

POLITECNICO DI MILANO

AA 2020-2021



Computer Science and Engineering

SOFTWARE ENGINEERING2

# RASD

**Requirement Analysis and Specification Document**

## Team Members

ID	Surname	Name
10609805	De Berardinis	Alessia
10611044	Dragoni	Arianna

Prof.ssa Elisabetta Di Nitto

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
1.1 Purpose	3
1.2 Scope	3
1.2.1 World Phenomena	4
1.2.2 Shared Phenomena	4
1.2.3 Goals	5
1.3 Definitions, Acronyms, Abbreviations	5
1.3.1 Definitions	5
1.3.2 Acronyms	5
1.3.3 Abbreviations	6
1.4 Reference Documents	6
1.5 Document Structure	6
<b>2. Overall Description</b>	<b>7</b>
2.1 Product Perspective	7
2.1.1 UML Description	7
2.1.2 State Charts	10
2.2 Product Functions	11
2.3 User Characteristics	14
2.4 Assumptions, Dependencies and Constraints	14
2.4.1 Domain Assumptions	14
<b>3. Specific Requirements</b>	<b>15</b>
3.1 External Interface Requirements	15
3.1.1 User Interfaces	15
3.1.2 Hardware Interfaces	17
3.1.3 Software Interfaces	17
3.1.4 Communication Interfaces	17
3.2 Functional Requirements	18
3.2.1 List of Requirements	18
3.2.2 Mapping	20
3.2.3 Use Cases	27
3.2.3.1 Use Cases Description	27
3.2.3.2 Use Case Diagram	35
3.2.4 Sequence Diagrams	36
3.2.5 Scenarios	41
3.3 Performance Requirements - Non Functional Requirements	43
3.4 Design Constraints – Non Functional Requirements	43
3.4.1 Standard Compliance	43
3.4.2 Hardware Limitations	43
3.4.3 Any Other Constraints	43
3.5 Software System Attributes – Non functional Requirements	43
3.5.1 Reliability and Availability	
3.5.2 Security	44
3.5.3 Maintainability	44
3.5.4 Portability	44
3.5.5 Scalability	44
<b>4. Formal Analysis Using Alloy</b>	<b>45</b>
4.1 Alloy Code	45
4.2 MetaModel	50
4.3 Results of Assertions	52
4.4 Results of Predicates	52
<b>5. Effort</b>	<b>53</b>

## 1. Introduction

### 1.1 Purpose

The purpose of the document is to provide a detailed description of CLup application, by highlighting its functionalities and services. The idea of developing this application starts from the need of managing the shopping experience during Coronavirus emergency, with the aim of avoiding lines that create crowds in front of the stores.

The application is meant to be used by the shop managers to regulate the influx of people in the buildings – through the scanning of QR codes at the entrances/exits – saving the customers from having to stand outside stores, which could be risky.

This is achieved thanks to a system that provides the customers with a position in the queue of a shop, by receiving a progressive number.

In this way, they should wait until their number is called, or close to being called, to get to the store.

For an effective success of the application, the system should give the customers a reasonably precise estimation of the time they should wait before their turn.

An additional service of the application allows people who do not have access to the required technology to hand out tickets on the spot, thanks to the presence of one or more physical totems.

CLup also allows users to “book” a visit: either a customer can indicate the approximate expected duration of the visit, or, in case of long-term customers, this time could be computed by the system through the analysis of their previous visits. In addition, to contain more people in the store, or in general to manage more effectively the affluence of people, users can indicate the categories of items they intend to buy.

### 1.2 Scope

The scope of the application is to allow store managers to regulate the number of people in the shops in order to prevent situations of gathering, that can be dangerous during the period of Coronavirus emergency.

On the other side, the application allows customers to do the shopping in a safer way both avoiding them queuing in front of the building with other people and letting them keep distances from the other customers. Specifically, according to the international rules, the distance between two people must be at least one meter. For this purpose, CLup is able to organize the entrances of people according to the capacity of the building, which is provided by the shop manager during his/her registration.

To do the shopping, the customers already registered must login to the application and “take a ticket”: in this way they obtain a position in the queue and the estimation of the waiting time.

In order to make the lining up mechanism effective, either the customer activates the localization, and the system calculates the time needed to get to the shop, or the customer estimates by himself/herself the time required from the place he/she is.

The customer has to scan the QR code generated with the ticket when arriving/leaving to/from the store to improve the efficiency of the system.

The users can also exploit the advanced functionality of “booking a visit”, indicating an approximate duration of the visit they intend to do. The system can infer this time for long-term customers by analyzing their previous visits.

The customers have also the possibility to indicate which kind of items they think they will buy, to allow the system to better organize the entrances by predicting which spaces will be fully occupied in the store and those who have not reached their maximum capacity. Also in this case the customers have to scan the QR code at the entrances/exits of the shop.

CLup is very simple to use because it includes all demographics, and to take into account visits from people who cannot use the application, the system provides the fallback option of handing out tickets on the spot through the presence of one or more totems positioned around the building.

### 1.2.1 World phenomena

<b>WP1:</b> store managers divide shops in departments
<b>WP2:</b> store managers organize the shops in order to avoid gathering due to coronavirus pandemic
<b>WP3:</b> the customer wants to do shopping in a safer way
<b>WP4:</b> the customer has a smartphone
<b>WP5:</b> people maintain a one-meter distance between each other
<b>WP6:</b> shops have totems at their entrance to allow people to take tickets on the spot
<b>WP7:</b> people arrive in front of their shop without queuing
<b>WP8:</b> shops have QR code scanners at the entrances/exits of the shop
<b>WP9:</b> shops have a maximum capacity

### 1.2.2 Shared phenomena

<b>SP1:</b> user takes the ticket online
<b>SP2:</b> user takes the ticket on the spot through the totem
<b>SP3:</b> user books a visit online
<b>SP4:</b> user checks when it is his/her turn
<b>SP5:</b> system assigns a position in the queue for each customer
<b>SP6:</b> system analyzes the data about the visits of the long-term customers
<b>SP7:</b> system provides the user the estimation of the waiting time
<b>SP8:</b> system provides the user the number of people above him/her in the queue
<b>SP9:</b> system provides to the user the time needed to reach the destination by using their position
<b>SP10:</b> the day of the booked visit the application reminds it to the customer through a notification
<b>SP11:</b> user provides to the system the estimated time necessary to reach the shop
<b>SP12:</b> user provides to the system the duration of the visit
<b>SP13:</b> user scans the QR code when he/she enters the shop
<b>SP14:</b> user scans the QR code when he/she leaves the shop
<b>SP15:</b> user provides to the system the categories of items he/she wants to buy
<b>SP16:</b> user receives from the system the QR code associated to a reservation
<b>SP17:</b> the system sends periodic notifications to the user about the remaining waiting time

### 1.2.3 Goals

The goals have direct impact on the world and so they are expressed either in terms of world phenomena or in terms of shared + world phenomena.

<b>G1:</b> allow people to take a ticket online to do shopping
<b>G2:</b> allow people to take the ticket on the spot
<b>G3:</b> allow people to maintain distance rules while doing shopping
<b>G4:</b> allow store managers to organize in a more efficient way the store
<b>G5:</b> allow people to avoid lines in front of the shop
<b>G6:</b> allow store managers to track the affluence of people in their shops
<b>G7:</b> allow people to book a visit online to do shopping
<b>G8:</b> allow the system to build statistics on data which are stored in the database

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

- **Customer:** who signs in the application with the purpose of doing shopping.
- **Store Manager:** who adds a store in the application and has the purpose of organizing it according to the new rules introduced to contrast the coronavirus pandemic.
- **User:** who signs in the application and uses the available services for him/her purposes. The user can be a customer or a store manager.
- **Demographic:** particular sector of population (children, seniors, adults, ...).
- **Ticket:** the number received which corresponds to the position in the queue.
- **Totem:** multimedial structure which allows people who cannot use the application to take a ticket.
- **QR Scanner:** digital structure where people have to scan their tickets in order to make the mechanism more efficient.
- **QR Code:** bidimensional matrix composed of black modules put in a square schema used to memorize information about a ticket.
- **Reservation:** is a general term used to indicate either a taken ticket or a booked visit.
- **Book a visit:** refers to the option for the user to book a visit in a selected shop for the following days.
- **Take a ticket:** refers to the option for the customer to have a position in the queue of a shop in the current day.
- **System:** it is the software that is the objective of the document.

### 1.3.2 Acronyms

- **QR code:** Quick Response Code
- **GPS:** Global Positioning System
- **API:** Application Programming Interface
- **UI:** User Interface
- **GDPR:** General Data Protection Regulation
- **RASD:** Requirement Analysis and Specification Document

### 1.3.3 Abbreviations

- **WPn**: World Phenomenon number n
- **SPn**: Shared Phenomenon number n
- **Gn**: goal number n
- **AF**: advanced function
- **Rn**: requirement number n
- **Dn**: domain assumptions number n

### 1.4 Reference Documents

- Specification Document: “R&DD Assignment A.Y. 2020-2021.pdf”
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications
- Available slides on beep

### 1.5 Document Structure

**Chapter 1:** it describes what the scope of the software is by showing the product, the application domain and the boundaries between the application domain and the external environment.

It identifies the world and shared phenomena and sets the goals of the application. It also explains what is included in the following sections, to guide the readers in the reading process. Moreover, for a better understanding it provides definitions, acronyms and abbreviations. This chapter also contains the reference documents, which are the sources useful for the creation of RASD document.

**Chapter 2:** it contains the overall description of the project.

First of all it provides the Product Perspective – which clarifies who are the actors the software interacts with – through the Uml description and the related class diagram and highlights in some State Charts the possible behaviors of the system in a series of events. This section is followed by the Product Functions, where the main functionalities provided by the application are shown, and the User Characteristics, with the users and their expected abilities. Finally the constraints that limit the developers are described and the domain assumptions (what is assumed to be true for the system to work properly) are shown.

**Chapter 3:** represents the main chapter of the document. In the first section it describes more in details how the software is going to interface with the external world through the specification of the External Interface Requirements (specifically the User Interfaces with the mockups, the Hardware Interfaces, the Software Interfaces and the Communication Interfaces). This section is followed by the one of the Functional Requirements, which provides: the specification of the Requirements and their mapping with the goals and the domain assumptions, the use cases and the related use case diagram, the sequence diagrams and some scenarios. The following three sections are about non-functional Requirements: Performance requirements, Design constraints and Software System Attributes (Reliability, Availability, Security, Maintainability, Portability).

**Chapter 4:** provides the description of the main goals through the formal modeling activity. The model is represented through Alloy language by using facts, predicates and assertions.

**Chapter 5:** defines the efforts (number of hours) each group member has spent for draft of the document.

## 2. Overall description

### 2.1 Product Prospective

#### 2.1.1 UML Description

The UML below shows the requirements, the interfaces of the machine and the interactions between the machine and the world.

CLup is going to be used as a mobile application and so both the customers and the shop managers have to download it on their devices to exploit the services provided. For this scope they need to be registered and based on their role, different kinds of information are requested to them.

The shop manager registers his/her shop by giving information about him/herself and his/her shop. The customer must provide personal information and, if he/she accepts, also his/her localization.

The UML highlights both the basic service of *taking a ticket online* to do shopping in a safer way and the advanced functionality of *booking a visit* for the following days.

To better understand the description of the UML provided above, it is necessary to pay particular attention to the **Data**, **Analysis** and **Schedule** classes.

Data class contains the information related to the customers, including all the taken tickets and the booked visits, and it is used to build Analysis.

Analysis are done in order to inform the user about real-time information and to define the schedule of each shop. The schedule contains the list of **Reservations** (ticket-visit) in a shop

- in the current day and past ones (“take a ticket” and “book a visit” option)
- in the future days (only in the case of “book a visit” option)

The system fills the white spaces of the current day when a customer takes a ticket and the ones of the future days when he/she books a visit.

Each **Shop** has a set of instances of the class Schedule, which contains the time slots (opening-closing time) for each day.

Both **Customers** and **Shop Managers** are **Users** and, based on their role, they use the application in different ways:

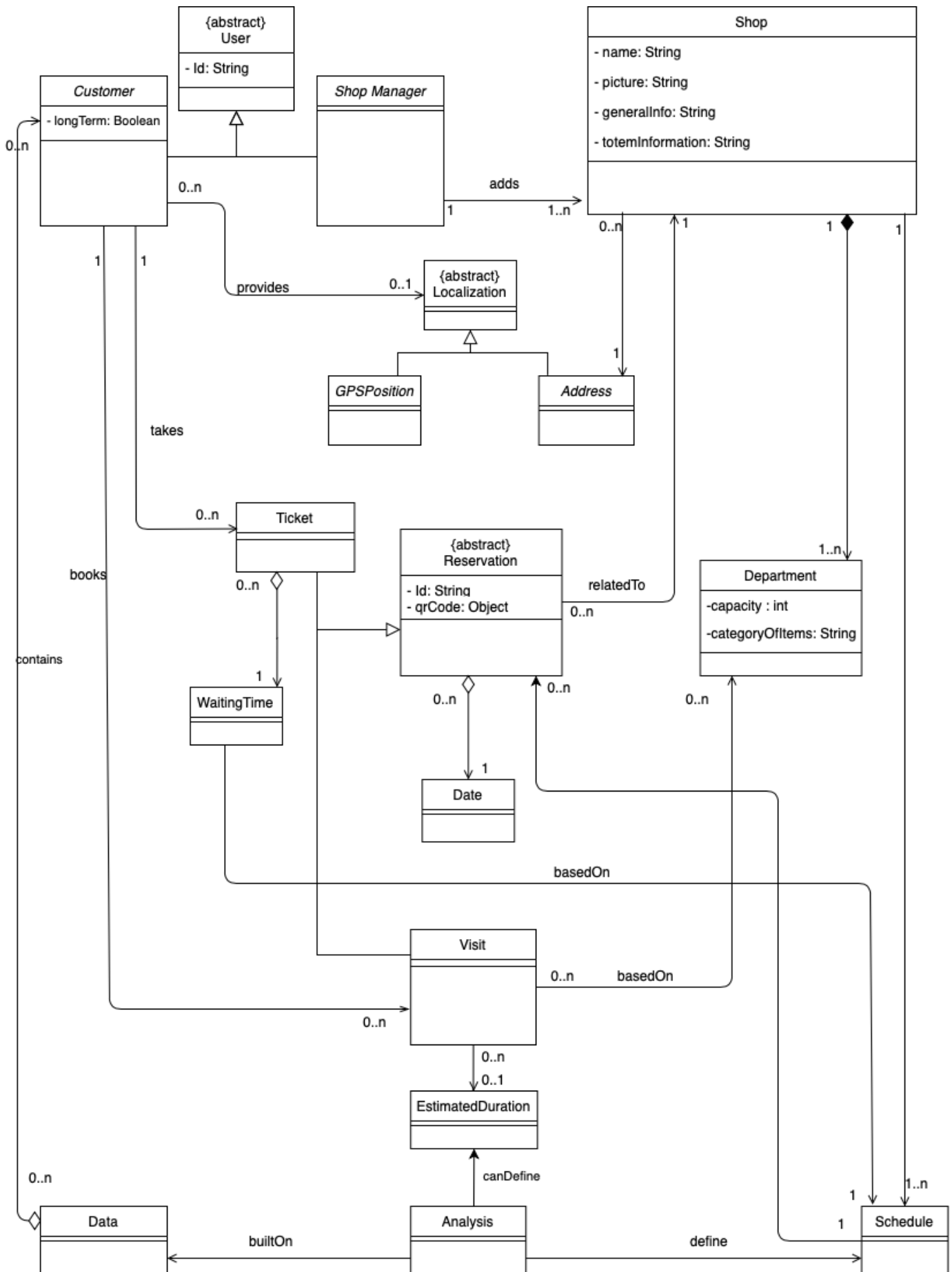
- The Shop manager can only add his/her shops to CLup Application with all the related information about the store. Each shop is composed of many **departments**, each of which has the capacity (provided by the shop manager) and the categories of items it is possible to find inside.
- The customer can take a ticket or book a visit. Both tickets and visits are Reservations.

Take a ticket: the **ticket**, characterized by the id and the associated QR code, contains the **date** and the **waiting time**, which is computed by the system by using the schedule based on the average duration of customers' permanence in the shop on the same day of the previous week. For each customer, the permanence durations in the shop are saved in the database. In the ticket there is also a reference to the related shop. To take a ticket the customer can provide his/her **localization**, that can be either a **GPS localization** or the **address**.

Book a visit: the customer can exploit the advanced functionality "book a visit" for the following days. The **visit**, like the ticket, is related to a shop and is characterized by an Id and the associated QR code. The customer can provide the category of items he/she has intention to buy and an approximate **expected duration** of the visit. Alternatively, for long term customers, the system can compute the duration (\*) by analyzing the data of their previous visits. Instead, for new customers who do not provide the duration of the visit, the system uses the average durations of all the other users of the same day of the previous week.

\*the duration is computed through the difference between the exit time and the entry time obtained thanks to the scanned QR codes.





### 2.1.2 State Charts

In this section it is provided an abstract description of the system's behavior, which is represented by a series of events and by the evolution over time of its states.

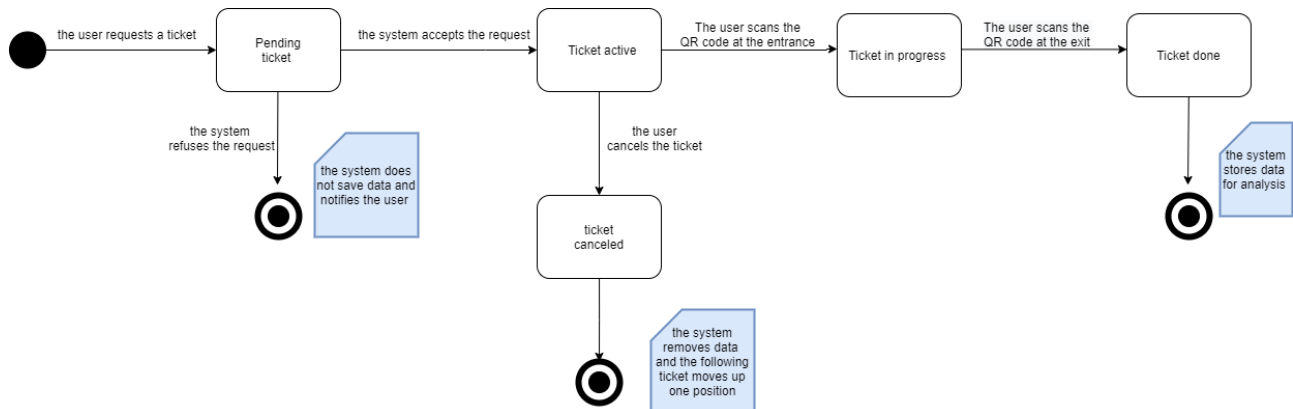


Figure 1 – State Diagram: user takes a ticket

The first state diagram describes the behavior of the system when the user wants to take a ticket online.

Once the request is received, the system elaborates it and decides whether to accept or refuse it.

The system refuses if the queue is full until the closing time of the selected shop, the user is already in another queue (he has also taken a ticket for another shop) or he/she takes a ticket in a time slot which overlaps with a booked visit. In this case the system notifies the user and does not save the data related to the ticket.

Instead, if the system accepts the request, the user can download the generated QR code and go to the store. When he/she enters/leaves the shop he/she has to scan it.

In any moment within his/her turn, the user can cancel his/her ticket. In this case the system removes the data about it from the schedule and the following tickets move up one position.

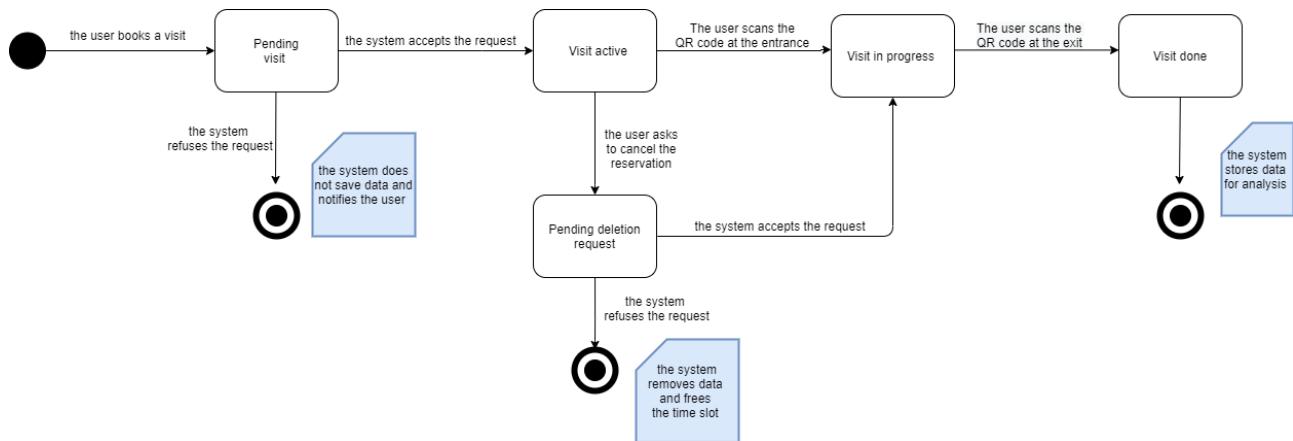


Figure 2 – State Diagram: user books a visit

The state diagram in figure 2 models the case in which the user decides to exploit the advanced functionality of *booking a visit*. Everything is the same of the *take a ticket* option but with two exceptions:

- The system can refuse the request of reservation's deletion in case the user tries to cancel it less than one hour before the time of the visit.
- The system refuses the request of reservation if the time of the ticket overlaps with the time slot of the visit or if the user tries to book two or more visits that overlap each other. Or again, if the user selects a day that has not available time slots

## 2.2 Product Functions

This section highlights the major functionalities of the software by underlining what it is able to offer.

### Take a ticket

It is the main functionality of the system.

It gives the user the possibility of taking a ticket, which corresponds to the position in a queue identified by a progressive number. With the ticket, the user also receives the QR code he/she must scan at the entrance/exit of the shop.

When the customer already registered and logged in to CLup opens the application, a list of shops - with the name, address, opening/closing time, departments and categories of items - is shown to him/her. In case of localization this list is sorted from the closest store to the furthest one.

Through this list and the number of people already present in the queue of a shop, the user can choose the best option for his/her purposes, take his/her position in the queue and move from the shop when it is almost his/her turn, also thanks to the periodic notifications sent by the system.

In case the queue is full until the closing time of the shop, the system does not allow to take a ticket for it. The same happens if the user selects a shop when it is closed (it is necessary to wait until the shop opens). Moreover, if a customer has already taken a ticket, and so he/she has already a

position in a queue, he/she cannot take another ticket, but he has the possibility to book a visit for a following day.

In any moment the user has the opportunity to check the status of the queue and to know about the remaining time he/she has to wait before his/her turn.

After the ticket request:

- If the user during the registration (or in any other moment through the application settings) has given access for the GPS localization, the system calculates and provides him/her the estimation of the time needed to reach the destination.
- The user who has not provided the GPS localization, is asked to write the address (city, street, civic number) in order to receive the estimated time by the system.
- The user can also skip the localization request without giving any information about his/her position and so he/she has to estimate by him/herself the time required to reach the destination.

In any moment the user can decide to activate GPS from the device's settings.

Moreover, when he/she wants, the customer can cancel the ticket and quit the queue he/she is positioned in, and consequently the following tickets move up one position in the queue.

## **Book a visit**

The user has also the possibility to “book a visit” for the following days.

Once selected the shop, the system first asks him/her to choose a day and select a time slot from the available ones, and then to provide the category of items he/she has intention to buy and an estimated approximate duration of the visit. The user can skip this request and directly receive the ticket, the related QR code and a recap with the day, time and the shop of the visit.

The user has the possibility to book a visit also when the shop is closed. Instead, the system can refuse the request of the visit's reservation:

- If the user tries to book two or more visits that overlap with each other or if he/she takes a ticket whose time overlaps with the time slot of the booked visit.
- If the user selects a day that has not available slot times.

The system sends a notification to the user two hours before the visit to remind him/her the reservation made.

The user has the possibility to cancel his/her reservation until one hour before the start of the time slot selected.

## Register a shop

When the shop manager decides to use the service provided by CLup, he/she opens the application and registers himself/herself as “Store manager”. With this kind of registration, the system requires:

- Personal information about him/herself
- The name and address of the shop
- Opening and closing time of the shop
- A picture and additional information of the shop (optional)
- The departments present in the building with the related capacity and the categories of items that can be found inside
- The position of the totem(s) where people can take the ticket if they do not have access to the application
- The scan of the certificate that validates his/her role in the shop

The store manager, once registered, must equip the shop with the necessary technologies:

- One or more **totems** - to allow people to take the ticket on the spot – that must be connected to the system to insert the ticket in the schedule of the day. It is able to print the ticket with the associated QR code.
- **QR code scanners** positioned at the entrances/exits of the shops to track the number of people inside the building and to modify the schedule real time.

## Show the status of the queue

The user - customer, after receiving the ticket, has always the possibility to enter the application and check the status of the queue in the dedicated field (“Check Status” field). Specifically, the system provides the user the number of people before him/her in the queue and the estimation of the time he/she has to wait.

## Show Shop’s Information and daily reports

The shop manager can open the application and visualize the list of his/her shops. For each shop the number of people that have accessed the store until that moment (*daily report*) is provided. Once selected a shop, the system shows him/her the information related to the shop (name, address, opening and closing time, picture, departments, categories of items for each department) and the *daily reports* of the current day and of the past ones.

## Show the list of shops with related information

Once opened the application the customer can see a list of shops. If he/she has provided the localization the list is sorted from the closest to the farthest one. For each shop the application shows how many people are already in the corresponding queue, which categories of items the store provides, the address and the opening and closing time of the store.

## Cancel a reservation

The user can cancel the reservation from the appropriate section of the application

- in any moment in case of *Take a ticket* option
- until one hour before the start of the slot time selected in case of *Book a visit* option

## Take a ticket on the spot

The user who cannot use CLup application on a device has the possibility to directly go to the shop and get the by using the totem outside the store, which prints the ticket with the QR code.

## 2.3 User Characteristics

This section provides the list of the actors and how the system interacts with them.

1. **User-Customer:** who downloads CLup application on his mobile phone, registers on it and logs in to *take a ticket* to do the shopping in a safer way or *book a visit* in a shop for the following days.
2. **User-Store manager:** who exploits CLup Application and adds his/him shop to it to give the customer the possibility to avoid queuing in front of the store with other people and letting them do the shopping keeping the right distances from the other customers.

## 2.4 Assumptions, dependencies and constraints

In this section it is explained everything that cannot be controlled by the system, but it is required by it to work properly and so assumed to be true.

### 2.4.1 Domain Assumptions

<b>D1:</b> shop managers have put totems at the entrances of the registered store
<b>D2:</b> shop managers have QR scanners at the entrances/exits of the registered store
<b>D3:</b> customer arrives to the store at the starting time of the booked visit without delay
<b>D4:</b> customer arrives at the shop at the time associated with his/her ticket without delay
<b>D5:</b> customer scans his/her QR code when he/she enters the shop
<b>D6:</b> customer scans his/her QR code when he/she leaves the shop
<b>D7:</b> the customer always reaches the shop if he/she has made a reservation(ticket-visit)
<b>D8:</b> if the customer cannot go to the shop, he/she remembers to cancel the reservation made
<b>D9:</b> if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
<b>D10:</b> the customer who wants to book a visit owns a device
<b>D11:</b> user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
<b>D12:</b> the customer who wants to take a ticket online owns a device
<b>D13:</b> each customer who does not take the ticket online, always takes the ticket through the totem before entering the store
<b>D14:</b> when active, the GPS provides the exact position with an error of ten meters at most
<b>D15:</b> the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
<b>D16:</b> the shop's address, the pictures, the categories of items, the various departments with associated capacities and any other shop's information provided by the store manager are real

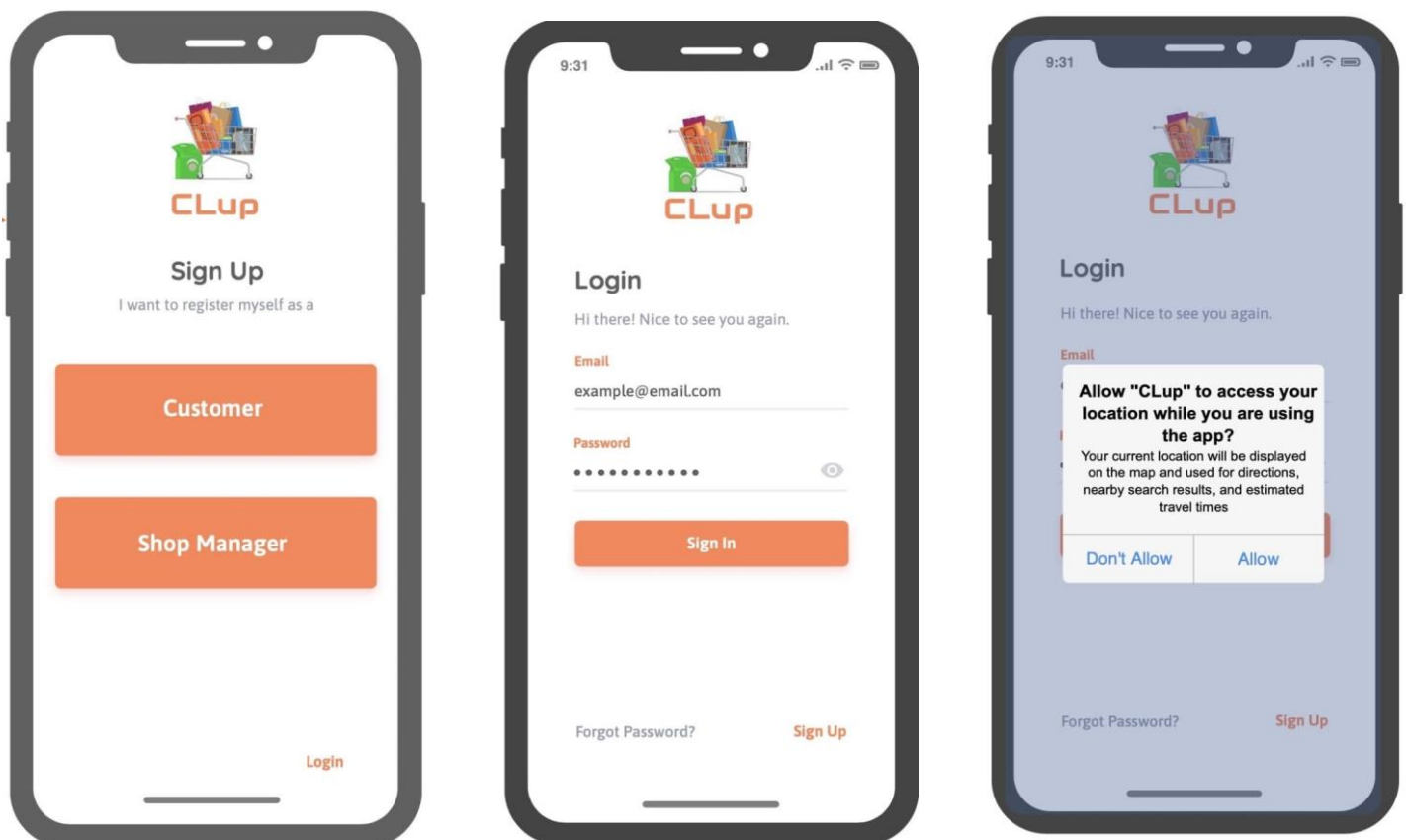
### 3. Specific Requirements

#### 3.1 External Interface Requirements

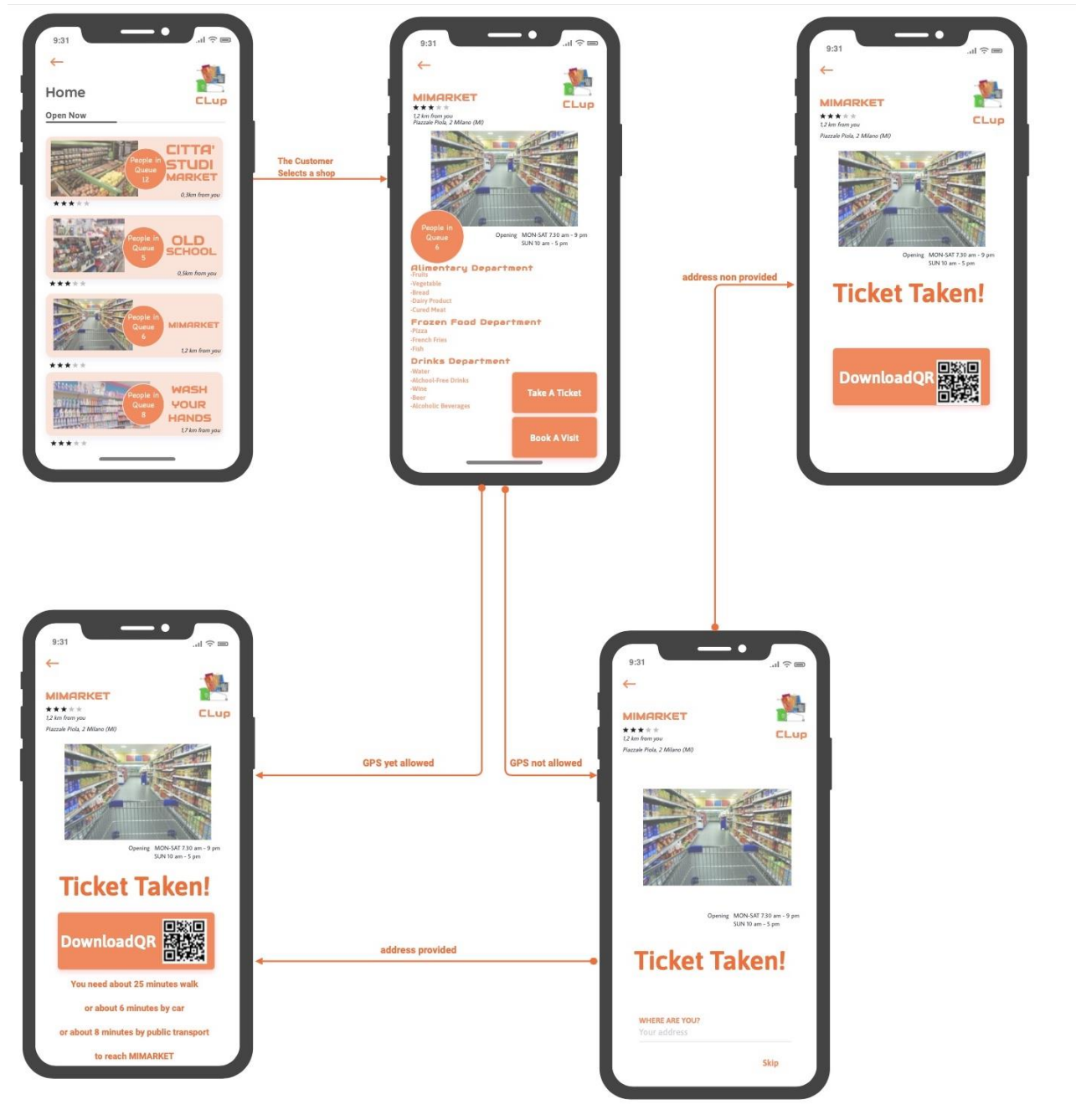
##### 3.1.1 User Interfaces

This section provides some examples of how the application will look like. More interfaces can be found in the User Interface Design section of the Design Document.

The following mockups show the Sign up of the user-customer and the Login of both the user-customer and user-Shop Manager



The following mockup shows the steps the customer should follow to take a ticket online.





### 3.1.2 Hardware interfaces

- **Mobile phone:** a mobile phone is required to the customer to take a ticket online (\*) or book a visit and to the store manager to register his/her shop in Clup Application.  
\*If the customer wants to take the ticket on the spot, owning a mobile phone is not necessary.
- **Totem:** the stores need to have at least one totem positioned in a visible place around the shop so that tickets taken by on the spot can be managed by the system. The totem represents a sort of proxy between the customers and the system.
- **QR scanner:** the scanners must be placed at the entrances and exits of the stores to make people scan the QR code associated with their tickets. Thanks to them it is possible to monitor the entrances and exits and so to update the schedule with real time data.

### 3.1.3 Software interfaces

- **City Map:** the use of a map of the city is necessary for the customer who decides to provide the position to choose a shop close to him/her and calculate the time and route to reach the destination. For this purpose, the system uses a public API.
- **Calendar:** the application allows customers to “book a visit” and for this reason it needs to show him/her a calendar to choose the day of the planned visit. For this scope there are several available APIs, like Google Calendar.

### 3.1.4 Communication interfaces

Devices connect and communicate with the server through Internet connection.

## 3.2 Functional requirements

### 3.2.1 List of Requirements

The functional requirements describe what the system provides to the user and highlight the interactions between the system and its environment. They are expressed in terms of shared phenomena.

<b>R1.</b> The system should provide the report of daily entrances/exits to the shop managers
<b>R2.</b> The system should ask the user-customer for authorization of GPS position
<b>R3.</b> The system should ask the user to tick privacy Terms of Services and Privacy Policy box
<b>R4.</b> The system should require the user- customer to be logged in to exploit the services related to Take a Ticket option provided by the application
<b>R5.</b> The system should require the user- customer to be logged in to exploit the services related to Book a visit option provided by the application
<b>R6.</b> The system should require the user-shop manager to be logged to exploit the services provided by the application
<b>R7.</b> The system should notify the user- customer (about the remaining waiting time) real time
<b>R8.</b> The system should ask the user- customer who want to book a visit to fill the mandatory fields of date and time of the visit
<b>R9.</b> The system notifies the user- customer two hours before the booked visit to remind it to him/her
<b>R10.</b> The system must save the data provided by the users when they register themselves
<b>R11.</b> The system must save the generated ticket taken online in the schedule
<b>R12.</b> The system should save the ticket of Book a visit real time in the schedule
<b>R13.</b> The system should generate automatically the QR code associated to the ticket
<b>R14.</b> The system should save real time the ticket taken through the totem in the schedule
<b>R15.</b> The system should allow the user-customer to decide between Take a ticket and book a visit options
<b>R16.</b> The system should allow the user-customer to provide in an optional field the category of items they want to buy (in Book a visit option)

<b>R17.</b> The system should allow the user-customer to provide in an optional field the estimated duration of the visit (in Book a visit option)
<b>R18.</b> The system should allow the user-customer to cancel the reservation of a visit until one hour before the starting of the visit
<b>R19.</b> The system should allow the user-customer to cancel a ticket within his/her turn
<b>R20.</b> The system should allow the user-customer to check the status of a queue of a shop
<b>R21.</b> The system should reserve a place in the selected time slot for each customer who books a visit
<b>R22.</b> The system should allow the user-customer to check in each moment the remaining time for his/her turn
<b>R23.</b> The system should allow the user-customer to check in each moment the number of people before him/her in a queue of a shop
<b>R24.</b> The system should assign a position in the queue for each customer who takes a ticket
<b>R25.</b> The system should allow the user-shop manager to edit information about their shop
<b>R26.</b> The system should ask the store manager to provide a certificate that validates his/her role
<b>R27.</b> The system should require the user- customer to be logged in to exploit the secondary services provided by book a visit (see information of the user, edit information...)
<b>R28.</b> The system should ask the user-customer to register himself/herself to the application by filling the form with mandatory fields
<b>R29.</b> The system should ask the user-store manager to register himself/herself to the application by filling the form with mandatory fields

### 3.2.2 Mapping

Goals	Domain Assumptions	Requirements
<b>G1</b>	D4, D5, D6, D7, D8, D9, D12, D11, D15	R2, R3, R4, R27, R7, R10, R11, R13, R15, R19, R20, R22, R23, R24
<b>G2</b>	D1, D4, D5, D6, D7, D11, D13	R14, R13, R 24
<b>G3</b>	D3, D4, D7, D16, D13, D15, D14	R7, R10, R11, R12, R13, R14, R16, R17, R20, R1, R22, R23, R24, R21
<b>G4</b>	D1, D2, D5, D6, D7, D8, D9, D11, D13, D15, D16	R1, R6, R10, R11, R12, R14, R18, R19, R25, R24, R21
<b>G5</b>	D3, D4, D11, D7, D14, D15, D16	R1, R2, R7, R9, R11, R12, R14, R16, R17, R20, R22, R23, R21, R24
<b>G6</b>	D1, D2, D5, D6, D7, D8, D9, D11, D13	R1, R6, R13,
<b>G7</b>	D3, D5, D6, D7, D8, D9, D10, D11, D15, D16	R2, R3, R5, R27, R8, R9, R10, R12, R13, R15, R16, R17, R18, R21
<b>G8</b>	D5, D6, D8, D9, D13, D15, D16	R1, R10, R8, R11, R12, R14, R16, R17, R24, R25, R21

<b>G1</b>	<b>ALLOW PEOPLE TO TAKE A TICKET ONLINE TO DO SHOPPING</b>
D4	customer arrives at the shop at the time associated with his/her ticket without delay
D5	customer scans his/her QR code when he/she enters the shop
D6	customer scans his/her QR code when he/she leaves the shop
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D8	if the customer cannot go to the shop, he/she remembers to cancel the reservation made
D9	if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
D12	the customer who wants to take a ticket online owns a device
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
R2	The system should ask the user-customer for authorization of GPS position
R3	The system should ask the user to tick privacy Terms of Services and Privacy Policy box
R4	The system should require the user- customer to be logged in to exploit the services related to Take a Ticket option provided by the application
R7	The system should notify the user- customer (about the remaining waiting time) real time

R10	The system must save the data provided by the users when they register themselves
R11	The system must save the generated ticket taken online in the schedule
R13	The system should generate automatically the QR code associated to the ticket
R15	The system should allow the user-customer to decide between <i>Take a ticket</i> and <i>book a visit</i> option
R19	The system should allow the user-customer to cancel a ticket within his/her turn
R20	The system should allow the user-customer to check the status of a queue of a shop
R22	The system should allow the user-customer to check in each moment the remaining time for his/her turn
R23	The system should allow the user-customer to check in each moment the number of people before him/her in a queue of a shop
R24	The system should assign a position in the queue for each customer who takes a ticket
R28	The system should ask the user-customer to register himself/herself to the application by filling the form with mandatory fields

<b>G2</b>	<b>ALLOW PEOPLE TO TAKE THE TICKET ON THE SPOT</b>
D1	shop managers have put totems at the entrances of the registered store
D4	customer arrives at the shop at the time associated with his/her ticket without delay
D5	customer scans his/her QR code when he/she enters the shop
D6	customer scans his/her QR code when he/she leaves the shop
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
D13	each customer who does not take the ticket online, always takes the ticket through the totem before entering the store
R13	The system should generate automatically the QR code associated to the ticket
R14	The system should save real time the ticket taken through the totem in the schedule
R24	The system should assign a position in the queue for each customer who takes a ticket

<b>G3</b>	<b>ALLOW PEOPLE TO MAINTAIN DISTANCE RULES WHILE DOING SHOPPING</b>
D3	customer arrives to the store at the starting time of the booked visit without delay
D4	customer arrives at the shop at the time associated with his/her ticket without delay
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D13	each customer who does not take the ticket online, always takes the ticket through the totem before entering the store

D14	when active, the GPS provides the exact position with an error of ten meters at most
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
D16	the shop's address, the pictures, the categories of items, the various departments with associated capacities and any other shop's information provided by the store manager are real
R1	The system should provide the report of daily entrances/exits to the shop managers
R7	The system should notify the user- customer (about the remaining waiting time) real time
R10	The system must save the data provided by the users when they register themselves
R11	The system must save the generated ticket taken online in the schedule
R12	The system should save the ticket of Book a visit real time in the schedule
R13	The system should generate automatically the QR code associated to the ticket
R14	The system should save real time the ticket taken through the totem in the schedule
R16	The system should allow the user-customer to provide in an optional field the category of items they want to buy (in Book a visit option)
R17	The system should allow the user-customer to provide in an optional field the estimated duration of the visit (in Book a visit option)
R20	
R21	The system should reserve a place in the selected time slot for each customer who books a visit
R22	The system should allow the user-customer to check in each moment the remaining time for his/her turn
R23	The system should allow the user-customer to check in each moment the number of people before him/her in a queue of a shop
R24	The system should assign a position in the queue for each customer who takes a ticket

<b>G4</b>	<b>ALLOW STORE MANAGERS TO ORGANIZE IN A MORE EFFICIENT WAY THE STORE</b>
D1	shop managers have put totems at the entrances of the registered store
D2	shop managers have QR scanners at the entrances/exits of the registered store
D5	customer scans his/her QR code when he/she enters the shop
D6	customer scans his/her QR code when he/she leaves the shop
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D8	if the customer cannot go to the shop, he/she remembers to cancel the reservation made
D9	if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)

D13	each customer who does not take the ticket online, always takes the ticket through the totem before entering the store
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
D16	the shop's address, the pictures, the categories of items, the various departments with associated capacities and any other shop's information provided by the store manager are real
R1	The system should provide the report of daily entrances/exits to the shop managers
R6	The system should require the user-shop manager to be logged to exploit the services provided by the application
R11	The system must save the generated ticket taken online in the schedule
R12	The system should save the ticket of Book a visit real time in the schedule
R18	The system should allow the user-customer to cancel the reservation of a visit until one hour before the starting of the visit
R19	The system should allow the user-customer to cancel a ticket within his/her turn
R21	The system should reserve a place in the selected time slot for each customer who books a visit
R24	The system should assign a position in the queue for each customer who takes a ticket
R25	The system should allow the user-shop manager to edit information about their shop
R29	The system should ask the user-store manager to register himself/herself to the application by filling the form with mandatory fields

<b>G5</b>	<b>ALLOW PEOPLE TO AVOID LINES IN FRONT OF THE SHOP</b>
D3	customer arrives to the store at the starting time of the booked visit without delay
D4	customer arrives at the shop at the time associated with his/her ticket without delay
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
D14	when active, the GPS provides the exact position with an error of ten meters at most
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
D16	the shop's address, the pictures, the categories of items, the various departments with associated capacities and any other shop's information provided by the store manager are real
R1	The system should provide the report of daily entrances/exits to the shop managers
R2	The system should ask the user-customer for authorization of GPS position
R7	The system should notify the user- customer (about the remaining waiting time) real time
R9	The system notifies the user- customer two hours before the booked visit to remind it to him/her
R11	The system must save the generated ticket taken online in the schedule

R12	The system should save the ticket of Book a visit real time in the schedule
R14	The system should save real time the ticket taken through the totem in the schedule
R16	The system should allow the user-customer to provide in an optional field the category of items they want to buy (in Book a visit option)
R17	The system should allow the user-customer to provide in an optional field the estimated duration of the visit (in Book a visit option)
R20	The system should allow the user-customer to check the status of a queue of a shop
R21	The system should reserve a place in the selected time slot for each customer who books a visit
R22	The system should allow the user-customer to check in each moment the remaining time for his/her turn
R23	The system should allow the user-customer to check in each moment the number of people before him/her in a queue of a shop
R24	The system should assign a position in the queue for each customer who takes a ticket

<b>G6</b>	<b>ALLOW STORE MANAGERS TO TRACK THE AFFLUENCE OF PEOPLE IN THEIR SHOPS</b>
D1	shop managers have put totems at the entrances of the registered store
D2	shop managers have QR scanners at the entrances/exits of the registered store
D5	customer scans his/her QR code when he/she enters the shop
D6	customer scans his/her QR code when he/she leaves the shop
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D8	if the customer cannot go to the shop, he/she remembers to cancel the reservation made
D9	if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
D13	each customer who does not take the ticket online, always takes the ticket through the totem before entering the store
R1	The system should provide the report of daily entrances/exits to the shop managers
R6	The system should require the user-shop manager to be logged to exploit the services provided by the application
R13	The system should generate automatically the QR code associated to the ticket
R29	The system should ask the user-store manager to register himself/herself to the application by filling the form with mandatory fields

<b>G7</b>	<b>ALLOW PEOPLE TO BOOK A VISIT ONLINE TO DO SHOPPING</b>
D3	customer arrives to the store at the starting time of the booked visit without delay
D5	customer scans his/her QR code when he/she enters the shop



D6	customer scans his/her QR code when he/she leaves the shop
D7	the customer always reaches the shop if he/she has made a reservation(ticket-visit)
D8	if the customer cannot go to the shop, he/she remembers to cancel the reservation made
D9	if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
D10	the customer who wants to book a visit owns a device
D11	user has the QR code associated with his/her ticket once arrived at the shop either on the device or in any other way (for instance he/she has printed it)
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
D16	the shop's address, the pictures, the categories of items, the various departments with associated capacities and any other shop's information provided by the store manager are real
R2	The system should ask the user-customer for authorization of GPS position
R3	The system should ask the user to tick privacy Terms of Services and Privacy Policy box
R5	The system should require the user- customer to be logged in to exploit the services related to Book a visit option provided by the application
R8	The system should ask the user- customer who want to book a visit to fill the mandatory fields of date and time of the visit
R9	The system notifies the user- customer two hours before the booked visit to remind it to him/her
R10	The system must save the data provided by the users when they register themselves
R12	The system should save the ticket of Book a visit real time in the schedule
R13	The system should generate automatically the QR code associated to the ticket
R15	The system should allow the user-customer to decide between <i>Take a ticket</i> and <i>book a visit</i> option
R16	The system should allow the user-customer to provide in an optional field the category of items they want to buy (in Book a visit option)
R17	The system should allow the user-customer to provide in an optional field the estimated duration of the visit (in Book a visit option)
R18	The system should allow the user-customer to cancel the reservation of a visit until one hour before the starting of the visit
R21	The system should reserve a place in the selected time slot for each customer who books a visit
R27	The system should require the user- customer to be logged in to exploit the secondary services provided by book a visit (see information of the user, edit information...)
R28	The system should ask the user-customer to register himself/herself to the application by filling the form with mandatory fields

<b>G8</b>	<b>ALLOW THE SYSTEM TO BUILD STATISTICS ON DATA WHICH ARE STORED IN THE DATABASE</b>
D5	customer scans his/her QR code when he/she enters the shop

D6	customer scans his/her QR code when he/she leaves the shop
D8	if the customer cannot go to the shop, he/she remembers to cancel the reservation made
D9	if the customer remembers to cancel the booking of the visit too late, he/she goes to the store anyway
D13	each customer who does not take the ticket online, always takes the ticket through the totem before entering the store
D15	the various information that can be provided by the users (position address, category of items that he/she wants to buy, approximate duration of the visit) is reliable
D16	the shop's address, the pictures, the categories of items, the various departments with associated capacities and any
R1	The system should provide the report of daily entrances/exits to the shop managers
R8	The system should ask the user- customer who want to book a visit to fill the mandatory fields of date and time of the visit
R10	The system must save the data provided by the users when they register themselves
R11	The system must save the generated ticket taken online in the schedule
R12	The system should save the ticket of Book a visit real time in the schedule
R14	The system should save real time the ticket taken through the totem in the schedule
R16	The system should allow the user-customer to provide in an optional field the category of items they want to buy (in Book a visit option)
R17	The system should allow the user-customer to provide in an optional field the estimated duration of the visit (in Book a visit option)
R21	The system should reserve a place in the selected time slot for each customer who books a visit
R24	The system should assign a position in the queue for each customer who takes a ticket
R25	The system should allow the user-shop manager to edit information about their shop
R28	The system should ask the user-customer to register himself/herself to the application by filling the form with mandatory fields
R29	The system should ask the user-store manager to register himself/herself to the application by filling the form with mandatory fields

### 3.2.3 Use Cases

#### 3.2.3.1 Use Cases Description

- The customer signs up to Clup Application

<b>Name</b>	Signup Customer
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has already downloaded and installed the application on the device
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user enters the application that shows the homepage</li> <li>2. the user clicks on the "Sign up" button</li> <li>3. the user chooses the "Customer" option (between "Customer" and "Shop manager" options)</li> <li>4. the user fills the mandatory fields (email, password, name, surname)</li> <li>5. the user fills the optional field (birthdate)</li> <li>6. the user ticks the "I agree to the Terms of Services and Privacy Policy" box</li> <li>7. the user clicks on the "Continue" button</li> <li>8. the user chooses between "Don't Allow" and "Allow" options with relation to the access of his/her GPS localization</li> <li>9. the system stores the user's data</li> </ol>
<b>Exit Condition</b>	The user is correctly registered to CLup Application and so he/she has the possibility to exploit the functionalities for the customer
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user was already registered and so the system asks the user to login</li> <li>2. the username already exists and so the system warns the user asking him/her to choose another one</li> <li>3. the user does not tick the "I agree to the Terms of Services and Privacy Policy" box and so the system warns the user to do it in order to confirm the registration</li> <li>4. the user does not fill all the mandatory fields and so the system asks him/her to insert missing information.</li> <li>5. the internet connection suddenly drops and so the system sends the "The connection is lost" message</li> </ol>

- **The Shop Manager signs up to CLup Application**

<b>Name</b>	Sign up Shop Manager
<b>Actor</b>	User-shop manager
<b>Entry Condition</b>	The user has already downloaded and installed the application on the device
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user enters the application that shows the homepage</li> <li>2. the user clicks on the "Sign up" button</li> <li>3. the user chooses the "Shop Manager" option (between "Customer" and "Shop Manager" options)</li> <li>4. the user fills the mandatory fields about himself (email, password, name, surname)</li> <li>5. the user fills the optional field (birthdate)</li> <li>6. the user ticks the "I agree to the Terms of Services and Privacy Policy" box</li> <li>7. the user clicks on the "Continue" button</li> <li>8. the user uploads the certificate that validates his/her role (mandatory)</li> <li>9. the user fills the mandatory fields about the store (Name and Address)</li> <li>10. the user uploads the picture of the store (not mandatory)</li> <li>11. the user clicks on "Finish" button</li> <li>12. the system validates the certification</li> <li>13. the user fills the mandatory fields about his/her shop (Departments' list with capacity and category of items, position of totems, opening and closing time)</li> <li>14. the user fills the optional fields about the additional information about his/her shop (description of the departments and /or of the product)</li> <li>15. the user clicks on the "End Registration" button</li> <li>16. the system stores the user's data</li> </ol>
<b>Exit Condition</b>	The user is correctly registered to CLup Application and so he/she has the possibility to exploit the functionalities for the shop manager. Now the shop is visible in the list of the stores of the application.
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user was already registered and so the system asks the user to login</li> <li>2. the username already exists and so the system warns the user asking him/her to choose another one</li> <li>3. the user does not tick on the "I agree to the Terms of Services and Privacy Policy" box and so the system warns the user to do it in order to confirm the registration</li> <li>4. the user does not fill all the mandatory fields and the system asks him/her to insert missing information</li> <li>5. the certification uploaded by the user is not valid and the system asks him/her to upload it again.</li> <li>6. the internet connection suddenly drops, and the system sends the "The connection is lost" message.</li> </ol>

- **The user logs in to the application**

In the following use case, we consider the general User (both Customer and Shop Manager) because the login phase is the same. The User Homepage is different for the customer and the shop manager.

<b>Name</b>	Log-in
<b>Actor</b>	User
<b>Entry Condition</b>	The application is correctly installed, and the user has already signed up
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user opens the application on his/her device</li> <li>2. the user enters the credentials (Email and Password)</li> <li>3. the user clicks on the “Sign In” button</li> <li>4. The system opens the User’s homepage</li> </ol>
<b>Exit Condition</b>	The user is successfully logged in and he/she has access to the offered services
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user enters a wrong email, and the system asks him/her to re-insert it.</li> <li>2. the user enters a wrong password, and the system asks him/her to re-insert it.</li> <li>3. the internet connection suddenly drops, and the system sends the “the connection is lost” message.</li> </ol>

- **The shop manager checks the information about his shop and monitor the daily affluence of people**

<b>Name</b>	See shop information and daily reports
<b>Actor</b>	User-shop manager
<b>Entry Condition</b>	The user has logged in and the application is open in the shop manager’s home page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user clicks on the “Menu” button</li> <li>2. the user clicks on the “About my Shops” button</li> <li>3. the system shows the user the list of his/her stores with the number of people that have accessed them until that moment</li> <li>4. the user selects a shop</li> <li>5. the system provides the user the details of the selected shop and the <i>Daily Reports</i> of the current day and of the past ones</li> </ol>
<b>Exit Condition</b>	The user knows all the information about the selected shop and its daily reports
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the internet connection suddenly drops and the system sends the “the connection is lost” message.</li> </ol>

- The user takes a ticket

<b>Name</b>	Take a ticket
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has logged in and the application is open in the home page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user selects a shop from the list of the available ones</li> <li>2. the system shows the information about the selected shop (people already in the queue, category of items, address, opening and closing time)</li> <li>3. the user chooses "Take a Ticket" between "Take a ticket" and "Book a visit" options</li> <li>4. the system saves the ticket in the schedule</li> <li>5. if the user has not already given access to the GPS localization, he/she decides whether to provide or not the address.</li> <li>6. if the system has access to the position of the user (either GPS or address), the user receives the estimated time to reach the shop.</li> <li>7. The system shows the estimated time, the number of people before him in the queue and the QR code associated to the ticket</li> <li>8. if he/she wants, the user can download the generated QR code by clicking on "Download QR code" button</li> </ol>
<b>Exit Condition</b>	The user has successfully taken a ticket
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user takes a ticket while he/she is already in the queue and the system warns him/her by sending the message "You are already in the queue of another shop"</li> <li>2. the user provides a nonexistent address and the system asks him/her to write another one.</li> <li>3. the user wants to take the ticket for a shop whose queue is full until the closing time and the system warns him/her by sending the message "This shop is unfortunately full! Please choose another one"</li> <li>4. the internet connection suddenly drops and the system sends the "The connection is lost" message.</li> </ol>

- The user books a visit

<b>Name</b>	Book a visit
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has logged in, the application is open in the home page

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user selects a shop from the list of the available ones</li> <li>2. the system shows the information about the selected shop (people already in the queue, category of items, address, opening and closing time)</li> <li>3. the user chooses “Book a visit” between the “Take a ticket” and “Book a visit” options</li> <li>4. the user selects the day he/she wants to do the visit in a calendar</li> <li>5. the user selects between the available time slots</li> <li>6. the user fills some/all of the optional fields (category of items he wants to buy/estimated duration of the visit)</li> <li>7. the user clicks on “Confirm reservation” button</li> <li>8. the system saves the information in the schedule</li> <li>9. the user receives the associated QR code and a recap of the visit (date, time and shop)</li> <li>10. if he/she wants, the user can download the generated QR code by clicking on “Download QRcode” button</li> </ol>
<b>Exit Condition</b>	The user has successfully booked a visit
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user selects a day which does not have available time slots and the system warns him/her by sending the message “This day has not available time slots. Please select another one!” and giving him/her the possibility to select another day</li> <li>2. the user gives an estimation of the duration of the visit which is longer than the remaining time before the closing of the shop. The system warns him/her with a notification and does not store the provided value of the duration (but the user can anyway book a visit).</li> <li>3. the user selects a slot time which overlaps with another booked visit or with a taken ticket and the system warns him/her with “Your visit is overlapping with another reservation!”</li> <li>4. the internet connection suddenly drops, and the system sends the “The connection is lost” message.</li> </ol>

- The user checks the status of the queue with the remaining waiting time and the number of people before his/her turn

<b>Name</b>	Check ticket's status
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has logged in, the application is open in the home page
	<ol style="list-style-type: none"> <li>1. the user clicks on the “Menu” button</li> </ol>

<b>Event Flow</b>	<ol style="list-style-type: none"> <li>the user clicks on the “Check Status” button</li> <li>the system shows the estimated remaining waiting time, the number of people above him/her in the queue, and the “Cancel Ticket ” button</li> </ol>
<b>Exit Condition</b>	The user visualizes a page which contains all the information about his/her taken ticket
<b>Exception</b>	<ol style="list-style-type: none"> <li>the user has not taken a ticket and the system shows a screen with “Take a ticket and then come back here to see all the related information!” message</li> <li>the internet connection suddenly drops and the system sends the “the connection is lost” message.</li> </ol>

- **The user downloads the QR code associated to the ticket he/she has taken**

<b>Name</b>	Download QR code
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has logged in, the application is open in the home page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>the user clicks on the “Menu” button</li> <li>the user clicks on the “My QR codes” button</li> <li>the system shows the list of his/her QR code(s)</li> <li>the user clicks on “Download QR code” button associated to the QR code he/she want to save in his/her device</li> </ol>
<b>Exit Condition</b>	The user has the QR code(s) saved in his/her device
<b>Exception</b>	<ol style="list-style-type: none"> <li>the user has neither taken a ticket nor booked a visit and the system shows a screen with “Take a ticket and/or book a visit and then come back here to download the associated QR code(s)!” message</li> <li>the internet connection suddenly drops, and the system sends the “the connection is lost” message.</li> </ol>

- **The user cancels the ticket he/she has already taken**

<b>Name</b>	Cancel ticket
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user is in the “Check ticket’s status” page



<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user clicks on the “Cancel Ticket” button</li> <li>2. the user clicks “Yes” to the “Do you want to confirm the cancellation of the ticket?” request</li> <li>3. the system removes the reservation from the schedule and from the “Check Status” page</li> </ol>
<b>Exit Condition</b>	The ticket is no longer present in the “Check Status” page and in the schedule
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the internet connection suddenly drops, and the system sends the “the connection is lost” message.</li> </ol>

- **The user checks the visits he/she has already booked**

<b>Name</b>	Check booked visit(s)
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user has logged in, the application is open in the home page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user clicks on the “Menu” button</li> <li>2. the user clicks on the “My booked visits” button</li> <li>3. the system shows the list of his/her booked visit(s)</li> <li>4. the user selects one booked visit</li> <li>5. the system shows the Name of the Shop (and the related information), the Date, slot time, the duration of the visit and the category of items he/she wants to buy (if provided) and the “Cancel Visit” button.</li> </ol>
<b>Exit Condition</b>	The user knows all the information about his/her booked visit(s)
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user has not booked any visit and the system shows a screen with “Book a visit and then come back here to see all the related information!” message.</li> <li>2. the internet connection suddenly drops, and the system sends the “the connection is lost” message.</li> </ol>

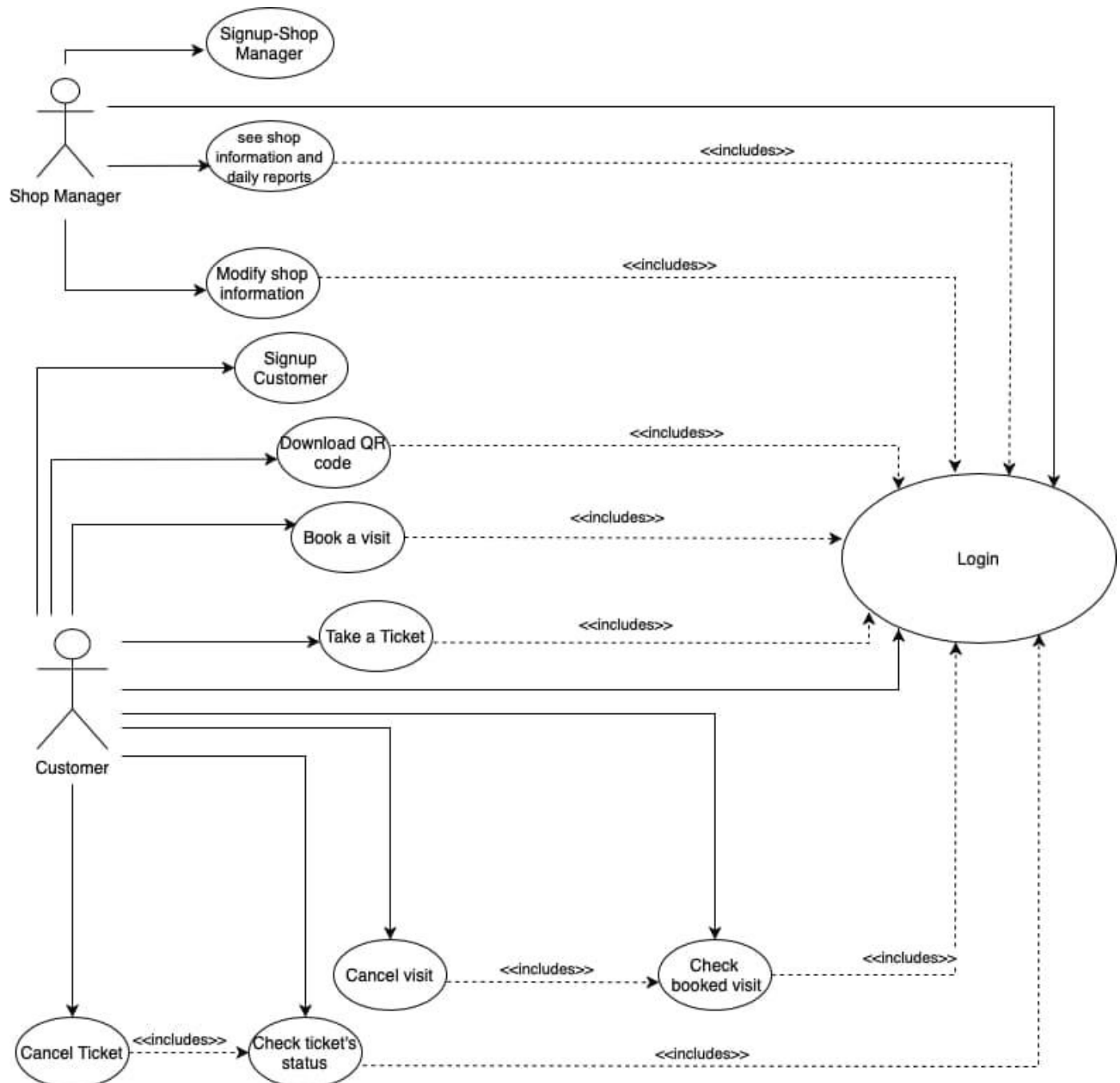
- The user cancels a visit he/she has already booked

<b>Name</b>	Cancel Visit
<b>Actor</b>	User-customer
<b>Entry Condition</b>	The user is in the “Check booked visits” page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. the user selects the visit he wants to cancel</li> <li>2. the user clicks on the “Cancel Visit” button</li> <li>3. the user clicks “Yes” to the “Do you want to confirm the cancellation of the ticket?” request</li> <li>4. the system removes the reservation from the schedule and from “Check booked visit(s)” page</li> </ol>
<b>Exit Condition</b>	The visit is no longer in the schedule and in the “Check booked visit(s)” page
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. The internet connection suddenly drops and the system sends the “the connection is lost” message.</li> <li>2. the user clicks on “Cancel Visit” button less than one hour before the time of the visit. The system warns him/her by sending “You can cancel a visit until one hour before the starting time of the visit!”</li> </ol>

- The shop manager wants to modify the information of his/her shop

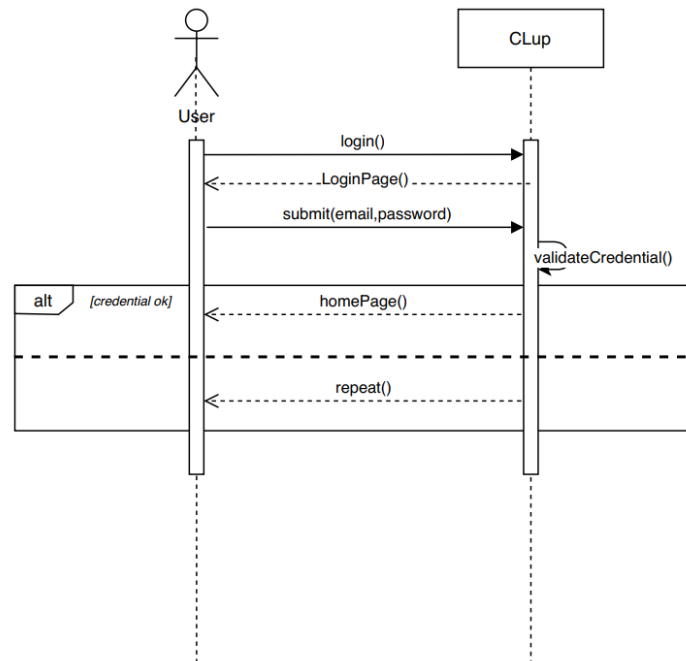
<b>Name</b>	Modify shop’s information
<b>Actor</b>	User-shop manager
<b>Entry Condition</b>	The user has logged in, the application is open in the home page
<b>Event Flow</b>	<ol style="list-style-type: none"> <li>1. The user clicks on the “Menu” button</li> <li>2. the user clicks on the “About my Shops” button</li> <li>3. the user clicks on the “Edit” button corresponding to the shop he/she wants to modify</li> <li>4. the user modifies some/all the information about his/her shop</li> <li>5. the user clicks on “Confirm” button</li> <li>6. the system saves the updated data</li> </ol>
<b>Exit Condition</b>	The information of the shop is modified
<b>Exception</b>	<ol style="list-style-type: none"> <li>1. the user provides a nonexistent address and the system asks him/her to write another one.</li> <li>2. the internet connection suddenly drops and the system sends the “the connection is lost” message</li> </ol>

## 3.2.3.2 Use Case Diagram

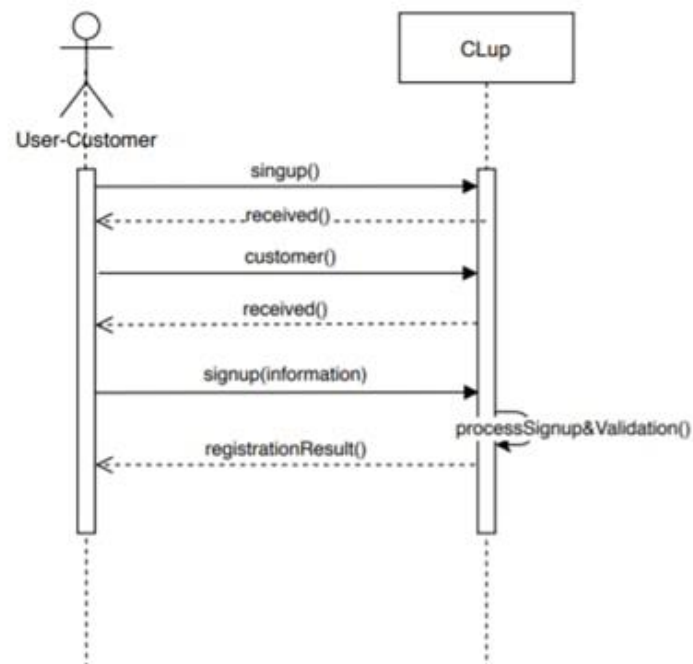


### 3.2.4 Sequence Diagrams

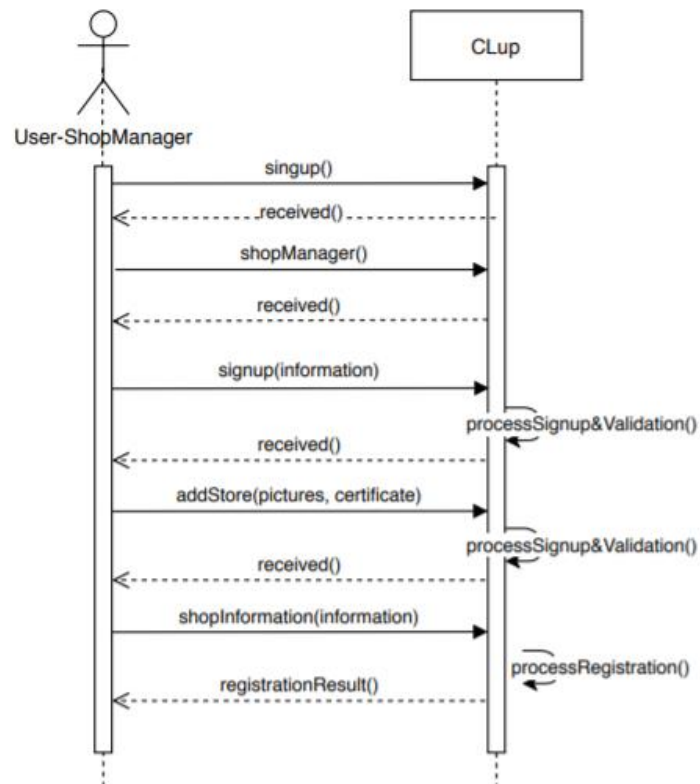
#### 1. Login



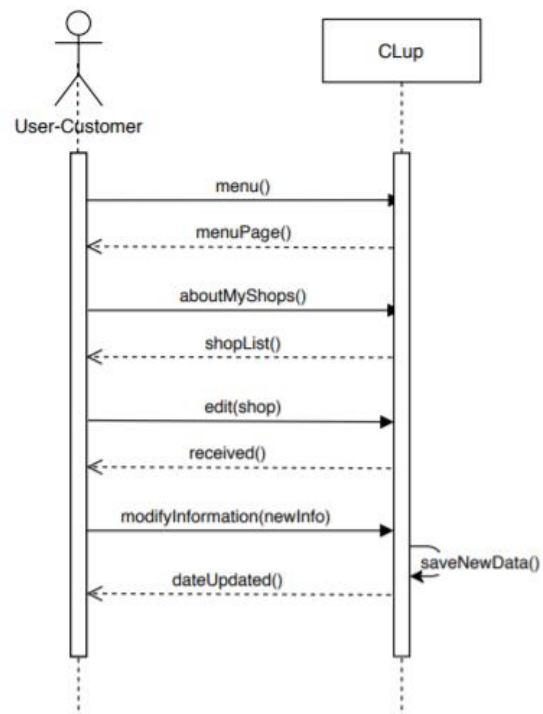
#### 2. Signup Customer



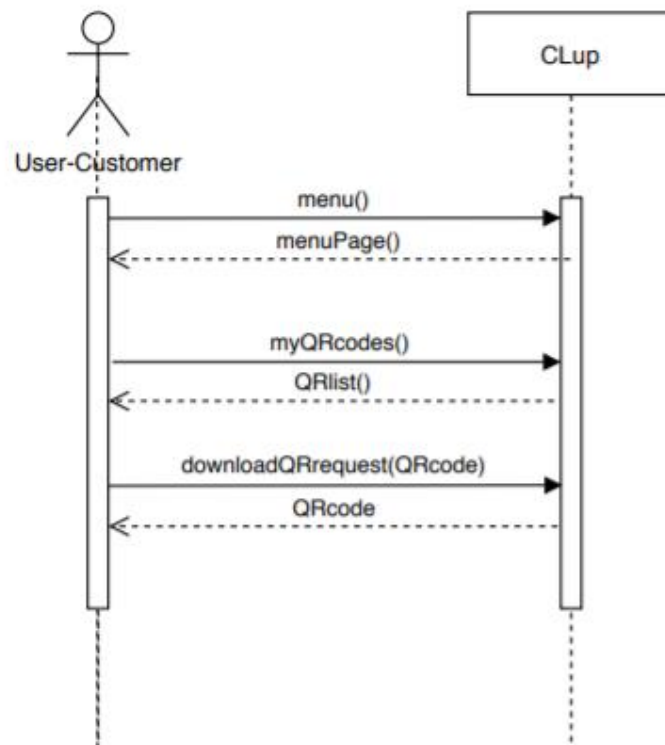
### 3. Signup Shop Manager



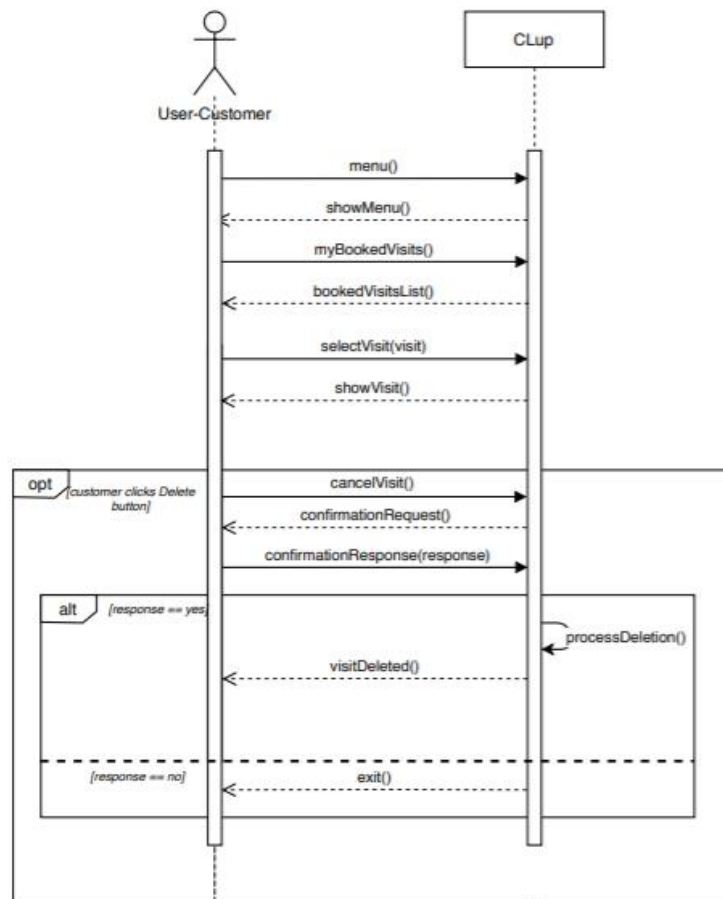
### 4. Modify Shop's Information



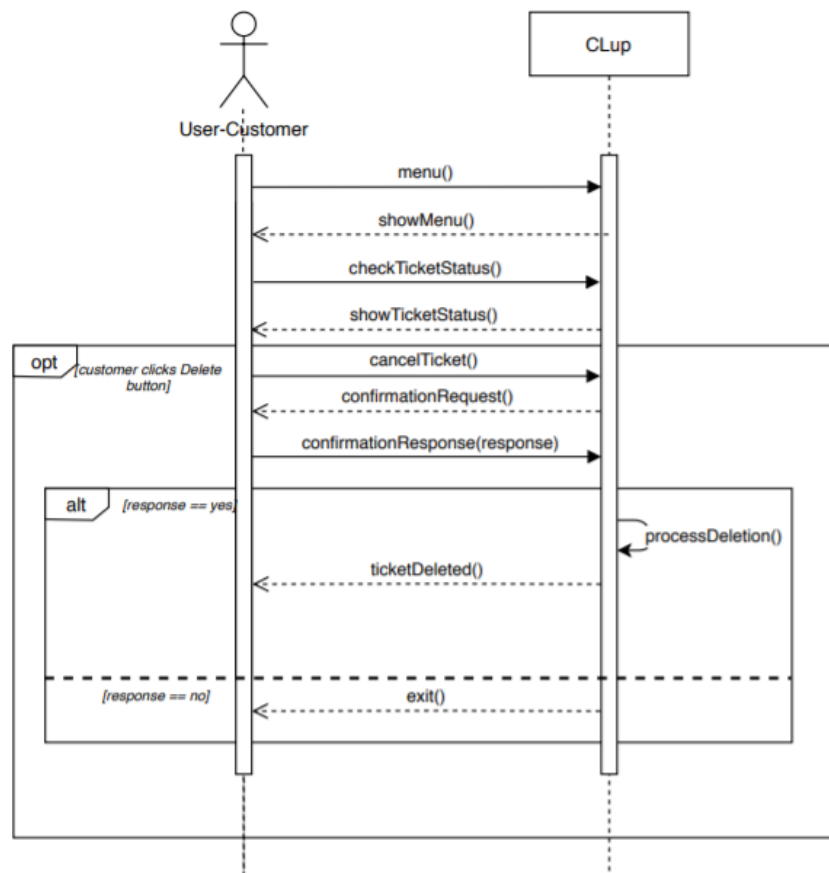
## 5. Download QR code



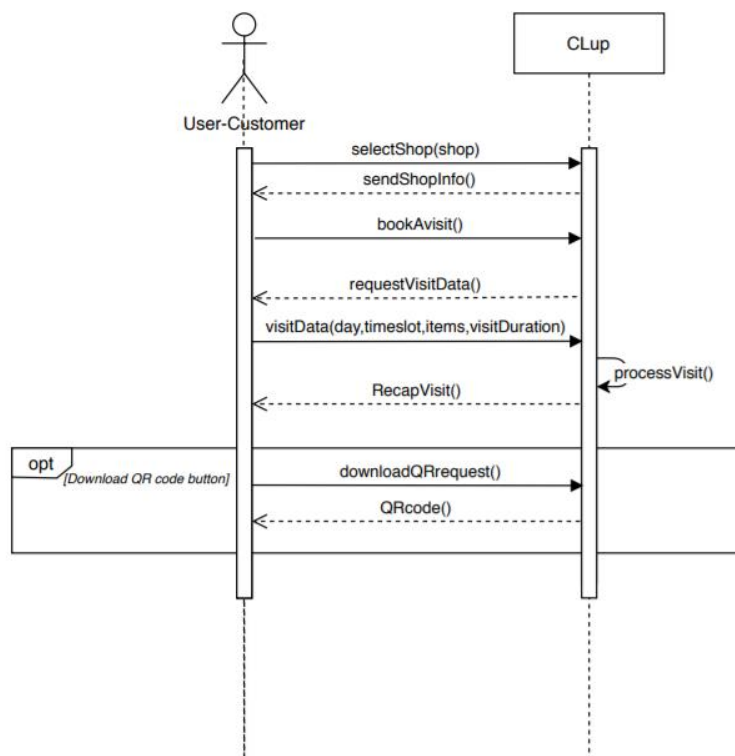
## 6. Check booked visit(s) and Cancel Visit option



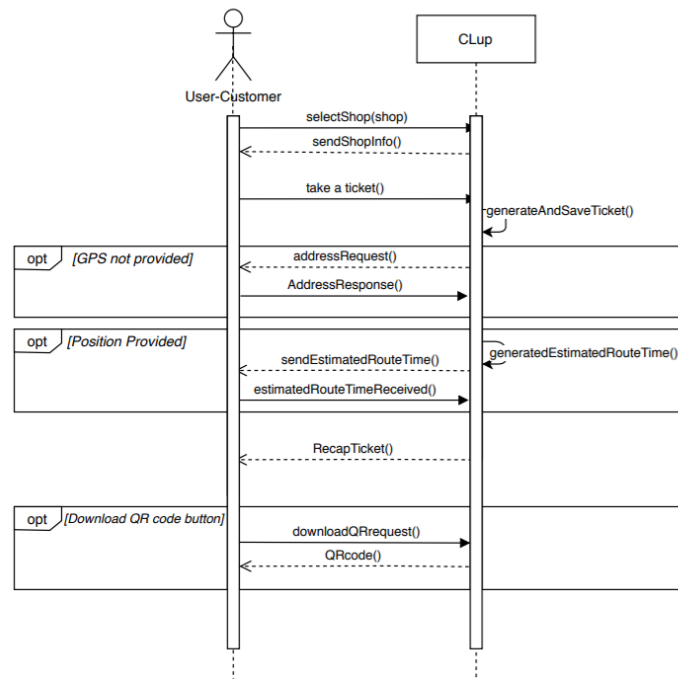
## 7. Check Status of the ticket and Cancel Ticket option



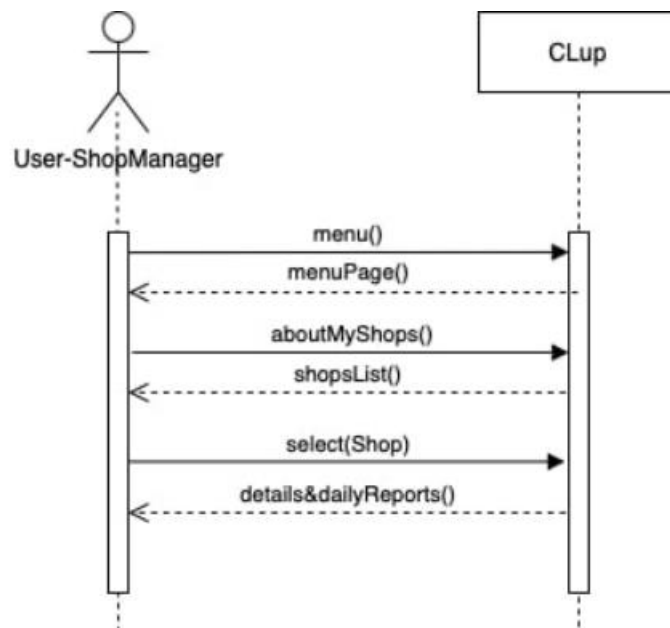
## 8. Book a visit



## 9. Take a ticket



## 10. See shop information and daily report





### 3.2.5 Scenarios

#### SCENARIO 1: Download CLup Application (general)

Simona is a young girl who visits her grandmother every weekend to give her help in the house and to make shopping for her. Due to the pandemic Simona is worried about going shopping and getting in touch with people, because in this way she could be affected by the virus and attach it to her grandma. For this reason, she decides to download CLup application, which allows her, thanks to the services provided, to keep the necessary distances from other people.

#### SCENARIO 2: User takes a ticket

Laura is a very successful businesswoman who works full time in a big company of cosmetics. She has two little sons, and she has to make lunch for them. On Tuesday she has a meeting which ends later than expected and she receives a call from one of her sons, who asks her to come back home because he and his brother are hungry. She has to buy food for lunch, so Laura decides to open CLup Application and see, in the list of shops provided, the closest shop and the number of people which are in the queue at that moment. She selects the more convenient shop and clicks on the "Take a ticket" button. Since she had provided her GPS localization once registered, she receives the time needed to reach the shop. So, she visualizes the QR code generated by the system, she downloads it and goes to the shop.

#### SCENARIO 3: User books a visit

Maria is a young student who is waiting for September, when the new iPhone is launched. She wants to buy it and so she decides to go to the Apple store with her father on Sunday, when they are both free. For this reason, Maria enters the CLup Application, logs in (she had already registered herself) and selects the Apple store in Piazza Liberty. She chooses the "Book a visit" option and selects from the calendar the date (Saturday 15 September) and the time slot 16.00-17.00, the last available one. She decides not to fill the optional field "Duration of the visit" but she adds information about the "Category of items" writing "mobile phone" in the appropriate field. She confirms the operation, and she receives the associated QR code with the recap of the just booked visit.

#### SCENARIO 4: User takes a ticket on the spot

Silvano is an old man who wants to go to the supermarket. During the Coronavirus emergency he is obliged to take a ticket to avoid people queuing in front of stores, which creates crowds. But unfortunately, Silvano does not have a device with internet connection. For this purpose, CLup Application gives the possibility to take the ticket on the spot. In this way Silvano can directly go to the supermarket. When he arrives at the supermarket, he finds the Totem where he can take the ticket. Once received the ticket with the associated QR code, he scans it at the entrance and starts shopping.

#### SCENARIO 5: The internet connection drops while taking a ticket

Giacomo is an off-site student. He is from Marche region, but he studies in Rome, at Sapienza University. Every day he goes to the campus and he takes with him the lunch, previously prepared at home. One day he is late and he does not have time to prepare it. So, at 13, during the lunch break, he wants to go to the supermarket to get something ready. He opens CLup application, he checks the list of the supermarkets and he selects the closest one. He is about to click the "Take a ticket" button but the internet connection suddenly drops. He immediately receives the message "The connection is lost" from the system. He enters the campus, and he tries to connect his device to the wifi of the University.

**SCENARIO 6: User cancels the reservation of a visit**

Luca is always busy since he works a lot during the week. He is the father of a young girl, Jessica, and every weekend he goes shopping with her in the center of Milan. Jessica wants to go to Walt Disney store, and so Luca decides to use CLup to book a visit for that Saturday. So he opens the application already installed, he registers himself, he logs in and after selecting the store placed in Corso Vittorio Emanuele, he books a visit. Unfortunately, on Saturday morning Jessica feels sick: she has a high temperature. For this reason Luca opens CLup, logs in and enters in the “Check booked visits” page. He chooses the visit programmed for Saturday and he clicks on the “Cancel Visit” button. The system asks him if he wants to confirm the operation and Luca clicks on “yes”. The reservation is canceled.

**SCENARIO 7: User adds a shop**

Niccolò is the owner of a small shop of house products in Udine (MI): “Wash your hands”. He wants to keep up with the times and so during Coronavirus Emergency he downloads from the Appstore CLup application to add his shop to the list of the available ones. He opens the application and he clicks on “sign up” button. He can select between the two options “Customer” and “Shop Manager” and so he clicks on “Shop Manager” button. He starts filling the mandatory fields but he does not see one of them and so when he wants to go on, the system reminds him to insert the missing information. Then he uploads the certificate, he fills the mandatory fields about his store, he uploads a shop’s picture he had taken the previous day and finally he clicks on “End Registration” button.

**SCENARIO 8: user checks the status of the ticket**

Lola is an old woman who is very skilled with technology. She has already taken the ticket for the shoes’ shop she wants to go to. She would like to know how many people are above her in the queue and the time she has to wait before her turns arrive. Consequently, she opens CLup Application and clicks on the “Menu” button and then on the “Check Status” option, where the user can check the status of the queue. She sees that she has to wait 10 minutes and that there are 3 people above her. Since she takes more or less 9 minutes to reach the shop, she takes her jacket and she leaves home.

### 3.3 Performance requirements – Non Functional Requirements

The system should be able to simultaneously serve 80000 individuals and to manage 200 shops. This is the start idea, but the system should be flexible to changes and to the number of requests of registration (both from the customers and from the shop manager users).

The response time for any action must be less or equal to 1 second. In addition, the system must generate the QR code within 3 seconds from the generation of the ticket or the “book a visit” request. The system should be able to run the algorithms implemented to manage the data in the schedule within two seconds from the action of the user.

### 3.4 Design Constraints – Non Functional Requirements

#### 3.4.1 Standards compliance

- The application is able to preserve the state of the system avoiding the accidental loss of data: the user should be able to have his/her state always preserved.
- According to the privacy of data the application processes sensitive information and so the project is subject to the GDPR (General Data Protection Regulation). These private data are stored in the internal storage of the application.
- To exploit all the services of the application absolute minimum permissions are requested.

#### 3.4.2 hardware limitations

All the shops must have totems and QR scanners in their buildings to manage the requests of tickets and the entrances/exits of people from/to the shops.

Both the user-customer (who decides not to take the ticket on the spot) and the user-shop manager must own a device to exploit the functionalities of the application. They also need a properly working internet connection. This must be available for all the possible actions that can be done with Clup Application.

Moreover, the device should have GPS sensors to provide the correct position of the user in case he/she decides to give the system access to it.

#### 3.4.3 Any other constraints

The application must respect all the privacy policies with relation to the user, according to the privacy Terms of Services and Privacy Policy: the personal information and the telephone number are not used for commercial scopes; the position, if provided, is used only to compute the estimation of the time necessary to reach the destination from the address of the user.

### 3.5 Software System Attributes

#### 3.5.1 Reliability

The system must be up 24 hours per day, every day without any interruption. In case of failure an error message must be displayed on the application within 15 minutes. To avoid data loss the system should periodically backup the data and the core services must be duplicated.

In this way the system guarantees high reliability.

### **3.5.2 Availability**

To make the system available full time and enable it to continue operating properly in the event of failure, it has a strong fault tolerance architecture.

Redundancy (switching to the duplicated system) must be ensured to make data available in case of breakdown of a portion of the system.

### **3.5.3 Security**

Since the users (both customers and shop managers) have to provide sensitive data, the system should protect it from any possible internal and external attack. For this purpose users' passwords and personal information must be encrypted to be protected during transmission. In this way data becomes useless if hacked from a server and the protection of privacy is guaranteed.

### **3.5.4 Maintainability**

The system must have a high level of maintainability. For this scope the code should be fully commented in order to better explain all the components present in the system.

Moreover the associated documentation should be extremely clear to make the project fully understandable by everyone.

Tests associated with the code must cover at least 80% of it. Specifically, automated tests should be used to make it easy to validate changes and integration tests must be continuous in order to build the code easily.

### **3.5.5 Portability**

The software must be implemented as a multiplatform application. It should support Android and iOS operating systems for mobile devices.

## 4. Formal Analysis Using Alloy

### 4.1 Alloy Code

```

sig Email{}
sig Picture{}
sig Items{}
sig Name{}
sig Day{}

--Represents the number of minutes spent until a specific moment
sig Time {
    minutes: one Int
}
--{ minutes >= 0 and minutes <= 1440 }
{ minutes >= 0 and minutes <= 6 }

--Represents a time slot between the starting minute and finishing one of a day
sig Slot {
    day: one Day,
    start: one Time,
    finish: one Time
}
{
    start.minutes < finish.minutes
}

--Represents the department of a shop with the related capacity and the categories of items that can be found inside
sig Department{
    departName: one Name,
    capacity: one Int,
    items: set Items,
}
{
    capacity > 0
}

--Represents the shop. Opening slots are the slot times in which the shop is open
sig Shop{
    id: one Int,
    shopManager: one ShopManager,
    picture: lone Picture,
    shopTickets: some Ticket,
    shopVisits: some Visit,
    departments: set Department,
    openingSlots: some Slot
}
{
    id > 0
}

--Represents the user Shop Manager who adds his/her shops (one or more)
sig ShopManager{
    id: one Int,
    shops: some Shop
}
{
    id > 0
}

--Represents the user Customer who exploit CLup services (take a ticket and book a visit)
sig Customer{
    id: one Int,
    email: one Email,
    customerTickets: some Ticket,
    customerVisits: some Visit
}
{
    id > 0
}

```

--Represents the ticket taken by a customer for a Shop

```
sig Ticket{
  id: one Int,
  ticketShop: one Shop,
  ticketCustomer: one Customer,
  ticketSlotTime: one Slot
```

```
{
  id > 0
}
```

--Represents the visit booked by a customer for a Shop. It can be also related to the various departments in which a visit takes place

```
sig Visit {
  id: one Int,
  visitShop: one Shop,
  visitCustomer: one Customer,
  visitDepartments: some Department,
  visitSlotTime: one Slot
}
```

-----FACTS-----

--the same customer cannot have a visit and a ticket overlapping

```
fact noTicketVisitCustomerOverlap{
  all c: Customer | all t: Ticket, v: Visit |( t in c.customerTickets and v in c.customerVisits
    and t.ticketSlotTime.day = v.visitSlotTime.day) implies
    (t.ticketSlotTime != v.visitSlotTime and t.ticketSlotTime.finish.minutes =< v.visitSlotTime.start.minutes
      or v.visitSlotTime.finish.minutes =< t.ticketSlotTime.start.minutes)
}
```

--the same customer cannot have two visits overlapping

```
fact noTwoVisitsCustomerOverlap{
  all c: Customer | all disj v1, v2: Visit |( v1 in c.customerVisits and v2 in c.customerVisits
    and v1.visitSlotTime.day = v2.visitSlotTime.day) implies
    (v1.visitSlotTime.finish.minutes =< v2.visitSlotTime.start.minutes
      or v2.visitSlotTime.finish.minutes =< v1.visitSlotTime.start.minutes)
}
```

--the same customer cannot have two tickets overlapping

```
fact noTwoTicketsCustomerOverlap{
  all c: Customer | all disj t1, t2: Ticket |( t1 in c.customerTickets and t2 in c.customerTickets
    and t1.ticketSlotTime.day = t2.ticketSlotTime.day) implies
    (t1.ticketSlotTime.finish.minutes =< t2.ticketSlotTime.start.minutes
      or t2.ticketSlotTime.finish.minutes =< t1.ticketSlotTime.start.minutes)
}
```

--a ticket must be used in a slot time in which the associated shop is open

```
fact ticketOpeningShopOverlap {
  all t: Ticket | some sl: Slot | (sl in t.ticketShop.openingSlots and sl != t.ticketSlotTime and sl.day = t.ticketSlotTime.day
    and sl.start.minutes =< t.ticketSlotTime.start.minutes
    and sl.finish.minutes >= t.ticketSlotTime.finish.minutes)
```



--a visit must be done in a slot time in which the associated shop is open

```
fact visitOpeningShopOverlap {
    all v: Visit | some sl: Slot | (sl in v.visitShop.openingSlots and sl != v.visitSlotTime and sl.day = v.visitSlotTime.day
                                and sl.start.minutes =< v.visitSlotTime.start.minutes
                                and sl.finish.minutes >= v.visitSlotTime.finish.minutes)
}
```

--a department in the departments' list of a visit must also be in the departments' list of the associated shop

```
fact isDepPresentInShop{
    all v: Visit | all d: Department | d in v.visitDepartments implies d in v.visitShop.departments
}
```

--a store cannot exist without the shop manager who registers it

```
fact shopShopManagerRelationship {
    all sm: ShopManager, s: Shop | s in sm.shops <=> sm = s.shopManager
}
```

--a ticket must be present in the tickets' list of its shop

```
fact ticketShop{
    all t: Ticket, s: Shop | t in s.shopTickets <=> t.ticketShop = s
}
```

--a visit must be present in the visits' list of its shop

```
fact visitShop{
    all v: Visit, s: Shop | v in s.shopVisits <=> v.visitShop = s
}
```

--a shop must be present in the shops' list of its shop manager

```
fact shopShopManager{
    all s: Shop, sm: ShopManager | s in sm.shops <=> s.shopManager = sm
}
```

--a ticket must be present in the tickets' list of its customer

```
fact ticketCustomer {
    all t: Ticket, c: Customer | t in c.customerTickets <=> t.ticketCustomer = c
}
```

--a visit must be present in the visits' list of its customer

```
fact visitCustomer {
    all v: Visit, c: Customer | v in c.customerVisits <=> v.visitCustomer = c
}
```

--the same department cannot be present in two different shops

```
fact oneShopForADepartment{
    no disj s1, s2: Shop | one d: Department | d in s1.departments and d in s2.departments
}
```

--a department does not exist without a shop that contains it

```
fact depForshop{
    all d: Department | one s: Shop | d in s.departments
}
```

--the same picture cannot be associated to two different shops

```
fact oneShopForAPicture{
    no disj s1,s2: Shop | one p: Picture | p in s1.picture and p in s2.picture
}
```

--an email cannot exist without an associated customer

```
fact emailCustomer{
    all e: Email | one c: Customer | c.email = e
}
```

```

--an item cannot exist without an associated department
fact itemsDepartment {
    all i: Items | one d: Department | i in d.items
}

--a picture cannot exist without an associated shop
fact pictureShop{
    all p: Picture | one s: Shop | s.picture = p
}

--a name cannot exist without the associated department
fact departmentName{
    all n: Name | one d: Department | d.departName = n
}

--an openingSlot cannot exists without the associated Shop or Ticket or Visit
fact slotShopTicketVisit{
    all os: Slot | (one s: Shop | os in s.openingSlots) or (one t: Ticket | os in t.ticketSlotTime) or (one v: Visit | os in v.visitSlotTime)
}

--a day cannot exists without the associated opening slot
fact daySlot {
    all d: Day | some s: Slot | d = s.day
}

--a time cannot exists without the associated slot time
fact timeSlotTime {
    all t: Time | some s: Slot | (s.start = t or s.finish = t)
}

```

```

--UNIQUE ID

fact uniqueDepartmentName{
    no disj d1,d2 : Department | d1.departName = d2.departName
}

fact uniqueShopId {
    no disj s1, s2 : Shop | s1.id = s2.id
}

fact uniqueTicketId {
    no disj t1, t2 : Ticket | t1.id = t2.id
}

fact uniqueShopManagerId {
    no disj s1, s2 : ShopManager | s1.id = s2.id
}

fact uniqueCustomerId {
    no disj s1, s2 : Customer | s1.id = s2.id
}

```



```

fact uniqueEmailCustomer{
  no disj c1,c2 : Customer | c1.email = c2.email
}

fact uniqueVisitId{
  no disj v1, v2 : Visit | v1.id = v2.id
}

-----PREDICATES-----

pred createATicket [t: Ticket, i: Int, s: Shop, c: Customer, sl: Slot]{
  t.id = i
  t.ticketShop = s
  t.ticketCustomer = c
  t.ticketSlotTime = sl
}

--run createATicket

pred createAVisit [v: Visit, i: Int, s: Shop, c: Customer, d: Department, d1: Department, sl: Slot] {
  d!=d1
  v.id = i
  v.visitShop = s
  v.visitCustomer= c
  d in v.visitDepartments
  d1 in v.visitDepartments
  v.visitSlotTime = sl
}
--run createAVisit

pred customersOfTheSameShop{
  #Shop = 1
  #Customer = 2
}

--run customersOfTheSameShop

pred customersWithTicketAndVisit {
  #Customer = 1
  #Visit = 1
  #Ticket = 1
}

run customersWithTicketAndVisit

-----ASSERTIONS-----

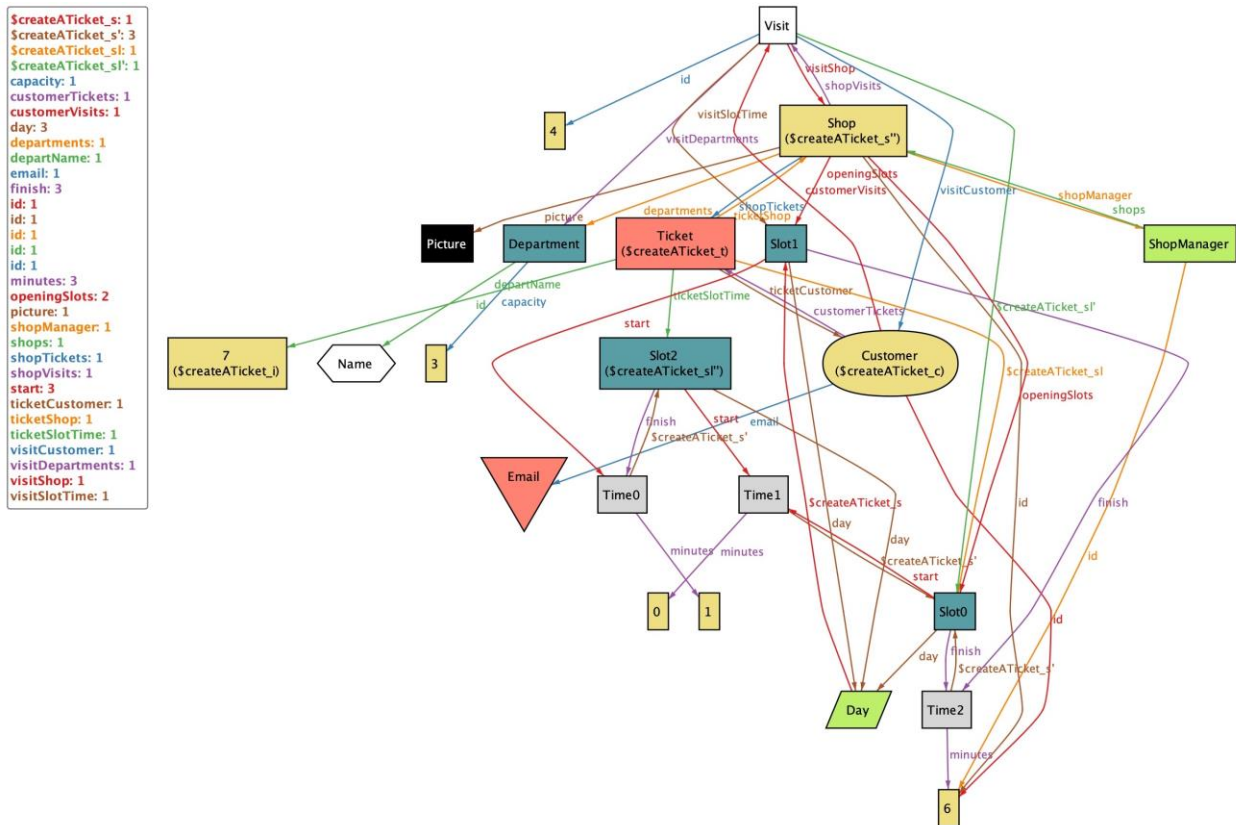
--the day of the ticket the shop is opened
assert checkTicket {
all t: Ticket | some sl: Slot | sl in t.ticketShop.openingSlots and sl.day = t.ticketSlotTime.day
}

--the day of the visit the shop is opened
assert checkVisit {
all v: Visit | some sl: Slot | sl in v.visitShop.openingSlots and sl.day = v.visitSlotTime.day
}

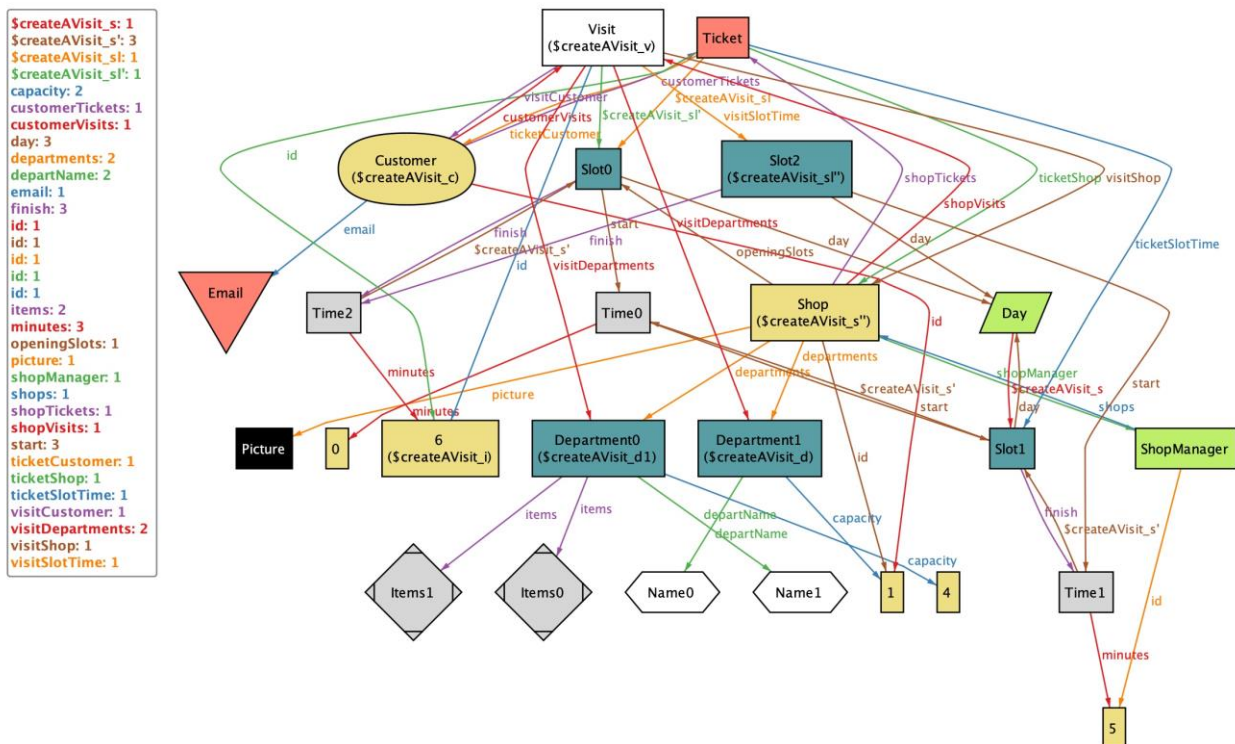
check checkTicket for 3
check checkVisit for 4

```

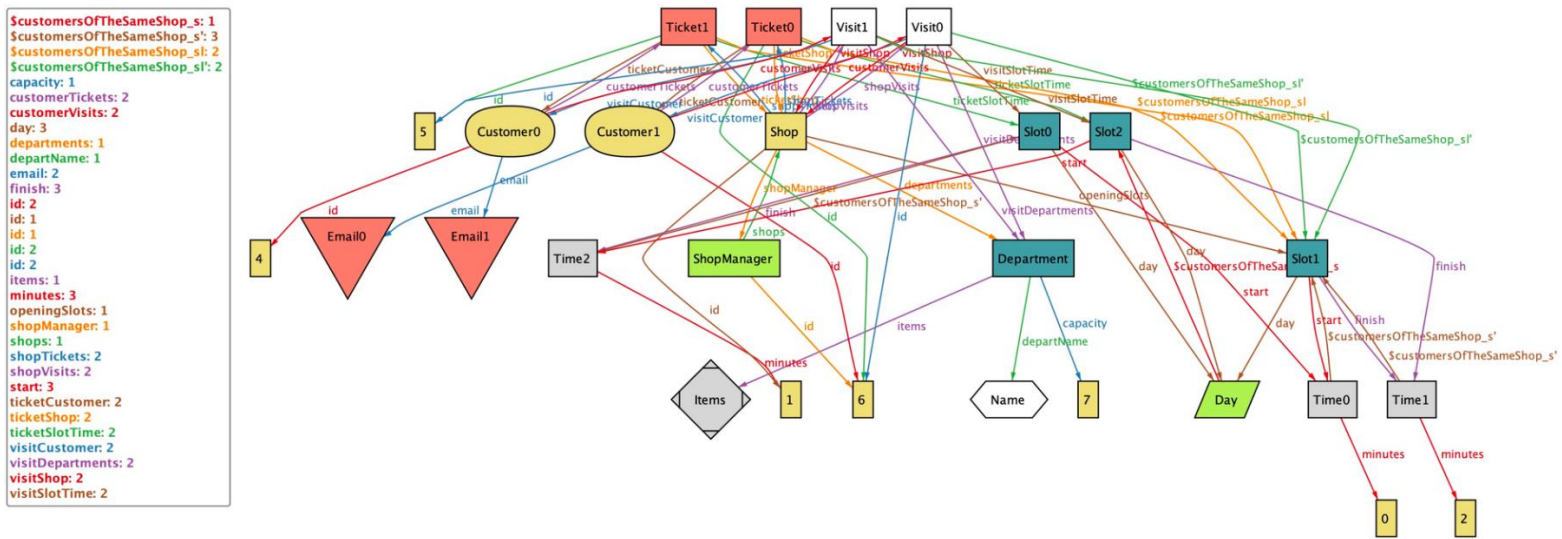
## 4.2 MetaModel



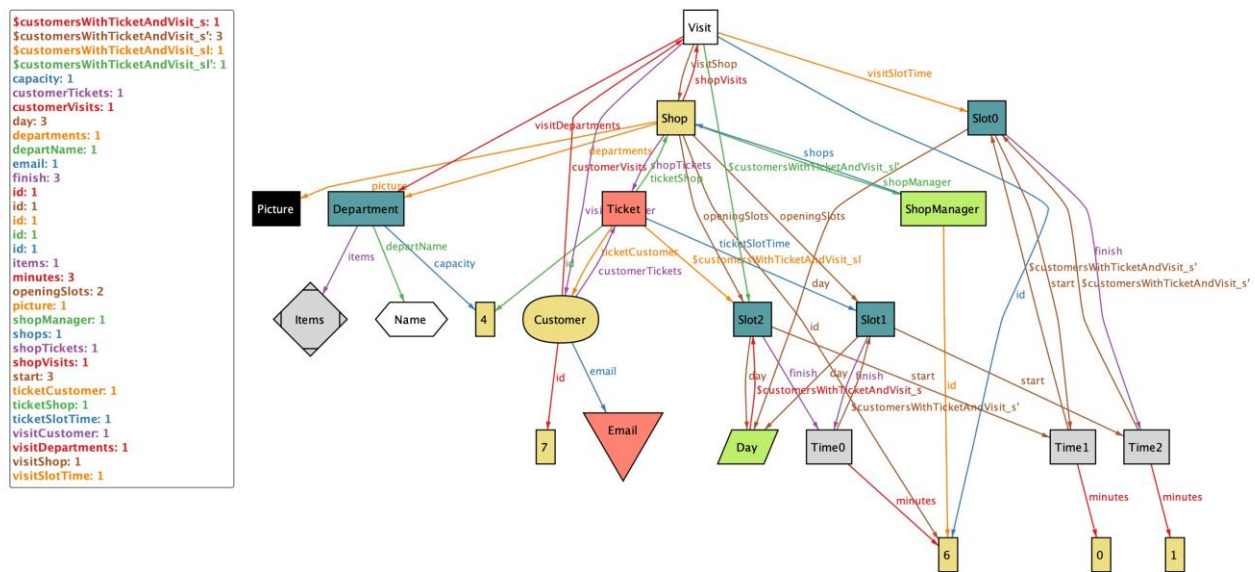
```
run createATicket
```



run createAVisit



run customersOfTheSameShop



run customersWithTicketAndVisit

### 4.3 Results of Assertions

#### Executing "Check checkTicket for 3"

Solver=sat4j Bitwidth=4 MaxSeq=3 SkolemDepth=1 Symmetry=20  
 16111 vars. 612 primary vars. 45878 clauses. 32ms.  
 No counterexample found. Assertion may be valid. 10ms.

#### Executing "Check checkVisit for 4"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
 25026 vars. 920 primary vars. 69713 clauses. 61ms.  
 No counterexample found. Assertion may be valid. 34ms.

### 4.4 Results of Predicates

#### Executing "Run createATicket"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
 16277 vars. 637 primary vars. 46275 clauses. 39ms.  
**Instance** found. Predicate is consistent. 111ms.

#### Executing "Run createAVisit"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
 16349 vars. 643 primary vars. 46391 clauses. 35ms.  
**Instance** found. Predicate is consistent. 60ms.

#### Executing "Run customersOfTheSameShop"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
 16020 vars. 609 primary vars. 45720 clauses. 33ms.  
**Instance** found. Predicate is consistent. 90ms.

#### Executing "Run customersWithTicketAndVisit"

Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20  
 16027 vars. 609 primary vars. 45743 clauses. 30ms.  
**Instance** found. Predicate is consistent. 164ms.



## 5. Effort spent

We have done about 80% of the work together in videocall, especially for the decision making. We have divided the remaining part equally.

We have also used a Google Doc to fix some aspects autonomously and in parallel.

Reading of the document	1h
Preliminary discussion	3h
Final reading	3h

Topics	Hours
Purpose and Scope	Member1: 2h Member2: 2h
World phenomena, shared phenomena, goals	Member 1: 2h 30min Member 2: 2h
Definitions, Acronyms, Abbreviations, Reference Documents, Document structure	Member 1: 1h 30min Member 2: 1h

UML description	Member 1: 4h Member 2: 4h 30min
State Charts	Member1: 2h Member2: 1h 30min
Product functions, User Characteristic	Member1: 2h 30min Member2: 2h
Domain Assumptions	Member1: 1h Member2: 1h

User Interfaces	Member 1: 1h Member2: 2h
Hardware, Software, Communication Interfaces	Member 1: 2h Member2: 2h
Functional Requirements and mapping	Member 1: 4h 30 min Member2: 4h
Mapping	Member1: 1h 30min Member2: 1h
Use Cases	Member1: 3h Member2: 2h

Use Case Diagram	Member1:1h Member2:1h
Sequence Diagrams	Member1: 2h Member2: 3h
Scenarios	Member1: 1h Member2: 1h
Performance Requirements, Design Constraints, Software System Attributes	Member1: 2h Member2: 2h
Alloy code, Metamodel	Member1: 3h Member2: 4h

Member 1: Alessia De Berardinis

Member 2: Arianna Dragoni