POLITECNICO
MILANO 1863

Requirements Analysis and Specifications Document of CLup

Version 3.0 - 06 January 2021

Pasquale Occhinegro
Giampiero Repole
Andrea Zanetti

Politecnico di Milano - A.A. 2020/2021

1. Introduction

1.1 Purpose

This document aims to explain both the application domain and the system to be developed. To reach this goal, the document completely describes the system in terms of functional and nonfunctional requirements, it analyzes the real needs of the customer in order to model the system, it shows the constraints and the limit of the software and indicates the typical use cases that will occur after the release. This document is addressed to the developers who have to implement the requirements and could be used as a contractual basis.

1.1.1 Goals

- G1 The application should allow a Customer to become a registered User after providing credentials and his address.
- G2 The application should allow a Customer to digitally line-up to enter a Supermarket from his home.
- G2.1 The application should allow a Customer to see their position in the queue.
- G3 The application should allow a Customer to book a visit to a selected store.
- G3.1 The application should allow a Customer to select departments in which they want to buy.
- G3.2 The application should allow a Customer to choose a proposed time slot for the visit.
- G4 The application should allow a Customer to retrieve a QR code in case of “virtual line-up” or “book a visit”.
- G5 The application should allow a Customer to physically line-up in front of the store.
- G6 The application should allow a Store Manager to manage the number of Customers allowed for each supermarket department
- G7 The application should allow the Store Manager to monitor entrances.
- G8 The application should allow a Customer to cancel a visit previously booked.
- G9 The application should send the Customer notifications when slots are available, if requested.

1.2 Scope

CLup (Customers Line up) is an application that allows people to safely approach an essential task such as grocery shopping during emergency times like the current pandemic.
People can sign-up on the app and provide their address to see a list of nearby stores from which they can choose one to line-up in a digital queue handled by the system.
When queuing, the system chooses the closest interval of time that satisfies the safety conditions of the store, and tells the Customer the estimated time of arrival and that remaining until his turn.
The Customer can also book the visit, specifying the store, their estimated time of staying and a general idea of the items they want to buy.
In this case the system sends the Customer a list of slots to choose from.
The Customer can always cancel a visit previously booked.
The system also memorizes the booking activity and during a new instance from the Customer the app will suggest the same time slots that were selected in his previous bookings.
All Customers are provided with a QR code, when they book a visit or line-up, that must be used at the store entrance to keep in check the number of people inside.
The Customers can also line-up in the virtual queue at the entrance of the store, where a physical QR code will be printed along with the estimated time remaining until his turn.
A Store Manager can submit their store in the app, specifying the number of people allowed in each department, and then monitor the number of people inside the store thanks to the QR code and the estimated time of staying.

The tables below analyze the phenomena involved:

WP1	Customer books a visit
WP2	Customer line-up for a visit
WP3	Store Manager provides information
WP4	Customer sign-up
WP5	Customer cancels a booking

SP1	Generate a QR code
SP2	Give alternative solutions
SP3	Suggest the last booking activity
SP4	Customer receives notification about available slot

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

Customer	A person that downloads the app or takes the physical ticket for lining up
Store Manager	A person that manages a store or a store chain that enters the system circuit
Map Provider	A third party activity that helps the system by providing his services
Time Slot	Interval of time of 30 minutes reserved for Customers that go grocery shopping using the “Book a visit” feature

1.3.2 Acronyms

RASD	Requirement Analysis and Specification Documents
GPS	Global Positioning System

1.3.3 Abbreviations

WPn	World Phenomenon number n
SPn	Shared Phenomenon number n
Gn	Goal number n
Rn	Requirement number n

1.4 Reference Documents

- Specification Document: “CLup Mandatory Project Assignment.pdf”
- Slides of the lectures

1.5 Overview

This RASD is divided in 6 chapters briefly described below:

Chapter 1: In this chapter the RASD is introduced, and a general description of the problem and the solution related is given.
Moreover the goals and the scope of the project are specified here.

Chapter 2: A more in-depth description of the project is given in this chapter.
A class diagram is used to visualize the concept and the structure of the solution given, and the product functions are presented.

Chapter 3: Here the external interface requirements are discussed, as well as the functional requirements, in which scenarios and use cases are provided.

Chapter 4: The alloy model of the project is described here.

Chapter 5: In this chapter is shown the amount spent by the members of the group on each

Chapter 6: External tools used during the making of this RASD are shown.

1.6 Revision History

Date	Version
22/12/2020	1.0 : Release version
03/01/2021	2.0 : - added Polimi logo; - small changes on muckup’s design;
06/01/2021	3.0 : - modified Polimi logo; - fixed typo in alloy assertion image;

2. Overall Description

2.1 Product Perspective

The purpose of CLup is to optimize as much as possible the influx of customers in grocery stores, letting them book their visit to the store through slots of limited capacity. In order to offer some core functionalities, the app relies on the GPS of the Customer's device to retrieve the position to calculate the path from home to the store. It also needs access to the phone camera in order to scan QR codes through the store's tablet so that Managers can monitor the influx of Customers in their store.
To better describe the domain model used, a class diagram is provided (Figure 1):

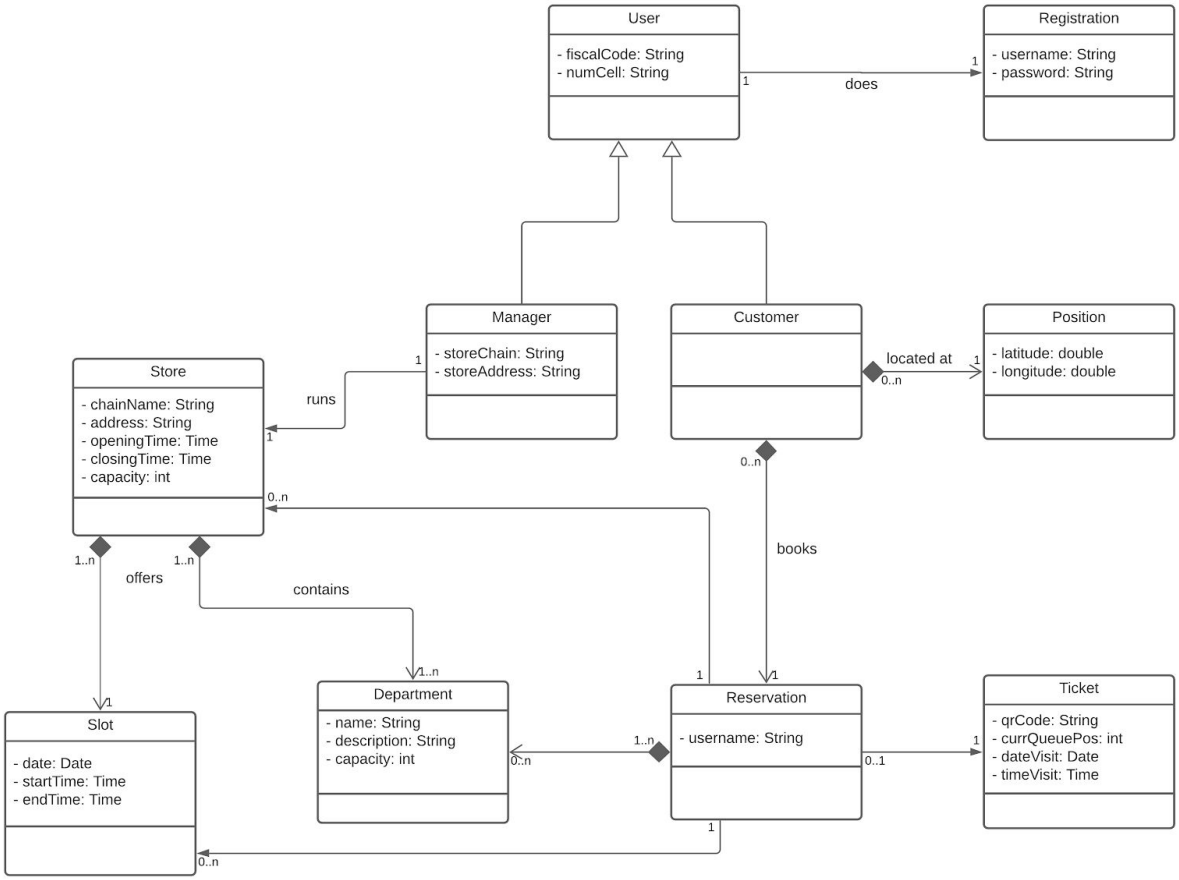

Figure 1: Class Diagram

2.2 Product Functions

2.2.1 Line-up for grocery shopping

The Customer can request a ticket by the app or from the physical device outside a store, and from that the system will check the closest interval of time in which the Customer can safely shop, considering an estimated time of staying and all the departments of that store occupied by one more Customer during that time.
It then sends (or prints, in case of the physical device) the time when the Customer should enter the store and a QR code to be scanned at the entrance.
It also receives from a provider the estimated travel time from the address of the Customer to the store, and sends a notification when they should start going.

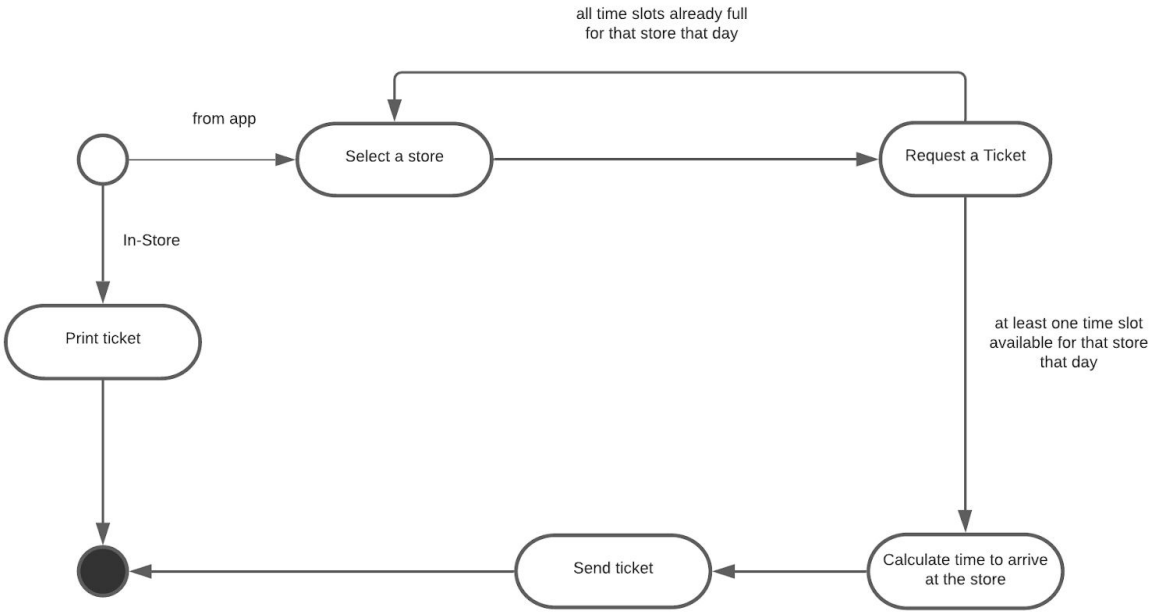

Figure 2: “Line-up” state chart

2.2.2 Book a visit

This function is only available on the app, and it differs from the lining-up function on different factors: the Customer must add a generic list of departments in which he intends to buy and then he can choose the number of time slots (half an hour per slot) that he wants to occupy.
If it isn't the first booking for the Customer, he can choose time slots selected in his last booked visit.
In case the Customer needs to book an unavailable time slot he can choose: to be notified if the selected slot goes back to available, or to select a different store.
When the Customer selects an available time slot and confirms, the system books the visit and sends him the ticket with the QRcode.

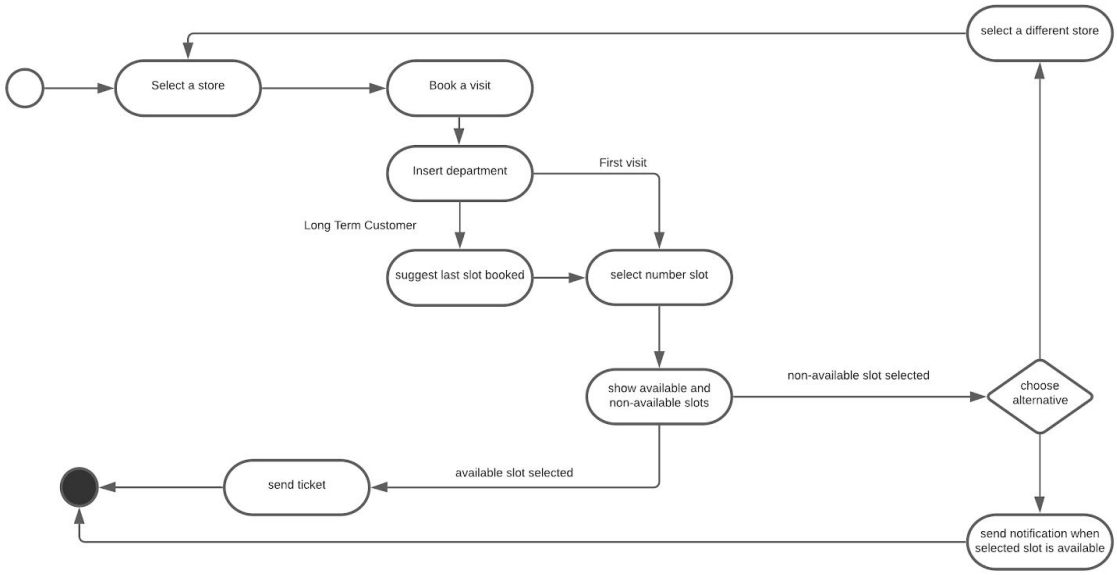

Figure 3: "Book a visit" state chart

2.2.3 Cancel a visit

This function is used only when the Customer has at least one session booked.
The system sends the list of the visits and asks which one the Customer wants to cancel, then, after the Customer confirmation, proceeds to free the slot previously occupied.

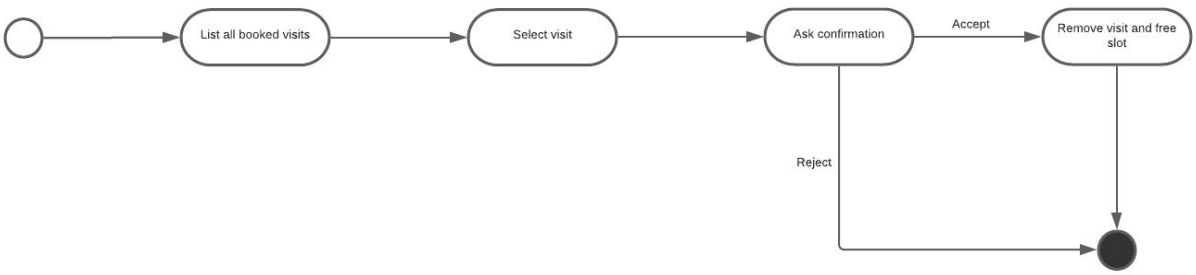

Figure 4: “Cancel a visit” state chart.

2.2.4 Check the store situation

This function can be performed only by the Store Manager.
Every time a QR code is scanned the system checks the time slots occupied in each department by the Customer, and updates the current number of people inside the store.
The Store Manager can always check the store situation by seeing this number or reducing the amount of people allowed in each department.

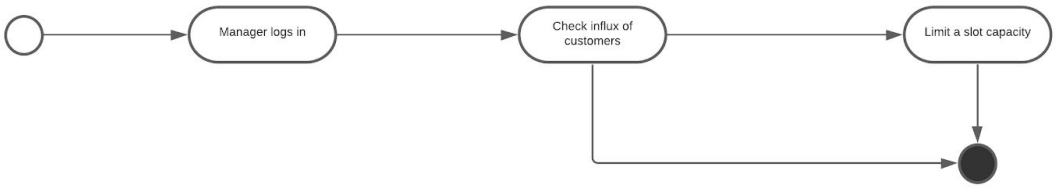

Figure 5: Store Manager functionality state chart

2.3 User characteristics

- Customer: a person registered on the CLup app who wants to book a visit or line-up to the grocery store.

- Store Manager: a manager from a grocery store chain subscribed to the CLup service, uses this app to monitor the influx of Customers or to change the number of people admitted in each department.

2.4 Domain Assumptions, Dependencies and Constraints

- D1: Each user has a smartphone;

- D2: The application is installed on a device;

- D3: The device is connected to the internet;

- D4: The position of the Customer is obtained through a GPS system integrated in the Customer's device and is trustworthy;

- D5: The position retrieved using the GPS system is accurate with an error of 5 meters at most;

- D6: GPS is active while the app is running;

- D7: The Customer grants the application to have access to GPS position.

- D8: There is a terminal on a Store entrance to physically line-up and print a ticket.

- D9: The Customer grants the application to send notifications.

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

Below are represented the user interfaces for a better understanding of the system functions:

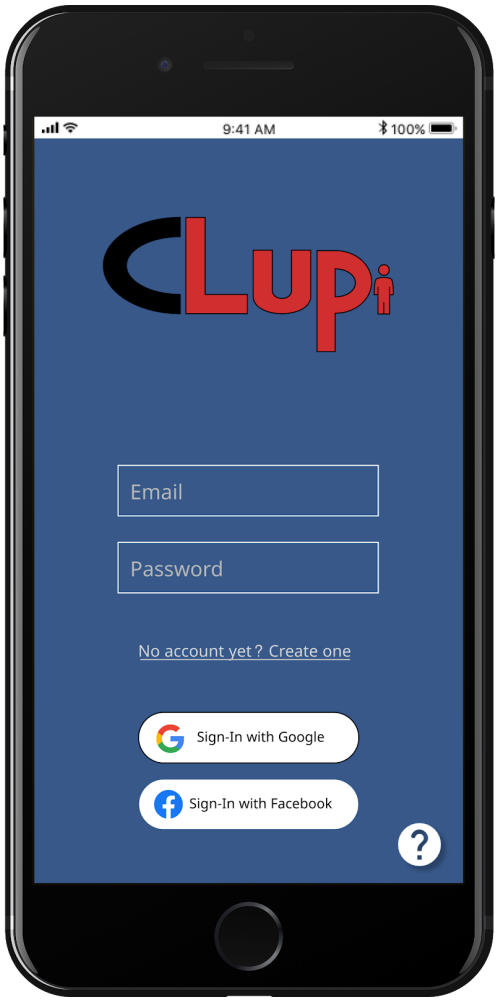

Figure 6: Login Page Mockup

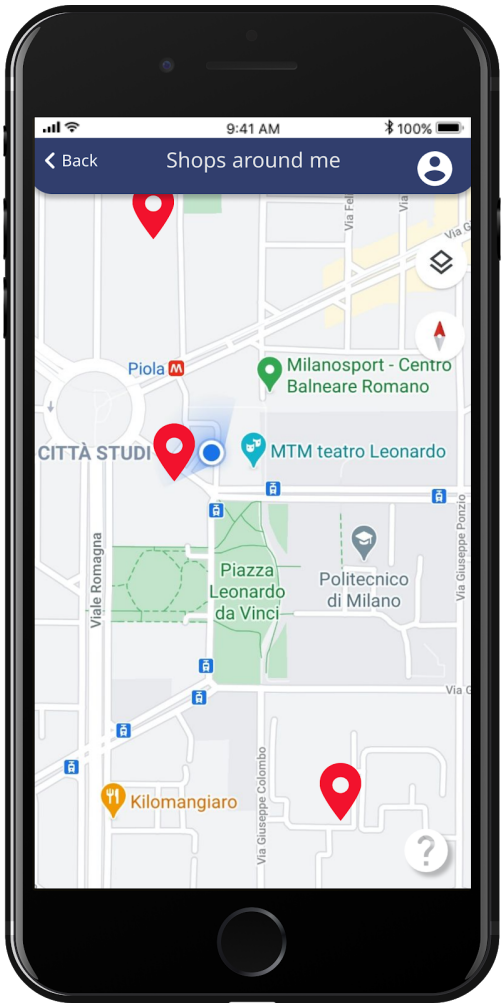

Figure 7: Store Selection Page Mockup

3.1.2 Hardware interfaces

The Customers that line-up from their home must own a smartphone with GPS enable to retrieve its location. Meanwhile, tablets with the app are available for the Customers that line-up physically at the entrance of the associated store, where they can retrieve their physical tickets, printed on the spot.

3.1.3 Software interfaces

The app makes use of external APIs such as a map provider to show the location of stores around the Customer and to calculate the estimated time for them to arrive at the store.

3.2 Functional requirements

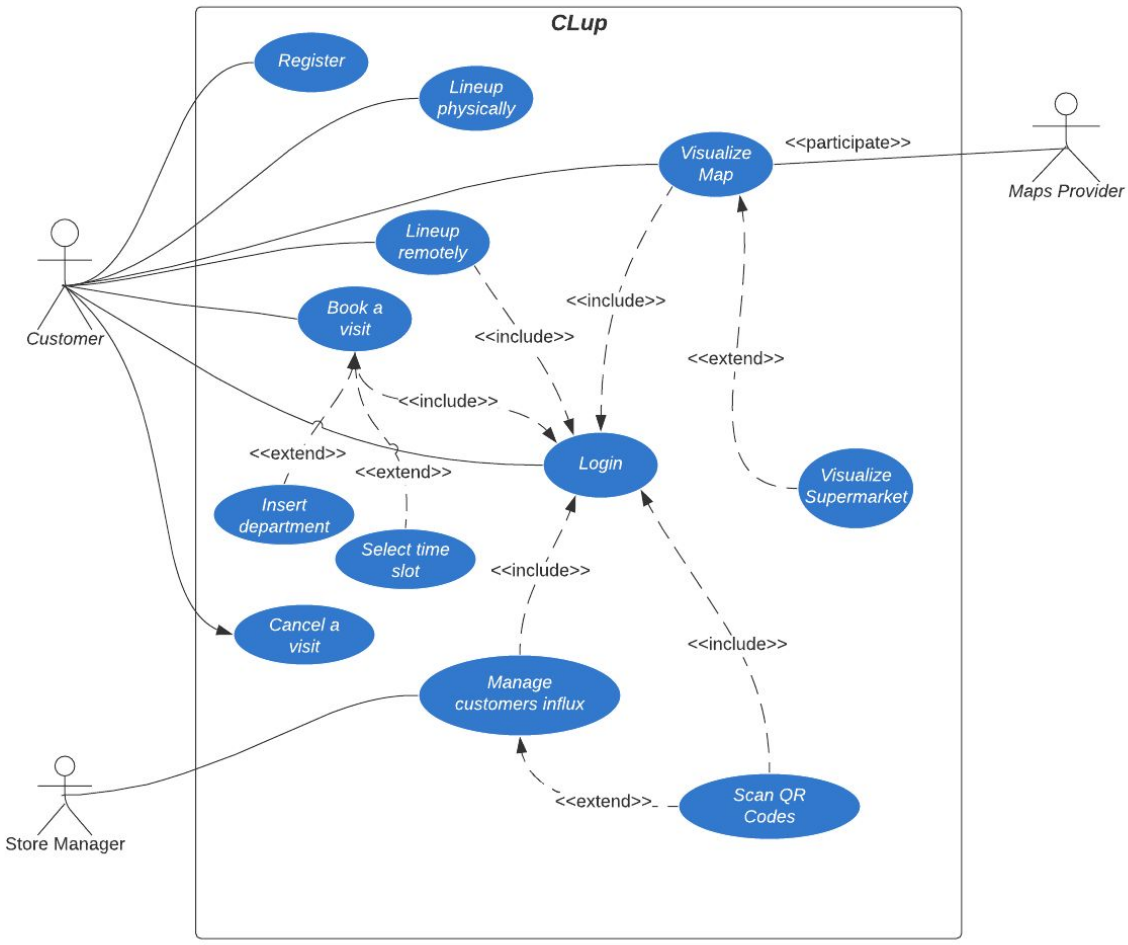

Figure 8: Use Cases diagram

3.2.1 Scenarios

Scenario 1:
John wants to go grocery shopping but during the coronavirus emergency only a limited number of people can enter the store. So he decides to book a visit through the app CLup in order to be sure to get access to the store without having to wait for it. John opens the app and looks for the nearest store to him and books a visit in a time slot specifying which kind of products he will buy once there.

Scenario 2:
Paul just got in front of a grocery store from a renowned chain but notices that a long line has formed outside of it. Some signs say to download the app CLup in order to take the virtual ticket for the line. So he downloads it, takes the ticket and waits for her turn. When his turn comes he uses the store tablet to scan his QR Code.

Scenario 3:
George wants to line up in the grocery store near his home but his internet connection doesn't work so came in front of the store and retrieves a physical ticket from the tablet using the application CLup. Then proceeds to wait for his turn.

Scenario 4:
Richard is the store manager of a big grocery store. He uses his business phone to monitor the influx of customers entering his store using the CLup application. At a certain point he notices that the gastronomy department is full so he regulates the influx on CLup giving priority to the customers that have to buy products from the more empty frozen food department.

3.2.2 Use Cases: Customer

- Name: Register
    - Actor: Customer
    - Entry Conditions: The Customer opens the app for the first time and wants to create an account.
    - Event Flow:
        1. The login page is shown by the application;
        2. The Customer clicks on the "sign up" button;
        3. The application shows the register form page;
        4. The Customer fills in the form with their personal information;
        5. The Customer clicks the "register" button;
        6. The system creates a new profile and saves the information given.
    - Exit Conditions: The Customer has now a personal account saved in the system and can login into the application
    - Exceptions:
        1. The information given by the Customer are associated to an already existing account, the system suggests to use the "login" function;
        2. The Customer does not fill the form completely, the system suggests to fill the remaining fields.
    - Special Requests: No special requests needed.

- Name: Login
    - Actor: Customer
    - Entry Conditions: The Customer has already created an account and opened the application.
    - Event Flow:
        1. The login page is shown by the application;
        2. The Customer enters his/her credentials and presses the Login button.
    - Exit Conditions: The Customer is now logged in and can see his own profile.
    - Exceptions:
        1. The Customer typed the wrong username or password and an error message popped up. The system now waits for the Customer to insert the correct credentials. Otherwise it can recommend to recover the credentials.
    - Special Requests: No special requests needed.

- Name: Line-up remotely
    - Actor: Customer
    - Entry Conditions: The Customer has opened the app and is logged in.
    - Event Flow:
        1. The Customer opens the map;
        2. The Customer selects a store;
        3. The Customer selects the option “request a ticket”;
        4. The system calculates the earliest available slot;
        5. The system sends the QR code and calculates the estimated time of arrival;
    - Exit Conditions: The Customer has a QR code to show at the entrance of the store
    - Exceptions:
        1. The system cannot find an available slot and asks the Customer to choose another store.
    - Special Requests: No special requests needed.

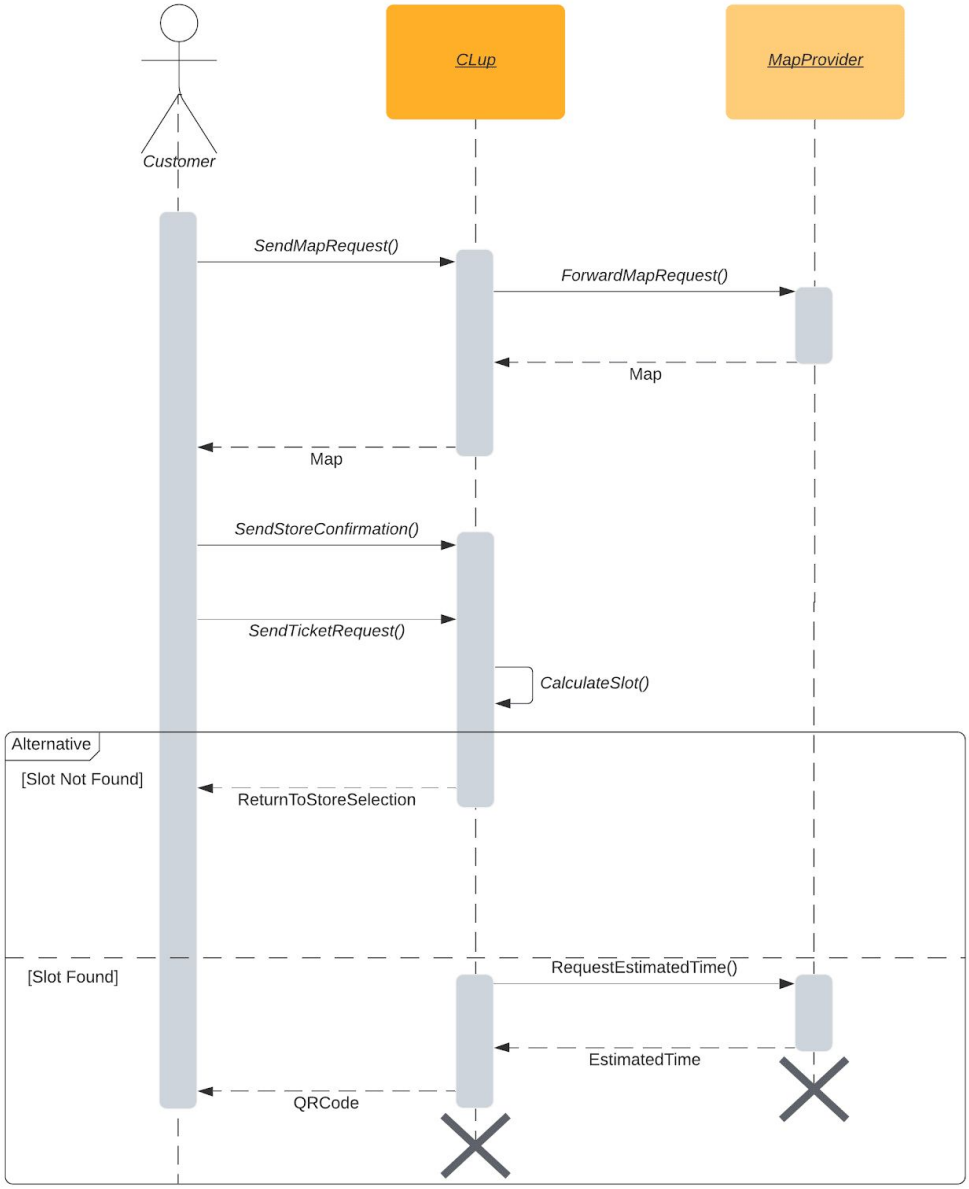

Figure 9: Sequence diagram of use case "Line-up remotely"

- Name: Line-up physically
  - Actor: Customer
  - Entry Conditions: The Customer is at the entrance of the Store in order to line-up from the device with Clup app installed made available by the Store.
  - Event Flow:
    1. The Customer is in front of the store without an internet connection
    2. The Customer goes to the nearest tablet displaying the CLup app
    3. The Customer prints the physical ticket and lines up
  - Exit Conditions: The ticket is printed without issues
  - Exceptions:
    1. There are too many Customers already in line and a ticket cannot be printed at the moment.
  - Special Requests: No special requests needed.

- Name: Book a visit
  - Actor: Customer
  - Entry Conditions: The Customer has opened the app and is logged in.
  - Event Flow:
    1. The Customer opens the map;
    2. The Customer selects a store;
    3. The Customer selects the option “book a visit”;
    3. The Customer inserts the list of departments in which he intends to buy;
       3.1. If the Customer is a long term user, the system suggests him his usual visit’s duration;
       3.2. The Customer select a new visit duration if doesn’t select the last;
    4. The system calculates available time slots for the visit and shows them to the Customer;
    5. The Customer chooses a slot;
    6. The system sends the QR code and the remainder of the time of visit;
  - Exit Conditions:
    The visit is successfully booked and the Customer has their ticket with QR code and time of visit.
  - Exceptions:
    1. Time slot chosen by the Customer is no longer available so he has to choose another one.
    2. The chosen slot is not free, in this case the system asks the Customer if he wants to be notified when this slot will be available or if he wants to try searching for another store, subsequently repeating the procedure of booking a visit.
  - Special Requests: No special requests needed.

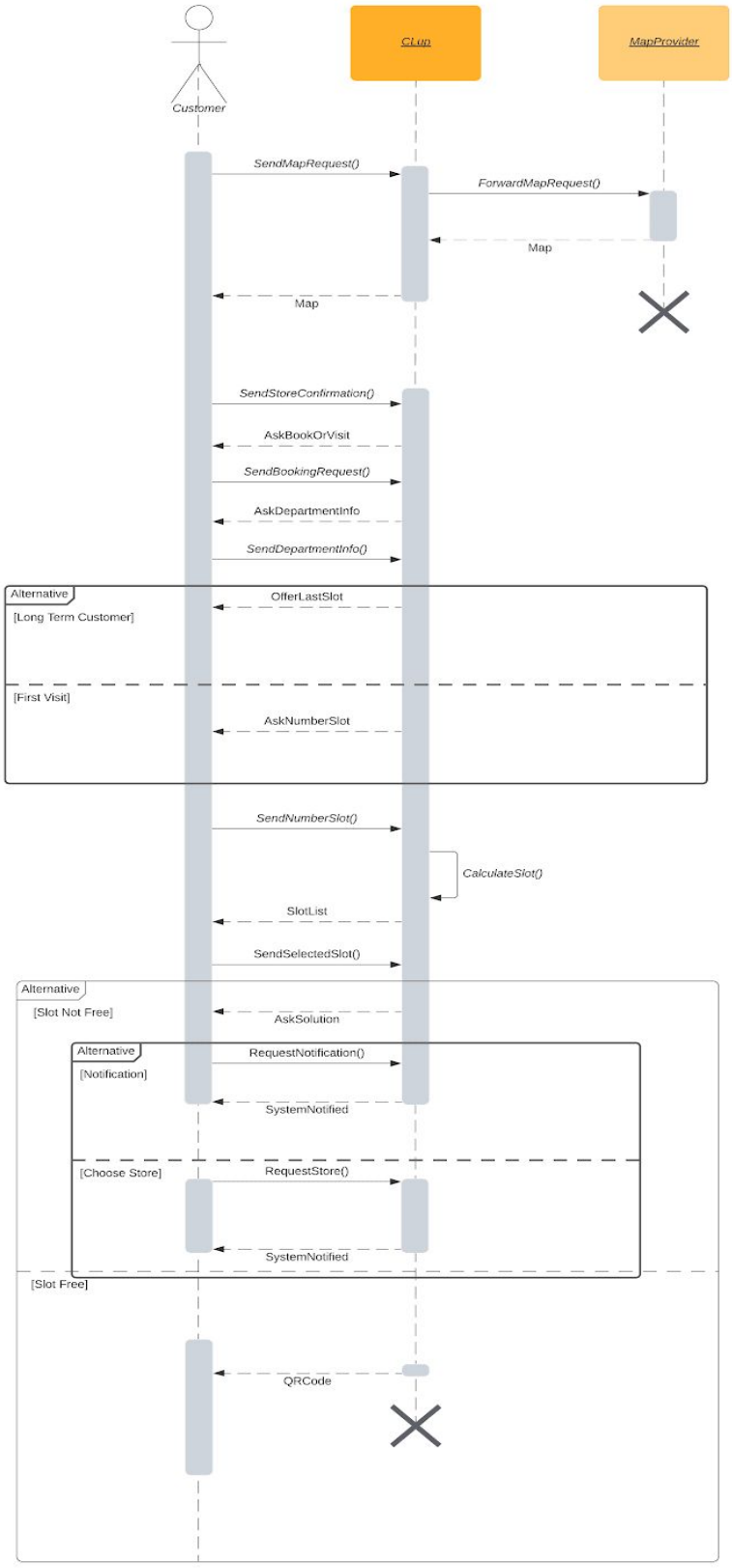

Figure 10: Sequence diagram of use case “Book a visit”

- Name: Cancel a visit
    - Actor: Customer
    - Entry Conditions: The Customer has opened the app, is logged in and has at least one visit booked
    - Event Flow:
        1. The application shows the list of all booked visits;
        2. The Customer chooses the visit he wants to cancel;
        3. The system asks confirmation;
        4. The Customer gives consent;
        5. The system frees the slot and cancels the visit;
    - Exit Conditions:
        The visit is successfully cancelled and the slot is now free.
    - Exceptions:
        1. The Customer denies the confirmation and the system returns to the previous state.
    - Special Requests: No special requests needed.

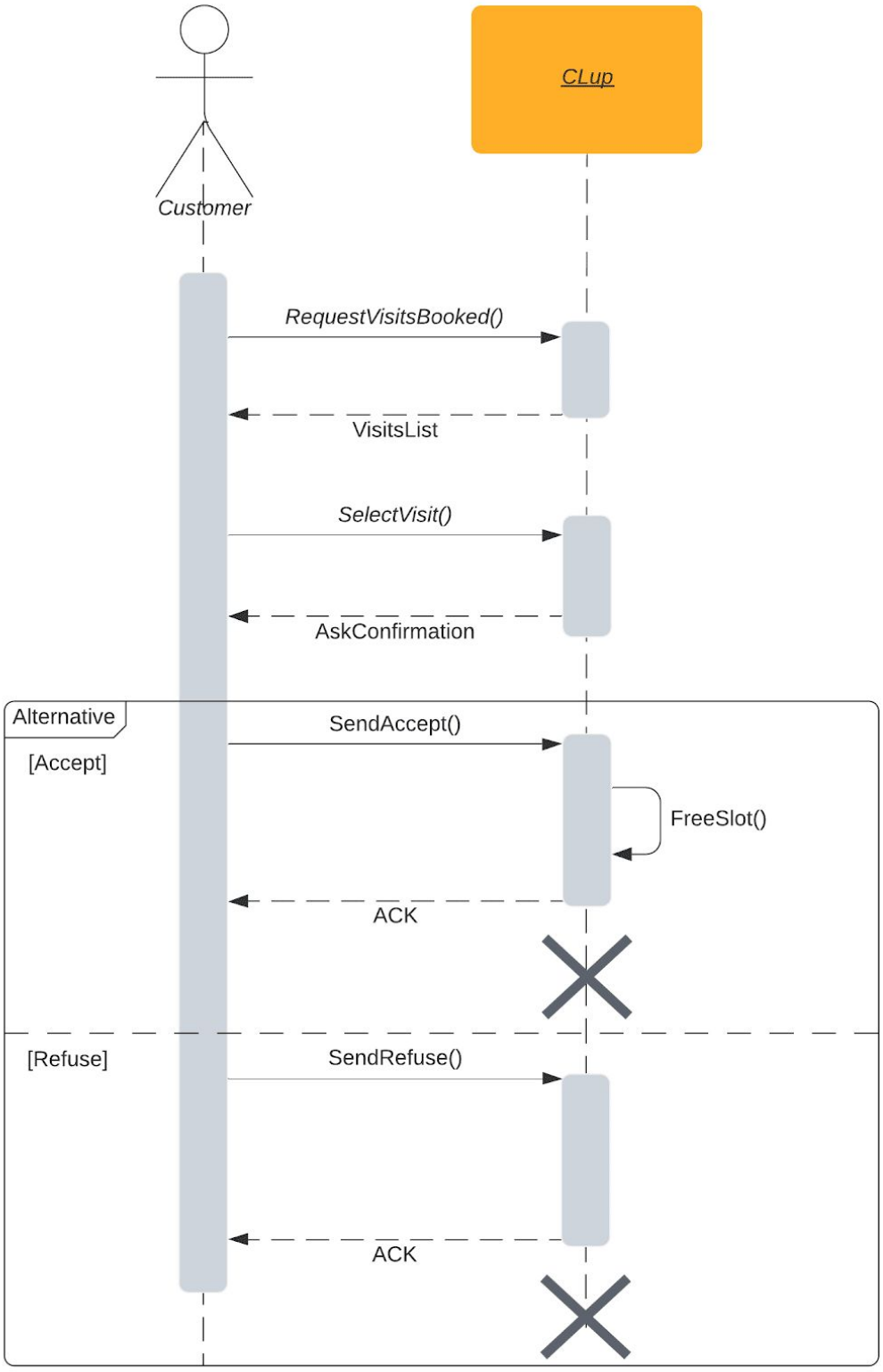

Figure 11: Sequence diagram of use case "Cancel a visit"

3.2.3 Use Cases: Manager

- Name: Manage influx of customers
    - Actor: Manager
    - Entry Conditions: The Manager is logged in.
    - Event Flow:
        1. The Manager clicks on the “check influx”;
        2. The system gives a summary of the total people inside the store and in each department;
        3. The Manager reduce the maximum number of people that can enter a department by clicking the “minus” button near each department name;
        4. The system saves the change and reduces the number of the slots.
    - Exit Conditions: The limit number has been successfully set.
    - Exceptions:
        1. The number of the department is set below 0, in this case the system prevents the change.
    - Special Requests: No special requests needed.

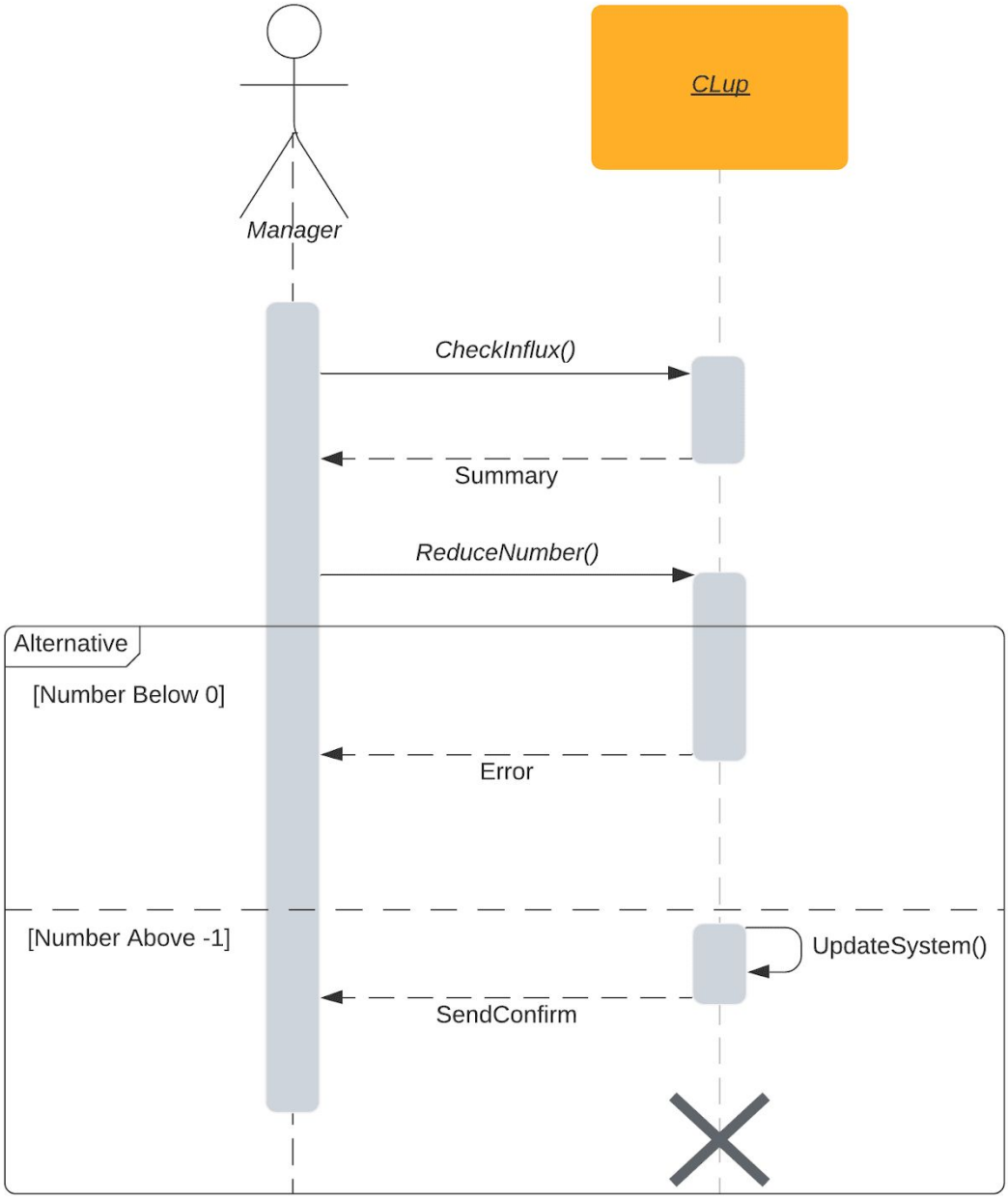

Figure 12: Sequence diagram of use case "Manage influx of Customers"

3.2.4 List of requirements

- R1: A Customer must be able to begin the registration process. During the process the system will ask the Customer to provide credentials.
- R2: A Customer must be logged in to the system using their credentials.
- R3: Store Manager must be logged in to the system using their credentials.
- R4: The system must be able to provide the list of stores on the map around the Customer
- R5: The system must be able to let the Customer select a store from the ones available on the map
- R6: A Customer must be able to choose to lineup for a store
- R7: A Customer must be able to choose to book a visit at a store
- R8: A Customer must be able to check the current position in the line to enter the store that he chose.
- R9: For each store, the system must be connected with a DB listing all the departments in that store.
- R10: During a “book a visit” process, the system must be able to suggest departments to the Customer based on previous visits to the store.
- R11: A Customer must be able to select a time slot from the available ones.
- R12: The system notifies the Customer 60 minutes before the booked visit to the store.
- R13: If the Customer chose “book a visit”, the system must be able to show the estimated time it takes to the Customer to reach the store.
- R14: The system must be able to connect to a printer and to print tickets.
- R15: The system must be able to show the ticket number and QR Code if a virtual ticket is taken.
- R16: Based on the number of people inside the store, an authorized store Manager must be able to regulate the influx of people coming in the store
- R17: The system must communicate with a DB that contains available time slots to “book a visit”
- R18: A Customer must be able to select a time slot from the non-available ones.
- R19: A Customer must be able to view previously booked visits.
- R20: A Customer must be able to cancel a previously booked visit.
- R21: A Customer must be able to receive notifications for available slots in the future.
- R22: A Store Manager must be able to monitor entrances through the “check influx” feature.

3.2.5 Traceability

In this section are provided two tables to keep track of the traceability of the main features of this project, in particular:

- Table 1: keep track of which goal is achieved through which Domain Assumptions and Requirements
- Table 2: keep track of the relationship between Use Cases and Requirements.

G1	The application must allow a Customer to become a registered User after providing credentials and his address.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R1	A Customer must be able to begin the registration process. During the process the system will ask the Customer to provide credentials
G2	The application must allow a user to digitally line up in a queue from his home.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
D4	The position of the Customer is obtained through a GPS system integrated in the Customer's device and is trustworthy
D5	The position retrieved using the GPS system is accurate with an error of 5 meters at most
D6	GPS is active while the app is running
D7	The Customer grants the application to have access to GPS position
R2	Customer must be logged in to the system using his/her credentials
R4	The system must be able to provide the list of stores on the map around the Customer
R5	The system must be able to let the Customer select a store from the ones available on the map
R6	A Customer must be able to choose to lineup for a store
R8	A Customer must be able to check the current position in the line to enter the store that he chose
R15	The system must be able to show the ticket number and QR Code if a virtual ticket is taken
G2.1	The application must allow a user to see his/her position in the queue.
D1	Each user has a smartphone

D2	The application is installed on a device
D3	The device must be connected to the internet
R2	Customer must be logged in to the system using his/her credentials
R4	The system must be able to provide the list of stores on the map around the Customer
R5	The system must be able to let the Customer select a store from the ones available on the map
R6	A Customer must be able to choose to lineup for a store
R8	A Customer must be able to check the current position in the line to enter the store that he chose
G3	The application must allow a registered user to book a visit to a selected store.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
D4	The position of the Customer is obtained through a GPS system integrated in the Customer's device and is trustworthy
D5	The position retrieved using the GPS system is accurate with an error of 5 meters at most
D6	GPS is active while the app is running
D7	The Customer grants the application to have access to GPS position
R2	Customer must be logged in to the system using his/her credentials
R4	The system must be able to provide the list of stores on the map around the Customer
R5	The system must be able to let the Customer select a store from the ones available on the map
R7	A Customer must be able to choose to book a visit at a store
R9	For each store, the system must be connected with a DB listing all the departments in that store
R10	During a "book a visit" process, the system must be able to suggest departments to the Customer based on previous visits to the store
R11	A Customer must be able to select a time slot from the available ones

R12	The system notifies the Customer 60 minutes before the booked visit to the store.
R13	If the Customer chose “book a visit”, the system must be able to show the estimated time it takes to the Customer to reach the store
R17	The system must communicate with a DB that contains available time slots to “book a visit”
G3.1	The application must allow a registered user to select departments in which he wants to buy.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R2	Customer must be logged in to the system using his/her credentials
R9	For each store, the system must be connected with a DB listing all the departments in that store.
R10	During a “book a visit” process, the system must be able to suggest departments to the Customer based on previous visits to the store.
G3.2	The application must allow a registered user to choose a proposed time slot for the visit.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R2	Customer must be logged in to the system using his/her credentials
R11	A Customer must be able to select a time slot from the available ones
R17	The system must communicate with a DB that contains available time slots to “book a visit”
G4	The application must allow a user to retrieve a QR code in case of “virtual lineup” or “book a visit”.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R2	Customer must be logged in to the system using his/her credentials

R15	The system must be able to show the ticket number and QR Code if a virtual ticket is taken.
G5	The application must allow a Customer to physically line-up in front of the store.
D8	There is a terminal on a Store entrance to physically line-up and print a ticket.
R14	The system must be able to connect to a printer and to print tickets.
G6	The application must allow a Store Manager to manage the number of customers for each supermarket department
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R3	Store Manager must be logged in to the system using his/her credentials.
R9	For each store, the system must be connected with a DB listing all the departments in that store.
R17	Based on the number of people inside the store, an authorized store Manager must be able to regulate the influx of people coming in the store
G7	The Application must allow the Store Manager to monitor entrances.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R3	Store Manager must be logged in to the system using his/her credentials.
R22	A Store Manager must be able to monitor entrances through “check influx” feature.
G8	The application should allow a registered Customer to cancel a visit previously booked.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
R19	A Customer must be able to view previously booked visits
R20	A Customer must be able to cancel a previously booked visit

G9	The application should send the Customer notifications when slots are available, if requested.
D1	Each user has a smartphone
D2	The application is installed on a device
D3	The device must be connected to the internet
D11	The Customer grants the application to send notifications
R18	A Customer must be able to select a time slot from the non-available ones
R21	A Customer must be able to receive notifications for available slots in the future

Table 1. Goals, Domain assumptions and Requirements mapping

User Type	Use Case	Goals
Customer	Register	G1, Mandatory steps to achieve most of the goals
	Login	
	Line-up remotely	G2, G2.1
	Line-up physically	G5
	Book a visit	G3, G3.1, G3.2, G9
	Cancel a visit	G8
Store Manager	Login	Mandatory step to achieve the goal for this actor
	Manage Customers Influx	G6, G7

Table2. Traceability matrix

3.3 Performance Requirements

The system has to manage lines in real-time and must be usable by a large number of Customers and Store Managers.
In order to achieve that it must guarantee to update lines' situation at least every 30 seconds. Also the available time slots to book a visit need to be updated in a few seconds every time a reservation is made.

3.4 Design Constraints

3.4.1 Standards compliance

The code should follow the requirements contained in this document. Furthermore, its comments should be clear and focused.
The application during the Register phase stores users' sensitive information, therefore the whole project is subject to the General Data Protection Regulation (GDPR).

3.4.2 Hardware limitations

The software application requires a mobile device able to capture the position and to scan QR codes (this last one only for the store's tablet).
The devices must be able to send data to the server side via internet connection.

3.4.3 Any other constraint

In order to manage efficiently the "book a visit" feature, the available time slots are organized in 30-minutes slots.

3.5 Software System Attributes

3.5.1 Reliability

The system must be running always in order to manage real-time lines, and also have to ensure integrity of data given the feature of showing to the Customer the list of its departments preferences.

3.5.2 Availability

The system requires an availability of 99.9% due to the real-time needing of the application, to notify positions in the queue to the Customers.
So to ensure that CLup works properly it suffices 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 year.

3.5.3 Security

The data stored in the system has to be encrypted, especially Customers' and Managers' credentials that have to be stored for months if not years.
The system should be protected against intrusion from agents that are not authorized to access it.

3.5.4 Maintainability

The system must guarantee a high level of maintainability. It has to be implemented using high level abstractions and avoiding hard-coding. Also, code must be well commented and documented.
Code must provide testing routines that cover at least 80% of the entire code.

3.5.5 Portability

The system must support Android and iOS operating systems for mobile devices.

4. Formal Analysis Using Alloy

4.1 Alloy Code

In this section we have described the Alloy model of CLup to check some of its crucial functionalities and their constraints.
In particular we prove that:

- a Customer can book only one visit for each TimeSlot;
- a single TimeSlot has a duration of 30 minute and a unique date;
- the capacity of a supermarket (and the capacity of each department) can't be violated for each timeslot;
- the capacity of a department can't exceed the capacity of its supermarket;
- a StoreManager is responsible for only one Supermarket;

```
sig Username{}

sig Date {}

abstract sig User {
	username : one Username
}

sig Customer extends User {
	bookings : set Booking
}

sig StoreManager extends User {}

sig Department {
	capacity: one Int
} {
	capacity >= 0
	one s : Supermarket | this in s.departments
}

sig Time {
	hour : one Int,
	minute : one Int,
}{
hour >= 0
hour <24
((minute = 0) or (minute = 30)) //constraints to model a 30 minute TimeSlot
}
```

```
sig TimeSlot {
        begin : one  Time,
        end :  one Time,
        date : one Date
}{
        //constraints to model TimeSlot duration
        (begin.hour = end.hour) or (end.hour = (1.add[begin.hour]))
}

sig Booking {
        Id: one Int,
        supermarket : one Supermarket,
        timeSlots : some TimeSlot,
        deps: some Department
} {
        all d: deps | d in supermarket.departments
        Id>0
}

sig Supermarket{
        bookings : set Booking,
        departments : some Department,
        capacity : one Int,
        managers : one StoreManager
}{
        capacity > 0
}

//each username belongs to only one User
fact UniqueUsernames {
        all disj u1, u2: User | u1.username != u2.username
}

//a booking belongs to only one Customer
fact OneBookingPerTimeSlot {
        all c : Customer |
        all t : TimeSlot |
        lone b : Booking |
        (b in c.bookings and
        t in b.timeSlots)
}
```

```
fact UniqueBookings {
	all c : Customer |
	all disj b, b' : Booking |
		(b in c.bookings and b' in c.bookings)
		implies
		(b.timeSlots & b'.timeSlots) = none
}

//one StoreManager for each supermarket
fact {all disj s1, s2 : Supermarket | s1.manager != s2.manager }

fact Tickets {
	all s: Supermarket |
		all b: s.bookings |
			b.supermarket = s
}

fact eachBookingConnectedToOneCustomer {
	all b : Booking | one c : Customer | b in c.bookings
}

//number of booking for each TimeSlot in a Supermarket can't exceed its capacity
fact RespectCapacitySupermarket {
	all s : Supermarket |
		all disj b, b': Booking |
	(b.supermarket = s and
		b'.supermarket = s and
		b.timeSlots - b'.timeSlots = none )
	implies
		#(b + b') <= s.capacity
}

//number of booking for each supermarket's department can't exceed its capacity
fact RespectDepartmentCapacity {
	all d : Department |
		all disj b, b': Booking |
	(d in b.deps and
		d in b'.deps and
		b.timeSlots - b'.timeSlots = none )
	implies
		#(b + b') <= d.capacity
}
```

```
//each supermarket's TimeSlot has a duration of 30 minutes, and it's modelling as below:
//        -if the timeslot begin with minute=0 then it ends with minute=30 and the hours at beginning and at the //end are the same;
//        -otherwise, if the timeslot begins with minute=30 then it ends with minute =0 and the hours at the end are //equals to the hours at the beginning plus one.
fact { all t : TimeSlot |
        ((t.begin.minute = 0) => ((t.end.minute = 30) and (t.begin.hour = t.end.hour)))
        and
        ((t.begin.minute = 30) => (( t.end.minute = 0) and (t.end.hour = (1.add[t.begin.hour]))))
}

//department capacity can't exceed supermarket capacity
fact CapacityRespected {
        all s : Supermarket |
        all d : Department |
        (d in s.departments)
        implies
        d.capacity < s.capacity
}


fact UniqueTimeSlots {
        all disj t, t' : TimeSlot |
        (t.date = t'.date)
        implies
        (t.begin != t'.begin)
}

fact ContiguousTimeSlotInBooking {
        all t : TimeSlot |
        all b : Booking |
        (t in b.timeSlots and not
        biggestTimeSlotInBooking[b, t])
        implies
        (one t1 : TimeSlot |
        t in b.timeSlots and
        contiguousTime[t, t1])
}


fact UniqueTimes {
   all disj t, t' : Time | t.hour != t'.hour or t.minute != t'.minute
}

pred contiguousTime (t1, t2 : TimeSlot) {
        t1.date = t2.date and
        t1.end = t2.begin
}
```

```
pred biggestTimeSlotInBooking (b : Booking, t : TimeSlot) {
        t in b.timeSlots and
        all t1 : b.timeSlots |
        t1 != t
        implies
        (t1.begin.hour < t.begin.hour) or (t1.begin.hour = t.begin.hour and t1.begin.minute <
t.begin.minute)
}



//assert that the TimeSlot is correctly modelled
assert minuteCheck {
   all t : Time | t.minute = 30 or t.minute = 0
}

//assert that there aren't bookings that don't belong to a Supermarket
assert eachBookingConnectedToSupermarket {
   all b : Booking | b.supermarket != none
}

check minuteCheck
check eachBookingConnectedToSupermarket

pred show {
        #Customer >= 3
        #Booking >= 4
        #Supermarket >= 2
        #TimeSlot >= 3
        some b : Booking |
        #(b.timeSlots) >= 2

}

run show for 6 but 6 Int
```

4.2 Metamodel

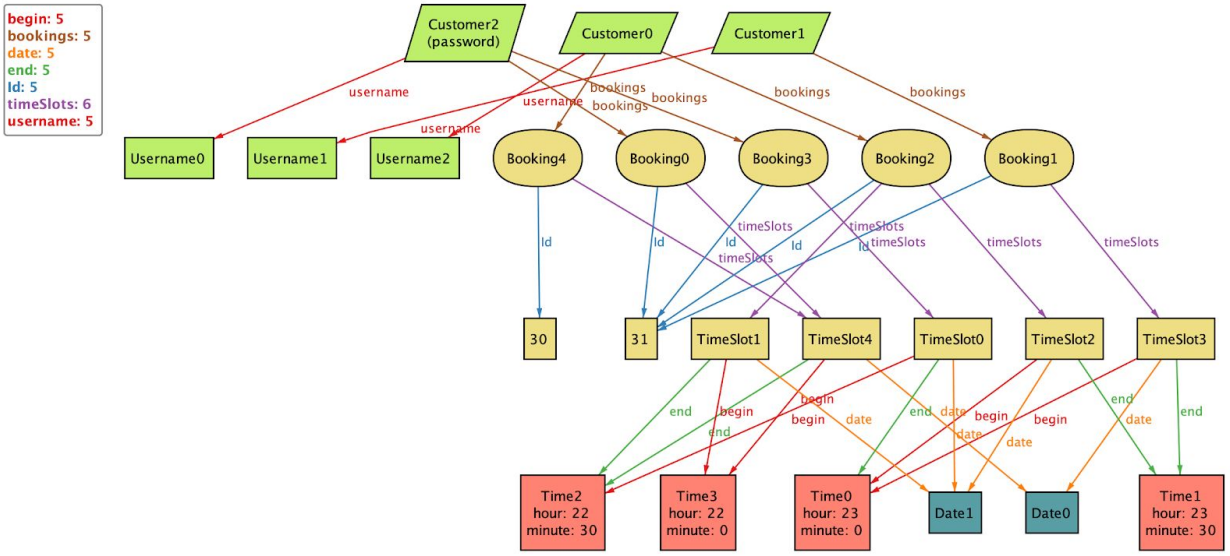

Figure 13: "Book a visit" seen from the perspective of Customers and Bookings

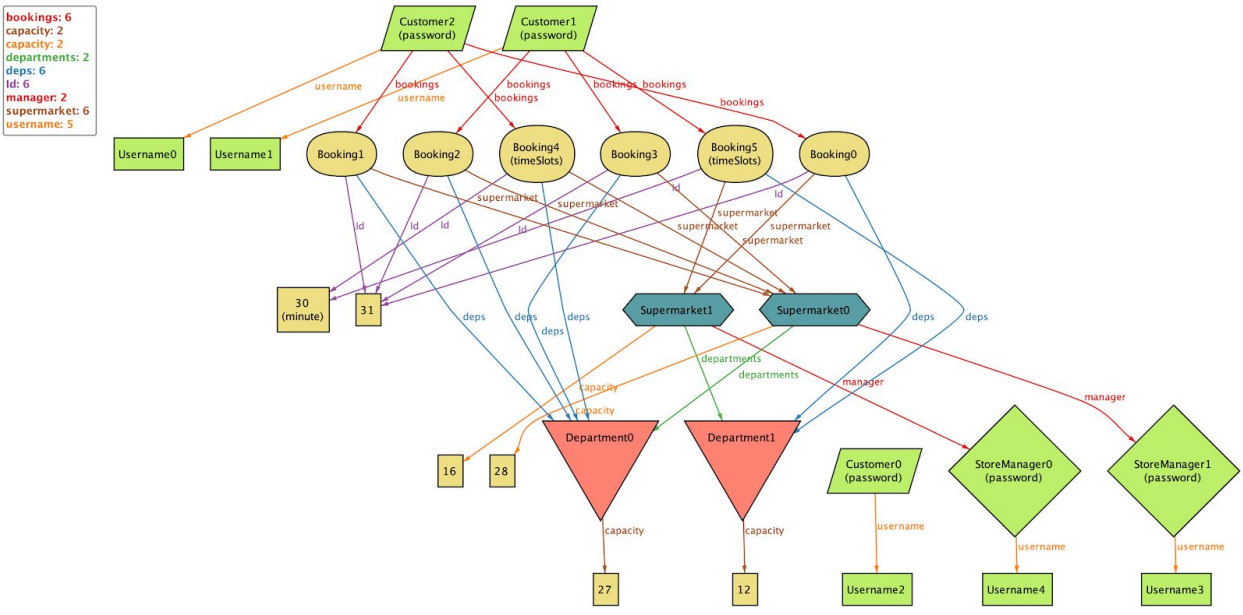

Figure 14: "Book a visit" seen from the perspective of Store Managers and Supermarkets

4.3 Result of Assertions

```
Executing "Check minuteCheck"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  0 vars. 0 primary vars. 0 clauses. 34ms.
  No counterexample found. Assertion may be valid. 0ms.

Executing "Check eachBookingConnectedToSupermarket"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  7698 vars. 381 primary vars. 18284 clauses. 29ms.
  No counterexample found. Assertion may be valid. 3ms.
```

5. Effort Spent

Topic	Hours
Introductory discussion on the assignment	3h
Purpose and Scope	2h
Overall description	6h
Functional Requirements	12h
Non-Functional Requirements	6h
Formal Analysis Using Alloy	10h

Table 3: Effort spent by Andrea Zanetti

Topic	Hours
Introductory discussion on the assignment	3h
Purpose and Scope	2h
Overall Description	5h
Functional Requirements	10h
Non-Functional Requirements	7h
Formal Analysis Using Alloy	12h

Table 4. Effort spent by Giampiero Repole

Topic	Hours
Introductory discussion on the assignment	3h
Purpose and Scope	4h
Overall Description	6h
Functional Requirements	8h
Non-Functional Requirements	6h
Formal Analysis Using Alloy	12h

Table 5: Effort spent by Pasquale Occhinegro

6. References

- All charts were made using Lucidchart https://lucid.app
- AlloyAnalyzer tool https://alloytools.org/download.html
- Google Docs
- Figma https://figma.com