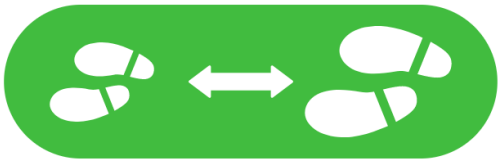
POLITECNICO
MILANO 1863

Computer Science and Engineering

Software Engineering 2 Project

Requirement Analysis and Specification Document

Version 2.0 - 08/01/2021



Customers Line-up

STAY SAFE

Authors

Enrico Gherardi
Ludovico Righi

Professor

Elisabetta Di Nitto

Tutor

Annalisa Binato

Contents

List of Figures

Chapter 1

Introduction

This chapter explains the universe where Customers Line-up application is located, all the goals, purposes and scopes of the system and gives some keys of interpretation of the document (abbreviations, definitions..).

1.1 Purpose

This document represents the Requirement Analysis and Specification Document. Goals of this document are to completely describe the system in terms of functional and non-functional requirements, analyze the real needs of the customer in order to model the system, show the constraints and the limit of the software and indicate the typical use cases that will occur after the release. This document is addressed to all the stakeholders, in particular the grocery shopping owners to understand the main features of the product and the developers who have to implement the requirements and could be used as a contractual basis.
This Section focuses on the description of the purpose of the system in the section 1.1.1 and lists all the goals in the section 1.1.2

1.1.1 General Purpose

Customer Line-Up is a crowdsourced application whose purpose is to create a service to support people, specially in lockdown time, in doing grocery shopping. The aim is, on the one side, to allow store managers to regulate the influx of people in the building and, on the other side, to save people from having to line up and stand outside of stores for hours on end.

Customer Line-Up is very simple to be usable by everyone. The software's main features are to allow customer to line-up from their home and to notify them when they have to go towards the store according the time they need to get to the supermarket. In addition, the system generates QR codes that would be scanned at the entrance and the exit of the store, thus allowing store managers to monitor entrances.
Moreover, CLup allows customers to book a visit to the supermarket indicating when and the approximate expected duration of the visit or, only for long-term users, this last one can be inferred by the system. In addition, the customer can indicate, if not the exact list of items that they intend to purchase, the categories of items that they intend to buy to allow the application to plan visits in a finer way.

1.1.2 Goals

In this section we indicate the main Goals of CLup, which are the high level objectives described in terms of Word Phenomena or Shared Phenomena controlled by the world, which will be presented in the next section.

- G_1 : Customers can line-up from their home to access the supermarket.
- G_2 : Customers can book in advance a visit to the supermarket.
- G_3 : Customers are notified when they have to go towards the store according the time they need to get to the supermarket.
- G_4 : Customers entrances are monitored and managed by the system.
- G_5 : Store managers can hand out tickets on the spot for customers who do not have access to the required technology.

1.2 Scope

This section lists World, Shared and Machine Phenomena in order to identify in a precise way the environment in which the application operates.
World Phenomena

- W_1 : Customers arrive to the supermarket.

- W_2 : Customers enter the supermarket.
- W_3 : The store managers hand out tickets on spot.

Shared Phenomena

Controlled by the World and observed by the Machine:

- S_1 : Customers make line-up requests.
- S_2 : Customers indicate with which mean of transport they intend to go the supermarket.
- S_3 : Customers scan their QR codes.
- S_4 : Customers book visits.
- S_5 : Customers input the duration of the visits.
- S_6 : Customers input items to buy or their categories.

Controlled by the Machine and observed by the World:

- S_7 : The system shows customers their waiting time.
- S_8 : The system notifies customers when they have to start.
- S_9 : The system generates tickets for the for customers who don't have the required technology.
- S_{10} : The system sends QR codes to customers.
- S_{11} : The system retrieves the GPS position of the customer.
- S_{12} : The system increases entrances monitoring the affluence of the departments.

Machine Phenomena

- M_1 : The system manages real-time queue.
- M_2 : The system calculate the waiting time for the single customer.
- M_3 : The system generates QR codes.

- M_4 : The system optimizes the planning of visits.
- M_5 : The system infer the visit time of long-term users.

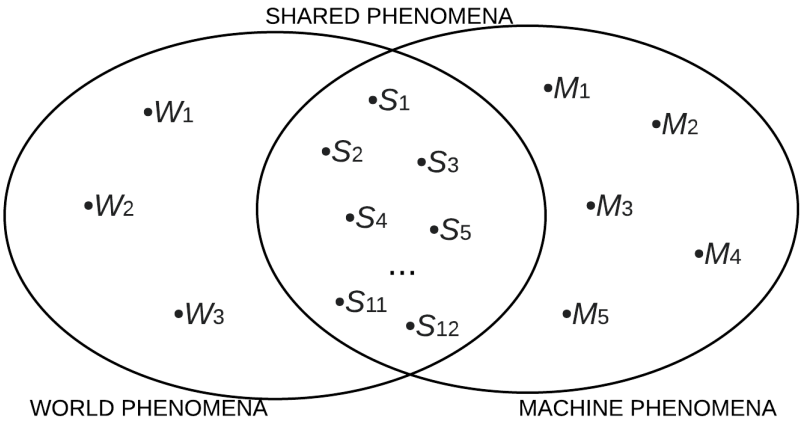


Figure 1.1: World, Shared and Machine Phenomena

1.3 Definitions, Acronyms, Abbreviations

This section gives some definitions in 1.3.1, acronyms in 1.3.2 and abbreviation in 1.3.3 which will be use in the document, in order to explain some concept and help the general understanding.

1.3.1 Definitions

- Customer: the user of the application who can line-up from his home or can book a visit.
- No-Tech Customer: the Customer of the supermarket that, for external reasons, cannot have an access to CLup in any way and has to enter the store with a ticket.
- Long-Term Customer: the customer who entered the supermarket using CLup at least 10 times.
- Store Manager: the user of the application that can monitor the affluence of his store, have to report to the system all the arriving No-Tech Customers and have to give them the tickets to enter.

- Waiting Time: the time the Customer have to wait before entering the supermarket.
- Real-Time Queue: the queue of all the Customers who had lined-up, who had booked and those without the required technology for a specific supermarket.
- Inferred Duration: the weighted average over the days and hours of the duration of a Long-term Customer's previous visits.
- Line-up Request: the request of a Customer for lining-up from his home in the selected supermarket.
- Booking Request: the request of a Customer to book a Visit in a specific supermarket.
- Visit: the result of a Booking Request.
- Ticket: a number generated by the System that permits to No-Tech Customers to enter the store.

1.3.2 Acronyms

- $RASD$: Requirement Analysis and Specification Document.
- API: Application Programming Interface.
- GPS: Global Positioning System.
- $GDPR$: General Data Protection Regulation.
- $MTTR$: Mean Time To Repair.

1.3.3 Abbreviations

- $CLup$: Costumers Line-up.
- W_N: World Phenomenon number N.
- S_N: Shared Phenomenon number N.
- M_N: Machine Phenomenon number N.

- G_N: Goal number N.
- R_N: Requirement number N.

1.4 Revision History

- Version 1.0 : 23/12/2020
- Version 2.0 : 08/01/2020
  In this version the following changes have been performed:
  - Updated definition of RealTimeQueue.
  - Updated Class diagram and added description.
  - Updated Statechart diagrams and added description.
  - Updated Product Functions.
  - Updated Text and Domain assumptions.
  - Added descriptions to Sequence diagrams.
  - Added connection between scenarios and corresponding use case.
  - Updated Effort Spent.

1.5 Reference Documents

- Specification document: "R&DD Assignment A.Y. 2020/2021".
- Lecture slides of professor M.Rossi and E.di Nitto of Politecnico di Milano.
- IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications

1.6 Document's Overview

The RASD document is structured by five chapters as describes below:

- Chapter 1: it represent an introduction. It is formed by a general description of the purpose of the application and a list of the goals which the system has to achieve. In addition, it presents a detailed description of the software's functionalities, specifying, in particular, the application domain through the "World and Machine phenomena" criterion.

- Chapter 2: it contains an overall description of the project such as a detailed characterization of the software's main functions. It presents also the general factors that can affect the system's behavior, such as user characteristics and project constraints.

- Chapter 3: it represent the body of the document. It contains the main requirements of the software, and the mockups related to the two different interfaces of the system. It lists some scenarios to show how the system acts in real world situations. They are followed by all the requirements necessary to reach the given goals and linked with the related domain assumptions. Lastly, the non-functional requirements are defined through performance requirements, design constraints and software system attributes.

- Chapter 4: it contains the Alloy model of some important aspects, with all the related comments and documentation in order to show in a formal way how the project has been structured.

- Chapter 5: it contains the effort, in terms of hours, which each member of the group spent working on the project.

- Chapter 6: it contains the list of any other documents or Web addresses to which this RASD refers.

Chapter 2

Overall Description

This chapter shows the system from an high level point of view.
The section 2.1 illustrates the perspective of the product and the section 2.2 describe its functions. The section 2.3 lists the actors of the application, while the sections 2.4 and 2.5 specify the constraints, assumptions and dependencies of the entire system.

2.1 Product Perspective

CLup is a new product whose intent is to help people and store managers during this pandemic caused by Covid-19. In particular, it is a self-contained product: it includes almost all subsystems needed to fulfil the requirements defined in this document, except for two APIs provided by Google, which will be briefly described later. However, any detailed definition of an external system is out of scope of this document.
In the following two subsections it is defined the domain through different models: in particular, the section 2.1.1 shows the Class Diagram, while in the section 2.1.2 the most relevant Statecharts Diagrams are represented.

2.1.1 Class Diagram

The following Class Diagram shows the main entities of the problem, from the requirements point of view. A more detailed and implementation-oriented Class Diagram will be provided in the Design Document.

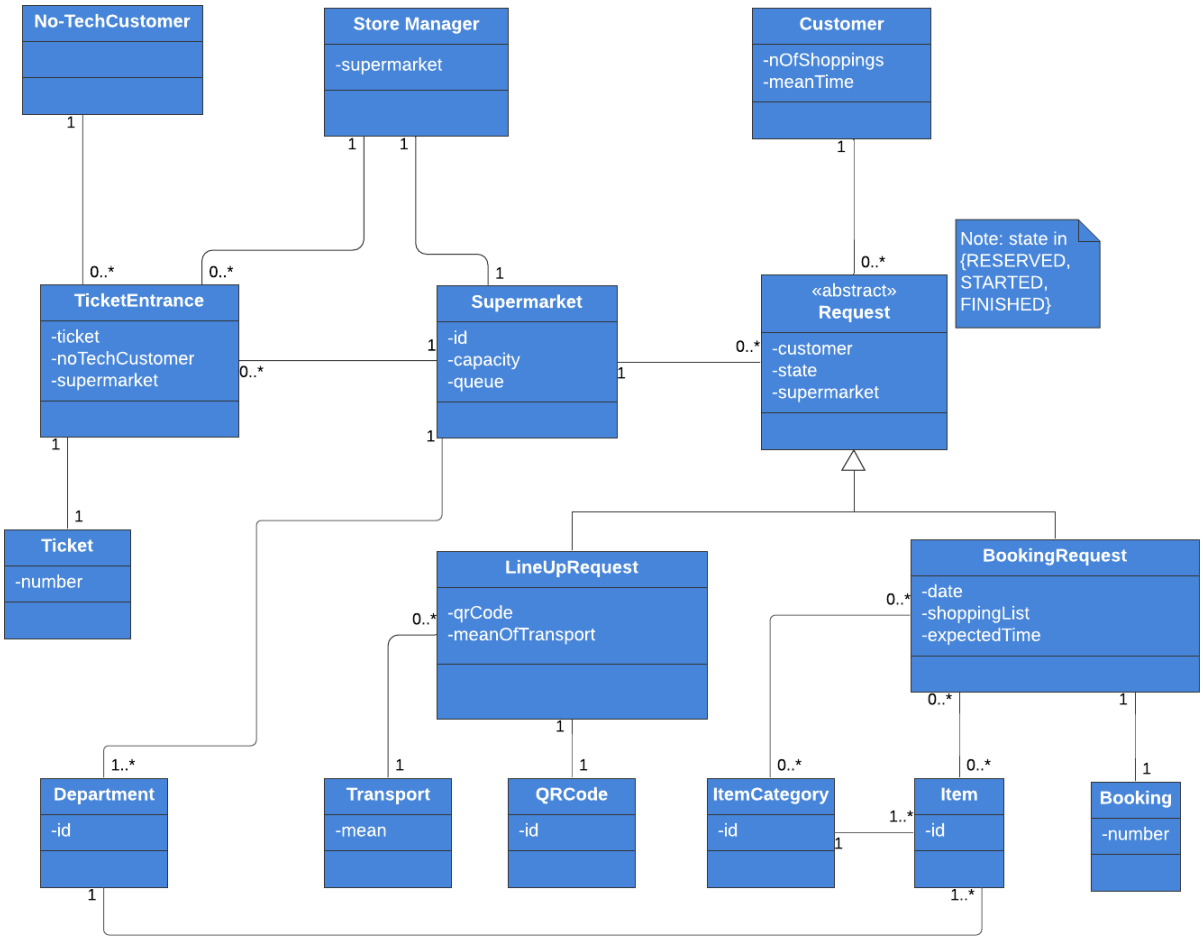


Figure 2.1: Class Diagram

2.1.2 Statechart Diagrams

Request State Diagram

This state diagram shows the three states of a Reuquest: the Request is always created ad Reserved and then he can pass into the Started state or into the Finished one, depending on whether he arrives at the supermarket.

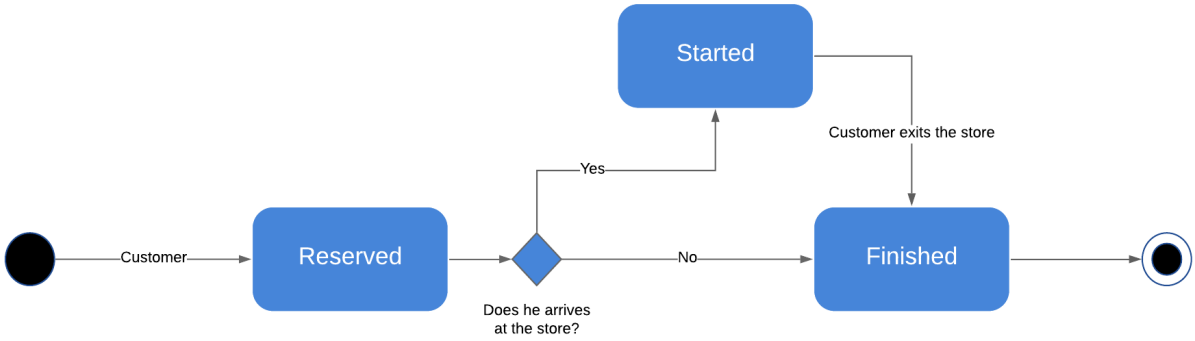


Figure 2.2: Request State Diagram

Ticket Entrance State Diagram This state diagram shows the different state of the entrance of a NoTechCustomer: when he arrives at the market, he is Reported by the StoreManager. If the supermarket is not full he can enter immediately the supermarket; otherwise, he has to wait for his waiting time.

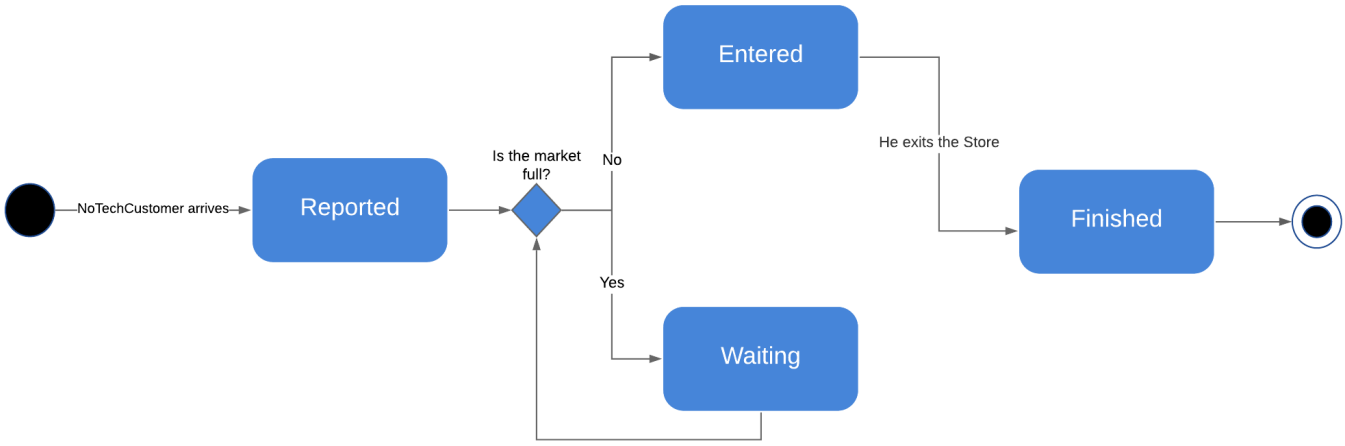


Figure 2.3: Ticket Entrance State Diagram

2.2 Product Function

This section lists and describes the main functions of Customer Line-Up.

2.2.1 Line-Up

This part of the whole system manages the action of making a Line-Up Request for a specific supermarket. The Customer uses the mobile phone interface to make a Line-up Request indicating the involved supermarket and the mean of transport he intends to use to reach it. Then, the system add the Customer in the Real-Time Queue of that store and shows him his Waiting Time. In addition, the system sends to the Customer a QR code that will be scanned both at the entrance and at the exit to monitor the affluence and to update the Real-Time Queue. For preventing Customers to arrive both too early or too late, the QR Codes have a validity which spans in an interval of time of 5 minutes with respect to the entrance time set by the system.
Finally, the system calculates the time needed by the Customer to reach the supermarket and sends him a notification when he has to start. It is important to underline that the system allows the Customer to make a single Line-Up Request at a time to permit a more fair distribution of the timeslots available. Also, for usage semplicity reasons, it permits only to cancel and not to modify the Request. Moreover, if the Line-Up Request is sent when the supermaket is closed or if the entrance time foreseen by the Waiting Time is after the closing time of the supermarket, the Request will be postponed to the first available slot.

2.2.2 Booking

This part of the whole system manages the action of booking a Visit for a specific supermarket. The Customer uses the mobile phone interface to make a Booking Request indicating the involved supermarket, the date and time of the Visit and its Expected Duration. If the Customer is a Long-Term one, the system can mine this last parameter from previous data.
Additionally, the Customer can indicate the exact list of items he intends to buy or their categories to enable the system to make a finer optimization allowing more people in the store.
The system shows only dates and times when the supermarket is open and the number of its Bookings is lower than its capacity. It is important to underline that the system allows the Customer to make a single Booking Request at a time to permit a more fair

distribution of the timeslots available. Also, for usage semplicity reasons, it permits only to cancel and not to modify the Request.

2.2.3 Entrance with Ticket

This modality is permitted only if the Customer does not have the possibility to reach the technology needed, for example, he has not the ability to use a smartphone or has not the financial possibility to buy it. When a No-Tech Customer arrives at the supermarket, the Store Manager uses his dedicated portal to report him to let the system update the queue. Then, if the supermarket is not full, the Store Manager hand out the ticket to the No-Tech Customer which can immediately enter; otherwise, the system generates the Ticket with the waiting time, which indicates when the No-Tech Customer is allowed to enter. Finally, the Store Manager has to indicate to the system when the No-Tech Customer exits the supermarket to update the number of people inside it.

2.2.4 Monitoring the Affluence

The system provides to the Store Managers a dedicated portal where they can check in real time the number of people inside their supermarket and the list of all the Bookings.

2.3 User Characteristics

The actors of the application are the following:

- Customer: a person who uses CLup to line-up or book a visit to a supermarket.
- Store Manager: the owner of a supermarket who takes advantage of CLup for monitoring the affluence of his store and for providing tickets to No-Tech Customers.

It is very important to underline the fact that users do not need to have any particular skill: since everyone needs to do the grocery shopping, the application it is thought to be extremely simple so that everyone can use it without problems.

2.4 Constraints

The current constraints on the project are related to the provision of hardware resources to run some of the software's main features.
Firstly, to guarantee the effectiveness of the Real-Time Queue, when Customers want to send a Line-Up Requests, they must provide their geographical position: the devices without a GPS are excluded from the lining-up mechanism. The same exclusion is also for Customers who, even if they have devices provided with a GPS, did not grant the localization permission to the application. Clearly, the system will have to ask for Customers' permission in order to retrieve and use their positions and will store those informations only for the period of time strictly needed, discarding them as soon as the Customer scans the QR code.
Secondly, all the services offered by the system are accessible only through an Internet connection, which is obviously indispensable for the different functionalities.
Finally, all the devices with processors that does not support the execution of inference algorithms cannot run the function that predicts the expected duration of the visit for Long-Term Customers.

2.5 Assumptions and Dependencies

In the specification document presented by the client we found some points that lacked in precision and would have led to some ambiguities. In order to better clarify those situations we decided to introduce the following assumptions.

2.5.1 Text Assumptions

- Every time that a Customer makes a Line-Up Request, he has to input the mean of transport he intends to use to reach the supermarket in order to be notified exactly in time when he has to start.
- Only a small percentage (2-5%) of Customer does not have the possibility to have the required technology.
- All the Customers who have lined-up have to scan the QR Code both when entering and exiting the supermarket in order to monitor the affluence.

- The expected duration of a visit has to be indicated for every Booking for non Long-Term Customers.
- The expected duration of a visit has to be indicated also for Long-Term Customers if it is different from the infered one.
- The indication of the list of items or their categories is optional, it can be done by Customers who want to help supermarkets to optimize their affluence.
- Customers who have indicated what they are intentioned to buy in their Booking are committed to do not go in other departments.

2.5.2 Domain Assumption

- D_1 : There is stable Internet connection.
- D_2 : The GPS provides the exact location with an error of 100 meters at most.
- D_3 : All the Customers, both those who have booked or lined-up, arrive at the supermarket.
- D_4 : The Customers arrive at the supermarket on time with an error of 5 minutes at most.
- D_5 : All the No-Tech Customers must obtain a ticket before entering.
- D_6 : Customers who have booked stay in the supermarket for the time they have indicated with an error of 5 minutes.

Chapter 3

Specific Requirements

This chapter focuses on requirements of the system. The section 3.1 lists all the external interfaces' requirements, the section 3.2 lists the functional requirements, the section 3.3 tells about performance requirements, the section 3.4 explains the design constraints and the final section 3.5 lists the attributes of the system.

3.1 External Interface Requirements

This section focuses on external interfaces' requirements, in particular the user, hardware and software ones.

3.1.1 User Interfaces

CLup contains two user interfaces: the first one is the mobile application, which is used by Customers to make Line-Up Requests and Bookings. The other one is the website interface, used by Store Managers to check the supermarket affluence and to request tickets for No-Tech Customers. The following mockups shows the idea of how the system should work. Further mockups will then be provided in the Design Document.

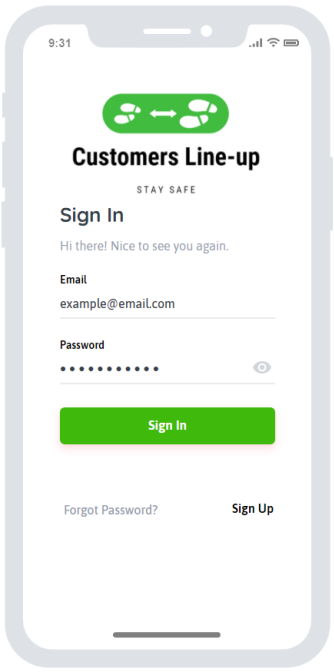


Figure 3.1: Login

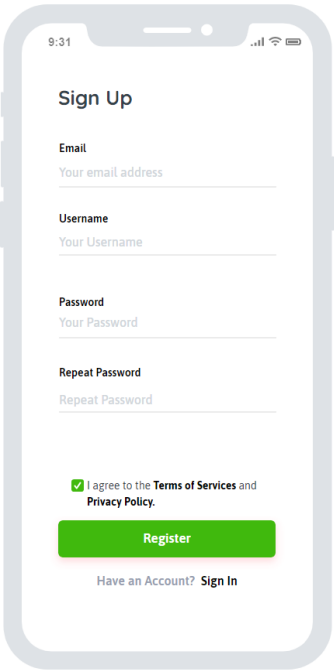


Figure 3.2: Registration

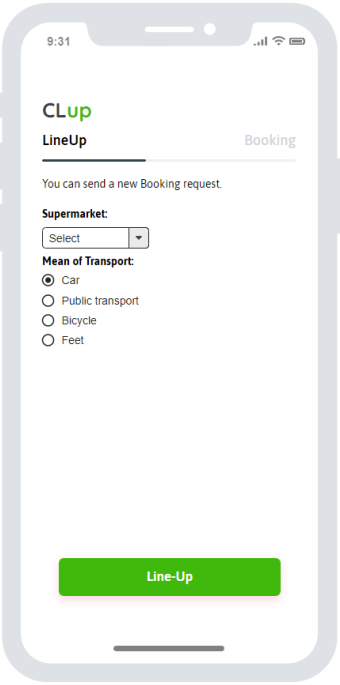


Figure 3.3: New Line-Up

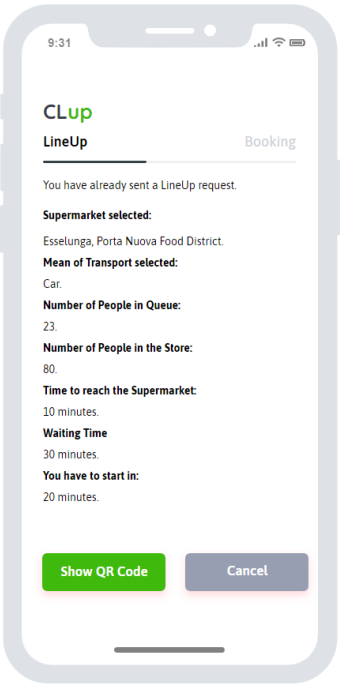


Figure 3.4: Line-Up sent

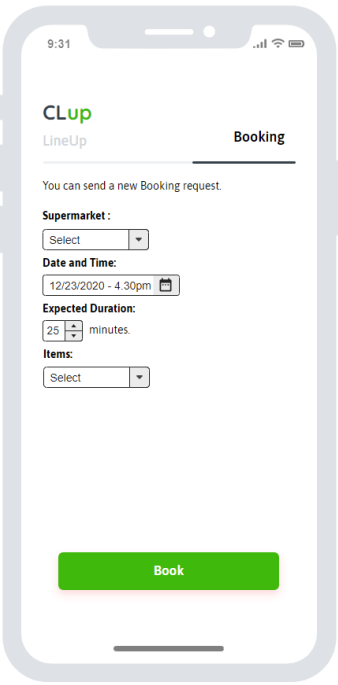


Figure 3.5: New Booking

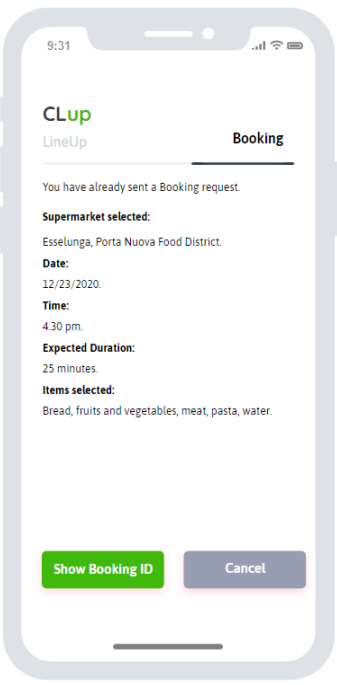


Figure 3.6: Booking sent

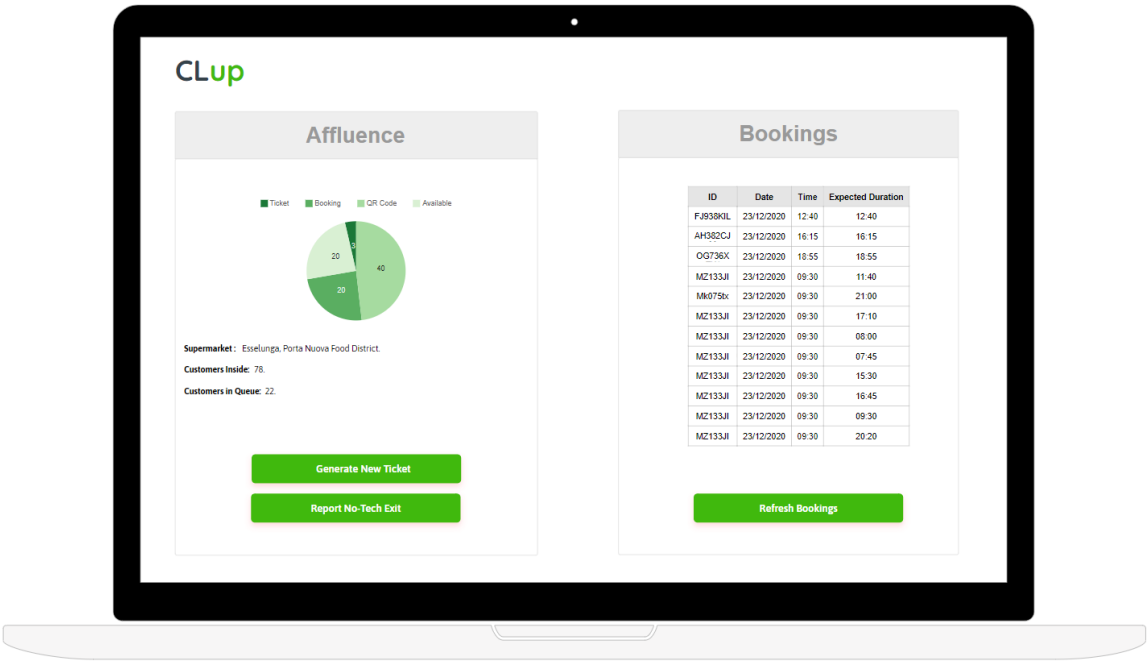


Figure 3.7: WebApp for Store Managers

3.1.2 Hardware Interfaces

The system requires a QR Code Scanner which is used to check the actual entrance and exit of the Customers. Every supermarket must be provided with two of these scanners, one placed at the entrance and the other at the exit.

3.1.3 Software Interfaces

The system uses the Google Map's APIs, in particular:

- GOOGLE MAPS GEOLOCATION API: to find the exact locations of the customers.
- GOOGLE MAPS DISTANCE MATRIX API: to get informations about the recommended route and the duration of the trip (driving, public transportation, walking or cycling) while considering the current traffic situation.

3.2 Functional Requirements

This section lists all the system requirements of CLup and their mapping with the goals. To better explain all the requirements the section starts with a list of use cases and scenarios for both Customers (3.2.1) and Store Manager (3.2.2) followed by a Use Case Diagram.

3.2.1 Customer

Scenario 1: Line-Up
Francesco is a young man who wants to take care of his grandparents during this lockdown period. He decides to grocery shop for them, but since he will have to be in contact with them he wants to minimize the risks of contagion. Luckily, he remembers that a friend told him of Customer Line-up and suddenly downloads it to line-up in the his grandparent's favorite grocery shop.
Scenario 2: Booking
Alessandra is the mother of two children, Luca and Paolo, and she is also a busy Product Manager. Her presence is fundamental for her company and she also wants to protect her family so Customer Line-Up is her favorite application to grocery shop in Milan. However, she has very few slots of time where she can grocery shop, so she finds the booking functionality of CL-Up very helpful.
Scenario 3: Entrance with ticket
Valentino is a 80 years-old man who is completely capable of taking care of himself. For him, it is very necessary to grocery shopping completely safe, but he has not the required technological skills to use a smartphone and to download Customer Line-Up. Therefore, every time that Valentino goes to his favorite supermarket the store manager give him a physical ticket to enter in total security. The store manager report Valentino to the system to update the real time queue and the waiting time of all his customer.
Scenario 4: Unexpected Event after a Line-Up request
Beatrice is a young woman that uses regularly Customer Line-Up to grocery shopping. She is a veterinarian and she must always be ready for emergency calls. One day, while she was going to her favorite shop, she received a call from an old client of hers saying that his dog has eaten some chocolate. Although she had a waiting time for the supermarket's entrance of only 10 minutes, she opened CLU and deleted her

Line-Up request to permit to the system to update the real-time queue and to offer a more precise service to its customers.

Name	Customer Registration
Actor	Customer
Entry Condition	Customer enters the Login page.
Event Flow	1. The Customer clicks on the "Sign In" button in the Login page to start the Registration process. 2. The Customer fills in all the mandatory fields. 3. The Customer clicks on the "Register" button. 4. The System saves the data. 5. The System sends and email to the new Customer with the confirmation.
Exit Condition	The Customer successfully ends the registration process and become a new User. From now on he/she can log in to the application providing his/her credentials and start using CLup.
Exceptions	1. The Customer is already registered. 2. The Customer inserts not valid informations in one or more mandatory fields. 3. The Customer chooses an username that has already been taken by another user. 4. The Customer chooses an email that has been associated with another user. 5. The Customer has not filled in one or more mandatory field. All exceptions are handled notifying the issue to the Customer and taking back the Event Flow to the point number 2.

Name	Customer Login
Actor	Customer
Entry Condition	The Customer has already registered.
Event Flow	1. The Customer inserts the username. 2. The Customer inserts the password. 3. The Customer clicks on the "Login" button.
Exit Condition	The Customer successfully logs in and have access to the CLup services.
Exceptions	1. The Custom has not filled in both the username and the password. 2. The Customer inserts an email which is not saved in the database. 3. The Custom inserts a wrong password. All exceptions are handled notifying the issue to the Customer through a warning message.

Name	Line-up
Actor	Customer
Entry Condition	Customer has already logged in.
Event Flow	1. The Customer chooses the supermarket in which he wants to go. 2. The Customer indicates with which mean of transport he intends to go to the supermarket. 3. The Customer clicks on the "Line-up" button. 4. The System inserts the request in the queue. 5. The System shows the Customer his Waiting Time.
Exit Condition	The Custom has successfully sent his line-up request.
Exceptions	1. The Customer has not chosen the supermarket. 2. The Customer has not indicated his mean of transport. 3. The Customer device is not provided with a GPS or the Customer has not given to CLup the permission to ulitize it. All exceptions are handled notifying the issue to the Customer through a warning message.

Name	Cancel Line-up
Actor	Customer
Entry Condition	Customer has already logged in and lined-up.
Event Flow	1. The Customer clicks on the "Cancel Line-up" button. 2. The System removes the request from the queue.
Exit Condition	The Customer has successfully canceled his line-up request.
Exceptions	There are no possible exceptions, since the "Cancel Line-up" button is shown only if there is actually a Line-up request in the System queue.

Name	Check Waiting Time
Actor	Customer
Entry Condition	Customer has already logged in and lined-up.
Event Flow	1. The Customer enters in the LineUp section. 2. The System updates the Waiting Time based on the Real-Time Queue. 3. The System shows the Waiting Time to the Customer.
Exit Condition	The Customer is showned his Waiting Time.
Exceptions	There are no possible exceptions, since the Waiting Time is shown only if there is actually a Line-up Request in the system queue.

Name	Entrance with QR Code
Actor	Customer
Entry Condition	Customer has already done a line-up request and is arrived at the supermarket.
Event Flow	1. The Customer opens the application. 2. The Customer clicks the button "Show QR Code". 3. The Customer scans the QR code at the entrance.
Exit Condition	The Custom enters succesfully the store and the real-time queue is updated.
Exceptions	1. The scanned QR Code invalid or expired. The customer is invited to use CL-Up and to leave the store in order to avoid gatherings.

Name	Booking
Actor	Customer
Entry Condition	Customer has already logged in.
Event Flow	1. The Customer chooses the supermarket in which he wants to go. 2. The Customer indicates the date and time of his visit. 3. The Customer inputs the expected time of the visit. 4. The customer indicates the list of items, or their categories, he intends to buy. 5. The Customer clicks on the "Book" button. 6. The System inserts the visit into the booking plan of the indicated supermarket. 7. The System shows the customer his booked visits.
Exit Condition	The Custom has successfully booked his visit.
Exceptions	1. The Customer has not chosen the supermarket. 2. The Customer has not indicated the date and time of the visit. 3. The short-term Customer has not indicated expected duration of the visit. All exceptions are handled notifying the issue to the Customer through a warning message.

Name	Cancel Booking
Actor	Customer
Entry Condition	Customer has already logged in and made a booking.
Event Flow	1. The Customer clicks on the "Cancel Booking" button. 2. The System removes the booking.
Exit Condition	The Customer has successfully canceled his booking request.
Exceptions	There are no possible exceptions, since the "Cancel Boooking" button is shown only if there is actually a Booking request in the System database.

3.2.2 Store Manager

Scenario 5: Monitoring the Affluence

Gianluca has been a store manager of his supermarket for 20 years. He takes really care of his customer's health and he decides to adopt Customer Line-Up for his store due to this reason. At every peak times of affluence, he can control the number of people inside his supermarket thanks to his dedicated portal. He also can control all the booked visits to monitor the future affluence of his store.

Name	Store Manager Registration
Actor	Store Manager
Entry Condition	Store Manager enters the Login page
Event Flow	1. The Store Manager clicks on the "Sign In" button in the Login page to start the Registration process. 2. The Store Manager fills in all the mandatory fields. 3. The Store Manager loads a certification document which proves that is a real Store Manager. 4. The Store Manager clicks on the "Register" button. 5. The System validates the certification document. 6. The System saves the data. 7. The System sends and email to the new Store Manager with the confirmation.
Exit Condition	The Store Manager successfully ends the registration process and become a new User. From now on he/she can log in to the application providing his/her credentials and start using CLup.

See the Exceptions in the next page.

Exceptions	1. The Store Manager is already registered. 2. The Store Manager inserts not valid informations in one or more mandatory fields. 3. The Store Manager chooses an username that has already been taken by another user. 4. The Store Manager chooses an email that has been associated with another user. 5. The Store Manager has not filled in one or more mandatory field. 6. The System does not validate the certification document. All exceptions are handled notifying the issue to the Store Manager and taking back the Event Flow to the point number 2.

Name	Store Manager Login
Actor	Store Manager
Entry Condition	The already registered Store Manager enters the Login page
Event Flow	1. The Store Manager inserts the username. 2. The Store Manager inserts the password. 3. The Store Manager clicks on the "Login" button.
Exit Condition	The Store Manager successfully logs in and have access to the CLup services.
Exceptions	1. The Store Manager has not filled in both the username and the password. 2. The Store Manager inserts an email which is not saved in the database. 3. The Store Manager inserts a wrong password. All exceptions are handled notifying the issue to the Store Manager through a warning message.

Name	Ticket Generation
Actor	Store Manager
Entry Condition	The No-Tech Customer arrives to the supermarket
Event Flow	1. The Store Manager enters his dedicated portal. 2. The Store manager reports the customer to the system. 3. The System updates the real-time queue of the supermarket. 4. The System generates the ticket for the No-Tech Customer.
Exit Condition	The No-Tech Customer enters successfully in the store immediately or after his waiting time if the supermaket is full.
Exceptions	There are no possible exceptions.

Name	Check Supermarket Affluence
Actor	Store Manager
Entry Condition	Store Manager has already logged in.
Event Flow	1. The Store Manager clicks the "Check Affluence" button. 2. The Store Manager selects the supermarket that he want to monitor. 3. The System shows the informations to the Store Manager.
Exit Condition	The Store Manager can see the informations regarding the affluences of the store.
Exceptions	There are no possible exceptions.

3.2.3 Requirements

G_1 : Customers can line-up from their home to access the supermarket.

- R_1 : The System retrieves the GPS position of the Customer.
- R_2 : The System sends the QR code for the entrance.
- R_3 : The Customer indicates with which mean of transport he intends to go to the supermarket.
- R_4 : The Customer indicates the supermarket in which he wants to go.
- R_5 : The System shows the Customer his Waiting Time.
- D_1 : There is stable Internet connection.
- D_2 : The GPS provides the exact location with an error of 100 meters at most.

G_2 : Customers can book in advance a visit to the supermarket.

- R_4 : The Customer indicates the supermarket in which he wants to go.
- R_6 : The Customer select date and time of his Booking.
- R_7 : The Customer can input the expected duration of his visit.

- R_8 : The System computes the Inferred Durations of the visits for Long-Term Customers who didn't input it.
- R_9 : The Customer can input the exact list of items to buy or their categories.
- D_1 : There is stable Internet connection.

G_3 : Customers are notified when they have to go towards the store according the time they need to get to the supermarket.

- R_1 : The System retrieves the GPS position of the Customer.
- R_{10} : The System computes the time needed for the Customer to reach the supermarket based on his currently GPS position and on his mean of transport.
- R_{11} : The System notifies the Customer when he have to start.
- D_1 : There is stable Internet connection.
- D_2 : The GPS provides the exact location with an error of 100 meters at most.

G_4 : Customers entrances are monitored and managed by the System.

- R_2 : The System sends the QR code for the entrance.
- R_{12} : Customers scan their QR code when they both enter or exit from the supermarket.
- R_{13} : The System computes the Waiting Time considering the Real-Time Queue.
- R_{14} : The System increases entrances considering the affluence of the departments.
- D_1 : There is stable Internet connection.
- D_3 : All the Customers, both those who have booked or lined-up, arrive at the supermarket.
- D_4 : The Customers arrive at the supermarket on time with an error of 3 minutes at most.

- D_5 : All the No-Tech Customers must obtain a ticket before entering.
- D_6 : Customers who have booked stay in the supermarket for the time they have indicated with an error of 5 minutes.

G_5 : Store Managers can hand out tickets on the spot for Customers who do not have access to the required technology.

- R_{15} : Store Manager sends to the System an Ticket Request for No-Tech Customers.
- R_{16} : The System generates the Ticket and sends it to the Store Manager.
- R_{17} : The System updates the Real-Time Queue.
- D_1 : There is stable Internet connection.

3.2.4 UML Description

Use Case Diagram

The figure below shows the Use Case diagram of CLup, which summarises use cases and the actors who perform them.

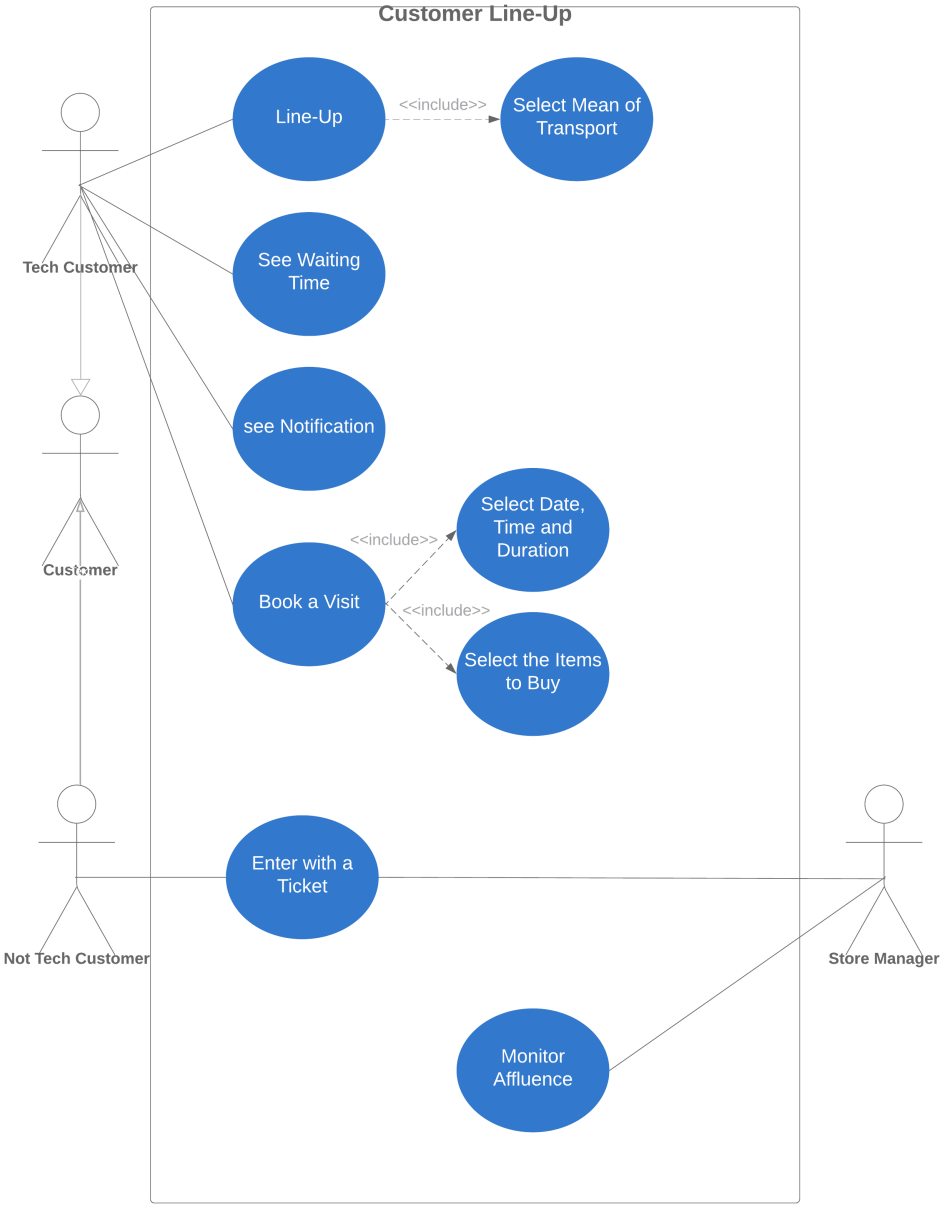


Figure 3.8: Use Case Diagram

Sequence Diagrams

Registration Sequence Diagram

This sequence diagram shows the process of registration made by the Customer: in particular he has to insert the email, username, password and its confirmation. Then, the system verifies the input and if these values are valid then the Customer

is successfully registered.

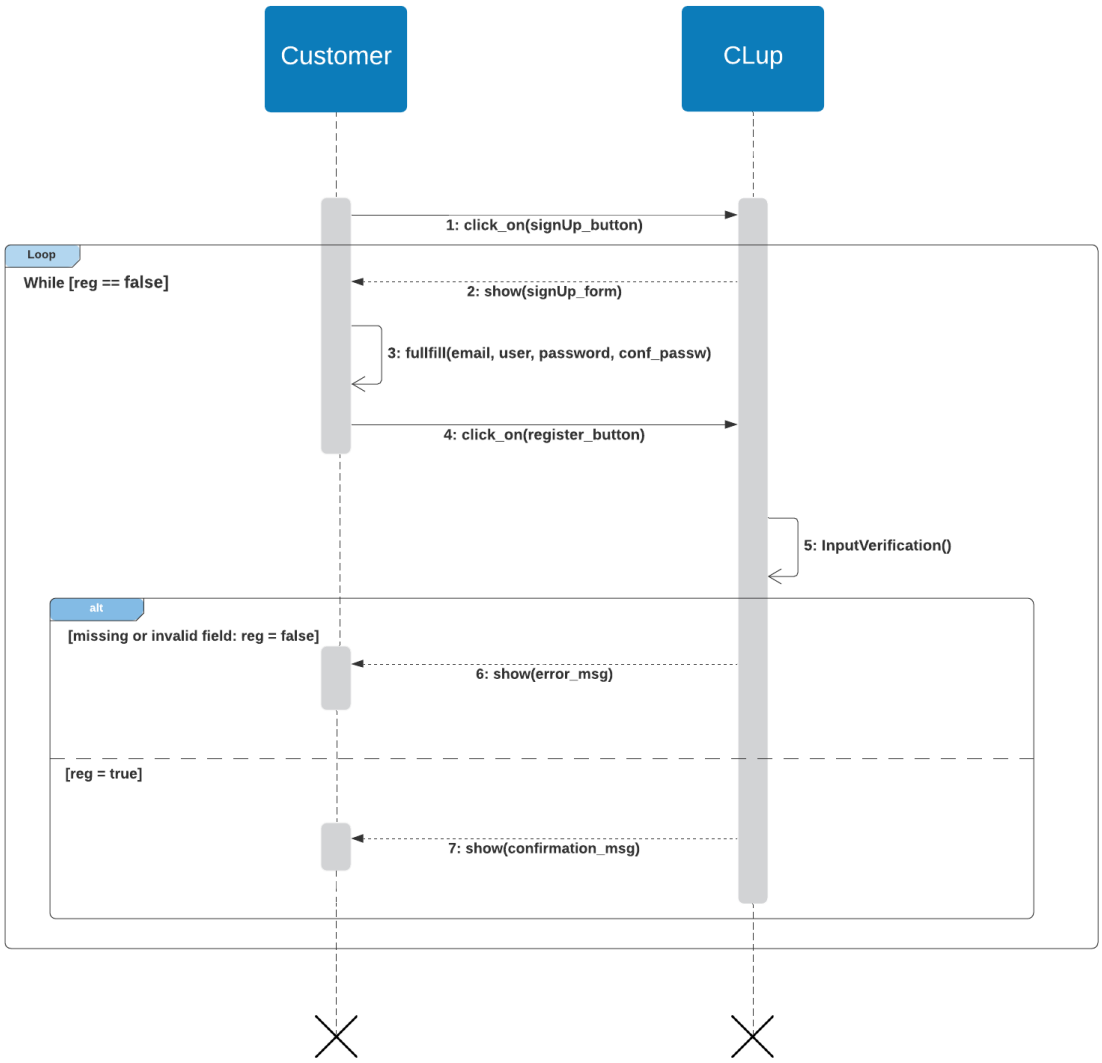


Figure 3.9: Registration Sequence Diagram

Login Sequence Diagram

This sequence diagram shows the process of login made by the Customer: in particular he has to insert his username and password. If the credentials are present in the database, then the Customer can log in.

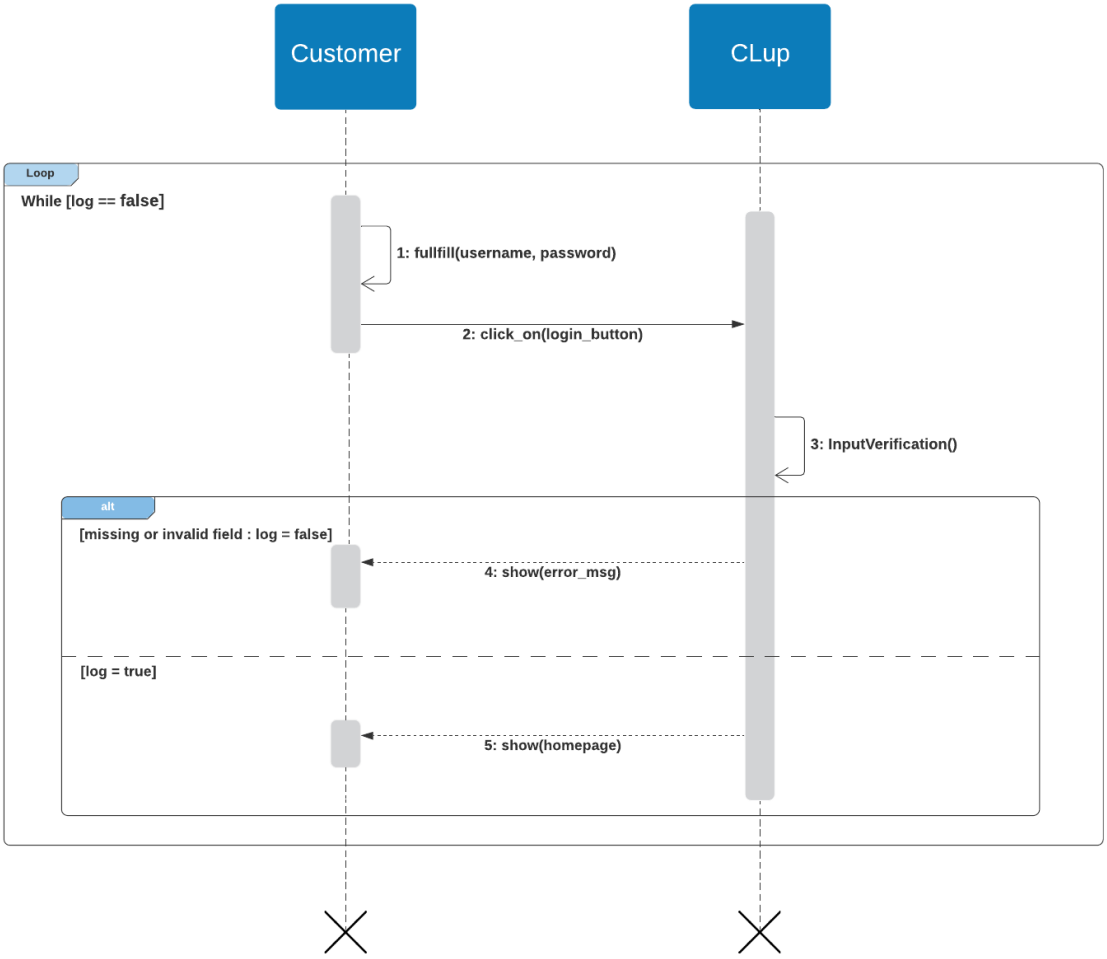


Figure 3.10: Login Sequence Diagram

LineUp Sequence Diagram

This sequence diagram shows how the Customer can send a new LineUpRequest: in particular, he has to only select the supermarket, between those available, and his mean of transport. If the parameters are valid, then the LineUpRequest is added to the queue.

Notice that it is always possible to send a new LineUpRequest, even if the supermarket is closed: in this case the request will be considered for the first available time slot.

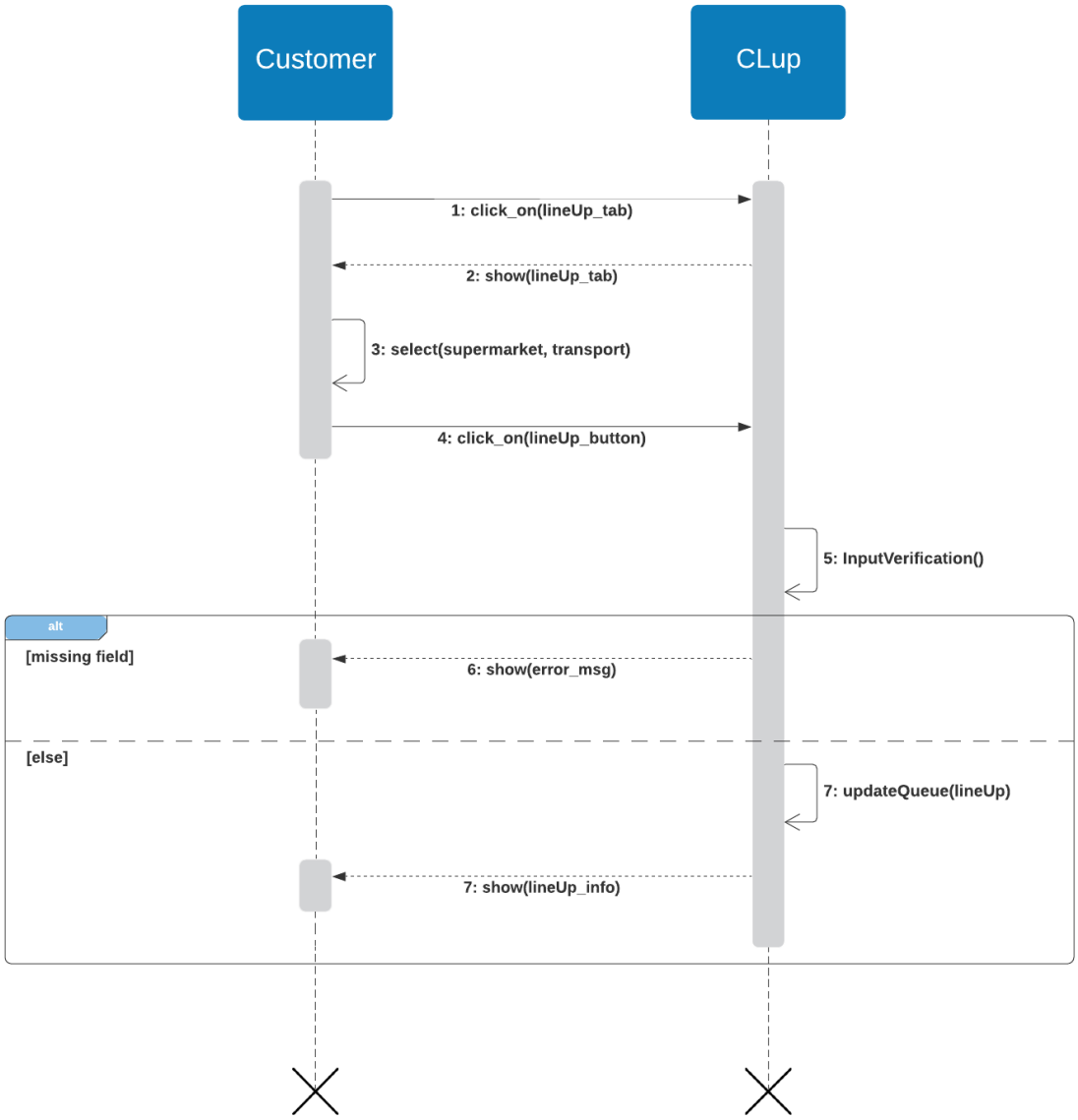


Figure 3.11: Line-Up Sequence Diagram

Booking Sequence Diagram

This sequence diagram shows how the Customer can send a new BookingRequest: in particular, he has to select the supermarket, between those available, and the day and time of the visit. Moreover, if he is not a longTermCustomer, he is oblied to input also the expectedDuration, while instead the itemsList is optional for all kinds of Customers. Then, the system check all the inputs and if they are valid it saves the BookingRequest.

Figure 3.12: Booking Sequence Diagram

Cancel Booking Sequence Diagram

This sequence diagram shows how the Customer can delete a BookingRequest: in

particular, he has only to click on the cancel button, which is shown only if he has an active Booking saved.
The analogous canceling procedure can be done for the LineUp.

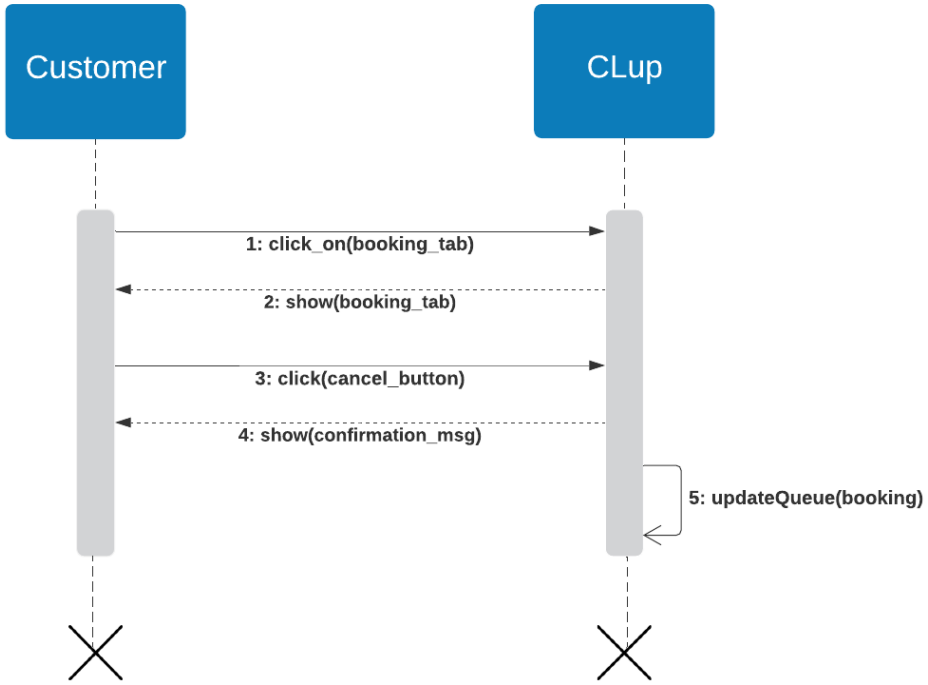


Figure 3.13: Cancel Booking Sequence Diagram

QR-Entrance Sequence Diagram

This sequence diagram shows how the entrance at the supermarket is performed: the Customer has to scan the QRCode and if the system recognized it as valid, then he is allowed to enter.

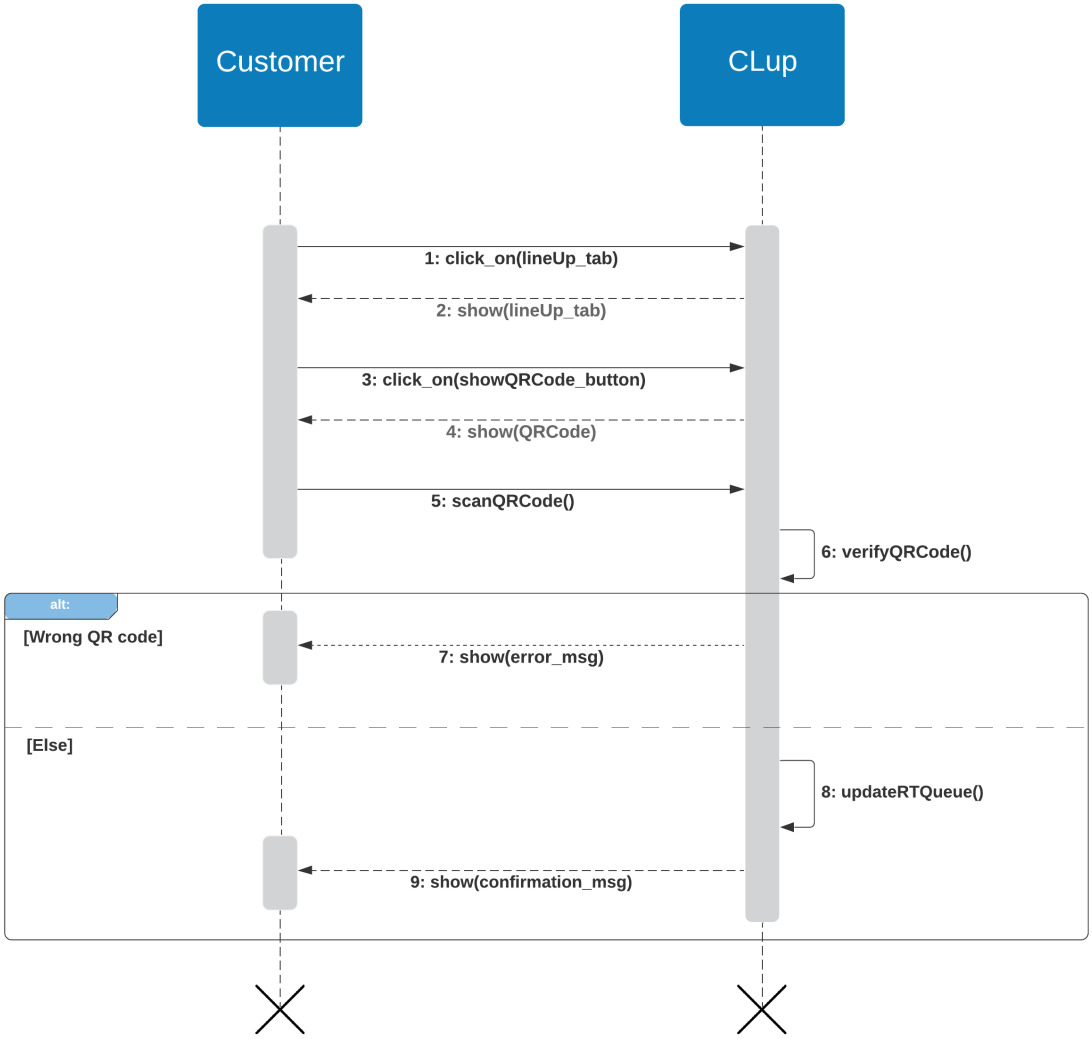


Figure 3.14: QR-Entrance Sequence Diagram

TicketGeneration Sequence Diagram

This sequence diagram shows the procedure with which the StoreManager can generate a new ticket for a NoTechCustomer. If the supermarket is full, the NoTechCustomer has to wait; otherwise, he can enter and the system updates the affluence data.

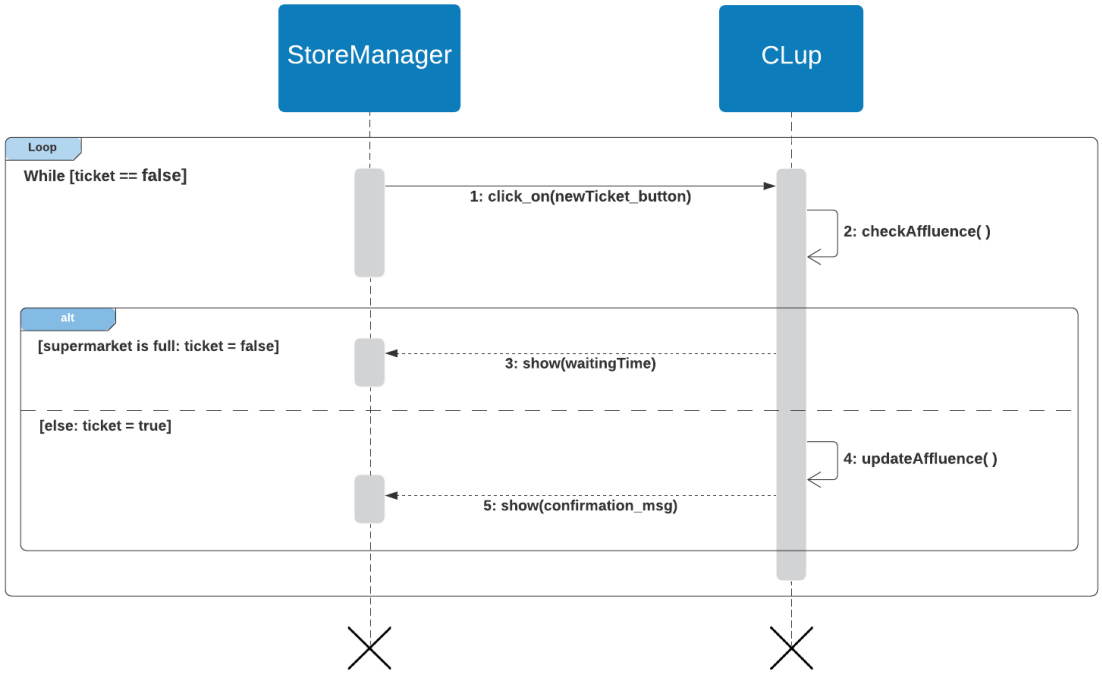


Figure 3.15: Ticket generation Sequence Diagram

3.3 Performance Requirements

The purpose of this section is to outline the Performance objectvies for the application to be implemented. In particular, with respect to the Response Time, the CLup must be able to send feedbacks within 2 seconds. Moreover, it is important to underline the fact that, since the application is thought for being used by a very large amount of people, ideally by every single person, the system must be capable to support at least 300 000 users simultaneously. This Workload problem will be addressed in the Design phase, but it is evident that a load balancer and multiple replicated servers will be needed.

3.4 Design Constraints

This section tells about the design constraints of the system, beginning with standard compliance, then with hardware limitations and finally with other constraints.

3.4.1 Standards Compliance

The system must be compliant with every possible standard that will be encountered during the entire development lifecycle. Also, since the application needs to store personal informations about the customers, for instance their emails and GPS position, the GDPR must be respected.

3.4.2 Hardware Limitations

- Mobile App:
  - iOS or Android smartphone
  - 3G/4G/5G Internet connection
  - GPS
- Web App:
  - Modern browser

3.4.3 Any Other Constraint

For the time being, there are no other constraints to be specified. If any new constraint will have to be introduced during the software development process, it will be added in the next versions of this document.

3.5 Software System Attributes

This final section lists the attributes that the system should try to achieve.

3.5.1 Semplicity

The application should be easy to use with a very intuitive interface, as the range of users include all demographics.

3.5.2 Reliability

Firstly, the line-up mechanism should be very effective in order to avoid that customers arrives to the grocery store after his/her number is called, or too early, as in this case we would get back into a physical line situation. For this reason, the system provides an estimation of the customer waiting time with a maximum error of 3 minutes and alert them taking into account the time they need to get to the shop from the place they are and their mean of transport.
Finally, it must guarantee a consistency of the bookings stored inside the system.

3.5.3 Availability

Grocery shopping is a primary need for the population. Considering that there are supermarket that are open 24h/7 and the preview reason the system should be up for a 99% of time. That means that the average time between the occurrence of a fault and service recovery (MTTR or downtime) should be contained around at 3.65 days per years. It does not need to provide a higher availability because the service is not strictly related with emergency situations.

3.5.4 Security

The system will use external services to keep the whole system secure by cyber attacks to protect the data of booking and the real-time queue of each supermarket.
The access to the Store Manager's portal is allow only for them with a unique password which is changed periodically.

3.5.5 Maintainability

The system must be modifiable for future implementations or upgrades, so the design of the structure will use appropriate design patterns to make any update easier and conflict-avoidable.

Chapter 4

Formal Analysis Using Alloy

In this chapter is shown the use of Alloy to analyze the critical parts of the system. In particular the most crucial points to analyze are:

- Only No-Tech Customer can enter with the Ticket.
- Customer can enter with a Line-Up or Booking Request.
- Customer can be involved in one started Request per time.

All the others Signatures and Facts are used to simplify the model in some cases or to help the building of the model in others. In particular section 4.1 lists and explains all the Signatures used , section 4.2 all the Facts of the model and the last section 4.4 shows an example of world with Alloy's Predicates.

4.1 Signatures

This section shows the piece of code in Alloy with all the signatures used for this model.

```
open util/time
open util/boolean

sig Customer {
   id: one Int,
   hasDevice: one Bool
}{
   id > 0
```

```
}
sig Supermarket {
   id: one Int,
   capacity: one Int
}{
   id > 0
}
abstract sig Request {
   custumer: one Customer,
   state: State one -> Time,
   supermarket: Supermarket
}
sig LineUpRequest extends Request {
   qrCode: one QRCode
}
sig BookingRequest extends Request {
   booking: one Booking
}
sig TicketEntrance {
   ticket: one Ticket,
   customer: one Customer,
   supermarket: one Supermarket
}
sig Ticket {
   number: one Int
}{
   number > 0
}
sig QRCode {
   id: one Int
}{
   id > 0
}
sig Booking {
   id: one Int
}{
   id > 0
}

abstract sig State {}
one sig Reserved extends State{}
one sig Started extends State{}
```

```
one sig Finished extends State{}
```

4.2 Facts

This section shows the piece of code in Alloy that contains facts. Some of the facts model the domain assumptions, others are needed to build a consistent model.

```
//Unique Istances
fact uniqueQRCode {
    all disj c, c': QRCode | c.id != c'.id
}
fact uniqueBooking {
    all disj b, b': Booking | b.id != b'.id
}
fact uniqueTicket {
    all disj t, t': Ticket | t.number != t'.number
}
fact uniqueCustomer {
    all disj a, a': Customer | a.id != a'.id
}

//Only not-Tech User can have a ticket
fact ticketForNoTech {
    all t: TicketEntrance | t.customer.hasDevice = False
}
fact RequestForTechCustomer {
    all q: Request | q.custumer.hasDevice = True
}

//A QRCode, Booking, Ticket has to be associated to its
   ↪ Request
fact allQRCodeInLineUPRequest  {
    all q: QRCode | one l: LineUpRequest | q.id = l.qrCode.id
}
fact allBookingInBookingRequest {
    all b: Booking | one r: BookingRequest | b.id = r.booking.
       ↪ id
}
fact allTicketInEntrance {
    all t: Ticket | one e: TicketEntrance | t.number = e.
       ↪ ticket.number
```

```
}
fact requestCreationStateChart {
    //A request is always created as "Reserved"
    all r: Request | one t': Time | r.state.t' = Reserved
        ↪
}
fact requestFinishedStateChart {
    //Once a Request is finished it cannot change status again
    all r: Request, t: Time |
      (r.state.t = Finished => all t': Time | gte[t',t] => r.
        ↪ state.t' = Finished)
}
fact requestStartedStateChart {
    //Once a Request is started it cannot get back to Reserved
    all r: Request, t: Time |
       (r.state.t = Started => all t': Time | gte[t',t] => r.
         ↪ state.t' != Reserved)
}
//A user can be envolved in one "Started" request per time
fact oneStartedPerUser {
    no disj r,r': Request |
       r.custumer = r'.custumer and
          some t: Time |
                r.state.t = Started and r'.state.t = Started
}
```

4.3 Predicates

This section shows the piece of code in Alloy that contains the Predicates.

```
pred isCustomerInARequest[c: Customer, t: Time] {
    one r: Request | r.state.t = Started and r.custumer = c
}

pred makeALineUpRequest[c: Customer, q: QRCode, t: Time, r':
    ↪ LineUpRequest]{
    //precondition
    not isCustomerInARequest[c,t]
    //postcondition
    r'.custumer = c
```

```
    r'.qrCode = q
    r'.state.t = Reserved
}
run makeALineUpRequest

pred makeABookingRequest[c: Customer, b: Booking, t: Time, r
    ↪ ': Request]{
    //precondition
    not isCustomerInARequest[c,t]
    //postcondition
    r'.custumer = c
    r'.booking = b
    r'.state.t = Reserved
}
run makeABookingRequest

pred startARequest[r: Request, t: Time]{
    //precondition
    not isCustomerInARequest[r.custumer,t]
    r.state.t = Reserved
    //postcondition
    r.state.(t.next) = Started
    isCustomerInARequest[r.custumer,t.next]
}
run startARequest

pred endARequest[r: Request, t: Time] {
    //precondition
    isCustomerInARequest[r.custumer,t]
    r.state.t = Reserved or r.state.t = Started
    //postcondition
    r.state.(t.next) = Finished
    not isCustomerInARequest[r.custumer,t.next]
}
run endARequest


//Pred Show
pred show {
    #Customer = 3
    #TicketEntrance = 1
    #BookingRequest = 1
```

```
    #LineUpRequest = 1
    #Supermarket = 1
}
run show for 3
```

4.4 Generated World

In this section it is shown the model generated by executing the Alloy code: in particular, three different Worlds are provided in order to underline the different states in which the Requests can be.

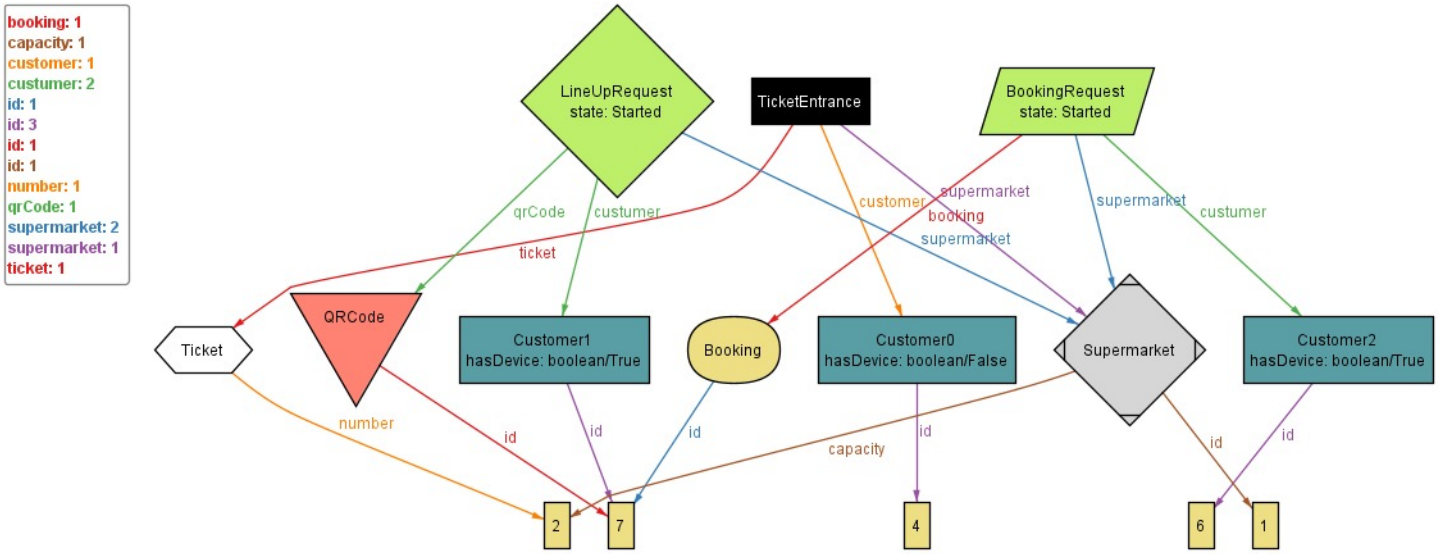


Figure 4.1: Alloy Model

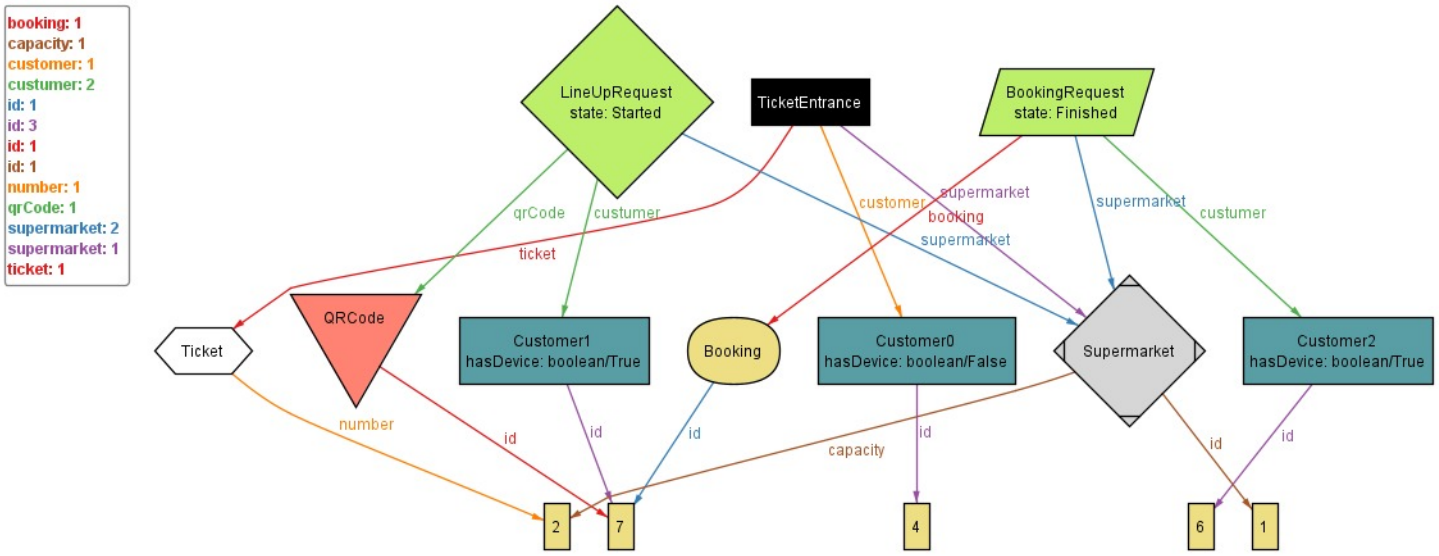


Figure 4.2: Alloy Model

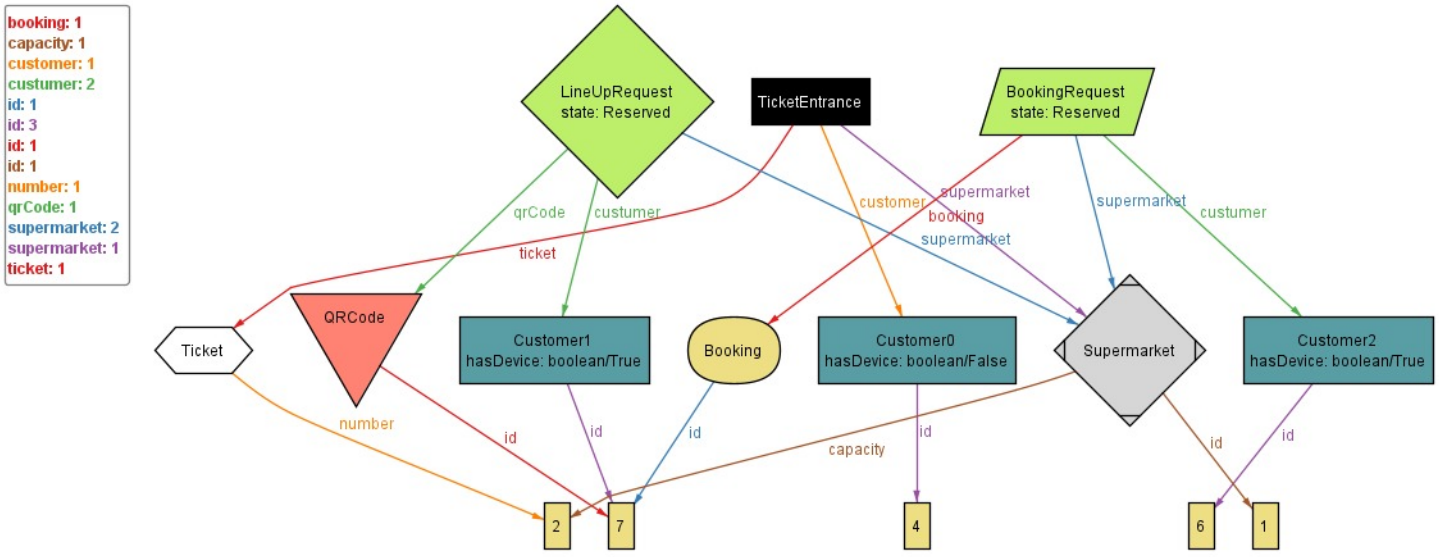


Figure 4.3: Alloy Model

4.5 Proof Of Consistency

```
Executing "Run makeALineUpRequest"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  6920 vars. 420 primary vars. 19513 clauses. 80ms.
  Instance    found. Predicate is consistent. 89ms.

Executing "Run makeABookingRequest"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  6920 vars. 420 primary vars. 19513 clauses. 64ms.
  Instance    found. Predicate is consistent. 65ms.

Executing "Run startARequest"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  6929 vars. 414 primary vars. 19595 clauses. 64ms.
  Instance    found. Predicate is consistent. 58ms.

Executing "Run endARequest"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  6932 vars. 414 primary vars. 19640 clauses. 70ms.
  Instance    found. Predicate is consistent. 125ms.

Executing "Run show for 3"
  Solver=sat4j Bitwidth=4 MaxSeq=3 SkolemDepth=1 Symmetry=20
  6734 vars. 408 primary vars. 19122 clauses. 83ms.
  Instance    found. Predicate is consistent. 104ms.
```

Chapter 5

Effort Spent

Ludovico Righi

Task	Hours
Chapter 1	7
Chapter 2	12
Chapter 3	13
Chapter 4	6
Version Update	5

Enrico Gherardi

Task	Hours
Chapter 1	8
Chapter 2	7
Chapter 3	9
Chapter 4	14
Version Update	5

Chapter 6

References

This document has been written referencing to the folllowing documents:

- IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications.
- Specification document: "R&DD Assignment A.Y. 2020/2021".
- Lecture slides of professor M.Rossi and E.di Nitto of Politecnico di Milano.
- Alloy Documentation.

The tools that have been exploited are the following:

- Alloy Analizer 5.1.0: to construct and prove the consistency of the Alloy model.
- LucidChart: to build the diagrams.
- LaTeX: to write the pdf file.
- GitHub: for versions controller.