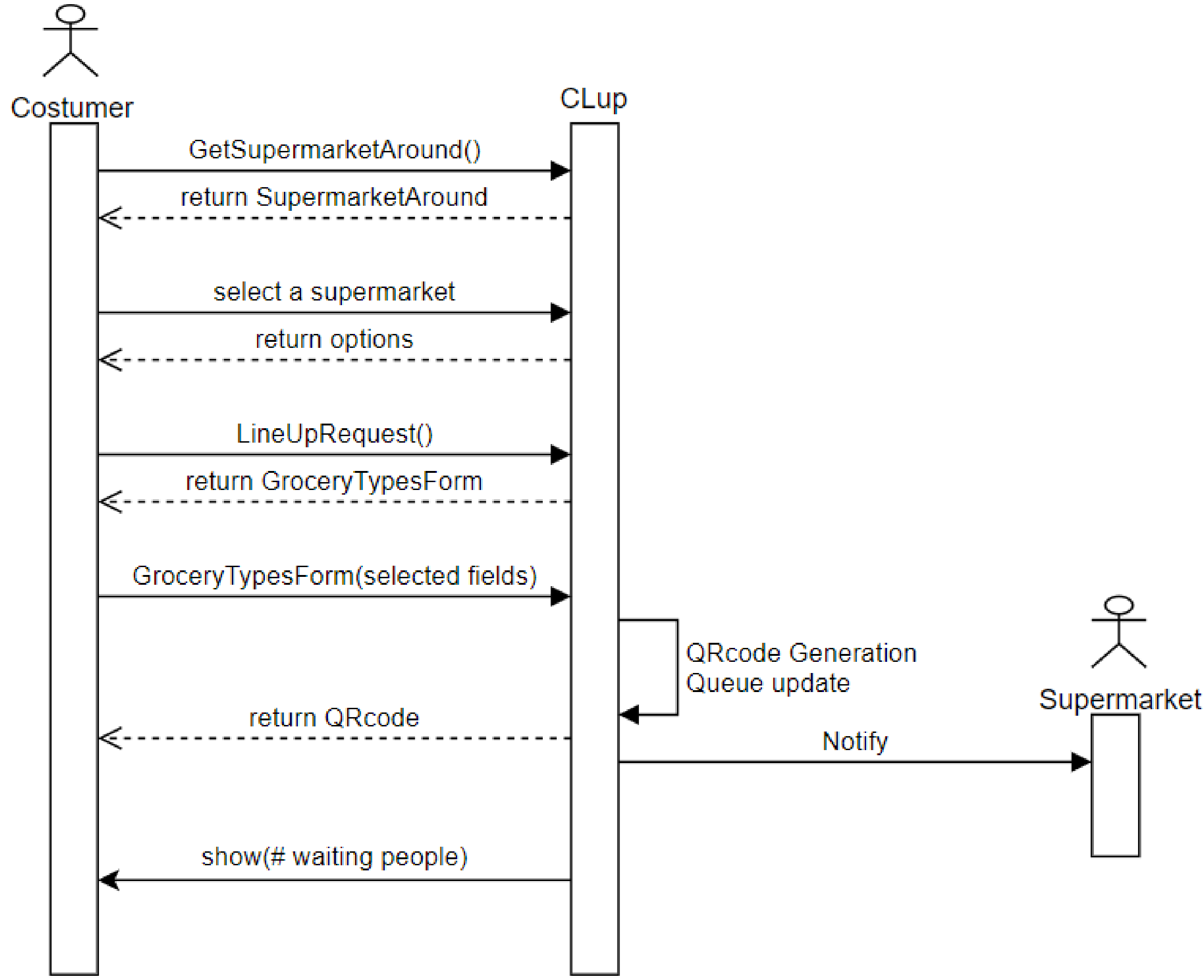


FIGURE 8 - UC5: LINE UP

6) **Costumer User books a visit to the supermarket for a specific date and time**

Name:	**Book a Visit**
Participating actors:	Every Costumer User
Entry conditions:	User is logged in and have selected a supermarket from the list
Flow of events:	- User makes a request for booking a visit - System returns a form in which the user can select the date and the time of the visit - User inserts date and time - System checks if there are available slots for the selected date and time - User selects the types of groceries he is going to buy - System creates the appointment and generates the QR code - System sends to the user the QR code and notifies the supermarket about the event
Exit conditions:	User's appointment d and the user can retrieve it in the
Exceptions:	- User is not logged in - User selects a day or a time with unavailable slots In this case the system sends back to the user a list of the available alternative slots for the selected day (suggestions) - Empty grocery types form - System fails - No available time slots for the selected date
Special Requirements:	The system must automatically update the queue and generate the QR code

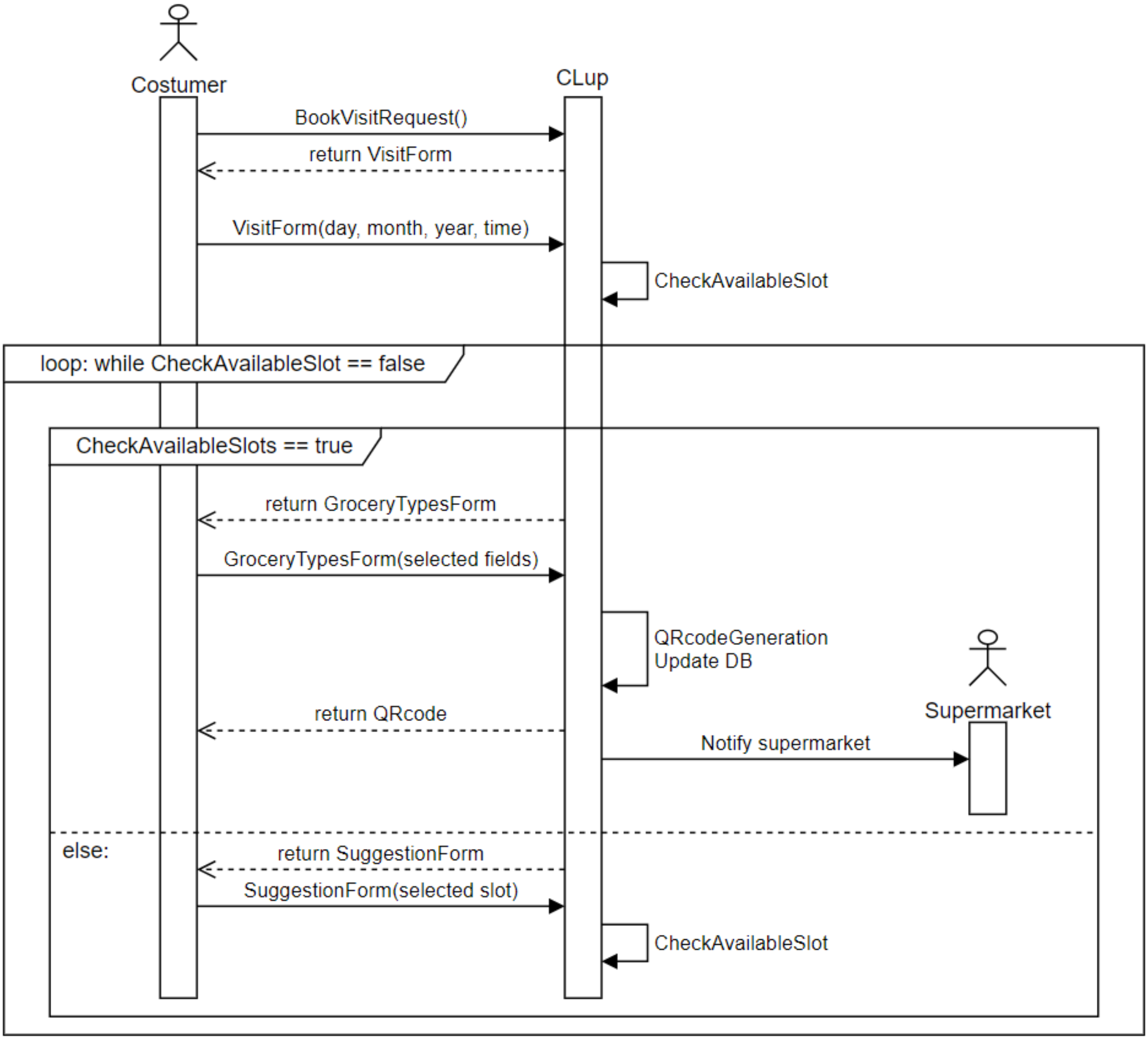


FIGURE 9 - UC6: BOOK A VISIT

7) **Costumer User can modify the time slot of his booked visit to the supermarket, assuming that time are available for the previously selected day.**

Name:	**User modifies his booked visit**
Participating actors:	Every costumer user
Entry conditions:	User has already booked a visit and user is logged in
Flow of events:	- User accesses booked visits page - System returns a list of all booked visits - User selects a booked visit - User clicks the modify button - System checks for alternative available slots - User selects the new time slot and submits - System updates the booked visit details and generate the new QR code - System send the new QR code to the user and notifies the
Exit conditions:	System returns the QR code to the user
Exceptions:	- User is not logged in - System fails during the uploading of the comment - No alternative available slots
Special Requirement:	After this operation, the queue is dynamically update

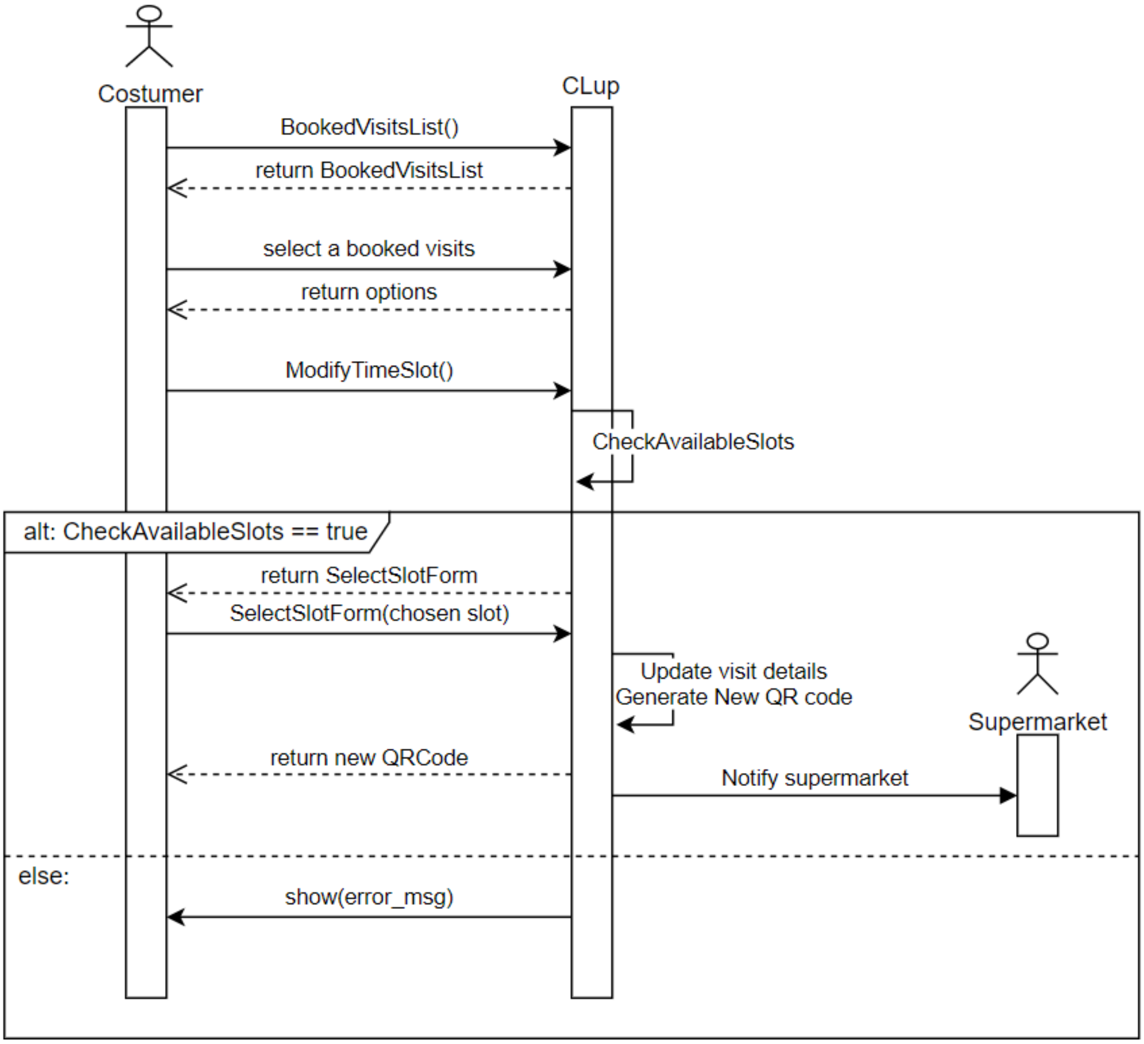


FIGURE 10 - U7: MODIFY BOOKED VISIT

8) **User Deletes a previously booked visit**

Name:	**Delete a visit**
Participating actors:	Every User
Entry conditions:	User has already booked a visit and user is logged in
Flow of events:	- User accesses booked visits page - System returns a list of all booked visits - User selects a booked visit - The user clicks the delete option and commit the operation - The system deletes the appointment and notifies the supermarket
Exit conditions:	The system returns to the user the updated list of booked visits
Exceptions:	- User is not logged in - The user did not book any visit to the supermarket - System fails
Special Requirement:	After this operation, the queue is dynamically update

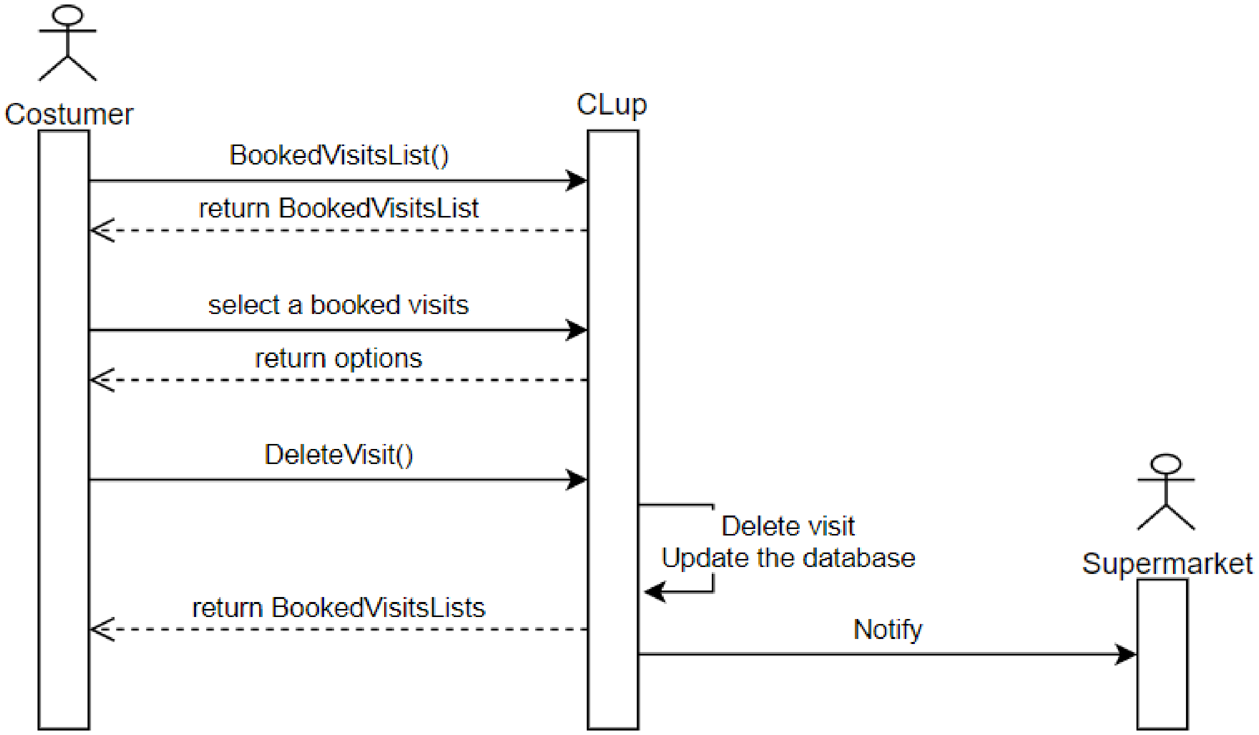


FIGURE 11 – UC8: DELETE VISIT

9) **Costumer User's entrance in the supermarket is possible only if the employee scans the QR code on entrance ticket.**

Name:	**User enters in the supermarket**
Participating actors:	Supermarket userand costumer user
Entry conditions:	Employee is in his work shift and is logged as Supermarket User
Flow of events:	- Employee selects "scan QR code" option - Employee frames the QR code with the application's camera and takes a picture - The device sends to the system the picture of the QR code - The system reads the QR code picture: elaborates it with an external software that retrieves the information contained in the QR code - The system checks if the costumer is allowed to get in - If the costumer is allowed to get in the system sends back to the employee's device a confirmation message - The system updates the queue - The system updates the current number of people inside the shop - The system notifies the user about the end of the operation
Exit conditions:	The system confirms that ticket is scanned successful
Exceptions:	- User is in late (time's up) - User shows up at the wrong shift - External software fails during the operation of scanning the QR code - System fails in execution

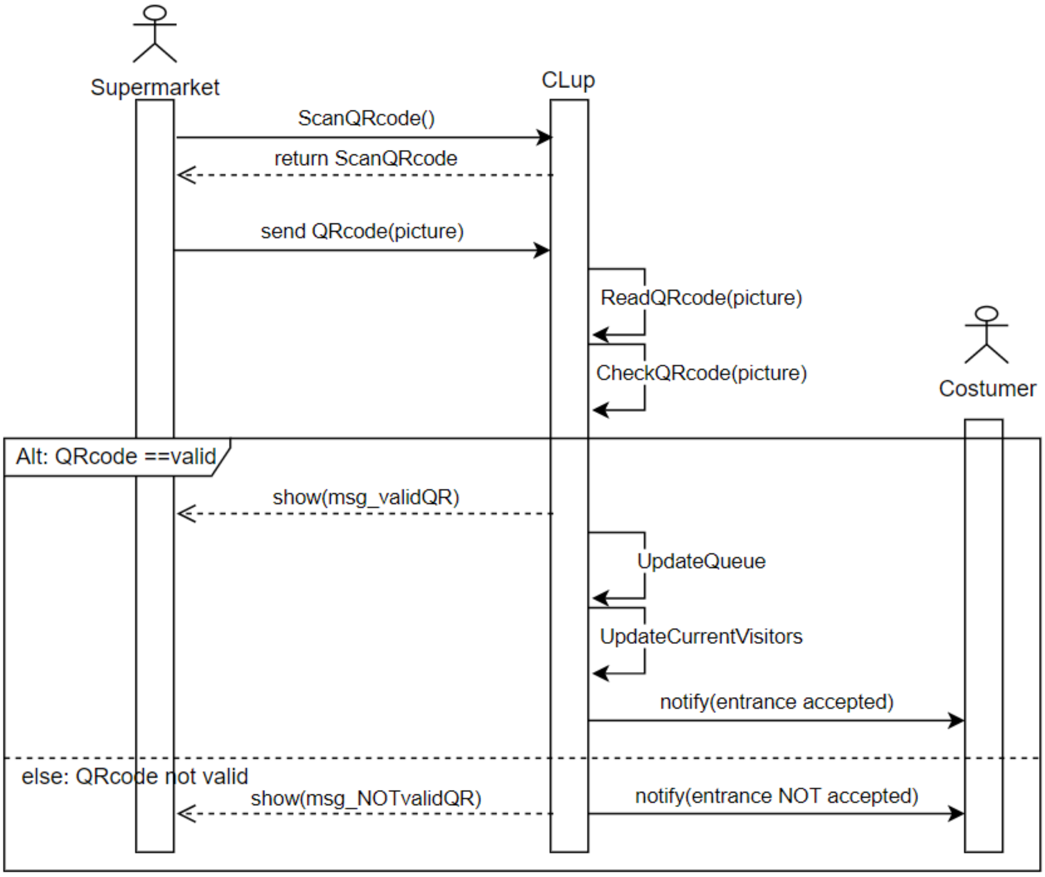


FIGURE 12 - U9: ENTRY

10) **Costumer User exits from the supermarket by scanning the QR code at a turnstile**

Name:	**Costumer exit**
Participating actors:	Every user
Entry conditions:	User has already scanned the entrance ticket User has finished his shopping
Flow of events:	- User frames the QR code with the camera of the turnstile - The QR code scanner takes a picture of the QR code - The picture is sent to the system - The system reads the information contained in the QR code with the external software - The system invalidates the QR code - The system updates the queue - The system updates the number of persons currently in the shop - The system notifies the user that his visit has finished and he is allowed to exit
Exit conditions:	The user exits from the supermarket
Exceptions:	- The QR code wasn't scanned at the entrance - External software fails during the operation - System fails

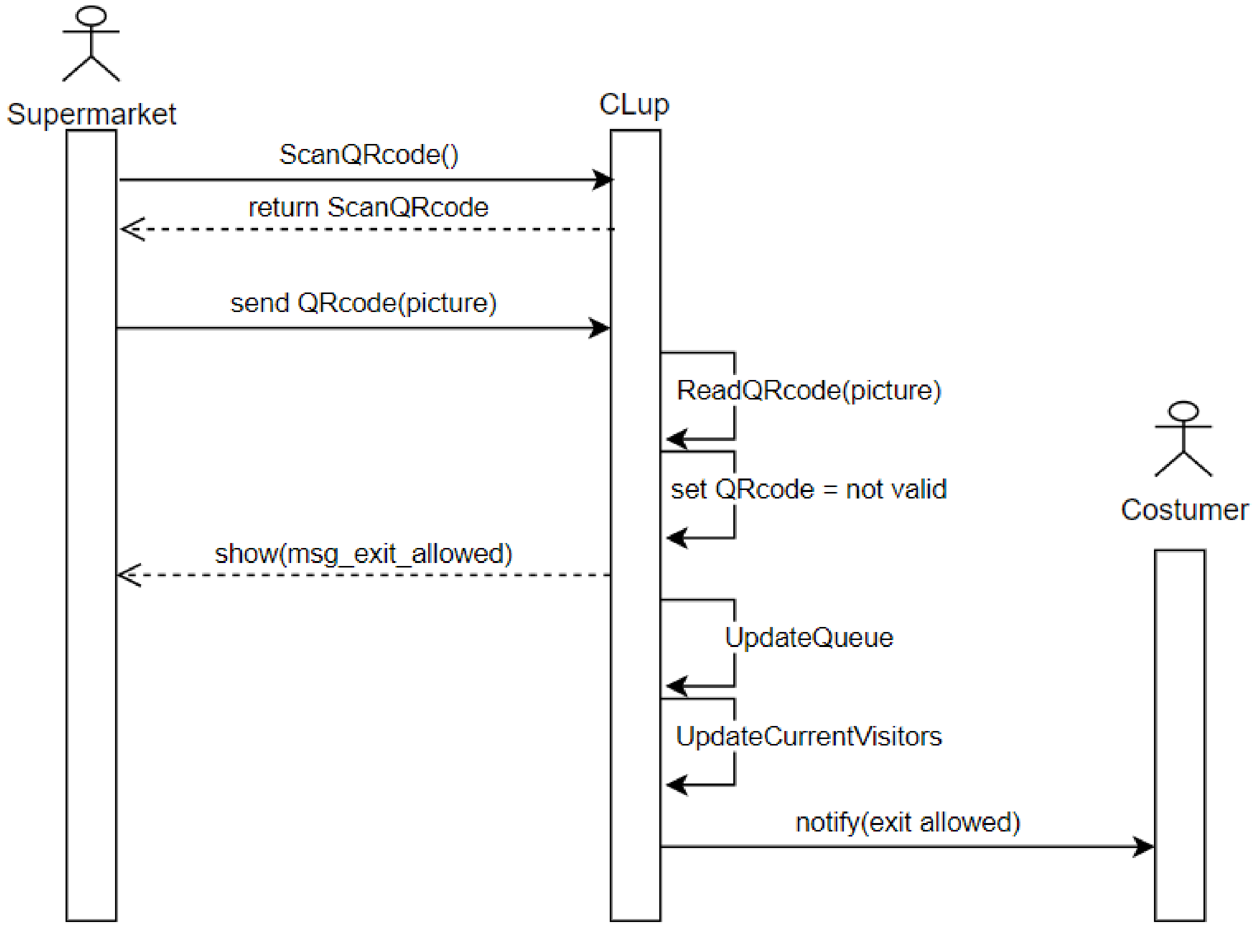


FIGURE 13 - UC10: EXIT

11) **Supermarket User observes results of statical analysis on the afflences for pre-defined time intervals.**

Name:	**Get statistical analysis**
Participating actors:	Supermarket User
Entry conditions:	Supermarket user is registered and logged in
Flow of events:	- User access the page of affluence statistical analysis - System returns the list of the available statistics - User selects a specific time interval between the availables - User visualizes information about the affluences
Exit conditions:	Statistical information are visible on the page
Exceptions:	- User is not logged in - System fails
Special Requirements:	User knows the basics of statistic to understand the usage of the tools and the meaning of the results obtained

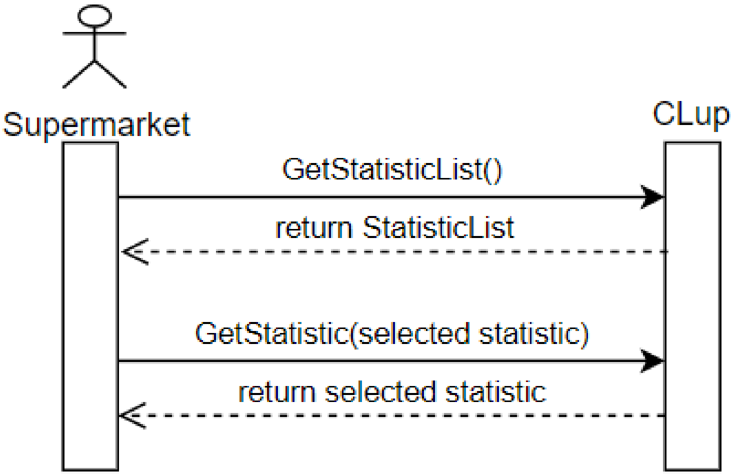


FIGURE 14 - U11: STATISTICAL ANALYSIS

12) **User goes physically to the supermarket and get in line**

Name:	**Physical Line**
Participating actors:	Supermarket User
Entry conditions:	User is registered and logged in, non-registered user goes to the supermarket
Flow of events:	- Employee logged as supermarket user selects the "Print QR code" - System generates the QR code - System set as time visit the maximum value - System update the queue - System returns the QR code to the user
Exit conditions:	The employee prints the new QR code and give it to the costumer
Exceptions:	- User is not logged in - System fails

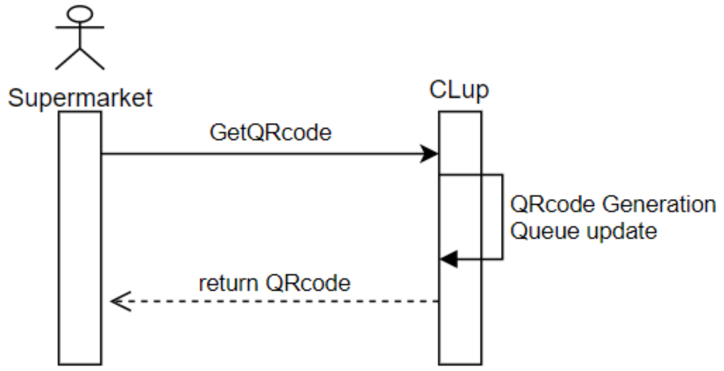


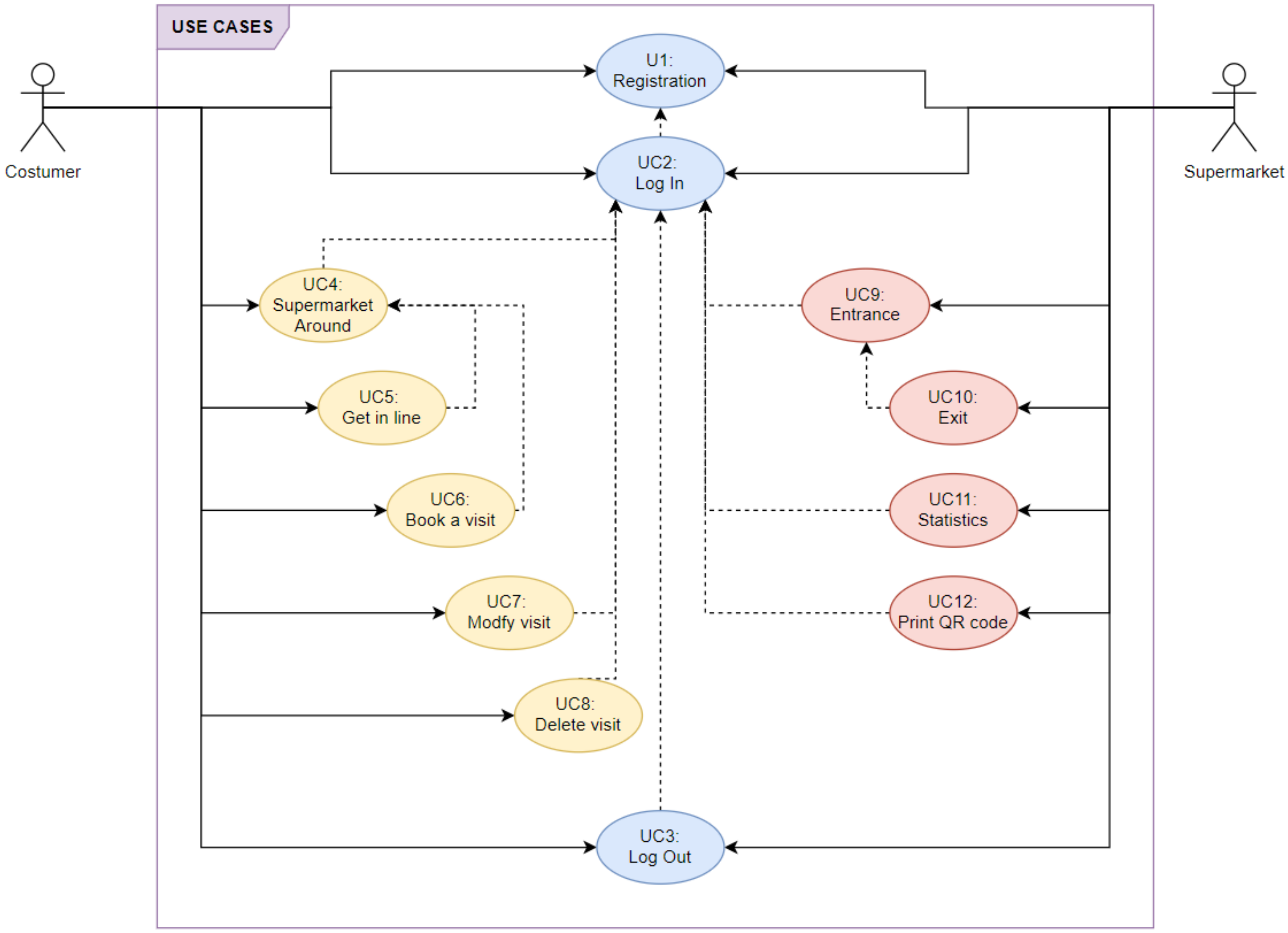
FIGURE 15 - U12: PHYSICAL LINE

3.2.4. Use Cases Diagram

The following diagram shows the relationship between the different use cases and how they are related to the different actors (Costumer and Supermarket).

The arrows connect the actor to the use cases he can perform, the dashed arrows connect use cases in relation to each other.

For istance, we can observe that no use cases from 3 to 12 can be executed if the user does not log in first and the log in cannot be performed since the user doesn't register to the application.



3.3. Performance Requirements

The system should be able to perform each function in no more than 5 seconds. It is required a response time of the system in retrieving information (for instance statistical analysis or the supermarket list) should not exceed the 3 seconds.

Moreover, the system must be able to guarantee the simultaneous connection of 100.000 individuals.

3.4. Design and constraints

3.4.1. Standards Compliance

The code of the application should strictly adhere to the standards defined in this document.

3.4.2. Hardware Limitation

The software application requires different types of hardware according to the type of User. Costumer Users must have a device with internet connection and a GPS sensor integrated. Supermarket Users require a mobile device with internet connection and a camera to scan the QR code. The statistical analysis consultation, performed by Supermarket User, requires a device with internet connection, it can be both a mobile or a laptop.

3.4.3. Any other constraints

Costumer User has limitations in executing functions:

- Costumer can queue only for a single supermarket at times (multiple queues for a single user are not accepted).
- Costumer can book only one visit per day.
- Line up requests are accepted starting half an hour before the opening of the shop until half an hour before closing.
- Costumer can book visits starting from the following day.

In addition, there must always be an employee outside from the supermarket that scans the QR code, checks the entrances and prints QR codes for the costumers that physically lines up for the shop.

3.5. Software System Attributes – Non-Functional Requirements

3.5.1. Availability and Reliability

Since the system helps in facing a national sanitary emergency, it requires a high availability: the system should be up to 99.9% of the time. This implies that the downtime, which is the average time between the occurrence of a fault and service recovery (known as MTTR – Mean Time To Recovery) should be around 8.76 hours per year.

In order to guarantee this time of downtime the system must have an appropriate infrastructure with a fully backup system.

3.5.2. Security

To register to the application, users must furnish sensitive data, for instance name, surname, e-mail and telephone number are required for costumer users' registration or the authentication cade for supermarket users. All the personal data and password must be stored in a secure way in the database. They must be hashed and encrypted before storing.

In addition, sensitive operation, such as password recovery or personal information update should not be in clear.

3.5.3. Maintainability

Purpose of maintainability is to improve effectiveness and efficiency of maintenance. The whole code should be written following the standard with a high level of abstraction. The code should be organized in different repositories in order to increase accessibility.

The code should contain comments an provide a testing routine that covers at least 80% of the whole code. Testing routine should be executed at nighttime, period in which the system is not used by the users.

3.5.4. Portability

The software should run in different environments, such as Windows OS, Linux OS and Mac OS. The mobile application should work for Android an iOS systems.

3.6. Additional Specifications

3.6.1. Grocery Types

When the customer user books a visit to the supermarket or is virtually queueing for entering, he is asked to declare the categories of items he inted to buy. The costumer will select the categories from a list of 10 types of goods. This operation allows the system to compute the average time of the user’s visit to the supermarket.

$$total\ time = \#grocery\ types\ \times 6\ min$$

The estimated time for shopping the goods for each category is 6 minutes. The shopping time ranges between a minimum time of 6 minutes to a maximum time of 1 hour. The grocery shopping types are the following:

1. Diaries
2. Meat and Fish
3. Fruit and Vegetables
4. Bread and Pastries
5. Gastronomy corner
6. Frozen Foods
7. Beverage
8. Personal Care
9. Household Products
10. Free Time

3.6.2. Statistical Analysis

Supermarket Manager, logged as Supermarket User, has the possibility to access to the “Statistical Analysis” page. This page contains:

- A counter that indicates the number of people currently inside the shop.
- A calendar that allows to select the time interval on which the system will perform the statistical analysis.

Once the user has selected the time interval the system will process the data collected for that time interval an will return to the user:

- A histogram showing the mean number of costumers for each day
- A line chart that shows the per hour affluences
- A pie chart showing the most requested types of goods

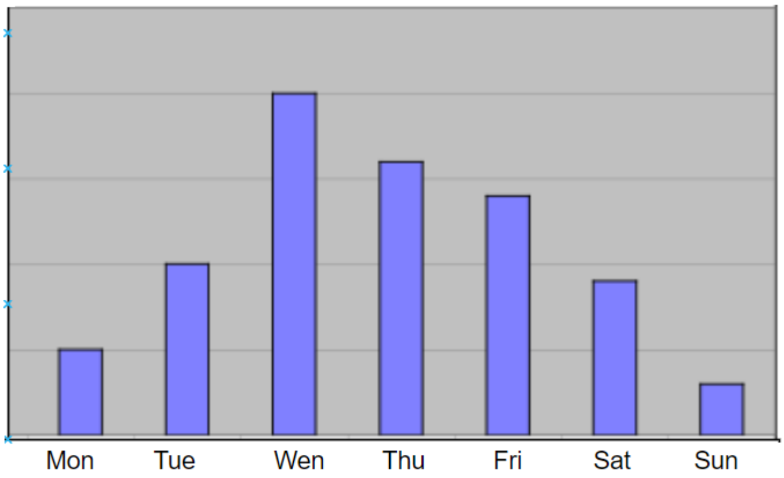


FIGURE 16 - HISTOGRAM

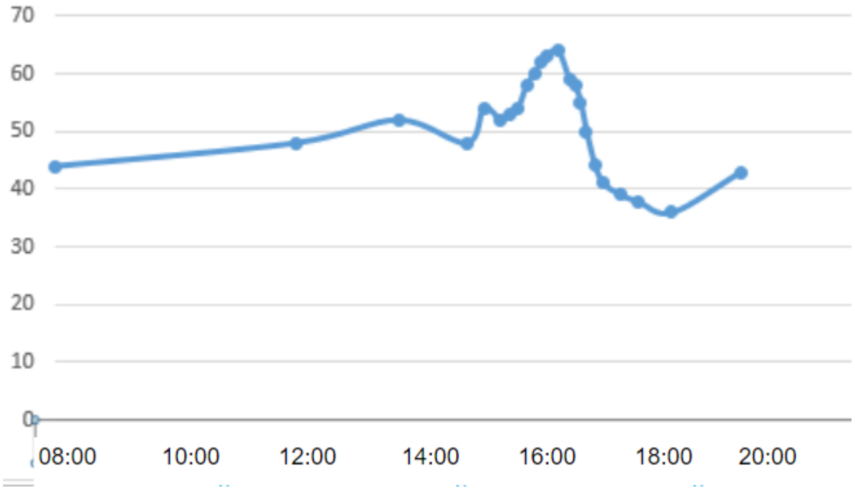


FIGURE 17 - LINE CHART

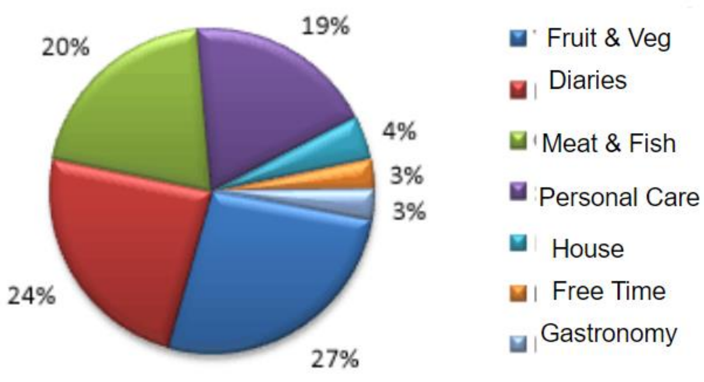


FIGURE 18 - PIE CHART

4. Alloy

This section is dedicated to the Alloy model of the Clup software, included all the functions of the application and their most important constraints. With this model we want to show:

- Possibiloty of the User to make a Book Request.
- Storage of the Book Request in the database if Validated.
- Generation of a QR code for every valid request
- All the users can do queue requests, costumers using the application through his own device and the employee out of the store for the customers that don't have technological possibilities.

In this model we assumed that all the requests done by the users are not fake.

4.1. Alloy Code

```
//The code starts with the definition of the signatures

abstract sig Position {}
sig City {}
sig Street {}

sig ExactPosition extends Position {
city: one City,
street: one Street,
civicnum: one Int
}

sig GPSPosition extends Position {
latitude: one Float,
longitude: one Float
}{  latitude.beforePoint<90 and latitude.beforePoint>-90
longitude.beforePoint<180 and longitude.beforePoint>-180
}

sig Float {
beforePoint: one Int,
afterPoint: one Int
}{  afterPoint>0  }

sig Date {
number: one Int,
month: one Int,
year: one Int  }{  number>0
month>0
year>0  }

sig Time {
hours: one Int,
minutes: one Int,
seconds: one Int
} {
hours>=0
minutes>=0
seconds>=0
}

abstract sig State{}
one sig VALIDATE extends State{}
one sig REJECTED extends State{}
one sig EVALUATION extends State{}

sig Area {
delimitedby: set ExactPosition
}

sig Code {
id: set String
}

sig IDCode{
id: set String
}

sig Supermarket {
id: one Code,
city: one City,
delimitedby: set Area
}

sig Queuenumber {
number:Int }{Int>0}
```

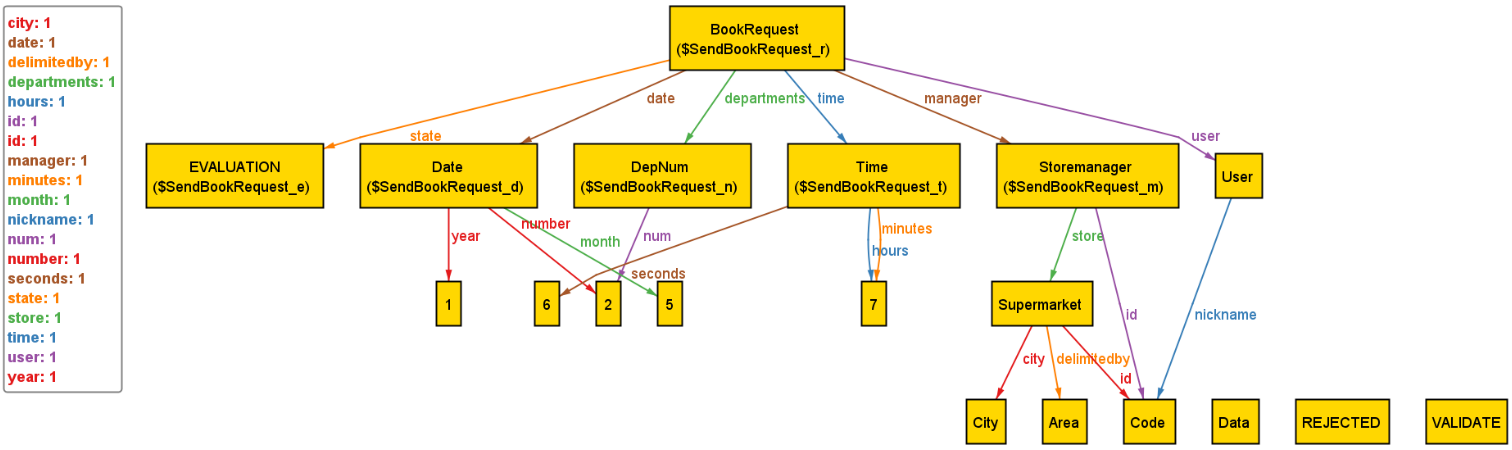
```
sig User {
nickname: one Code
}

sig QRCode {
code: one IDCode,
request: one Request,
user: one User
}

sig Storemanager {
id: one Code,
store: one Supermarket
}

sig DepNum{
num: Int
}{num>0}

abstract sig Request {}


one sig Data {
request: set Request
}

sig QueueRequest extends Request {
manager: one Storemanager,
customer: one User,
departments: one DepNum,
state: one State}{((manager=none and customer!=none) or (manager!=none and customer=none))
}

sig Suggestions {
data: one Data
}

sig BookRequest extends Request {
manager: one Storemanager,
user: one User,
departments: one DepNum,
date: one Date,
time: one Time,
guaranteedby: set Suggestions ,
state: one State }

sig RequestValidated {
request: one Request
}

sig Statistics {
data: one Data
 }

sig StatisticalRequest extends Request {
manager: one Storemanager,
guaranteedby: set Statistics
}

sig DataSent {
manager: one Storemanager,
statistics: one Statistics
}

//Once having defined all the signatures we proceed with the definition of the facts.
//All accepted Requests are in Data, all rejected Requests are not in data.
//We consider BookRequest as example, but it works for all kind of requests.

fact allAccepredRequestAreInData {
all r: BookRequest | one d: Data | r.state = VALIDATE implies r in d.request
}

fact allRejectedRequestAreNotInData {
all r: BookRequest | one d: Data | r.state = REJECTED implies r not in d.request
}

//Storemanager can retrieve only statistics related to his supermarket

fact storeManagerRetrieveCorrectData{
all d : DataSent | one s : Storemanager | d.manager = s
}
```

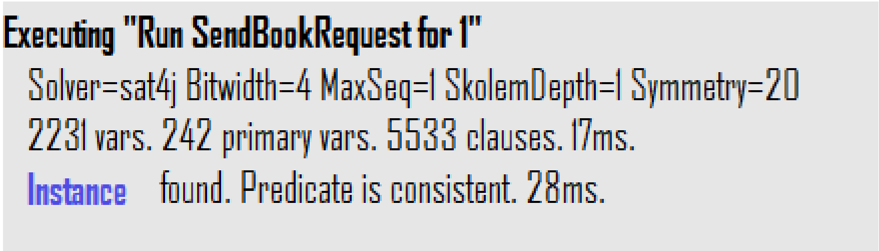
```
//To each QR code correspond a unique request

fact oneQRCodeToEachRequest {
all q : QRCode | one r: BookRequest | q.request = r
}

//To each supermarket correspond and unique Area

fact oneAreaToEachSupermarket {
all s : Supermarket | one a : Area | s.delimitedby = a
}


//Predicates


// Costumer user send a BookRequest

pred SendBookRequest [r: BookRequest, d: Date, t: Time, m: Storemanager, n: DepNum, e:
EVALUATION] {
r.state=e
r.date=d
r.time=t
r.manager=m
r.departments=n
}
//run SendBookRequest for 1



//The request is validated

pred Validation(u:User, r:BookRequest, d:Data, rv: RequestValidated) {
r.state=VALIDATE
r.user=u
r in d.request
r in rv.request
}
//run Validation for 1
```

```
//A QR code is generated for each valid request
pred QRCodeGeneration(r:BookRequest, q: QRCode, c: IDCode ){
r.state = VALIDATE
q.request = r
q.user = r.user
q.code = c
}
// run QRCodeGeneration for 1
```

4.2. Execution Results

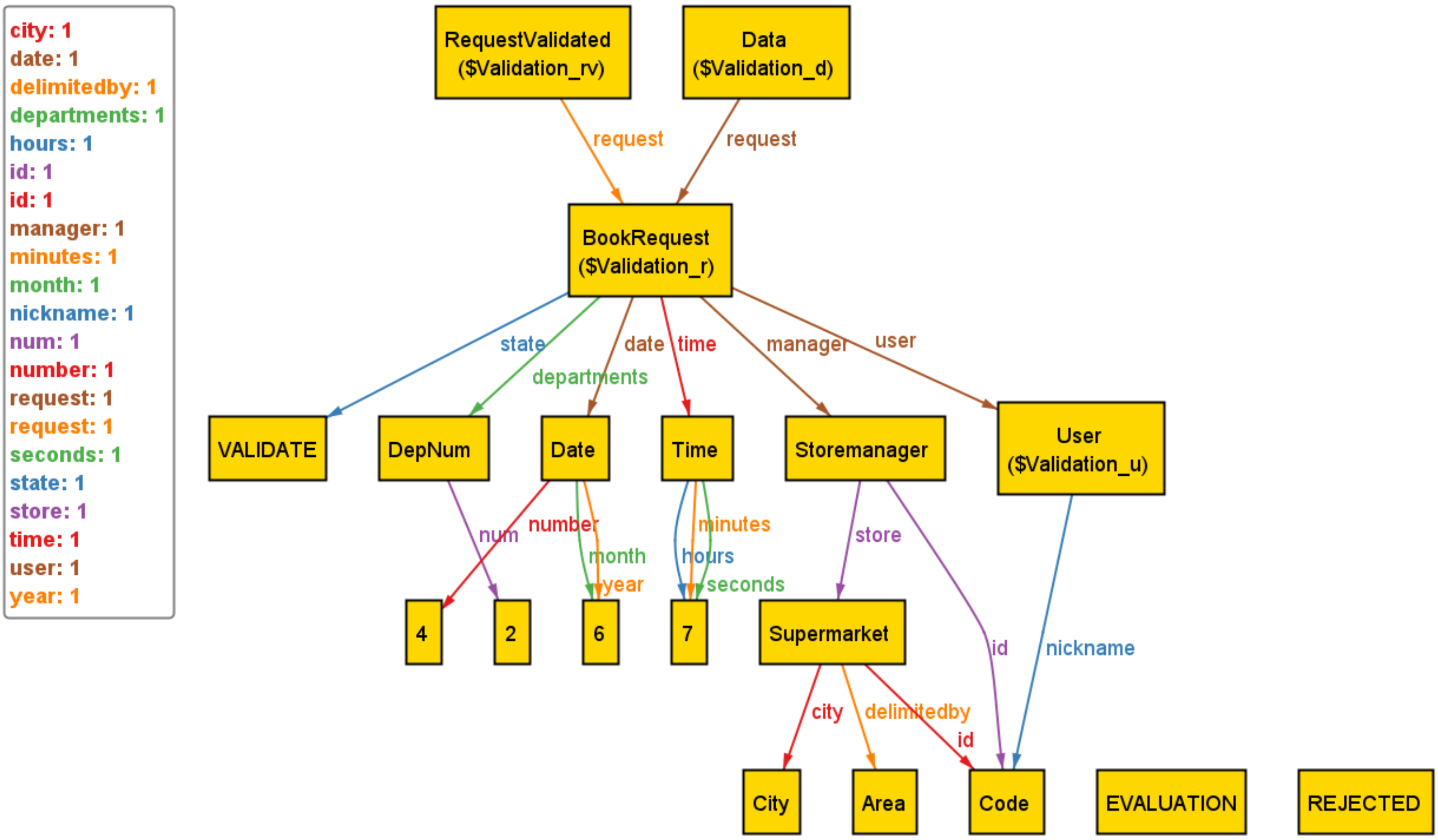
Predicate: SendBookRequest

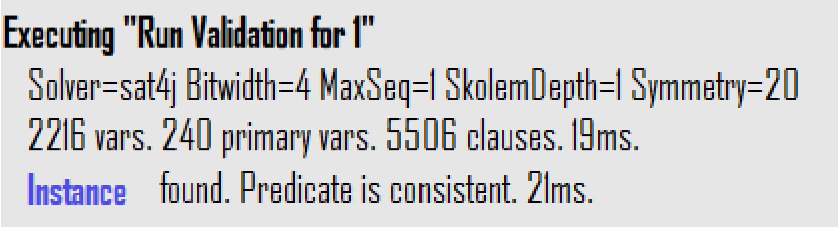




```
Executing "Run SendBookRequest for 1"
   Solver=sat4j Bitwidth=4 MaxSeq=1 SkolemDepth=1 Symmetry=20
   2231 vars. 242 primary vars. 5533 clauses. 17ms.
   Instance   found. Predicate is consistent. 28ms.
```

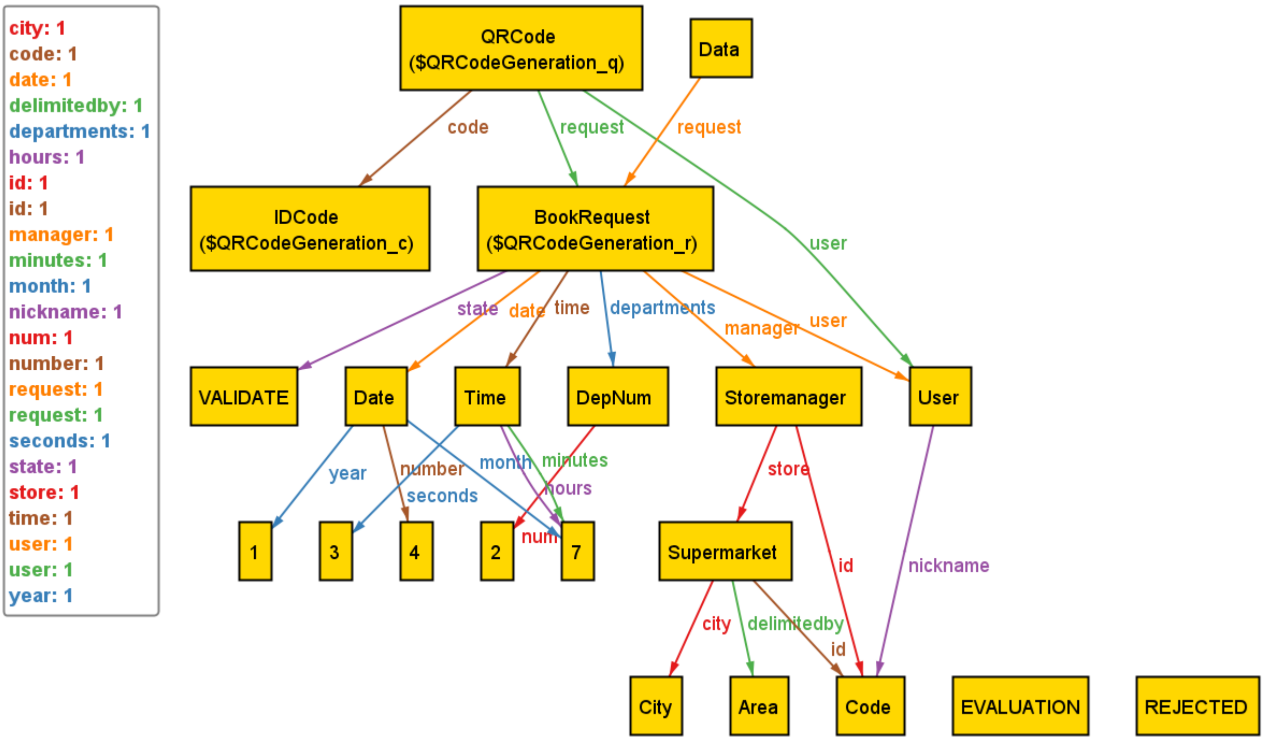
Predicate: Validation

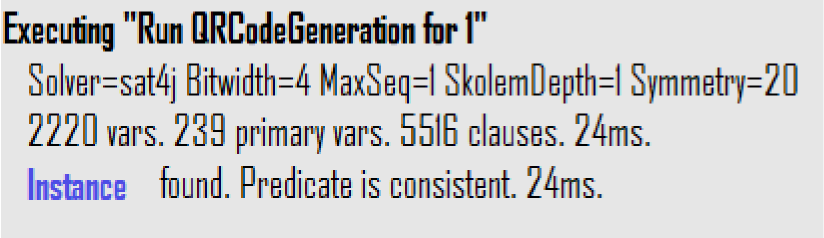




```
Executing "Run Validation for 1"
   Solver=sat4j Bitwidth=4 MaxSeq=1 SkolemDepth=1 Symmetry=20
   2216 vars. 240 primary vars. 5506 clauses. 19ms.
   Instance  found. Predicate is consistent. 21ms.
```

Predicate: QRCodeGeneration





```
Executing "Run QRCodeGeneration for 1"
   Solver=sat4j Bitwidth=4 MaxSeq=1 SkolemDepth=1 Symmetry=20
   2220 vars. 239 primary vars. 5516 clauses. 24ms.
   Instance  found. Predicate is consistent. 24ms.
```

5. Effort Spent

Alghisi Maarianna

TOPIC	HOURS
Discussion on first part	3h
Purpose and scope	3h
Phenomenas and goals	4h
Discussion on second part	3h
Product functions	2h
Statecharts	1h
Discussion on third part	3h
External Interface Requirements	4h
User Interface Mockups	2h
Requirement List and mapping	5h
Scenarios	1h
Use cases and UC diagrams	12h
Non-functional requirements	1h
Specification on grocery types and statistical analysis	1h
Alloy Code	8h
Document Composition	6h

Gabriele D'Ascoli

TOPIC	HOURS
Discussion on first part	3h
Domain assumption	2h
Product functions	3h
UML Diagram	6h
Discussion on second part	3h
UML description	5h
Statecharts	1h
State diagram	2h
Discussion on third part	3h
User characteristics	2h
Software System Attributes	3h
Document composition	2h
Alloy code	6h

Martina Pasturensi

TOPIC	**HOURS**
Discussion on first part	3h
Domain assumption	1h
UML Diagram	1h
Discussion on second part	3h
UML description	1h
Statecharts	1h
Discussion on third part	3h
Requirements	4h
Design and constraints	3h
Attributes and Non-Functional Requirement	1h
Use cases	4h
Document composition	2h
Alloy	8h

6. References

- All the diagrams have been made with https:// www.draw.io
- Alloy code development has been supported by org.alloytools.alloy.dist (Alloy Analyzer 5.1.0)