Politecnico di Milano

School of Industrial and Information Engineering

SOFTWARE ENGINEERING 2 PROJECT

Requirement Analysis and Specification Document (RASD)

Customer Line-up

Version 1.0

Authors
Hamza Haddaoui
Giuseppe Piccirillo
Alessandro Restifo

Supervisor
Dr. Elisabetta di Nitto

December 23, 2020

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

1.1.1 General Purpose

The purpose of this product is to help preventing crowds into stores and supermarkets, by planning each customer's visit. Before the coronavirus pandemic outbreak, crowds were not a problem inside shops: customers could go to buy goods at any time, without having to worry. COVID-19 has undeniably changed the way most people live their day-to-day lives. Controlling distancing between people and ensuring observance of the hygienic guidelines are both essential parts of managing the impact of the disease.

CLup is born for this purpose: helping managers to regulate the admissions to stores, preventing people to queue up outside and therefore lowering the risk of infection. When a customer has the necessity to buy something, all she/he has to do is to select the preferred store from the application, and then get a ticket. This ticket is nothing more than the digital counterpart of the situation where people line up for a service, but with more advantages. Furthermore, the product will have an additional function: we want customer to have the opportunity to book a visit in another day or time, without having to queue up. Finally, CLup can also build up statistics, collecting some data for a better service: for example, the best forecast for new customers can be predicted by checking the average permanence of a customer.

1.1.2 Goals

[G1] A user can book a ticket

[G2] A user can book a visit

[G3] The app can alert users when it's time to start going to shop, according to their GPS position

[G4] customers who don't have access to a smartphone, can get ticket on the spot

[G5] User can get suggestions by the application about different time slots and the duration of a visit based on previous data

[G6] The users validate the booking by scanning their QR code

[G7] Shop managers can visualize statistics about the booking requests

1.1.3 Table functions-goals

In the table shown below, the stakeholders needs are reported in three different sections (basic, advanced 1, advanced 2) in relation to the goals. The basic service is relative to the possibility to obtain tickets from the application, while the advanced function 1 is relative to the visit booking service. The last function which is the advanced 2, is about suggestions and other tips that can be given to the user to enhance its experience.

	Goals						
	G1	G2	G3	G4	G5	G6	G7
Basic service	X			X		X	X
Advanced function 1		X				X	
Advanced function 2			X		X	X	

Table 1: Goal function mapping

1.2 Scope

The application comes in two versions: for citizens a mobile application is available on the major app stores, and for business owners, a web-based application. The main feature available for customers is to obtain a ticket or book a visit to access the preferred store. Store managers on the other hand, can monitor entrances and statistics. In case a user has no access to a mobile phone, the ticket can be booked using a device located at the shop. At the end of the booking, the device will print out a booking digest containing among other information, an estimated time at which the user must show up at the store.

1.2.1 World and shared phenomena

Here are listed the phenomena related to the "machine", which means the software-to-be with the required working hardware and the "world", which is the real environment affected by the "machine". A phenomena can be shared by both machine and world if it's controlled by the world and observed by the machine, or controlled by the machine and observed by the world.

Phenomenon	**Shared**	**Who controls it**
A citizen wants to go buying in a shop	N	W
Grocery's and supermarkets want to avoid queues and gatherings in their markets because of covid	N	W
The store managers monitors and regulates the entrances	Y	W
A citizen retrieves a ticket or books a visit	Y	W
A citizen creates an account	Y	W
QR code of the book is scanned and validated	Y	W
Users see the available time slots of a store	Y	W
The user specifies the time required for the visit and the products to buy	Y	W
The app computes the duration of a visit if not specified and if enough data is available	N	M
The system checks if a shop is going to be too much crowded to suggest other time slot or even similar shop	N	M
The book is saved on the system	N	M
The app generates QRs to be scanned at the entrance for monitoring purposes	N	M
The app computes the "time to leave" to the chosen market	N	M

Table 2: World and Machine Table

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

- User: a citizen registered to CLup service
- Guest: a citizen using CLup services without being registered
- Ticket: reservation of a place on a virtual queue
- Visit: reservation of a place on scheduled day and time
- Timestamp: time and day at which a certain operation is performed
- Basic services: CLup functionality of booking a ticket, accessible both on the app and at the shop without registration
- Advanced services: CLup functionalities accessible only on app, upon registration
- Time to leave: estimated time at which the user should leave to get at shop at the right time

1.3.2 Acronyms

- GPS: Global Positioning System
- QR: Quick Response, refers to QR Code
- API: Application Programming Interface
- OS: Operative System
- TTL: Time To Leave

1.3.3 Abbreviations

- Gn: n-th goal
- Dn: n-th domain assumption
- Rn: n-th requirement
- UCn: n-th use case

1.4 Revision history

- Version 1.0: Initial release of the RASD

1.5 Reference documents

- Document: "R&DD Assignment A.Y. 2020-2021.pdf", Requirement Engineering and Design Project: goal, schedule, and rules
- ISO/IEC/IEEE 29148 dated Dec 2011, Systems and software engineering — Life cycle processes — Requirements engineering

1.6 Document structure

1. **Introduction** The first section identifies the product, the application domain, the goals related to the product. It discerns among world, shared and machine phenomena considering the application domain. It defines the terms used in this Specification Document. It describes the content and structure of the RASD.

2. **Overall Description** The second section describes external interfaces: system, user, hardware, software. It summaries the major product functions. It describes the user characteristics and the application constraints. It analyses the domain assumptions and the required dependencies.

3. **Specific Requirements** The third section takes a closer look to the aspects identified in section 2. This represents the body of the document, whose main points are requirements and constraints. The key topics covered are the definition of the external interfaces, the functional requirements of the application and the performance requirements. Moreover, this section provides a detailed view about the design constraints (from standards compliance to hardware limitations) and about the most important software system attributes.

4. **Formal Analysis Using Alloy** The fourth section presents a formal model of the application created through Alloy. The model allows a better visual understanding of the application domain thanks to the visual representation of the worlds generated from the code, while assuring the correctness of the conceptual model developed in the requirements document.

5. **Effort spent** The fifth section reports the details of the time spent by each member of the group on the various parts of the RASD.

6. **References** The sixth section reports the notes found in the RASD for a better explanation.

2 Overall Description

2.1 Product perspective

Description
CLup is a mobile application that tries to answer the need for distancing born during the coronavirus emergency. People can exit their houses only for essential needs and even grocery shopping can become a challenge in the presence of such strict rules. Supermarkets need to restrict access to their stores to avoid having crowds inside, which are a source of hazard. CLup allows its users to remotely book a ticket or a visit to any grocery or supermarket which supports CLup.

Users
The application is meant to be used by the widest stage of people possible, since its purpose is born and built on the basic needs of every citizen. Therefore, the average user is everyone who needs to fulfil her/his basic needs in the safest and most time-efficient way. Furthermore, the application also allows the store managers to better manage the accesses to the store itself, through the use of specific QR codes the user needs to scan upon every access to the store.

System, Software and Hardware
Since nowadays most of the people own a smartphone, the application will be released as a mobile application for the mobile OSs Android and iOS. In addition to that, the store managers will be able to access the data for analysis purposes through the use of a web based application. The smartphones shall be equipped with a mobile data network. An embedded GPS receiver is also necessary in order for some advanced features of the app to work as intended. For the web application to properly run, a PC with an active internet connection is necessary. Furthermore, shops should be equipped with a guest device able to print tickets right away, as a backup option.

2.1.1 Class Diagram

Here is represented a high-level view of the application, through a UML class diagram. Notice that it does not contain every element that the system needs. The most important classes needed are the Visit and Ticket Classes, which both inherit from the Booking abstract Class. Both Visit and Ticket share a few common attributes as the ID and the QR code that has to be scanned upon entrance to the store. While Visits can specify a particular time slot and a possible shopping list, Tickets need to take into account the time needed to get to the store. Bookings are created by Users, which retrieve Suggestions based on the data available to CLup (the previous duration of visits, the date and the available time slots). Every Booking is associated to a Store, which can retrieve and analyze the Statistics related to the affluence of the customers.

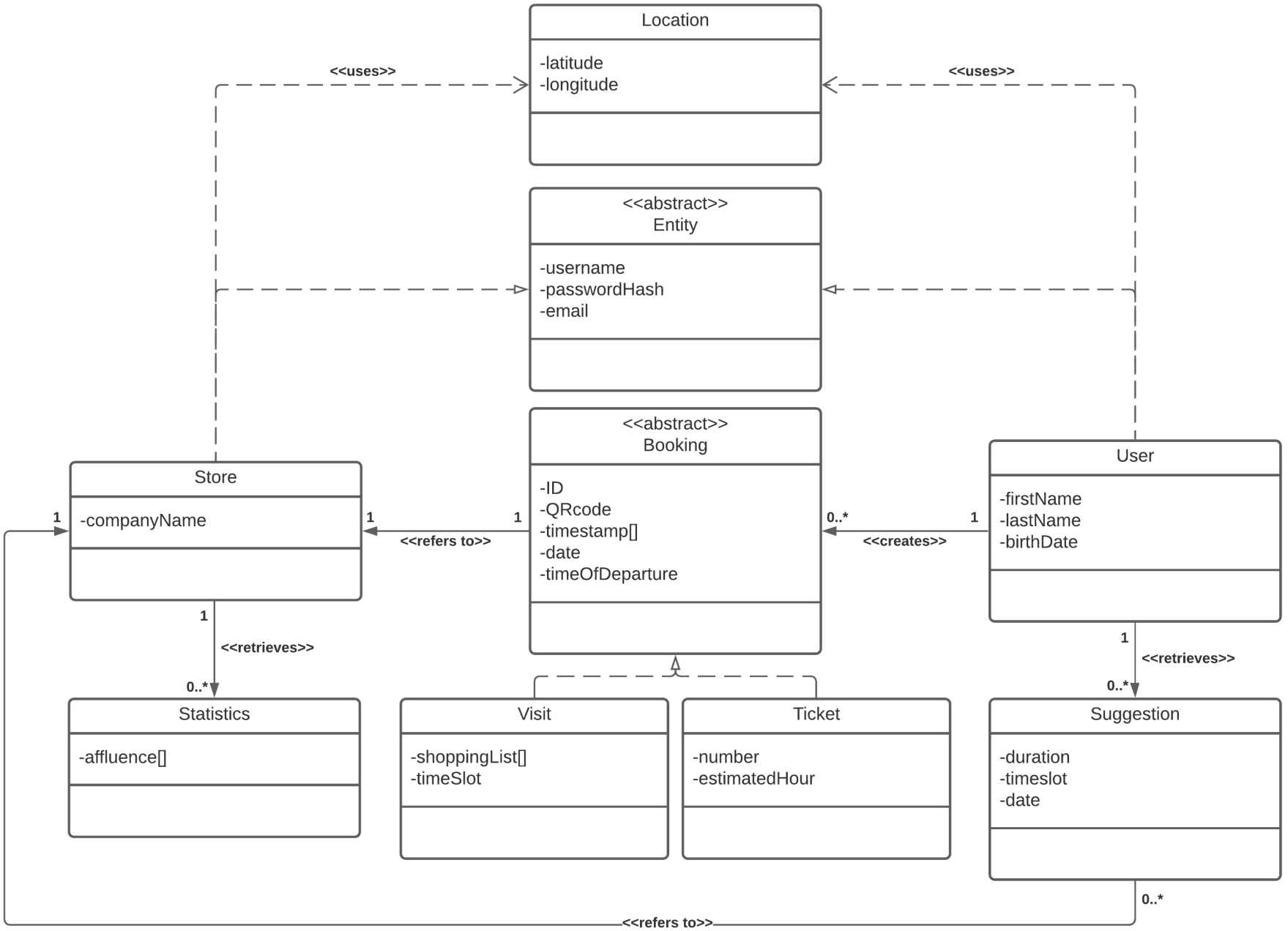

Figure 1: High-level Class Diagram

2.1.2 Flowchart

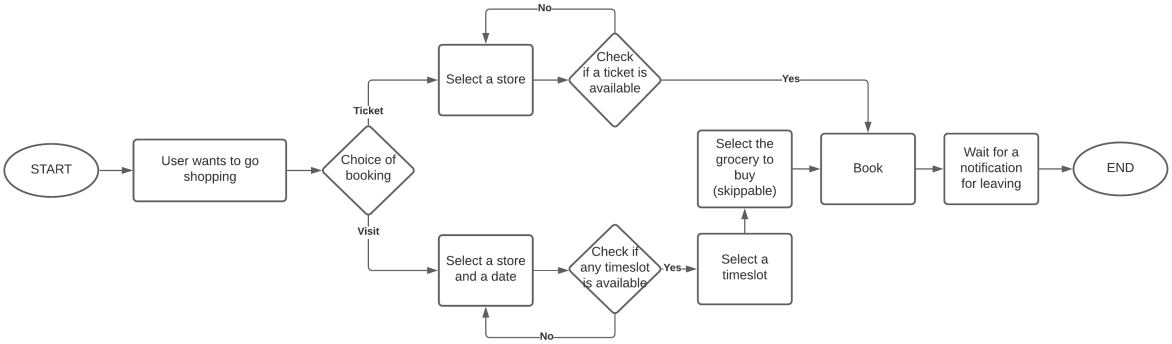

Figure 2: Flow Chart representing the Booking process

In order to understand the process of booking a ticket, here is a UML flowchart representing its key steps. The process starts with the decision of a customer equipped with CLup application to go shopping. The user has to decide whether to choose a ticket or a visit. If the user decides to retrieve a ticket, it is then prompted with the selection of the desired store. The system then checks if the store is available. If not, the user is prompted again with the decision of the store. When a store with enough free tickets is selected, the place is booked. The app then computes the required time to leave, and the user waits for the notifications to advise him to set out for the store. If the user decides instead to visit a store, a selection of the store and the date is necessary. The systems checks if any timeslot is available. If not, the user has to change store or date. If a timeslot is available the user has to select one of the prompted timeslots. Then, a selection of the grocery that has to be bought is necessary. Once these steps are completed, the visit can be correctly booked, and the user will wait for the notifications to remind her/him to leave.

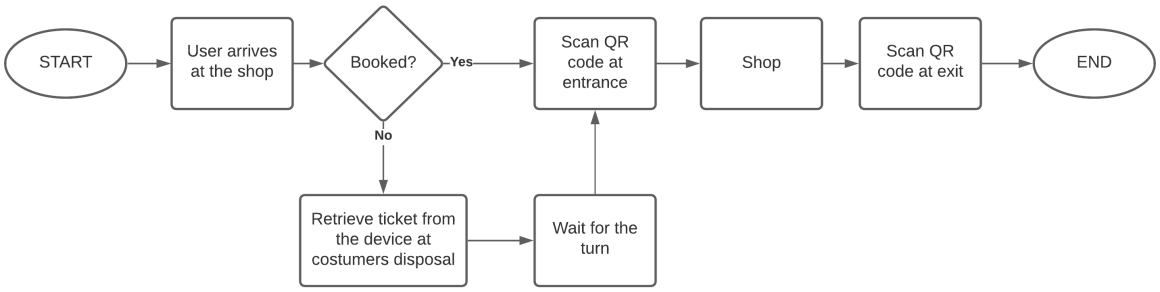

Figure 3: Flow Chart representing the QR code scanning process

This second flowchart represents the queuing mechanism in action, in a real life context. The process starts with a user arriving at a shop. The customer may or may not have the application CLup, therefore she/he may not have a ticket. If the customer has no previously booked ticket, there is the possibility to get one on an on-location device at customers disposal, at the entrance of the shop. As soon as the client gets one, she/he has to wait for the turn. As soon as the client's turn arrives, the QR code on the ticket can be scanned at the entrance of the shop through a special scanner, and the client has now access to the store. Upon exit, the client has

to scan the QR code on the ticket again. The power of this methodology relies in the fact that many customers are already used to scan a ticket in order to get out of a shop, if they shop at the automatic cash machine. Though CLup powerful infrastructure, this data (timestamps of entrance and exit among the others) will be used to estimate the duration of any visit to a store and correlate it to the type of grocery bought or the type of store, thus allowing managers to better manage queues and entrances.

2.2 Product functions

2.2.1 Booking a ticket

CLup works as a digital counterpart to the common situation where people who are in line for a service retrieve a number that gives their position in the queue. Physically retrieving a number forces people to first approach the building, and then wait in close proximity until their number is called, which is a less than ideal situation in a lockdown situation. In order to book a ticket, the user needs to tap the corresponding button in CLup home page. Upon touch, CLup will allow the user to access the different markets available, providing a function to actually send the request for the ticket. CLup will inform the user if tickets are or are not available for the selected market. User info, as well the request, the chosen market, the timestamp, the ticket details and the associated unique QR code will be saved in a database in CLup online infrastructure. These data will be used by CLup to prompt the user with the suggested time to leave in order to arrive on time. At the market, the QR code is scanned and the request is marked as completed. Moreover, the details of the QR scanner and the time of the scan are saved within the system.

2.2.2 Booking a visit

In order to book a visit, the user needs to tap the corresponding button in CLup home page. Upon touch, CLup will allow the user to access the different markets available. Then, the user will be asked to indicate also the approximate expected duration of the visit. Alternatively, for longterm customers, this time can be inferred by the system based on an analysis of the previous visits data. The user can choose whether to indicate the duration or let the system provide it. Alongside with the duration, the application also allows users to indicate the exact list of items or the categories of items that they intend to buy. Before actually sending the request, CLup will suggest alternative time slots for visiting the store, to balance out the number of people, if the preferred one is not available. When the user confirms the booking, CLup sends the request to the server. The request is saved in a database and processed. The data will be used by CLup to prompt the user with the suggested time to leave in order to arrive on time for the visit. At the market, the QR code is scanned and the request is marked as completed, with the details of the QR scanner and the timestamp of the scan. The QR code will be scanned at the exit too, in order for the system to infer the real duration of the visit.

2.2.3 Entrances monitoring and statistics

The store managers can access CLup web interface with a browser, allowing them to see real-time online queues - accessing the online database information regarding only their particular store - and the possible future visits. Furthermore, store managers are given data analysis tools that allow them to see the inflow of customers in the market thanks to QR scanners and knowledge of assigned time slot in a given day. This feature allows the managers to better monitor the entrances and get useful data for running the market.

2.3 User characteristics

There are two kinds of users that take advantage of the system:

- Citizens: the average user is everyone who needs to fulfil her/his basic needs of getting goods in the safest and most time-efficient way. This user takes advantage of the reduction in length of the queues of supermarkets and shops, that allows both a faster and a safer way of buying the products needed. The application helps the user by telling her/him when to leave in order to reach the location in time.
- Store managers: store managers can take advantage of data analysis tools available on the web application. Information about issued ticket, booked visits, as wells as QR code scans, are available in a user friendly interface. Shop managers can therefore trace the users activity and monitor the entrances.

2.4 Assumptions, dependencies and constraints

2.4.1 Assumptions

[D1] Every citizen who wants to access CLup advanced services has a smartphone with the CLup application installed

[D2] GPS can provide a reasonably high-accuracy location of the mobile phone

[D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth.

[D4] Every grocery store or supermarket who wants to take advantage of the CLup application has the correct infrastructure to access its functionalities

[D5] The user has granted permission for GPS, disk usage and notifications

[D6] Every shop that joins CLup services, has a device equipped with a printer available for guest user who wants to get a ticket

On modern mobile phones, assumptions D2 and D3 are commonly satisfied, since every modern smartphone SoC provides both GPS antennas and mobile network antennas. Assumptions D1 and D4 are reasonable, since they represent the modern approach to applications utilization. Assumption D5 must be valid for the correct working of the "time to leave" functionality and to download the PDF digest of the booking. Assumption D6 is required for users who do not own a device to get a ticket with.

2.4.2 Dependencies

CLup needs to be able to alert the user taking into account the time they need to get to the shop from the place they currently are. In order to implement this feature, CLup application needs to access "Directions API" service[1] offered by Google which, through the starting and ending point coordinates can return a .json file with the duration of the travel. CLup will then use this information to advise the user.

2.4.3 Constraints

CLup application will be implemented through a client-server approach. CLup client-side should be responsive and fast. It should provide reliable information, especially in the calculation of the "time to leave" to the markets and in the requests sent. The connection between client and server

has to be secure. CLup server-side should be highly reliable. The integrity of the data stored in the databases on the server is critical. The server should also be highly parallelized, since it has to handle multiple requests at the same time. Other details constraint-wise are reported in section 3.5.

3 Specific Requirements

3.1 External Interface Requirements

This section gives more details about the user interfaces of the mobile application, as well as hardware, software and communication interfaces.
The web application interface and the in-shop ticket hand out application are shown and discussed in the Design Document.

3.1.1 User Interfaces

This section will present some sample interfaces taken from the application. Further mockups and UX flowcharts will be shown in the Design Document.

3.1.2 App Mockups

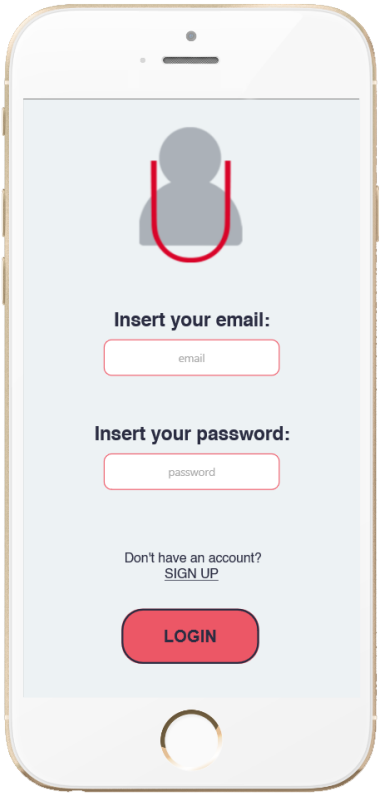

Figure 4: User Login Mockup

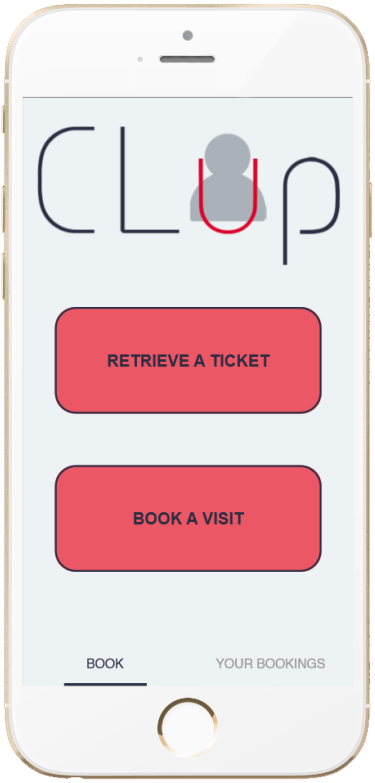

Figure 5: Home Mockup

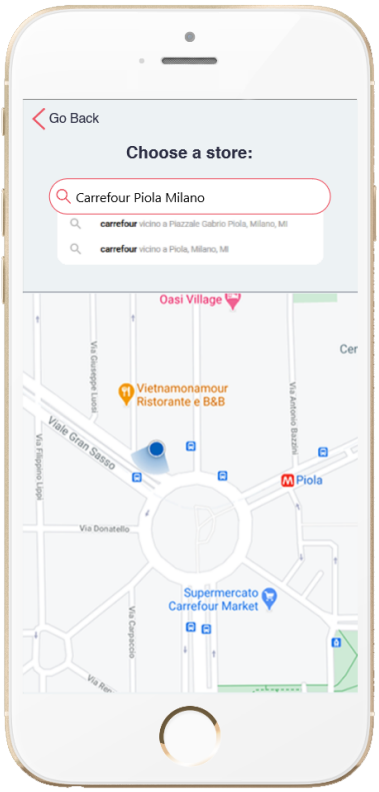

Figure 6: Choosing a store to take the ticket Mockup

Figure 7: Retrieve Ticket Mockup

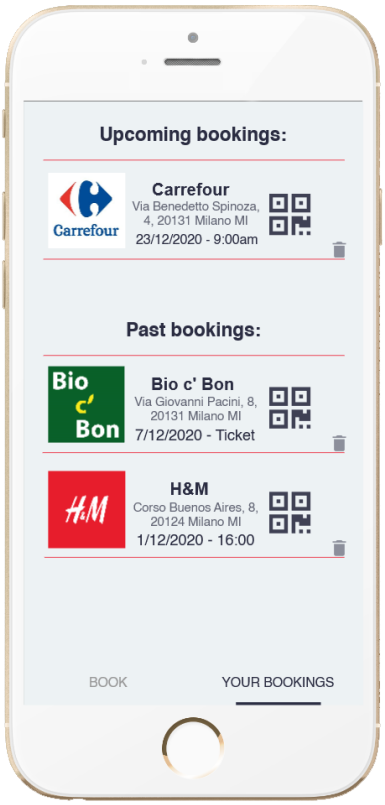

Figure 8: User Bookings Mockup

Figure 9: Example of a user booking

Figure 10: Example of a Push Notification

3.1.3 Hardware Interfaces

No external hardware interfaces are needed for users. Users will able to use the application through their own mobile phones with a internet connection and the GPS localization. The store must be equiped with a device (e.g. a tablet computer) connected to a printer to hand out tickets to customers.

3.1.4 Software Interfaces

Clup application will run on mobile application and web application: the first one is developed to allow user to take a ticket or book a visit, through their own mobile phone (having a CLup account) or through a device located outside the desired shop (in this way, guests can take ticket even if they don't have the application). The web application is built for store manager wants to keep track the shops' statistics.

3.1.5 Communication Interfaces

Application - Server communications are handled by standard TCP/IP network interfaces.

3.2 Functional Requirements

3.2.1 Citizen

Scenario 1: citizen registration Sophy is a single mom that works for an insurance company. Since she is very busy, especially because of her children, she used to buy her lunch in a cafeteria next to the office. After the Covid-19 pandemic outbreak, admissions to indoors are restricted, so she always has to queue up for 30 to 40 minutes to get her lunch. She also tried to change eatery, but the situation is pretty similar. This is a huge problem, since she doesn't have time to cook lunch for herself at home and she only have one-hour break. One day, scrolling Instagram, Sophy finds out about a new application called CLup. She immediately installs it and creates an account, simply putting in some basic information and her email. She finds out that many shops and cafeterias near her office have joined this fantastic network, and she immediately books her next visits. Thanks to the booking functions, now Sophy can enjoy the lunch break, instead of queuing up all the time.

Name	Sign up
Actor	Citizen
Entry condition	The citizen has installed the application and opens it
Event flow	1. The citizen presses the "sign up" button at the bottom of the screen 2. The citizen inserts the required fields: first name, last name, email, username and a password 3. The citizen confirms the registration by pressing the "sign up" button 4. The CLup server registers the user into the database
Exit condition	The user, identified by her/his email is now enabled to access the booking section, by logging in with the email and the password inserted during the subscription
Exceptions	• The inserted email is already registered in the system: the user will therefore be asked to insert another address • Some fields are left blank, so the missing information is highlighted, asking the user to insert them • The password inserted is too weak, or the password re-inserted does not match with the first field. The user is therefore asked to check the password.

Table 3: Use case 1: Sign up

3.2.2 User

Scenario 1: Booking tickets and visits Gloria usually goes to local shops to get primary goods: bread, vegetables, meat, and so on. Before Covid-19 pandemic, she had to queue up multiple times, wasting a lot of time. Now thanks to CLup, she can easily get information of how many people are queueing up at the shops she has to visit, getting suggestions about the best time slots and days to visit. Gloria can efficiently program the journey, by booking the tickets and keeping track of them. Furthermore, the application sends a notification to Gloria, when her turn is approaching, based on her current location. Sometimes, when Gloria knows exactly when she will be free, instead of booking instant tickets, she books a visit in advance. This way, she can be more time efficient, being sure that she will find what she needs. When Gloria arrives at the store all she has to do is open the application and show the QR code to the shop assistant, who will scan it, by double checking her visit.

Scenario 2: Using statistics George is a young student living with his parents. Since his mom and dad are pretty old, to avoid any risk of infection they don't go out except for very strong reasons. George is in charge of the groceries and he really enjoys helping his family, to face this pandemic. Before discovering CLup, George was very worried to get infected inside the supermarket, primarily because he will be responsible of infecting his parents. Now, he is very happy since the application let him consult statistics based on his and other users shopping habits, suggesting him less crowded days and hours, firstly, to lower the probability of meeting someone infectious inside the store, but also to enhance his shopping experience. This is possible thanks to those who fill in their grocery needs in the booking relative section, and also scanning by the QR code generated by the app, both at entrance and exit of the supermarket by automatic scanner.

Scenario 3: Shop assistant booking Joe's grandfather, Michael, is a 60 years old man, living alone in an apartment near the city center. He always does the grocery on Tuesdays, but most of the times, he finds a lot of people queuing outside the store. One day, talking with his grandson, he discovers about CLup. Joe is a CLup user since day one and suggests his grandpa to buy a brand-new smartphone to use it. Michael is not capable of using this kind of technologies and refuses to do so. However, he asks at the supermarket where he usually does the grocery, if there is any alternative option to avoid the queue without having to buy a phone to install the application. Linda, the store manager, tells Michael that thanks to a CLup functionality, he can get a ticket or a visit using the tablet at the entrance: he just has to press the button and the tablet will print out the booking digest. Michael is so happy that thanks to this technology he can shorten his time away from home.

Name	User log-in
Actor	User
Entry condition	The user opens the application
Event flow	1. The user presses the "log-in" button in the bottom of the screen 2. The user inserts his credentials (email and password) 3. The user presses the log-in button
Exit condition	The system accepts the credentials, and the user is now able to perform actions inside the application
Exceptions	• The email is not associated with any account, therefore the user will be asked to check her/his input • The password is not correct, so the user will be asked to insert the correct password. After three tries, the account is temporarily blocked, and a reset mail is sent

Table 4: Use case 2: User Log-in

Name	Ticket booking
Actor	User
Entry condition	The user has opened the app and is logged in
Event flow	1. The user presses the "retrieve a ticket" button 2. The user inserts the name of the shop/supermarket 3. The user confirms the chosen shop 4. The number of people queuing up is shown: the user is asked if she/he wants to proceed 5. When the user forwards the request, the CLup server process it and sends an acknowledge
Exit condition	The ticket booked prospectus is inserted, jointly with the QR code, in the "your bookings" section by the system
Exceptions	• The application does not find the searched market/shop: the application asks the user to check the query • The tickets are currently not available. A popup message is shown: the user is asked if she/he wants to choose a similar shop or go back to the starting screen

Table 5: Use case 3: Ticket booking

Name	Visit booking
Actor	User
Entry condition	The user has opened the app and is logged in
Event flow	1. The user presses the "book a visit" button 2. The user inserts the name of the shop/supermarket 3. The user confirms the chosen shop 4. The user selects the date 5. The user is asked which kind of groceries among the ones shown, she/he wants to buy (can be skipped) 6. The application shows the timeslot in which is better going to shop: the user can accept the advice, or she/he can select the timeslot in which she/he wants to shop, among the free ones 7. The user is shown a digest of the booking, and asked if she/he wants to proceed 8. When the user forwards the request, the CLup server process it and sends an acknowledge
Exit condition	The visit booked prospectus is inserted, jointly with the QR code, in the "your bookings" section by the system
Exceptions	• The application does not find the searched market/shop: the application asks the user to check the query • There are no visits currently available. A popup message is shown: the user is asked if she/he wants to choose a similar shop or go back to the starting screen

Table 6: Use case 4: Visit booking

Name	"Time to leave" notification
Actor	User
Entry condition	The user has booked a ticket or a visit
Event flow	1. The application periodically checks the current GPS position of the user, the estimated time to reach the shop, and the estimated time left for the user's turn (in case the booking is a ticket) 2. When the "time to leave" condition is satisfied, a notification is sent to the user
Exit condition	The user leaves her/his home, and reaches the shop
Exceptions	• GPS is not working, so a popup message after the booking is shown, saying that this functionality cannot be used • Notification permission is not granted, so the user will never receive any notification

Table 7: Use case 5: "Time to leave" notification

Name	Guest booking
Actor	Guest
Entry condition	The guest is at the shop in front of the tablet at disposal for booking
Event flow	1. The guest presses the button to get a ticket 2. The device will print out a receipt with the booking digest
Exit condition	The guest takes the receipt and use it as a counterpart of the electronic booking receipt
Exceptions	• Printer has run out of paper / ink • There are no available tickets

Table 8: Use case 6: Guest Booking

Name	QR code usage
Actor	User
Entry condition	The user is at the shop at entrance or exit
Event flow	1. The user opens the application, and navigate to the "your bookings" section 2. The user opens the current booking 3. The user shows the QR code to the shop assistant/automatic scanner which will scan it, double checking the user's visit
Exit condition	The user leaves the shop to go home
Exceptions	• The QR code is not recognized

Table 9: Use case 7: QR code usage

3.2.3 Store managers

Scenario 1: Monitor entrances Phil is the manager of a huge supermarket, OssoCenter. Recently, after a meeting with other colleagues and stakeholders, he decided to join the CLup network to give more services to the customers. Thanks to CLup insights about people, Phil has a clearer idea of which are the most used entries, and also the busiest days of the week. Thanks to this information he can arrange Covid-19 sanitations and also cleaning of the bathrooms and hallways. This has brought a huge economic saving; therefore, money can be invested in more useful services. Moreover, Phil has used this insight to know what the most used entry is: therefore, he charged up advertising in that area.

Scenario 2: Monitor consumptions Monica holds a bakery shop, near the main city street. She recently joined the CLup application and discovered functionalities that brought huge improvements to her business. Thanks to CLup, she can predict how much bread and cakes to cook, without having to throw away anything. Monica can see how many visit bookings are requested, and she can also know how many people are queueing in a certain day in real time. Also, when Monica knows that in a certain day, the amount of work will be higher, she asks her son, Jacob to help her in delivering orders.

Name	Shop manager log-in
Actor	Shop manager
Entry condition	The shop manager opens the browser and accesses CLup website
Event flow	1. The manager inserts the given credentials (email and password) 2. The manager presses the log-in button
Exit condition	The system accepts the credentials, and the manager is now able to perform actions inside the application
Exceptions	• The email is not associated with any account, therefore the user will be asked to check the input • The password is not correct, so the user will be asked to insert the correct password. After three tries, the account is temporarily blocked, and a reset mail is sent

Table 10: Use case 8: Shop manager log-in

Name	Statistics consultation
Actor	Shop manager
Entry condition	The shop manager opens the browser and logs in
Event flow	1. The manager selects the section in which he wants to consult statistics
Exit condition	The user is able to see graphics and other useful information about her/his shop/supermarket
Exceptions	• There are no data available to be shown

Table 11: Use case 9: Statistics consultation

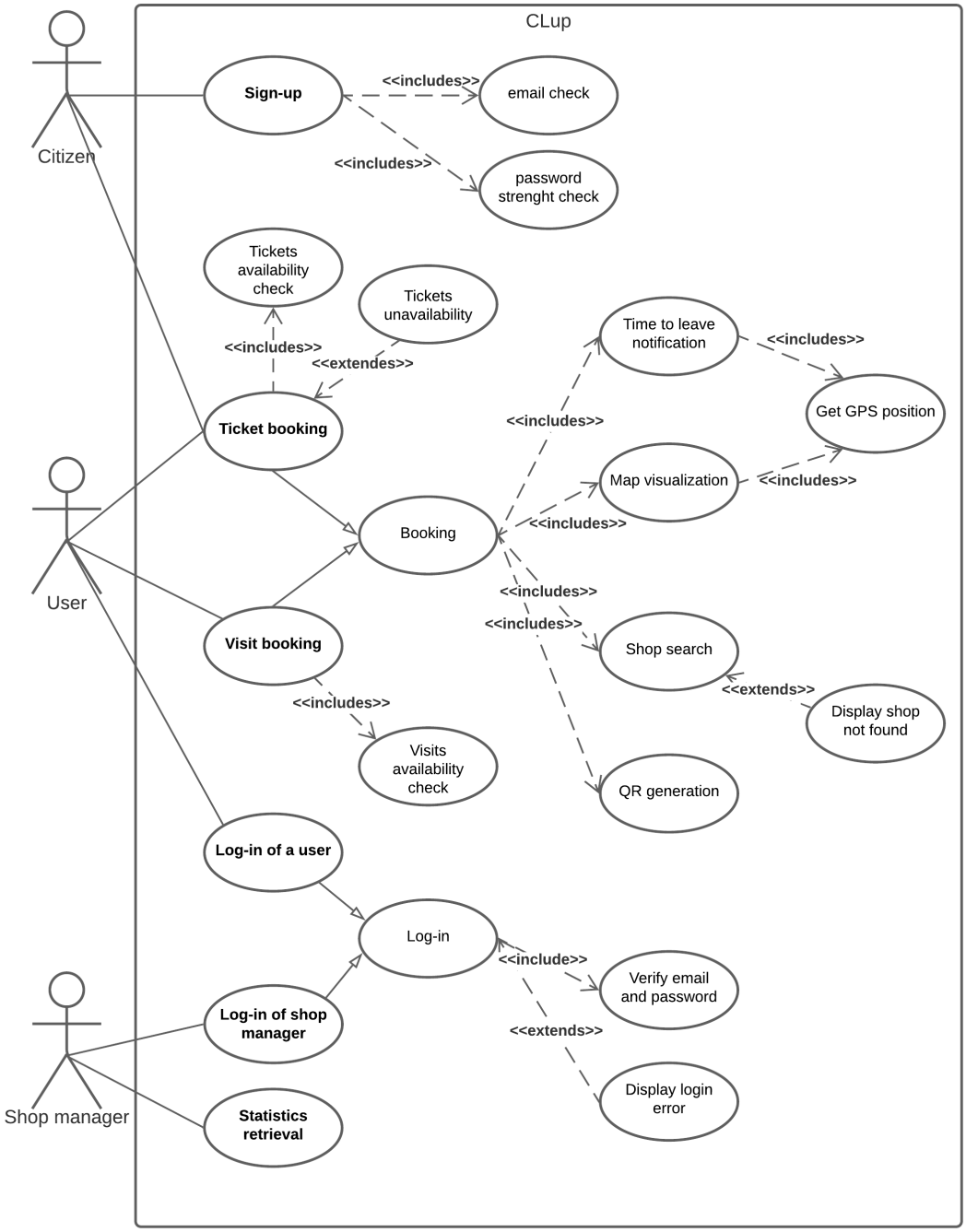

Figure 11: Use case diagram

The use case diagram shown above in **[Fig.11]** shows the three main actors that interact with the application: the citizens (which are not registered), the user (which is registered) and the shop manager. On the diagram, it is clear which functionalities each actor can access, and also the dependencies of these modules with other modules.

3.2.4 Sequence Diagrams

The first sequence diagram shown **[Fig.12]** represents the sequence of operations a user has to perform in order to book a ticket or a visit. The first operation is the login: if the credentials are accepted then the user is allowed to perform the booking, else an error message is shown, depending on the problem (email not found or password wrong). The booking process is straightforward: the user selects a shop, querying data from it. It receives the response and from time to time it can receive suggestions suited for the user. At the end the user sends the booking and the server replies with the booking digest including the QR code.

The second sequence diagram **[Fig.13]** shows the process by which a shop assistant can consult data on the web application. The login process is similar to the first sequence diagram, while the data are fetched through HTTP queries.

The last sequence diagram **[Fig.14]** represents a noteworthy situation involving the TTL notification algorithm. If there is any booking pending, the TTL component wakes up from time to time, and checks if it is time to leave. It requests the GPS position, and then it calls an external API (Google Directions) that given the user and the shop position, returns the estimated travelling time. If it is time for the user to leave, a notification is sent, otherwise the interrupt is rescheduled again.

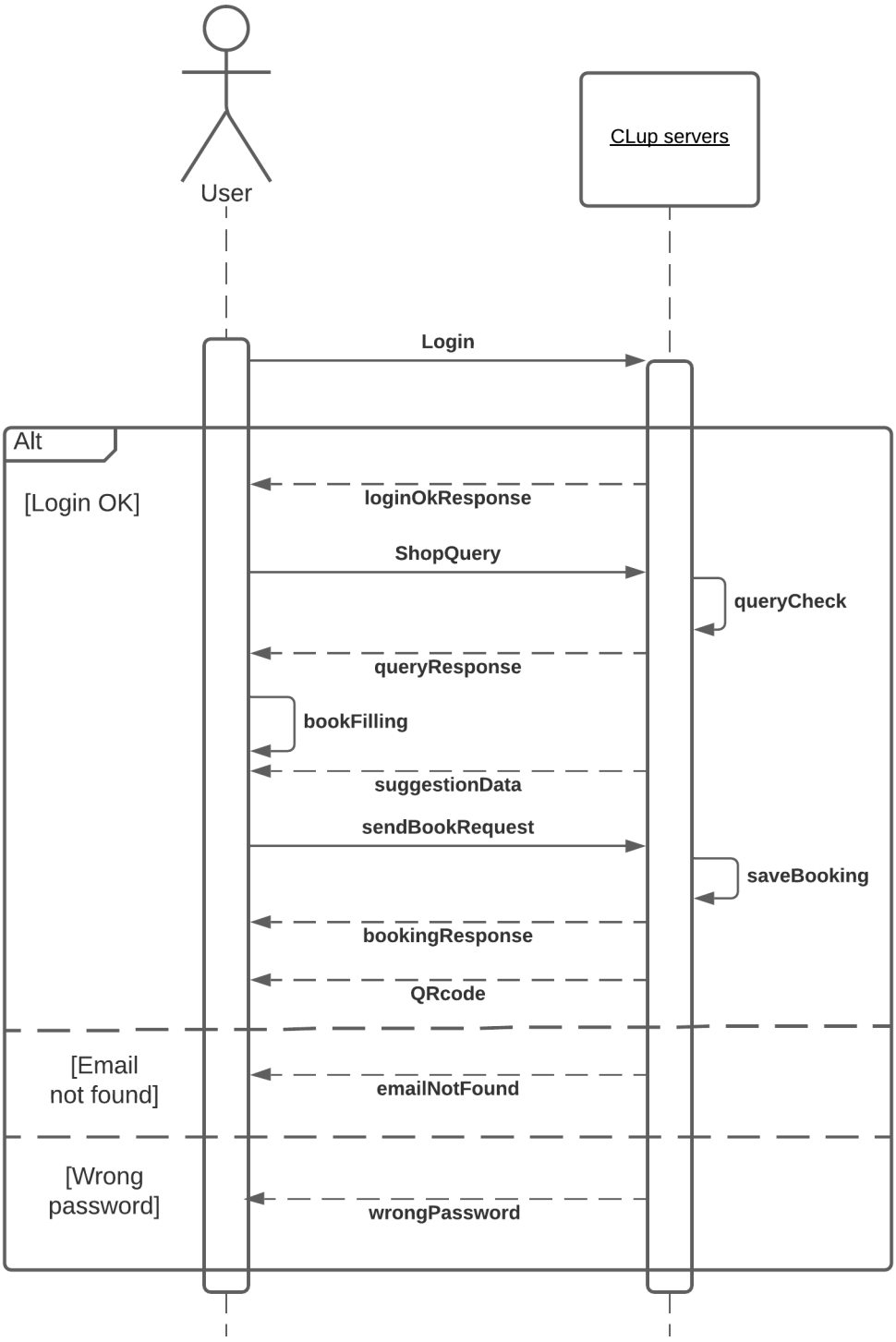

Figure 12: User Login sequence diagram

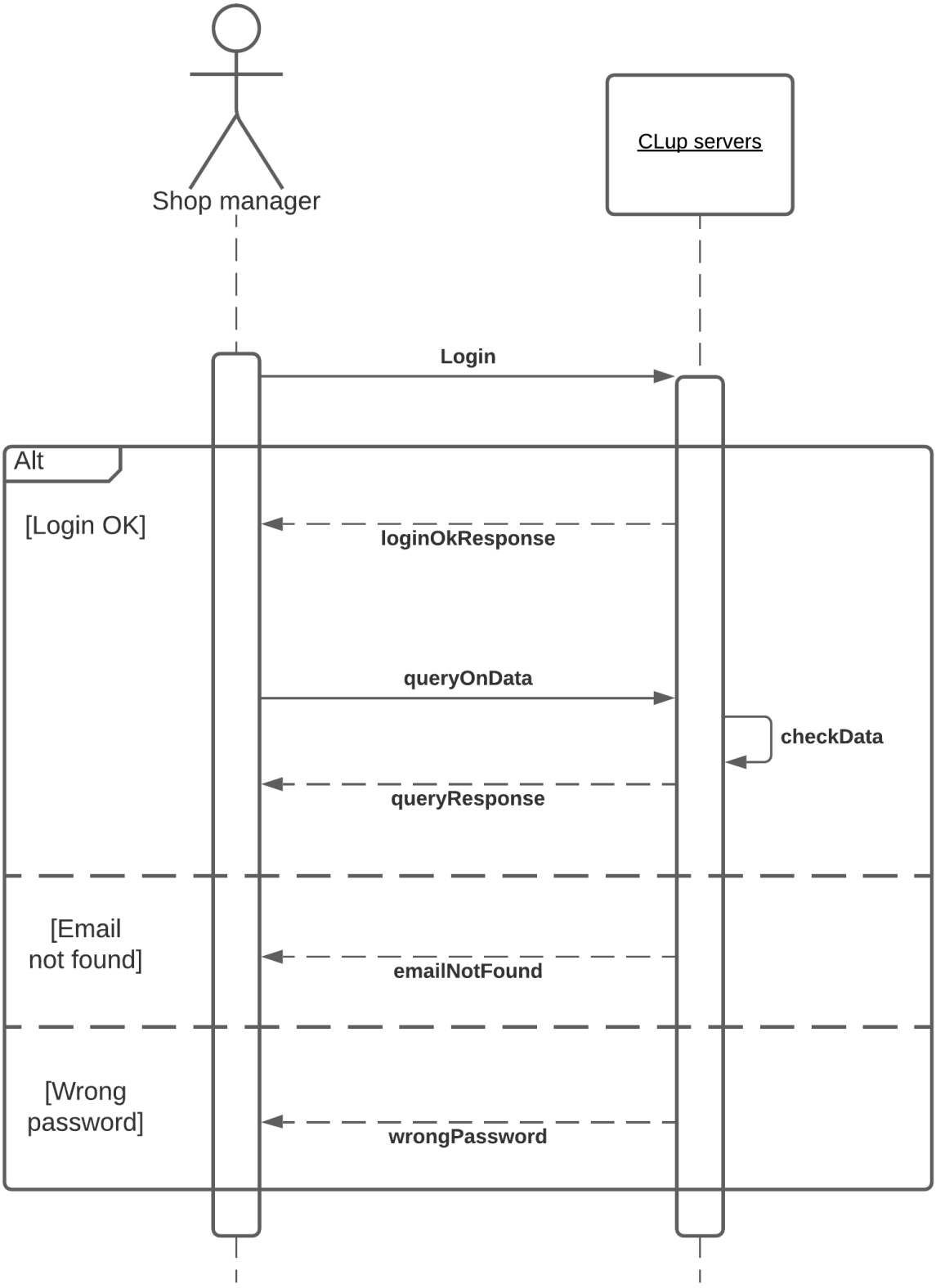

Figure 13: Shop manager sequence diagram

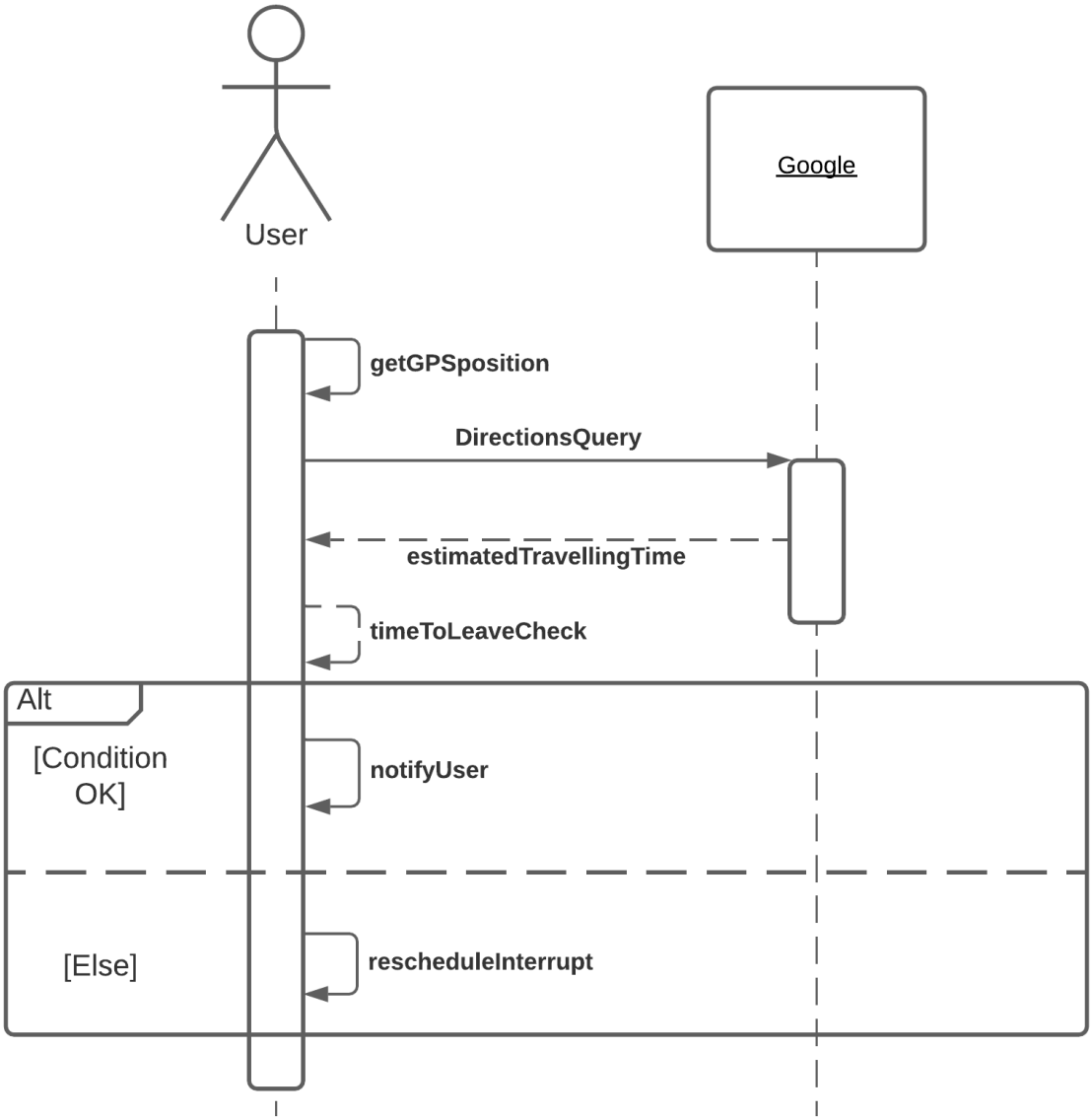

Figure 14: Time to leave sequence diagram

3.2.5 Requirements

Requirements must ensure satisfaction of the goal given the context of the domain assumption.

[G1] **A user can book a ticket**

- [D1] Every citizen who wants to access CLup services has a smartphone with the CLup application installed.
- [D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth
- [D5] The user has granted permission for GPS, disk usage and notifications
- [R1] A citizen not yet registered must be able to sign up and become a user
- [R2] The application must allow users to authenticate
- [R3] The application must allow the user to search for the desired store
- [R4] The application must allow user to book after filling the required data
- [R5] The system must correctly save data regarding issued bookings
- [R6] The application must offer an interface to retrieve issued bookings
- [R17] The application must allow the user to book only if there is an available place

[G2] **A user can book a visit**

- [D1] Every citizen who wants to access CLup services has a smartphone with the CLup application installed.
- [D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth
- [D5] The user has granted permission for GPS, disk usage and notifications
- [R1] A citizen not yet registered must be able to sign up and become a user
- [R2] The application must allow users to authenticate
- [R3] The application must allow the user to search for the desired store
- [R4] The application must allow user to book after filling the required data
- [R5] The system must correctly save data regarding issued bookings
- [R6] The application must offer an interface to retrieve issued bookings
- [R7] The application must allow the user to select a date and a time for the visit
- [R17] The application must allow the user to book only if there is an available place

[G3] **The app can alert users when it's time to start going to shop, according to their GPS position**

- [D1] Every citizen who wants to access CLup advanced services has a smartphone with the CLup application installed
- [D2] GPS can provide a reasonably high-accuracy location of the mobile phone
- [D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth
- [D5] The user has granted permission for GPS, disk usage and notifications
- [R8] When a booking is pending, the application must be able to wake up at a certain frequency to check when it is time to leave for the user

[R9] The application must send a notification when it is time to leave

[R16] The system must be able to interface with third party services

[G4] **customers who don't have access to a smartphone, can get ticket on the spot**

[D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth

[D4] Every grocery store or supermarket who wants to take advantage of the CLup application has the correct infrastructure to access its functionalities

[D6] Every shop that joins CLup services, has a device equipped with a printer available for guest user who wants to get a ticket

[R5] The system must correctly save data regarding issued bookings

[R10] The application must print out the booking digest

[G5] **User can get suggestions by the application about different time slots and the duration of a visit based on previous data**

[D1] Every citizen who wants to access CLup services has a smartphone with the CLup application installed

[D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth

[R11] The application must keep data about previous visits to shops

[R12] The application must popup suggestion during the booking process

[R14] The system must be able to analyze data and show statistics in order to monitor the entrances

[G6] **The users validate the booking, by scanning the QR code at the entrance of the store**

[D4] Every grocery store or supermarket who wants to take advantage of the CLup application has the correct infrastructure to access its functionalities, such as a PC

[R6] The application must offer an interface to retrieve issued bookings

[R13] The generated QR code must contain all useful data for the transaction

[G7] **Shop managers can visualize statistics about the booking requests**

[D3] Every device that wants to take advantage of CLup services has an internet connection with a reasonable network bandwidth.

[D4] Every grocery store or supermarket who wants to take advantage of the CLup application has the correct infrastructure to access its functionalities

[R11] The application must keep data about previous visits to shops

[R13] The generated QR must contain all useful data for the transaction

[R14] The system must be able to analyze data and show statistics in order to monitor the entrances

[R15] The system must allow the shop manager to see only data related to her/his store

3.2.6 Traceability matrix

	Use cases									Goals						
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	G1	G2	G3	G4	G5	G6	G7
R1	X		X	X						X	X					
R2		X	X	X				X	X	X	X					
R3			X	X						X	X					
R4			X	X						X	X					
R5			X	X		X	X	X	X	X	X		X			
R6			X	X				X	X	X	X				X	
R7				X							X					
R8					X							X				
R9					X							X				
R10						X							X			
R11					X			X	X					X		X
R12														X		
R13			X	X		X	X	X	X						X	X
R14								X	X					X		X
R15								X	X							X
R16					X							X				
R17			X	X						X	X					

Table 12: Traceability matrix

3.3 Performance requirements

CLup application is expected to be used mainly during the day. Through a quick search in the data collected by Google[2] and made public within Google Maps we can expect the highest usage of the ticket retrivial feature to be around the 10:00 - 12:00 and 15:00 - 18:00 time slots. Hence, we expect the service to reach the highest usage in those hours. During night time, a great reduction in the service utilization is expected. A great degree of parallelization is therefore needed to run the service properly within every time slot. The maximum number of users expected could be around 10-15% of customers for every smaller grocery store and 5-10% of customers for every supermarket that adopts CLup application.

3.4 Design constraints

3.4.1 Standards compliance

CLup application should be as safe and secure as possible. CLup users and their data must be protected. These standards must be taken into account from the ground of the application up:

- ISO/IEC 27001:2013: specifies the requirements for establishing, implementing, maintaining and continually improving an information security management system within the context of the organization.

- ISO/IEC 20648:2016: details the requirements for use of the Transport Layer Security (TLS) protocol in conjunction with data storage technologies. The requirements set out in this specification are intended to facilitate secure interoperability of storage clients and servers as well as non-storage technologies that may have similar interoperability needs.
- General Data Protection Regulation (EU) 2016/679 (GDPR)

3.4.2 Hardware limitations

CLup application is made of two front-ends, one through the mobile application for the customers, and one through the web app for the store managers. Both the front-ends require internet access with a reasonable data bandwidth. For the mobile application to take advantage of all the available features, a GPS antenna is also necessary, though not strictly required for the basic functions to work.

3.4.3 Other constraints

The application will need to take care of the constraints of the OSs hosting it too, both from UI and application logic sides.

3.5 Software System Attributes

3.5.1 Reliability

The application should perform even when unexpected or unanticipated events occur. In order to create a reliable system, an accurate testing suite should be developed and the application should be extensively tested. The service should present an extimated MTBF larger than at least 1'500'000 hours[3].

3.5.2 Availability

CLup service represents an active countermeasure to the spreading of a severe pandemic as COVID-19. Given the vital importance of such measures and services, the service is expected to have at least 99.9% availability. Therefore during one year, at most 8.76 hours of downtime are expected.

3.5.3 Security

CLup service is expected to interact with highly sensitive and private data. Users and their data must be a top priority for CLup service. On top of applying the GDPR standards, ISO/IEC 20648:2016 and ISO/IEC 27001:2013, CLup has to implement further measures to ensure that data remains protected and unaccessible to malicious users. Data on the drives should always be encrypted with strong encryption algorithms as AES 256. Connections from and to the service servers should always be encrypted as well, through the latest TLS 1.3 protocol. Furthermore, in order to better protect users, passwords must always be saved after being hashed through SHA-2 algorithm, and only the digest should be saved. These measures and protocols, when correctly implemented and along with a well written code, should provide the safest service.

3.5.4 Maintainability

In order to be maintainable, code must follow the naming conventions of the appropriate programming language. Code should be clear and well commented. Moreover, design patterns and best-practices should be used whenever is possible.

3.5.5 Portability

CLup application is expected to work in a variety of OSs. This is due to the double use cases of the service (both customers and store managers will need to access CLup) and the presence of a number of competing platforms such as iOS and Android in the mobile market or Windows and Linux-based distros server-side.

4 Formal Analysis Using Alloy

In this section a formal model of the application "CLup" is presented. The model has been created through Alloy modelling language. The model allows a better visual understanding of the application domain thanks to the visual representation of the worlds generated from the code, while assuring the correctness of the conceptual model developed in the requirements document. The .als file can be found in the GitHub repository. In order to make the model more clear, the code is separated with comments in "Signatures", "Facts", "Predicates" and "Assertions".

4.1 Alloy code

```
//Signatures

abstract sig Account {
        username: one Username,
        email: one Email,
}

sig Username {}
sig Email {}

sig User extends Account {
        bookings: set Booking,
}

sig Manager extends Account {
        shop: one Shop
}

sig Guest {}

sig ShopType {}
sig ShopName {}

sig Shop {
        name: one ShopName,
        shopType: one ShopType,
        position: one Position,
        dailyAvailability: one Int,
        reservationsAvailability: set Slot
}
{
 dailyAvailability ≥ 0
}

sig Date {}
sig Time {}

sig Slot {
        date: one Date,
```

```
        time: one Time
}

sig Position {}

abstract sig Bool {}
one sig TRUE extends Bool {}
one sig FALSE extends Bool {}

sig Timestamp {}

sig QRcode {
        scanned: one Bool
           ↪
}

abstract sig Device {
        internetConnection: one Bool
}

sig PersonalDevice extends Device {
        user: one User,
        position: lone Position,
        GPS: one Bool
}

sig GuestDevice extends Device {
        shop: one Shop
}

abstract sig validationBooking {}
   ↪
one sig VALIDATED extends validationBooking {}
one sig PENDING extends validationBooking {}

abstract sig Booking {
        user: one User,
        shop: one Shop,
        date: one Date,
        qrcode: one QRcode,
        status: one validationBooking
}

sig Ticket extends Booking {
        queuePosition: one Int
}
{
queuePosition > 0
}
```

```
sig Visit extends Booking {
        visitSlot: one Slot,
        durationTime: one Time,
        list: one Bool
}

sig Suggestion {
        user: one User,
        shop: one Shop,
        visitSlot: one Slot,
        duration: one Int,
        timeToLeave: one Bool
}


//Facts


//The position should be available if and only if the GPS signature
   ↪  is available
fact positionOnlyIfGPS {
        all p: PersonalDevice | p.GPS = FALSE implies #p.position =
           ↪  0
}

//Each device that takes advantage of CLup services must have an
   ↪ active internet connection
fact DeviceHasConnection {
        all p: PersonalDevice | p.internetConnection = TRUE
        and
        all g: GuestDevice | g.internetConnection = TRUE
}

//Two tickets emitted on the same day, for the same shop, must not
   ↪ have the same position
fact noSameTicketsSameDaySameShop {
        no disj t1, t2 : Ticket | t1.shop = t2.shop and t1.date =
           ↪ t2.date and t1.queuePosition = t2.queuePosition
}

//There cannot be a user sharing the same position of other users
fact differentPositionsUser {
        no disj p1, p2 : PersonalDevice | p1.position = p2.position
}

//Each booking is unique, given its qrcode
fact uniqueBooking {
        no disj b1, b2 : Booking | b1.qrcode = b2.qrcode
}
```

```
// Each booking is owned by a unique user
fact uniqueBookingProperty {
        no u1, u2 : User | some b : Booking | b in u1.bookings and
           ↪ b in u2.bookings
}

// Each email present in the database is associated with an entity
fact allEmailsAssociatedToEntity {
        all m : Email | one e : Account | e.email = m
}

// Each suggestion on the system is linked to a specific and unique
   ↪  user
fact uniqueSuggestions {
        no disj s1, s2 : Suggestion | s1.user = s2.user
        no disj u1, u2 : User | some s : Suggestion | s.user = u1
           ↪ and s.user = u2
}

// The issued tickets number must never exceed the total daily
   ↪ availability of the shop
// The ticket number must be lower than the queue maximum number
   ↪ given by the daily availability of the shop
fact ticketConsistentCoherent {
        all t : Ticket | #{t1:Ticket | t1.date = t.date and t1.shop
           ↪  = t.shop} ≤ t.shop.dailyAvailability
        and
        all t : Ticket | t.queuePosition ≤ t.shop.dailyAvailability
}

// Two tickets issued in the same day and shop, must never have the
   ↪  same number
fact ticketUnique {
        no disj t1, t2: Ticket | t1.shop = t2.shop and t1.date = t2
           ↪ .date and t1.queuePosition = t2.queuePosition
}

// There cannot be two or more accounts sharing the same email
   ↪ address
fact noDuplicateAccount {
        all a, b: Account | a ≠ b iff a.email ≠ b.email
}

// There cannot be two or more shop sharing the same name and
   ↪ position
fact noMultipleShops {
        no disj s1, s2 : Shop | s1.name = s2.name and s1.position =
           ↪  s2.position
}
```

```
// Each shop has a unique device allowed to issue tickets for
   ↪ guests
fact oneDevicePerShop {
        no disj d1, d2 : GuestDevice | d1.shop = d2.shop
}

// Each timeslot is uniquely identified by a time and a date
fact uniqueSlot {
        no disj s1, s2 : Slot | s1.date = s2.date and s1.time = s2.
           ↪ time
}

// Each shop has a unique manager allowed to access the system
fact managerForEachShop {
        all s: Shop | some m : Manager | m.shop = s
}

// Each personal device have a unique account associated with it
fact AccountForEachPersonalDevice {
        all u: User | one p: PersonalDevice | p.user = u
}

// If the GPS is ON, then the position must be available
fact PositionAvailableIfGPSIsActive {
        all p : PersonalDevice | p.GPS = TRUE    #p.position = 1
}

// A scanned QR code makes a ticket VALIDATED
fact scannedMakesValidated {
        all b : Booking | b.qrcode.scanned = TRUE implies b.status
           ↪ = VALIDATED
}

// Predicates


// This predicate converts the status of a booking to VALIDATED. It
   ↪  should be called when the QR code is scanned
pred validateBooking [b: Booking] {
        b.qrcode.scanned = TRUE
        b.qrcode.scanned = TRUE    b.status = VALIDATED
}

// This predicate builds up a ticket to be assigned to a certain
   ↪ user
pred bookTicket [u: User, ticket: Ticket, s: Shop, qc: QRcode,
   ↪ queuepos: Int] {
        ticket.user = u
        ticket.shop = s
        ticket.qrcode = qc
```

```
        ticket.qrcode.scanned = FALSE
        ticket.queuePosition = queuepos
        ticket.status = PENDING

        addBookingToUser[u, ticket]
}

// This predicate builds up a visit to be assigned to a certain
   ↪ user
pred bookVisit [u: User, visit: Visit, s: Shop, qc: QRcode,  qp:
   ↪ Int, vS: Slot, dT: Time, ls: Bool] {
        visit.user = u
        visit.shop = s
        visit.qrcode = qc
        visit.qrcode.scanned = FALSE
        visit.visitSlot = vS
        visit.durationTime = dT
        visit.list = ls
        visit.status = PENDING

        addBookingToUser[u, visit]
}

//This predicate adds the booking to the user's booking list
pred addBookingToUser [user: User, booking: Booking] {
        user.bookings = user.bookings + booking
}

// Assertions

// This assertion is checking the [G1] - The user should be able to
   ↪  book a ticket
assert ticketBooked {
        all u: User, t: Ticket, s: Shop, qr: QRcode, qp: Int |
           ↪ bookTicket[u, t, s, qr, qp] implies t in u.bookings
           ↪ and t.user = u
        and t.shop = s and t.qrcode = qr and t.queuePosition = qp
}
check ticketBooked for 7

// This assertion is checking the [G2] - The user should be able to
   ↪  book a visit
assert visitBooked {
        all u: User, v: Visit, s: Shop, qr: QRcode, qp: Int, vS:
           ↪ Slot, dT: Time, ls: Bool| bookVisit[u, v, s, qr, qp,
           ↪ vS, dT, ls]
        implies v in u.bookings and v.user = u and v.shop = s and v
           ↪ .qrcode = qr and v.visitSlot = vS
        and v.durationTime = dT and v.list = ls
}
```

```
check visitBooked for 7

// This assertion is checking the [G6] - The users validate the
   ↪ booking by scanning their QR code
assert userValidateScanningQR {
        all u: User, t: Ticket, s: Shop, qr: QRcode, qp: Int |
           ↪ bookTicket[u, t, s, qr, qp] and validateBooking[t]
           ↪ implies
        t.status = VALIDATED
}
check userValidateScanningQR for 7
```

4.2 Assertions

The Alloy analyzer provided these results: showing that it wasn't able to find any counterexamples for the three assertions, we demonstrated that the system we implemented is consistent and coherent.

```
Executing "Check ticketBooked for 7"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  23385 vars. 1724 primary vars. 51527 clauses. 276ms.
  No counterexample found. Assertion may be valid. 34ms.

Executing "Check visitBooked for 7"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  23504 vars. 1740 primary vars. 51722 clauses. 99ms.
  No counterexample found. Assertion may be valid. 23ms.

Executing "Check userValidateScanningQR for 7"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  23378 vars. 1724 primary vars. 51417 clauses. 80ms.
  No counterexample found. Assertion may be valid. 3ms.
```

Figure 15: Alloy execution of the assertions

4.3 Worlds Generated

Here are presented two worlds generated using Alloy modelling language.

4.3.1 First world

In the first world presented in the next page [fig.17], what can be appreciated is the presence of a user owning two bookings: a ticket and a visit for a certain shop. These two booking are in two different states, depending on the QR code scan: the first is VALIDATED because the QR code has been scanned, while the second is PENDING because the QR code has not been scanned yet. Again, it can be seen that the Ticket number is "1", which is lower than the maximum daily availability of the shop. Finally, the last thing noticeable in this world is the fact that the visit slot is exactly in the set of slot made available by the shop. In sight of this, we can say that this world, given the facts declared in the alloy model, is coherent and consistent with the requirements previously identified.

4.3.2 Second world

In the second world [fig.18], we augmented the generated entities to prove other facts. First of all, it can be noticed the number of issued tickets by a shop is never bigger than the total number of places available. Furthermore, two tickets for the same day and shop cannot have the same number. The user profiles are identified by the email, so it is legit that some users share the same username. Finally, we can see that position is available only for users that have a GPS connection available. In this world, we can see that user1 has two tickets in her/his list: ticket0 is an old ticket, and is still saved in the system. The status of the ticket is VALIDATED, since the user has checked her/his presence at the shop, by scanning the QR code. The second ticket in the list, is ticket2. It is noticeable that the user has not yet scanned the QR code, because the status of the booking is still PENDING. However, we can infer that this will happen soon because the user is at the shop, given the fact that both these entities share the same position. Again, we can say that this world, given the facts declared in the alloy model, is coherent and consistent with the requirements previously identified.

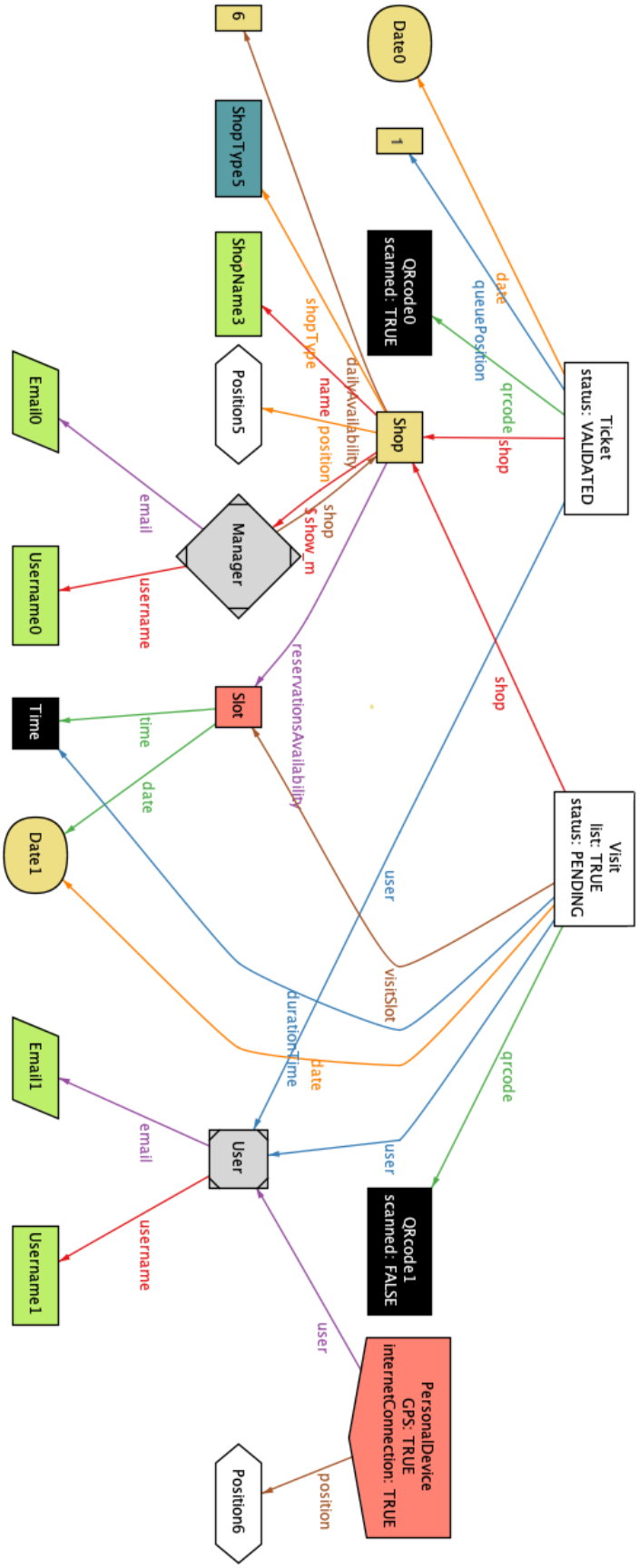

Figure 16: First world generated by Alloy

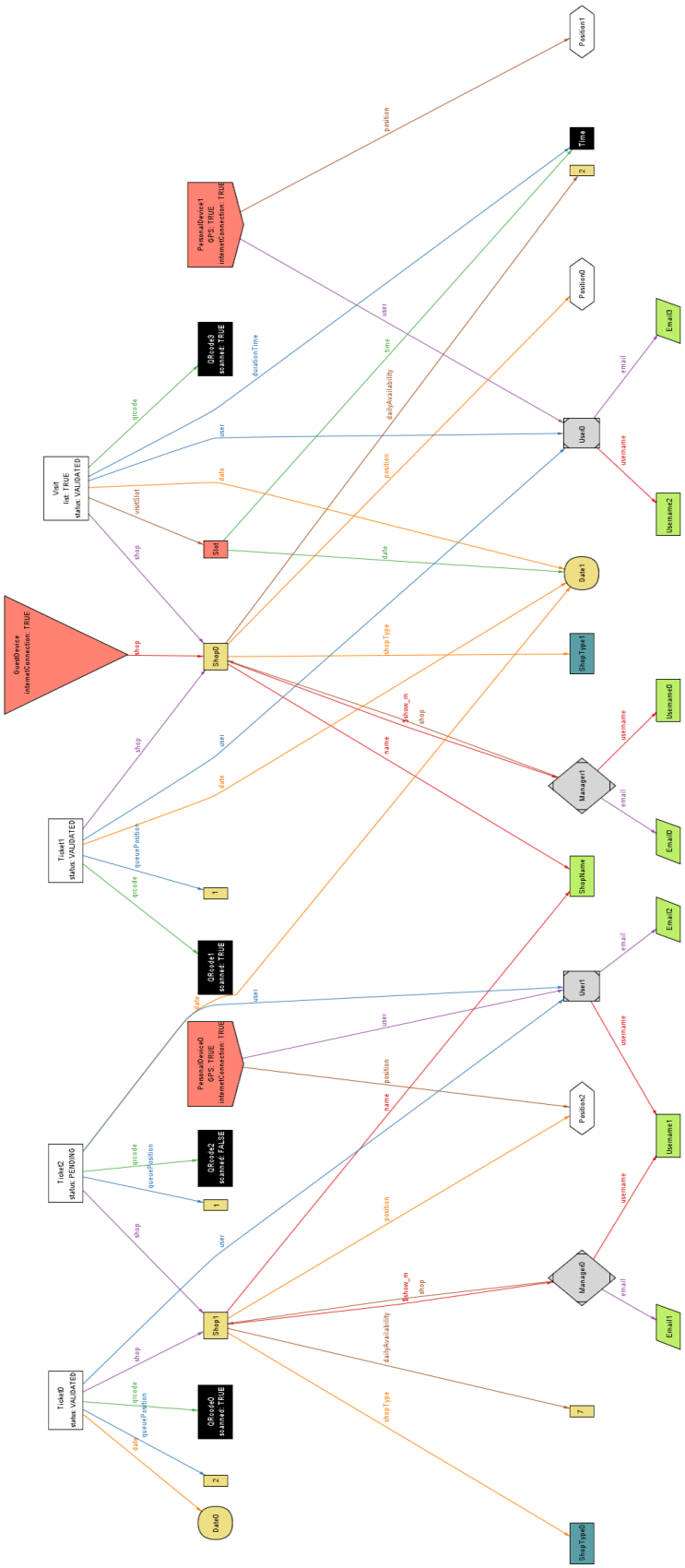

Figure 17: Second world generated by Alloy

5 Effort Spent

Hamza Haddaoui		
Task	**Time**	**Date**
First Meeting	1.5h	2020/10/16
Second Meeting	2.0h	2020/10/23
Review	2.0h	2020/11/29
Third Meeting (Ch1,2 rev)	3.0h	2020/11/30
UML making	3.0h	2020/12/01
Functional Requirements	6.0h	2020/12/06
Chapter 3 MEETING A	3.0h	2020/12/07
Functional Requirements	3.0h	2020/12/08
Chapter 3 MEETING B	3.0h	2020/12/08
Alloy meeting	3.0h	2020/12/12
Alloy	4.0h	2020/12/16
Chapter 4 Meeting	3.0h	2020/12/16
Alloy Testing	2.5h	2020/12/17
Latex RASD import	6.0h	2020/12/18
Alloy refinement	2.5h	2020/12/18
Alloy working	1.5h	2020/12/19
Alloy and Latex refining	2.5h	2020/12/20
RASD check	1.5h	2020/12/20
RASD finishing	3.0h	2020/12/21
RASD final review	2.0h	2020/12/22
Total	**58 h**	

Giuseppe Piccirillo		
Task	**Time**	**Date**
First Meeting	1.5h	2020/10/16
Second Meeting	2.0h	2020/10/23
First chapter	3.5h	2020/11/28
Third Meeting (Ch1,2 rev)	3.0h	2020/11/30
UML making	3.0h	2020/12/01
Mobile app mockups A	6.0h	2020/12/04
Mobile app mockups B	4.0h	2020/12/05
Chapter 3.1 making	0.5h	2020/12/06
Chapter 3 MEETING A	3.0h	2020/12/07
Chapter 3 MEETING B	3.0h	2020/12/08
Alloy meeting	3.0h	2020/12/12
Alloy Signatures	1.5h	2020/12/12
Alloy	1.5h	2020/12/16
Chapter 4 Meeting	3.0h	2020/12/16
Alloy Testing	2.5h	2020/12/17
Alloy refinement	2.5h	2020/12/18
Alloy working	1.5h	2020/12/19
Alloy and Latex refining	2.5h	2020/12/20
RASD check	1.0h	2020/12/21
RASD finishing	3.0h	2020/12/21
RASD final review	2.0h	2020/12/22
Total	**53.5 h**	

Alessandro Restifo		
Task	**Time**	**Date**
First Meeting	1.5h	2020/10/16
Second Meeting	2.0h	2020/10/23
Chapter 2 making	7.0h	2020/11/28
Chapter 1 refinement	1.2h	2020/11/29
Third Meeting (Ch1,2 rev)	3.0h	2020/11/30
UML making	3.0h	2020/12/01
Chapters 3.3, 3.4, 3.5	5.5h	2020/12/06
Chapter 3 MEETING A	3.0h	2020/12/07
Chapter 3 MEETING B	3.0h	2020/12/08
Alloy meeting	3.0h	2020/12/12
Alloy	2.5h	2020/12/16
Chapter 4 Meeting	3.0h	2020/12/16
Alloy Testing	2.5h	2020/12/17
Alloy refinement	2.5h	2020/12/18
Alloy working	1.5h	2020/12/19
Alloy and Latex refining	2.5h	2020/12/20
RASD check	1.5h	2020/12/21
RASD finishing	3.0h	2020/12/21
RASD final review	2.0h	2020/12/22
Total	**53.2 h**	

6 References

[1] - par 2.4.2: Google Directions API Documentation at https://developers.google.com/maps/documentation/directions/overview?csw=1
[2] - par 3.3: Google Business Help page at https://support.google.com/business/answer/6263531hl=en#:~:text=To%20determine%20popular%20times%2C%20wait,in%20to%20Google%20Location%20History
[3] - par 3.5.1: Maxon technical support on "Reliability analysis, Failure rate, MTBF" at https://support.maxongroup.com/hc/en-us/articles/360017808654-Reliability-analysis-Failure-rate-MTB