



POLITECNICO

MILANO 1863

SOFTWARE ENGINEERING 2 PROJECT

Requirement Analysis and Specification Document

CLup - Customer Line-up

Version 1.1

Authors

Galzerano Arianna
Lampis Andrea
Leone Monica

Supervisor

Dr Di Nitto Elisabetta

Sommario

1. INTRODUCTION.....	4
1.1 PURPOSE.....	4
1.1.1 Description of the given problem.....	4
1.1.2 Goals	5
1.1.3 Manager Service.....	5
1.2 SCOPE.....	6
1.2.1 World phenomena.....	6
1.2.2 Machine Phenomena.....	6
1.2.3 Shared phenomena	7
1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS.....	7
1.3.1 Definitions	7
1.3.2 Abbreviations.....	7
1.3.3 Acronyms.....	8
1.4 REVISION HISTORY.....	8
1.5 REFERENCE DOCUMENTS.....	8
1.6 DOCUMENT STRUCTURE.....	8
2. OVERALL DESCRIPTION.....	9
2.1 PRODUCT PERSPECTIVE	9
2.1.1 Scenarios.....	9
2.1.2 UML Description.....	11
2.1.3 State Charts.....	14
2.2 PRODUCT FUNCTIONS.....	15
2.2.1 Line up.....	15
2.2.2 Booking a visit	15
2.2.3 Suggest alternatives when the desired store is not available.....	15
2.2.4 Display of the stores' waiting time	16
2.2.5 Monitoring people inside a store.....	16
2.2.6 Allow to monitor statistics	16
2.3 USER CHARACTERISTICS	17
2.3.1 Actors.....	17
2.4 ASSUMPTION, DEPENDENCIES, AND CONSTRAINTS	18
2.4.1 Domain assumptions	18
2.4.2 Dependencies.....	18
3. SPECIFIC REQUIREMENTS	19
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	19
3.1.1 User Interfaces.....	19
3.1.2 Hardware Interfaces.....	22
3.1.3 Software Interfaces.....	22
3.1.4 Communication Interfaces.....	22

3.2 FUNCTIONAL REQUIREMENTS.....	23
3.2.1 List of Requirements.....	23
3.2.2 Mapping.....	24
3.2.3 Use Cases.....	28
3.2.3.1 Use Cases Description.....	28
3.2.3.2 Use cases diagram.....	34
3.2.4 Sequence diagrams	35
3.3 PERFORMANCE REQUIREMENTS.....	40
3.4 DESIGN CONSTRAINTS.....	40
3.4.1 Standards compliance.....	40
3.4.2 Hardware limitations	40
3.5 SOFTWARE SYSTEM ATTRIBUTES	40
3.5.1 Reliability and Availability.....	40
3.5.2 Security.....	40
3.5.3 Maintainability	41
3.5.4 Portability	41
3.5.5 Scalability	41
4. FORMAL ANALYSIS USING ALLOY.....	42
5. EFFORT SPENT	52
5.1 GALZERANO ARIANNA.....	52
5.2 LAMPIS ANDREA.....	52
5.3 LEONE MONICA.....	53

1. Introduction

1.1 Purpose

This document is the *Requirements Analysis and Specification Document (RASD)* of the *CLup – Customers line-up* application. It contains the main goals, the domains, and its representation through some models, in order to define and entirely describe the system in terms of functional and non-functional requirements. It also analyzes the main needs of the customers and the scenarios with the most typical uses cases.

The document represents the baseline for project planning and estimation of costs and size, for software evaluation and for change controls. It is addressed both to the project managers and the developers, in order to better understand the whole application and how to develop it.

1.1.1 Description of the given problem

CLup is an application with two main purposes: the first one is to allow store managers to regulate the flow of people in the building, and the second one is to save people from having to line up and stand outside of stores for a huge amount of time.

The application allows the user to see a map of the city on which the nearest stores are highlighted with the opening hours and the current time that approximately must be waited to enter. After choosing a store, the user has two options to enter it:

- Line up retrieving a number from his home and waiting until the number is called to approach the store. If the user selects an unavailable store, where the current waiting queue is longer than two hours, the application suggests him the nearest store with a shorter waiting time or which is currently available.
- Book a visit to the supermarket, guaranteeing that the client will be able to enter the store in a certain time. In this case, the system will send a notification to the client two hours before the booking as a remainder and in order to have a confirmation of the booking. If the user wants to select an unavailable slot of time, the application suggests him the nearest stores where the chosen slot is still available.

In the case LineUp reservation, the system will periodically calculate the distance of the customer from the store to send him a notification when it is time to get close to the store. In order to making this time estimation as exact as possible, the user can also indicate the means of transportation (on foot, car, public transport) through which he will go to the supermarket.

Moreover, the user must indicate if he will come alone or with another person and the amount of time he will stay in the store (0.15, 0.30, 0.45, 1, 1.15, 1.30, 1.45, 2 hours). Alternatively, he can point out which categories of items he needs to buy (fruit and vegetables, meat, pasta and rice...) and the system will calculate itself an estimated

permanence time. Additionally, if a user already visited the same store many times, the system can automatically define this time through the statistics of his previous visits.

When the reservation has been successfully completed, the user receives a QR Code which identifies it (to manage his access to the store and to retrieve information about the amount of time he stayed in the supermarket and how often) and which must be scanned to enter and leave the store. It enables to check how many people are in each store all the time. For the people who do not have access to the required technology, the stores hand out tickets also at the entrance. Even in this case the tickets present a QR Code that must be scanned in the same way, and the user must indicate how much time he will stay in the store and how many people there will be (1 or 2).

1.1.2 Goals

In this section the goals we aim to accomplish through the different functionalities we plan to implement are listed out.

NAME	BRIEF DESCRIPTION
G1.1	Allows OnlineUser to line up
G1.2	Allows PhysicalUser to line up
G1.3	Allows OnlineUser to book a slot of time
G2	Allows OnlineUser to visualize the map of the city with the stores
G3	Allows OnlineUser to see statistics about the stores
G4	The system gives suggestions about alternative stores when the chosen one is not available
G5	The system notifies the onlineUser when he should start coming closer to the supermarket
G6	The system notifies the onlineUser to remember him about his slot reservation and asks for a confirmation
G7	The system interacts with the Manager Service

NAME

BRIEF DESCRIPTION

G1.1	Allows OnlineUser to line up
G1.2	Allows PhysicalUser to line up
G1.3	Allows OnlineUser to book a slot of time
G2	Allows OnlineUser to visualize the map of the city with the stores
G3	Allows OnlineUser to see statistics about the stores
G4	The system gives suggestions about alternative stores when the chosen one is not available
G5	The system notifies the onlineUser when he should start coming closer to the supermarket
G6	The system notifies the onlineUser to remember him about his slot reservation and asks for a confirmation
G7	The system interacts with the Manager Service

1.1.3 Manager Service

The S2B has to interact with the manager service. It is a software developed separately and which is not dealt in this document. It has two different functions which are connected to the S2B: the first one is to register a store to the service, inserting all the mandatory details such as the store capacity, the departments, the opening hours; the second one is to monitor the store (further details on the section 2.2.6). The way the two systems interacts is briefly explained in the Design Document.

1.2 Scope

In this section, the phenomena related to the *machine* – which is the software to be developed – and to the *world* – which is the real environment in which CLup will be used – are enumerated. A phenomenon can be shared if it is controlled by the world and observed by the machine or vice versa.

1.2.1 World phenomena

NAME	BRIEF DESCRIPTION
WP1	User needs to go grocery shopping
WP2	OnlineUser takes a smartphone with himself
WP3.1	OnlineUser goes to the store with a reservation
WP3.2	PhysicalUser goes to the store without a reservation
WP4	Each store has a certain capacity to contain people
WP5	Costumer lines up out of the store
WP6	OnlineUser is interested about statistics on the stores
WP7	Manager is interested in complying with the Covid rules in his store

NAME

BRIEF DESCRIPTION

WP1	User needs to go grocery shopping
WP2	OnlineUser takes a smartphone with himself
WP3.1	OnlineUser goes to the store with a reservation
WP3.2	PhysicalUser goes to the store without a reservation
WP4	Each store has a certain capacity to contain people
WP5	Costumer lines up out of the store
WP6	OnlineUser is interested about statistics on the stores
WP7	Manager is interested in complying with the Covid rules in his store

1.2.2 Machine Phenomena

NAME	BRIEF DESCRIPTION
MP1	Generation of QR Code
MP2	Calculation of the estimated time to arrive to the chosen supermarket
MP3	Computation of the open and currently available stores
MP4.1	Esteem of the residence time inside the store for a usual OnlineUser
MP4.2	Esteem of the residence time inside the store through the analysis of the type of items that the OnlineUser needs to buy
MP5	Encryption of sensitive data
MP6	The system stores and reads data
MP7	The system retrieves data from the database

NAME

BRIEF DESCRIPTION

MP1	Generation of QR Code
MP2	Calculation of the estimated time to arrive to the chosen supermarket
MP3	Computation of the open and currently available stores
MP4.1	Esteem of the residence time inside the store for a usual OnlineUser
MP4.2	Esteem of the residence time inside the store through the analysis of the type of items that the OnlineUser needs to buy
MP5	Encryption of sensitive data
MP6	The system stores and reads data
MP7	The system retrieves data from the database

1.2.3 Shared phenomena

NAME	BRIEF DESCRIPTION
SP1	User shows the QR Code entering the supermarket
SP2	User shows the QR Code leaving the supermarket
SP3	The system sends a notification that invite him to come closer to the store
SP4	The system sends a notification that remind him about his booking
SP5	OnlineUser makes a reservation
SP6	OnlineUser deletes a reservation
SP7	The systems track the position of OnlineUsers
SP8	OnlineUser is able to look at the map and find the stores near to him

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

NAME	BRIEF DESCRIPTION
QR Code	It is a type of matrix barcode which contains information about a reservation
User	He is a generical costumer of the store
OnlineUser	He is a customer of the store who goes to the supermarket using CLup
PhysicalUser	He is a customer of the store who goes to the supermarket not using CLup
Observer Pattern	The observer pattern is a software design pattern in which an object, named the <i>subject</i> , maintains a list of its dependents, called <i>observers</i> , and notifies them automatically of any state changes, usually by calling one of their methods

1.3.2 Abbreviations

NAME	BRIEF DESCRIPTION
S2B	Software to be
Gn	Goal number n
WPn	World phenomena number n
SPn	Shared phenomena number n
MPn	Machine phenomena number n
Dn	Domain assumption number n
Rn	Functional requirement number n

1.3.3 Acronyms

NAME	BRIEF DESCRIPTION
API	Application Programming Interface
GPS	Global Positioning System

1.4 Revision history

Date	Version	Changelog
09/12/2020	1.0	First Release
22/12/2020	1.1	Second Release with final revision

1.5 Reference documents

- [1] Slide of the lectures
- [2] Specification document “R&DD Assignment A.Y. 2020-2021”
- [3] GOOGLE LLC, Google Maps Platform Documentation:
<https://developers.google.com/maps/documentation>
- [4] Firebase Cloud Messaging: <https://firebase.google.com/docs/cloud-messaging>

1.6 Document structure

The document is divided in five parts.

- **INTRODUCTION:** It gives an overview on the purpose and scope of the document, defining the main goals and phenomena and the definitions, acronyms, and abbreviations of the most used terms. It also contains the revision history and the reference documents to better underline how it has been developed.
- **OVERALL DESCRIPTION:** It contains scenarios and further details on the shared phenomena given through class diagrams and state charts. Moreover, it describes the major functions of the application and their constraints. There is also information about users and their main characteristics in order to clarify their needs. Finally, the domain assumption that can be deducted from the assignment are included.
- **SPECIFIC REQUIREMENTS:** it includes more details on all the aspect in section 2 which can be useful for the development team. It represents the core of the document as it contains the functional and non-functional requirements, described through the definition of the use cases and their associated sequence/activity diagrams. There are also described the requirements on the external interfaces and the design constraints.
- **FORMAL ANALYSIS USING ALLOY:** It consists of the model designed using Alloy and the corresponding metamodel generated from the code, in order to describe formally the application.
- **EFFORT SPENT:** It includes information about the number of hours each group member has worked for the development of the document

2. Overall Description

2.1 Product perspective

2.1.1 Scenarios

- **Scenario 1: BOOKING A SLOT**

Elisa and Gianmarco, a young nurse and a doctor, are a family with two little children and live in the center of Milan. Due to the Coronavirus emergency, their turns became longer and now they have not so much time to go grocery shopping. Trying to find a solution, they discover that CLup gives the possibility to book a slot of time to go shopping. This made them so much happy, because it finally provided them to go to the supermarket without any waste of time. In fact, through the reservation modality, they are now able to do the shopping while going back home, avoiding having to queue, in order to be back home to their children as soon as possible.

- **Scenario 2: AN INTERESTING SUGGESTION**

Marianna is a matricula of Politecnico di Milano who has just started her Master of Science. She has never lived in a big city before and, due to the Coronavirus emergency, she doesn't have the possibility to visit her new district. However, through CLup she became able to discover all the nearest supermarkets in order to find the less crowded one, as she is very careful and tries to meet as few people as possible until the emergency won't be over.

- **Scenario 3: THE ADVANTAGE OF A PHYSICAL USER**

Melania is an old lady who lives alone in her house in Florence. Due to the Coronavirus emergency, her family is not able to visit her anymore and is very concerned about the health of their lovely grandma. For the lady, even go grocery shopping has become difficult because of the huge amount of time she is forced to stand in line. Nevertheless, thanks to CLup now she is able to go to the store, get her ticket and go back home waiting for her turn, if the queue is very long. However, as a percentage of spaces is dedicated to the customers that do not have access to the CLup's required technology, the waiting time is always very short, and she is able to enter immediately the supermarket.

- **Scenario 4: THE SOLUTION WHEN TOO BUSY**

Antonio has been working in an important Bank in Turin for a few months. Since he is a new entry inside the place, a lot of office work is assigned to him and his performance is constantly evaluated by his coworkers. For this reason, Antonio often arrives home really late and since he starts working at 8 am, he never finds the time to go grocery shopping and always end up eating junk food during the week because he can't find the time to buy fresh fruits and vegetables to eat for lunch or dinner. However, with the CLup functionality that allows to book a visit, Antonio will be able to book a visit to the store next to the office and go grocery shopping in the slot of time which coincides with the 2-hours-break he has for lunch.

- **Scenario 5: HURRY UP FOR THE NAP**

Silvia lives in the countryside near Rome and has a young daughter which she daily brings to kindergarten in the city.

After school, the child is really tired but once a week Silvia needs to stop at the supermarket in order to do the grocery shopping. She chooses the grocery store in the center because it is more furnished and significantly cheaper than the small ones in the suburbs. Thanks to CLup, she is able to book a visit to the supermarket next to the school 15 minutes after the end of classes, avoid the queue which annoys the young child and be home in the shortest time possible. Moreover, thanks to the functionality of the app she is able to point out that she will be with another person when visiting and knowing that the number of people in the store is monitored makes her feels safer for the health of her daughter, given the current sanitary situation.

- **Scenario 6: A SECOND OPTION WHEN TIME IS UP**

Monica is the new maid of an esteemed family in Venice and she is new in the city. She had a busy working day in the house and suddenly remembers that she still has to go grocery shopping and be back home in time to cook dinner. She searches on the CLup app for the grocery store recommended to her by Mrs. Foscolo but when clicking for the line-up functionality the app informs her that the waiting time is almost of two hours. Fortunately, the woman is advised thanks to the additional functionality that there are other nearby stores in which she can line up with almost no waiting time. Eventually, she is happy to go in an equally well-stocked and less crowded supermarket and manages to complete all the household chores, included the shopping, in time.

2.1.2 UML Description

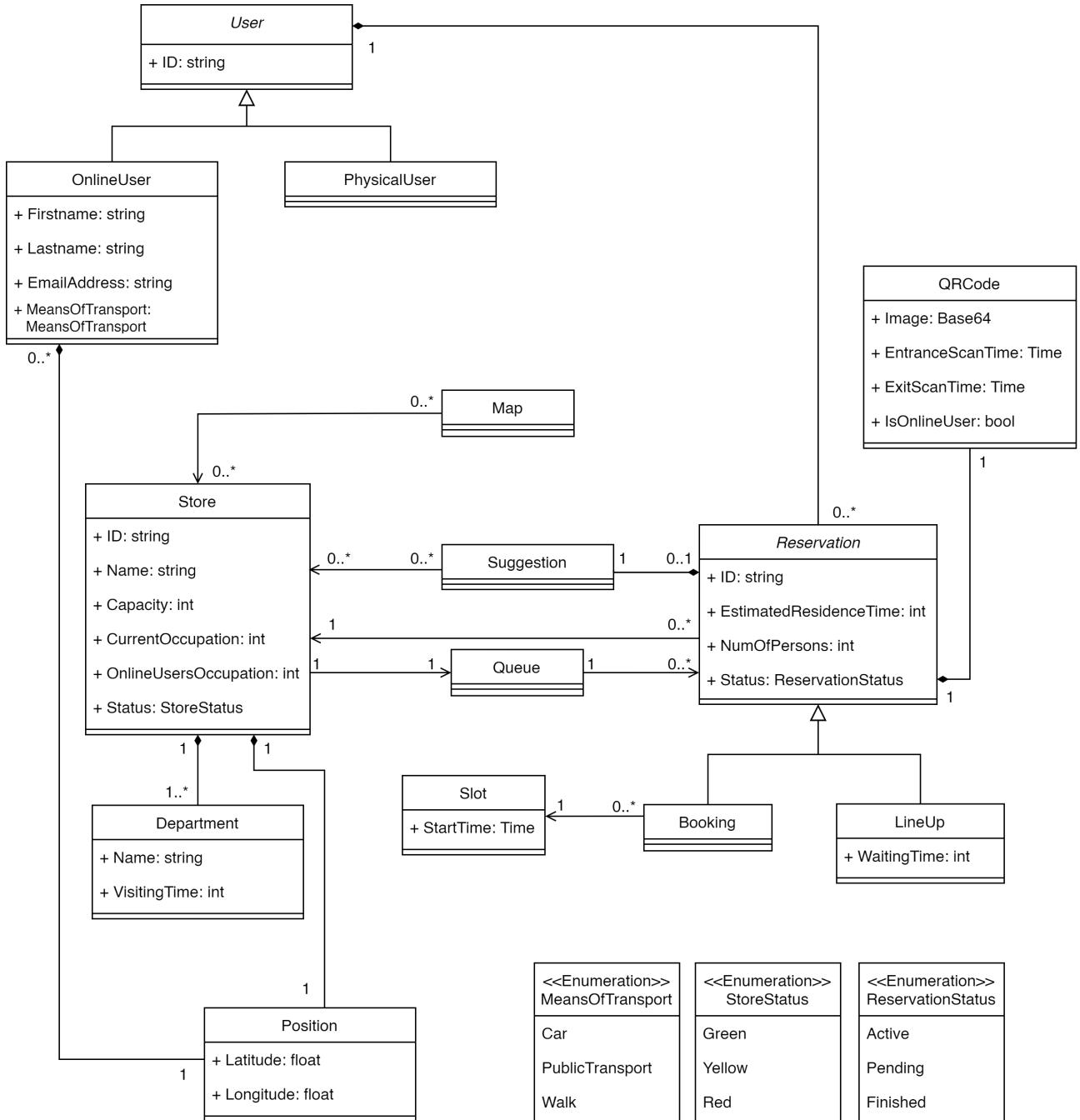


Figure 1 - UML 1: High-level Class Diagram

The UML diagram provided in Figure 1 provides a model of the application structure, focusing on the main classes and attributes of the system to be developed. It does not include every class that will be necessary to define the complete architecture of the system. Their roles will be further explored in the Design Document.

Down here there is a brief explanation of each class with its role and attributes. They have been underlined to a better readability and understanding.

- **User:** this class is the father of two possible kind of users, which are OnlineUser and PhysicalUser. The first one contains all the useful data about the user, which are acquired during the registration process and necessary for the login, his real-time position, and the means of transport that he chose on the map. Both of them contain the ID, which is a unique primary key for each user, and the set of reservations that has been made. Through this attribute it is possible to retrieve information to build statistics on user habits.
- **Store:** this class contains all the information about a store, as its name and the ID, which is a primary key for each store. First of all, there is its position, which is used to compute the distance from each onlineUser. Furthermore, the status represents the current status of the store working as a traffic light: it is green if the waiting time is lower than 1 hour, yellow if the waiting time is between 1 and 2 hours, red if the waiting time has reached a value of 2 hours or if the store is temporarily closed or it's going to close. Moreover, the capacity (which is a fixed value), currentOccupation (which must be update real-time and represents the number of customers currently into the store) and the onlineUsersCapacity (which represents the number of onlineUsers into the store) are used to calculate how many customers are able to enter the store and to build statistics. In addition, it contains the list of departments which are present in the store. Finally, the queue indicates the current customers' queue for the store and must be updated real-time.
- **Department:** this class represents a department of the store, with its name (ex: "fruit and vegetables", "frozen food section") and the estimatedTime, which is the approximate time to do grocery shopping in this specific section. This last attribute is essential for the estimation of the time that the user will spend inside the store; as a matter of fact, an additional computation in order to find the total time for the entire visit is also necessary.
- **Reservation:** It contains all the information about a reservation: the store which it is associated to, the estimatedResidenceTime (which is the approximated time that the onlineUser will spend inside the store) and the numOfPersons (the number of customers which will enter the store, a value between 1 and 2). They are all the data acquired during the reservation process. This class also contains the status, a description of the current state of the reservation, which can be active, pending or finished. Moreover, it is extended by the two types of reservations which can be made: Booking and LineUp. While for the first one it is essential to include the slot of time chosen for the visit, in the second one there is the waitingTime, an attribute that represents the estimated time left to enter the shop and that is updated by real-time checking the state of the queue.

- **Slot:** this class represent a slot of time which can be reserved. The only attribute is the startTime, that represents the starting time of the visit. Inserting also the ending time would be useless as it can be computed through the residence time. The type used for the attribute is Time which includes not only the hour but also the date on which the visit is booked.
- **Suggestion:** it contains a list of stores which represent the suggested ones when the chosen store is not available or too crowded.
- **Queue:** this class represent the queue associated to a certain store. For this reason, there is a one-to-one correlation between these two classes. It contains a list of reservations, which is updated real-time.
- **QRCode:** it indicates the QRCode that must be shown to enter and exit from the store during a specific reservation. It is unique for each reservation and it is identified through the image (encoded in base64), which contains information about the user and the reservation. The isOnlineUser boolean is used to distinguish if it is associated to a reservation made by an OnlineUser or by a PhysicalUser. The entranceScanTime and exitScanTime are the specific times at which the user has scanned the code.
- **Map and Position:** these two classes are important to understand how the application interacts and contains information coming from the real world

Moreover, Department has been associated through the composition to the Store because it has no sense (on a logical level) to exist if it is not referenced by its Store. The same argument holds respectively for Reservation and User, Suggestions and Reservation, QRCode and Reservation.

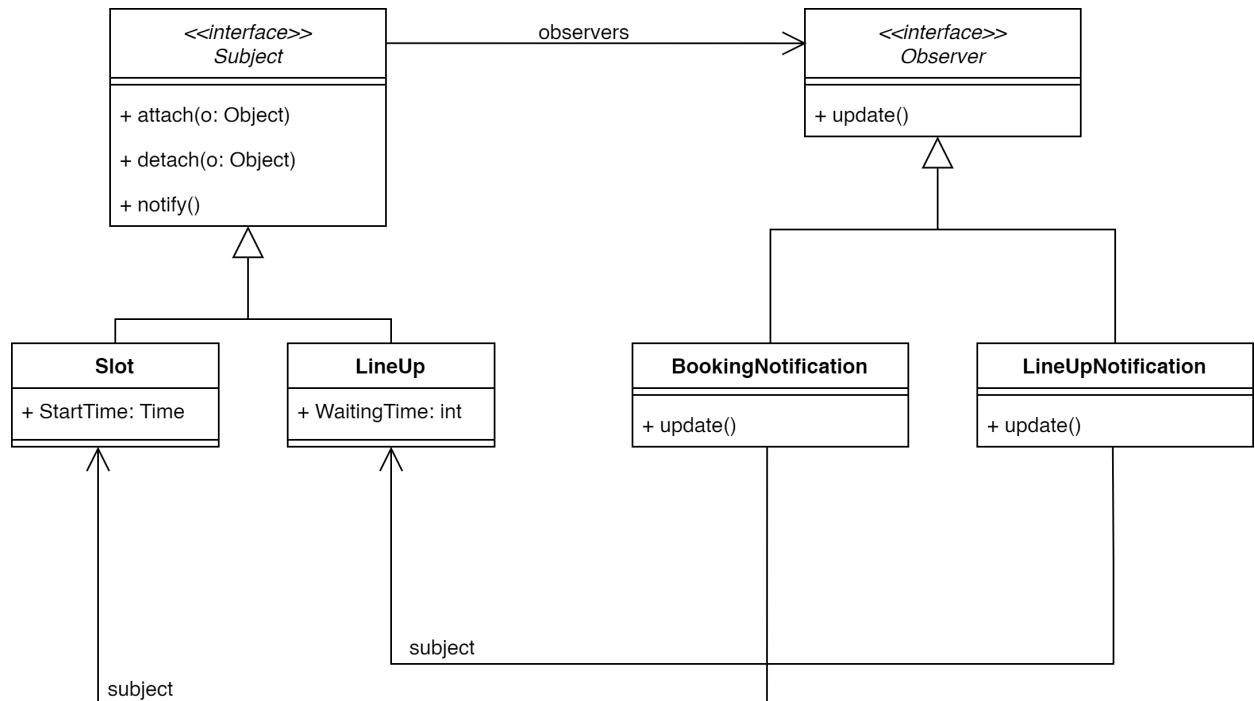


Figure 2 - UML 2: Observer pattern

In Figure 2, a UML diagram is illustrated to explain the observer pattern applied. There are two classes (**BookingNotification** and **LineUpNotification**) which are responsible to send an advice to the **OnlineUser** when the **StartTime** differs by two hours from the current time and when the **WaitingTime** becomes equal to the time to reach the chosen store. This pattern was introduced inside the application structure because automatically notifying the user is a good practice for a correct flow of events that the app has to take into consideration and control.

2.1.3 State Charts

Now we are going to analyze the most critical feature of the application - the reservation process - modelling its behavior and showing the evolution over time of its states through an appropriate state diagram, which are reported below.

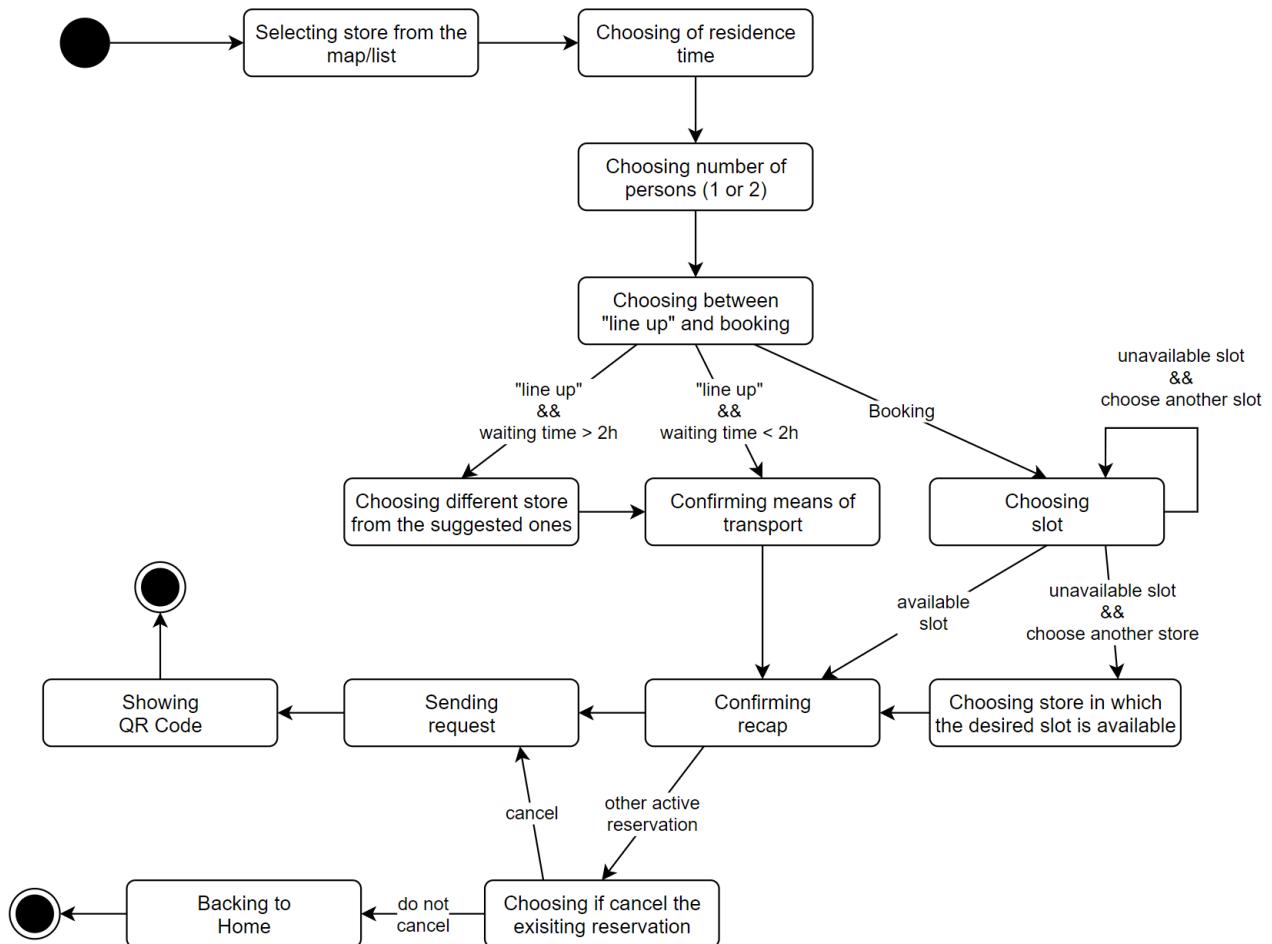


Figure 3 - State Diagram 1: creating a reservation

2.2 Product functions

In this section the main functionalities of CLUp are listed and explored in detail.

2.2.1 Line up

The main functionality of the app focuses on save people from having to line up and stand outside of stores for a huge amount of time. Line up represents the first alternative which is proposed to the user. When opening the app, the OnlineUser is able to visualize the map of a region and to select the store he wants to go to. At this point, after the insertion of the requested information (residence time and number of people who will come to the store), the OnlineUser can choose to line up. Through this action, it is possible for the OnlineUser to add himself to a virtual queue which has the purpose of substituting the physical one which would be formed outside the store. Therefore, as it were a real one, the OnlineUser is able to check through the app how many people are in front of him and the estimated time after which he will be able to enter the store. In this way, he does not have to lose his time waiting near the store, but he can come closer only when it is time. The app will help him by sending a notification when the estimated time of entrance coincides with the estimated time computed by the system to reach to the store from his current position.

2.2.2 Booking a visit

Booking a visit represents the second alternative which is proposed to the user. It follows the same mechanism of the first one, giving the user an additional benefit. As a matter of fact, while through the lineup choice the OnlineUser is put in a virtual queue and is not able to choose the exact time of entrance in the store, as it changes dynamically according to the people in the queue, through this second option the OnlineUser is able to select the specific time he wants to enter the store, choosing a slot of time that, once selected, does not change anymore. However, the first available time slot cannot be within the next two hours, so this option is not recommended to a customer who needs to go to the supermarket as soon as possible. In this case too, the OnlineUser will be reminded of the booking with a notification sent from the app two hours before the entrance time which will also ask him to confirm his presence.

2.2.3 Suggest alternatives when the desired store is not available

When selecting a store in the map, the OnlineUser is able to choose between all the displayed stores, as the booking option doesn't take care of the current waiting time. However, at the end of the procedure of the booking, in case the chosen store is not currently available for the line-up option, or if the chosen slot of time is not available anymore, the system suggests the OnlineUser an alternative store, selecting the one which is the nearest to the one previously chosen. This represents another helpful function of the application as it allows the OnlineUser to have multiple choices, providing a grocery store that the customer might have not taken in account and which might be less a crowded store.

2.2.4 Display of the stores' waiting time

When clicking the map to find the desired store, a useful functionality that the application gives to the OnlineUsers is the display of the current waiting time for all of the stores visible. As a matter of fact, in the visualization of the city map, each icon of a store can be colored in 3 different ways: green for availability with very low waiting time for the lineup; yellow for availability with medium waiting time before entering the store; red for unavailability (the store is temporarily closed or it is going to close) or for a waiting time close to 2 hours. By clicking on a store, the exact time (live) in minutes is displayed and updated constantly thanks to the monitoring of the queue done automatically by the system. In this way, the OnlineUser will be able to have a better customer experience, avoid long queues and pick the most suitable store for his needs and preferences.

2.2.5 Monitoring people inside a store

After the OnlineUser approaches the chosen store and scans the QR code, the application will keep track of some aspects of his visit to the shopping point. The application will check the time that the customer spends inside the store, compare it to the estimated time of residence indicated on the booking. The system, by tracking when the OnlineUser is inside, is able to give the exact number of persons in a store at any moment and show it to OnlineUsers which are using the application when clicking on a store icon on the map. All this information will be used by the application for future predictions on the behavior of a specific customer and to give updates to the other OnlineUsers waiting in the lineup or that want to book a visit.

2.2.6 Allow to monitor statistics

CLup is useful also to monitor various statistics, and this is possible thanks to the information retrieval about users and their habits.

- a) Users point-of-view: when selecting a store, the onlineUser is able to click the "Statistics" button and obtain various types of information about the store: the number of customers at any time, the percentage of occupied store, the days of the week when the store is more crowded. In this way they can choose to go grocery shopping in a more comfortable way, going to the store which is less crowded or in a less popular time.

All the information showed to the users are completely anonymized.

- b) Managers point-of-view: managers use a dedicated Web App to access some more detailed statistics about their store. These stats include all the ones described for the users, but also: influx in each department, best loyal customers (e.g., to provide dedicated discounts).

In this way, managers can also verify the safety of their store and be sure they adhere to anti-covid rules.

The Web App will be developed independently from the S2B and thus it will be not discussed in this document. The only part in common is the database where all the information is stored.

2.3 User characteristics

CLup is an application suitable for every person that has a mobile phone with an integrated GPS. The OnlineUsers have accepted all the terms and conditions imposed by the company during registration phase. They have received valid credentials to login and start searching for available stores. Moreover, as S2B will be easy to use, onlineUsers do not need to have any particular technical skill, except for being able to properly use the main functions of their phones.

2.3.1 Actors

Guest: a person who has the application on his device but still has to register; he cannot use any functionality of the application.

PhysicalUser: a person which is a customer of one of the stores available on the application but has not booked a visit or lined up online.

OnlineUser: a person who needs an effective and fast way to go grocery shopping, he has successfully registered and now owns an account with a valid username and a password. The OnlineUser has also accepted to give the application access to his current location.

System: it represents CLup and the software in a generic way.

Note that *Managers* are not included in the actors since they are not in the scope of the S2B and they interact through an external system.

2.4 Assumption, dependencies, and constraints

2.4.1 Domain assumptions

NAME	BRIEF DESCRIPTION
D1	If the store chosen by OnlineUser is currently not available, he searches for an alternative instead of going to the supermarket pretending to enter as a PhysicalUser
D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D3	If the smartphone of the OnlineUser has an integrated GPS sensor, it is always active while using CLup and is activated in background each five minutes
D4	About 80% of the customers goes to the supermarket through the use of CLup
D5	OnlineUser comes to the supermarket with the means of transportation that he selected on the app
D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
D7	A high percentage of OnlineUsers confirm their reservation when it is asked
D8	A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation

2.4.2 Dependencies

Since the S2B is mainly a mobile application, the main dependency is to have a smartphone, which must provide the following features:

1. Internet connection, possibly a mobile connection (3G/4G/5G) to be able to use the app even where there is no Wi-Fi. Note that the QR Code, once generated, will be cached and accessible offline, so an internet connection is not strictly required to approach the store and scan the code.
2. GPS sensor

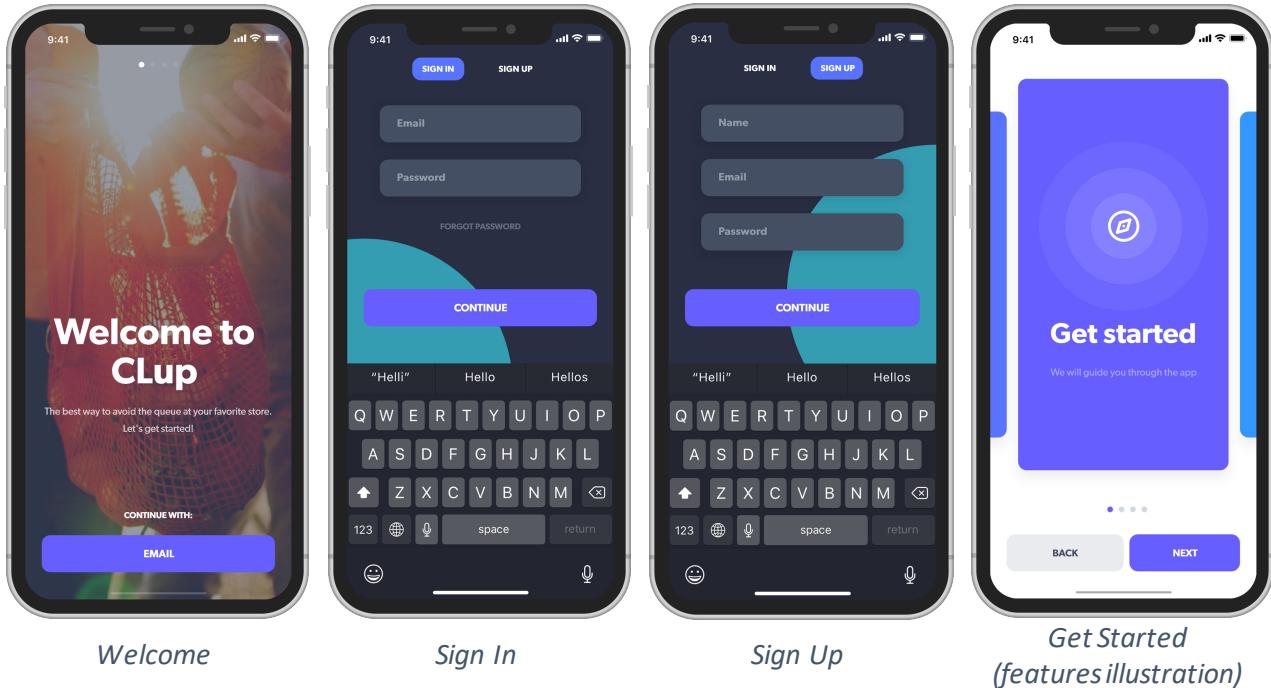
Also, there is the need to use some external software or APIs:

1. Map: the app will be dependent to some Maps API used to show the map and to calculate the travel time. An example of this service is Google Maps API [3].
2. Reverse Geocoding: the app will be dependent on some Maps API to get the full address, knowing the coordinates of location coming from the GPS of the device, or convert addresses to geographic coordinates. An example of this service is Google Maps API [3].
3. Notification Service: the app will use an external provider to send notifications to the devices. An example of this service is Firebase Cloud Messaging [4].

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

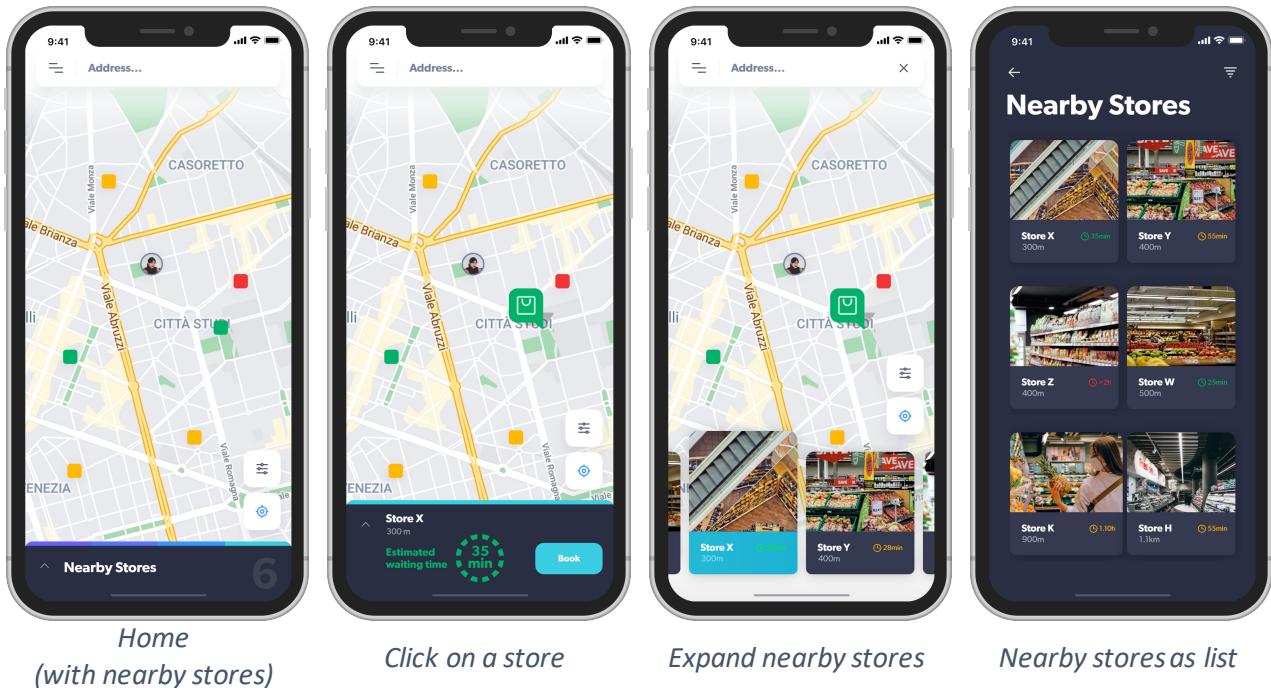


Welcome

Sign In

Sign Up

Get Started
(features illustration)

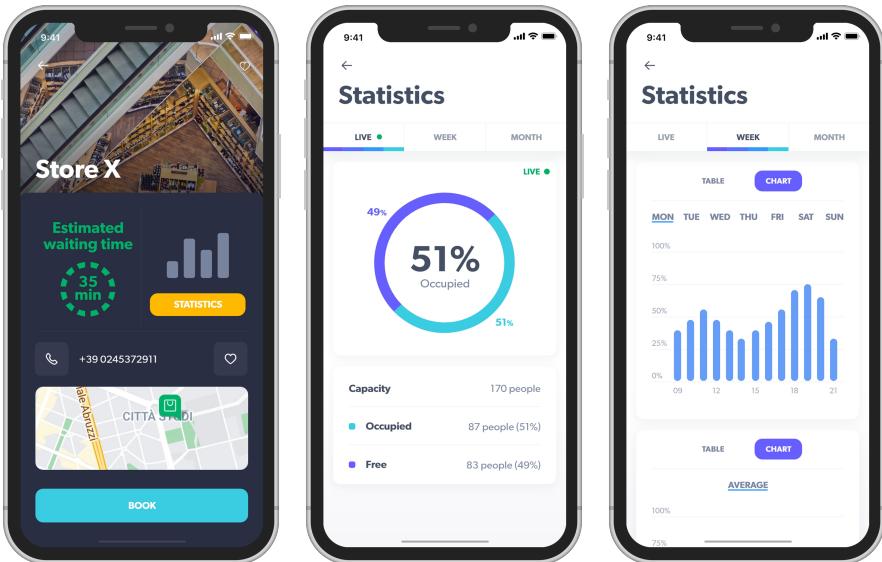


Home
(with nearby stores)

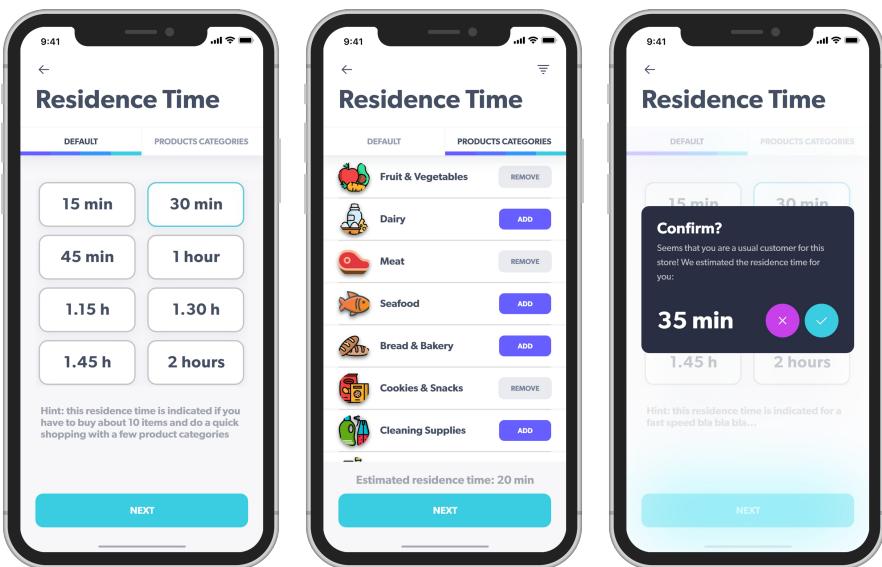
Click on a store

Expand nearby stores

Nearby stores as list

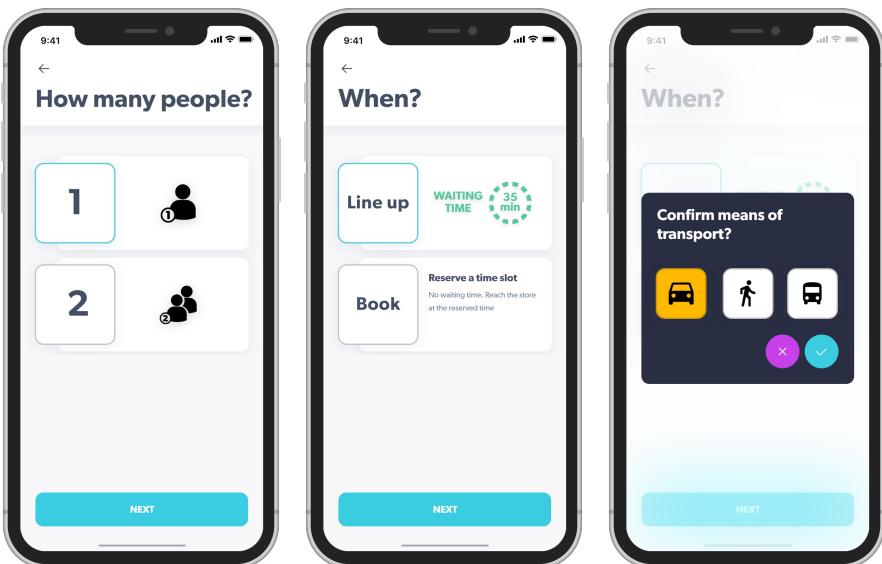


Store Details and Statistics:
Clicking on a store from the list of nearby stores or from the preview, will open the page about the store details. Here we can open the Statistics page or start the Booking process, other than visualize several info (phone number, estimated waiting time, address).

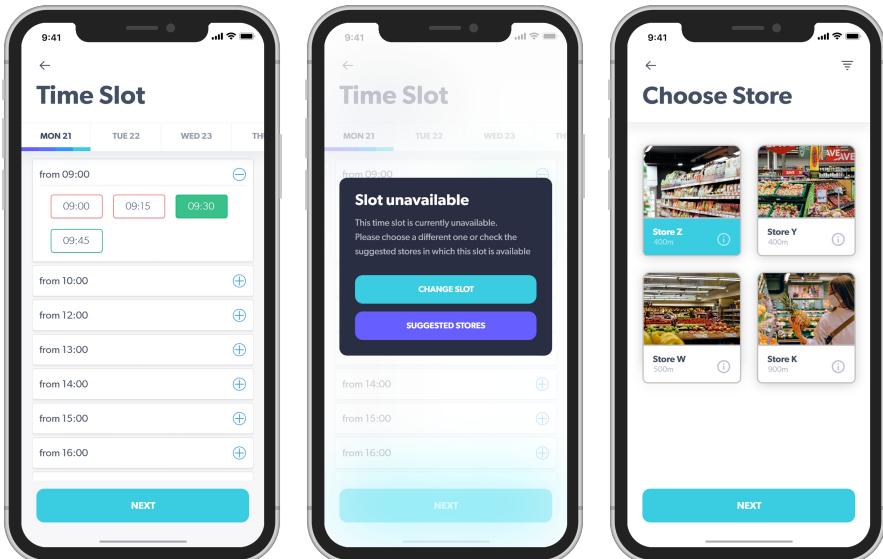


Reservation – Residence Time:
The first step of a reservation is the choice of the Residence Time.
There are three possibilities:

- Choice between predefined times
- Estimations through the choice of the products categories
- Automatic estimation for usual customers

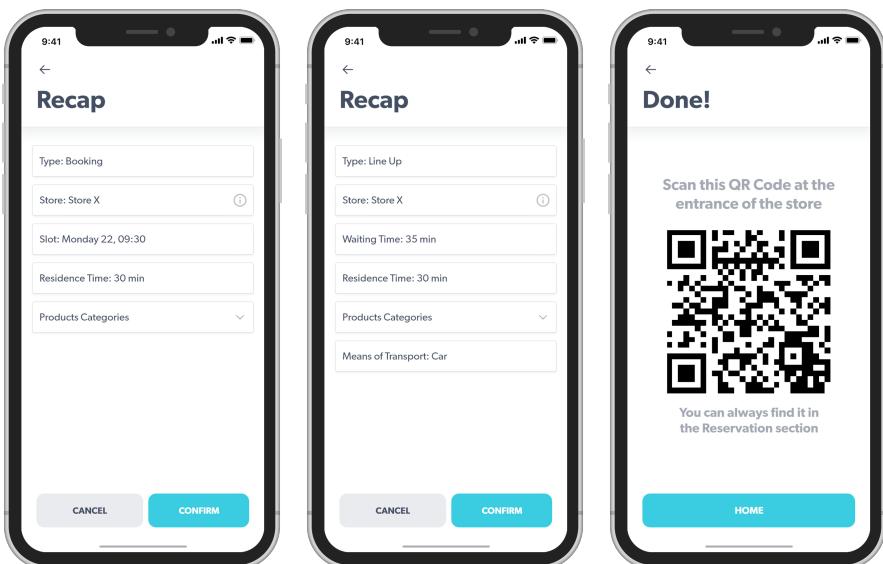


Reservation – How many, When, Means of transport:
The reservation process continues with the choice of how many people will take part in the shopping.
Then it is asked the choice between "Line Up" and "Booking".
If Line Up is chosen, then it will be asked to confirm the means of transport that will be used to reach the store.



Booking – Time Slot:

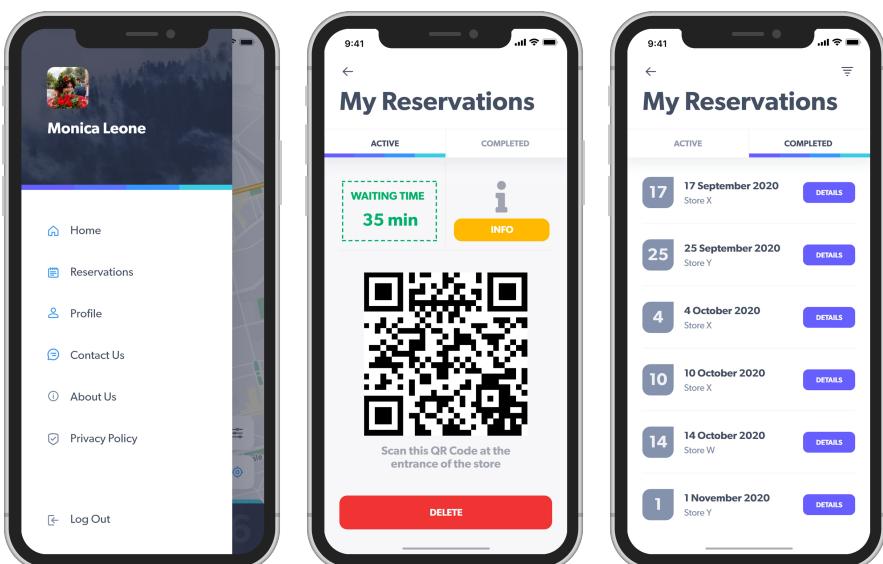
If, in the previous step, “Booking” has been chosen, then it will be asked to choose the desired Time Slot. Time slots will be in different colors according to their availability: red means that the slot is not available, green that it is available. The red slots can still be chosen, and in that case the system will ask to change the slot or to choose one of the suggested stores.



Reservation – Recap, Done:

At the end of the reservation process (Booking or Line Up), it will be shown the Recap with all the details of the reservation.

If the recap is confirmed, the QR Code associated to that reservation will be shown.



Menu, My Reservations:

From the home page, the menu can be opened. In it, several sections can be found.

In the Reservations section will be shown the Active Reservation (if any), with all the relative details, and the list of the completed reservations.

3.1.2 Hardware Interfaces

OnlineUsers and Guests are the only types of actors to whom CLup is addressed. Both need to own a smartphone with an integrated GPS and an Internet connection to make the app work properly.

It is not provided a web app, except for managers. Nevertheless, it is not the subject of the RASD, as it deals with different functions, and it will not be discussed in this document.

The stores that are registered to the service need to have at their entrance a machine which is able to scan a QRcode and send it to the system. Moreover, they must have a machine which is able to hand out tickets for the PhysicalUsers, where is possible to select the estimated residence time and the number of persons visiting the store (1 or 2). The printed ticket will contain the estimated waiting time and a QRcode that works in the same way as the ones provided by the app.

3.1.3 Software Interfaces

- CLup must interface with the smartphone's GPS via the operating system API in order to localize the OnlineUser position and show the according map.
- CLup must interface with a public API service that allows to retrieve a map given a GPS position/an address.
- CLup must interface with a public API service that allows to send push notifications to all the supported mobile OS.
- Since there are some external systems that need to interact with the S2B (i.e.: the QRCode scanner and the QRCode dispenser), the CLup backend will provide some dedicated API.

3.1.4 Communication Interfaces

The S2B will only make use of the usual communication protocols (TCP/IP) to guarantee the connection between the OnlineUsers' smartphones and the backend system.

The way information is exchanged over this protocol can be chosen among the different de-facto standard formats, and it will be discussed in the Design Document.

3.2 Functional Requirements

3.2.1 List of Requirements

R1.1	The onlineUser must be able to point out the duration of his visit by choosing a defined period of time
R1.2	The onlineUser must be able to point out the duration of his visit by selecting which categories of items he will buy
R1.3	The system must be able to calculate the duration of an onlineUser's visit by statistics on his previous visits
R2	The onlineUser must be able to see in real-time an estimated waiting time to enter for each store
R3	The onlineUser must be able to see in real-time the waiting time for his booking
R4	The onlineUser must be able to delete a booking within an hour from its start
R5.1	The onlineUser must be able to select the means of transportation
R5.2	The onlineUser must be able to change the means of transformation at any moment
R6	The onlineUser is not allowed to line up to a store where the current queue has a waiting time longer than two hours
R7	Information must be used to build statistics about waiting time, duration of visits and most busy times of the day
R8	The system must be able to count the number of people in each store in real-time
R9	The system must be able to compute the amount of time of each visit inside the store
R10	The internet connection of the OnlineUser must work properly
R11	The smartphone of the OnlineUser must have an integrated GPS sensor
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser
R13	The onlineUser is not allowed to book a new reservation if he already has an active one
R14	The onlineUser must be able to choose the number of people who will come to the store (1 or 2)
R15	The system must be able to retrieve all the information about each past reservation
R16	The onlineUser is not allowed to book a slot of time which is already full
R17	The onlineUser must be able to confirm his reservation when he receives the related notification
R18	The onlineUser is not allowed to book a slot for a time which is closer than two hours from the moment he makes the reservation
R19	The onlineUser is not allowed to book a slot for a time which is farther than seven days from the moment he makes the reservation
R20	The system must be able to send notifications to onlineUsers
R21	The smartphone of the OnlineUser must be able to receive notifications
R22	The system must be able to hand out physical tickets
R23	The system must be able to register a new store or modify an existing one

3.2.2 Mapping

Goals	Domain assumptions	Requirements
G1.1	D1, D2, D4, D5, D6, D7, D8	R1.1, R1.2, R1.3, R3, R4, R5.1, R5.2, R6, R8, R9, R10, R11, R12, R13, R14
G1.2	D2, D4, D7, D8	R8, R9, R22
G1.3	D1, D2, D4, D5, D6, D7, D8	R1.1, R1.2, R1.3, R4, R5.1, R5.2, R8, R9, R10, R11, R12, R13, R14, R16, R18, R19
G2	D2, D6	R2, R5.1, R10, R12
G3	D6	R7, R8, R9, R10, R15
G4	D1, D2, D4, D5, D6, D7, D8	R1.1, R1.2, R1.3, R5.2, R6, R8, R10, R11, R12, R16, R18, R19
G5	D3, D5	R10, R11, R12, R20, R21
G6	D7	R10, R17, R20, R21
G7		R7, R15, R23

The next tables represent the mapping of each goal with its description and the correspondent domain assumptions and requirements, in order to make the mapping easier to read and understand.

G1.1 Allows OnlineUser to line up

D1	If the store chosen by OnlineUser is currently not available, he searches for an alternative instead of going to the supermarket pretending to enter as a PhysicalUser
D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D4	About 80% of the customers goes to the supermarket through the use of CLup
D5	OnlineUser comes to the supermarket with the means of transportation that he selected on the app
D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
D7	A high percentage of OnlineUsers confirm their reservation when it is asked
D8	A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation
R1.1	The onlineUser must be able to point out the duration of his visit by choosing a defined period of time
R1.2	The onlineUser must be able to point out the duration of his visit by selecting which categories of items he will buy
R1.3	The system must be able to calculate the duration of an onlineUser's visit by statistics on his previous visits
R3	The onlineUser must be able to see in real-time the waiting time for his booking
R4	The onlineUser must be able to delete a booking within an hour from its start
R5.1	The onlineUser must be able to select the means of transportation
R5.2	The onlineUser must be able to change the means of transformation at any moment
R6	The onlineUser is not allowed to line up to a store where the current queue has a waiting time longer than two hours
R8	The system must be able to count the number of people in each store in real-time
R9	The system must be able to compute the amount of time of each visit inside the store
R10	The internet connection of the OnlineUser must work properly

R11	The smartphone of the OnlineUser must have an integrated GPS sensor
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser
R13	The onlineUser is not allowed to book a new reservation if he already has an active one
R14	The onlineUser must be able to choose the number of people who will come to the store (1 or 2)

G1.2 Allows PhysicalUser to line up

D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D4	About 80% of the customers goes to the supermarket through the use of CLup
D7	A high percentage of OnlineUsers confirm their reservation when it is asked
D8	A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation
R8	The system must be able to count the number of people in each store in real-time
R9	The system must be able to compute the amount of time of each visit inside the store
23R22	The system must be able to hand out physical tickets

G1.3 Allows OnlineUser to book a slot of time

D1	If the store chosen by OnlineUser is currently not available, he searches for an alternative instead of going to the supermarket pretending to enter as a PhysicalUser
D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D4	About 80% of the customers goes to the supermarket through the use of CLup
D5	OnlineUser comes to the supermarket with the means of transportation that he selected on the app
D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
D7	A high percentage of OnlineUsers confirm their reservation when it is asked
D8	A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation
R1.1	The onlineUser must be able to point out the duration of his visit by choosing a defined period of time
R1.2	The onlineUser must be able to point out the duration of his visit by selecting which categories of items he will buy
R1.3	The system must be able to calculate the duration of an onlineUser's visit by statistics on his previous visits
R4	The onlineUser must be able to delete a booking within an hour from its start
R5.1	The onlineUser must be able to select the means of transportation
R5.2	The onlineUser must be able to change the means of transformation at any moment
R8	The system must be able to count the number of people in each store in real-time
R9	The system must be able to compute the amount of time of each visit inside the store
R10	The internet connection of the OnlineUser must work properly
R11	The smartphone of the OnlineUser must have an integrated GPS sensor
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser
R13	The onlineUser is not allowed to book a new reservation if he already has an active one

R14	The onlineUser must be able to choose the number of people who will come to the store (1 or 2)
R16	The onlineUser is not allowed to book a slot of time which is already full
R18	The onlineUser is not allowed to book a slot for a time which is closer than two hours from the moment he makes the reservation
R19	The onlineUser is not allowed to book a slot for a time which is farther than seven days from the moment he makes the reservation

G2 Allows OnlineUser to visualize the map of the city with the stores

D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
R2	The onlineUser must be able to see in real-time an estimated waiting time to enter for each store
R5.1	The onlineUser must be able to select the means of transportation
R10	The internet connection of the OnlineUser must work properly
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser

G3 Allows OnlineUser to see statistics about the stores

D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
R7	Information must be used to build statistics about waiting time, duration of visits and most busy times of the day
R8	The system must be able to count the number of people in each store in real-time
R9	The system must be able to compute the amount of time of each visit inside the store
R10	The internet connection of the OnlineUser must work properly
R15	The system must be able to retrieve all the information about each past reservation

G4 The system gives suggestions about alternative stores when the chosen one is not available

D1	If the store chosen by OnlineUser is currently not available, he searches for an alternative instead of going to the supermarket pretending to enter as a PhysicalUser
D2	The user stands in the store for an amount of time which deviates by a maximum of five minutes from the one that has been indicated
D4	About 80% of the customers goes to the supermarket through the use of CLup
D5	OnlineUser comes to the supermarket with the means of transportation that he selected on the app
D6	When OnlineUser selects the types of items that he will buy, he sticks to this indication
D7	A high percentage of OnlineUsers confirm their reservation when it is asked
D8	A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation
R1.1	The onlineUser must be able to point out the duration of his visit by choosing a defined period of time
R1.2	The onlineUser must be able to point out the duration of his visit by selecting which categories of items he will buy
R1.3	The system must be able to calculate the duration of an onlineUser's visit by statistics on his previous visits

R5.2	The onlineUser must be able to change the means of transformation at any moment
R6	The onlineUser is not allowed to line up to a store where the current queue has a waiting time longer than two hours
R8	The system must be able to count the number of people in each store in real-time
R10	The internet connection of the OnlineUser must work properly
R11	The smartphone of the OnlineUser must have an integrated GPS sensor
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser
R16	The onlineUser is not allowed to book a slot of time which is already full
R18	The onlineUser is not allowed to book a slot for a time which is closer than two hours from the moment he makes the reservation
R19	The onlineUser is not allowed to book a slot for a time which is farther than seven days from the moment he makes the reservation

G5 The system notifies the onlineUser when he should start coming closer to the supermarket

D3	If the smartphone of the OnlineUser has an integrated GPS sensor, it is always active while using CLup and is activated in background each five minutes
D5	OnlineUser comes to the supermarket with the means of transportation that he selected on the app
R10	The internet connection of the OnlineUser must work properly
R11	The smartphone of the OnlineUser must have an integrated GPS sensor
R12	The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser
R20	The system must be able to send notifications to onlineUsers
R21	The smartphone of the OnlineUser must be able to receive notifications

G6 The system notifies the onlineUser to remember him about his slot reservation and asks for a confirmation

D7	A high percentage of OnlineUsers confirm their reservation when it is asked
R10	The internet connection of the OnlineUser must work properly
R17	The onlineUser must be able to confirm his reservation when he receives the related notification
R20	The system must be able to send notifications to onlineUsers
R21	The smartphone of the OnlineUser must be able to receive notifications

G7 The system interacts with the Manager Service

R7	Information must be used to build statistics about waiting time, duration of visits and most busy times of the day
R15	The system must be able to retrieve all the information about each past reservation
R23	The system must be able to register a new store or modify an existing one

3.2.3 Use Cases

3.2.3.1 Use Cases Description

1. Registration of an OnlineUser

Name	Registration of an OnlineUser
Actors	Guest
Entry Condition	1) Guest has downloaded the app onto the personal device 2) Guest has a working internet connection 3) Guest has not an account yet
Event Flow	1) Guest clicks on “Sign Up” button 2) Guest fills all the mandatory fields with personal information and clicks the Confirm button 3) The system checks the validity of the input 4) The system saves the data and sends a confirmation email
Exit Conditions	Guest has successfully completed the registration and becomes an OnlineUser
Exceptions	1) The Guest is already registered 2) The user did not fill all the mandatory fields 3) Username is already taken 4) Email is already linked to another account
	All the exceptions are handled by notifying the user and taking him back to the sign-up page.

2. Login of an OnlineUser

Name	Login of an OnlineUser
Actors	OnlineUser
Entry Condition	OnlineUser has previously successfully signed up to the application service
Event Flow	1) OnlineUser opens the app on the device 2) OnlineUser clicks on the “Login” button and inserts username and password 3) The system checks the validity of the input and sends back a session Token
Exit Conditions	OnlineUser is redirected to the home page of the App and can access all the functionalities of the application
Exceptions	1) The username inserted does not exist 2) The password inserted does not match the username
	All the exceptions are handled by notifying the user and taking him back to the login page.

3. Booking

Name	Booking
Actors	OnlineUser
Entry Condition	OnlineUser has already logged in the application
Event Flow	<p>1) OnlineUser clicks on a store on the interactive map</p> <p>2) OnlineUser:</p> <ul style="list-style-type: none"> 2.1) inserts the desired residence time between predefined intervals of time 2.2) inserts the categories of the items he needs to buy 2.3) accepts the suggested residence time or goes to 2.1) or 2.2) (<u>only</u> for regular customers) <p>3) OnlineUser inserts the number of person (max 2)</p> <p>4) The system calculates an estimated waiting time</p> <p>5) OnlineUser chooses between “Line up” and Booking:</p> <p><u>Line up:</u></p> <ul style="list-style-type: none"> 5.a.1) if waiting time < 2h, OnlineUser confirms the means of transport 5.a.2) if waiting time > 2h, OnlineUser chooses a different store from the suggested ones <p><u>Booking:</u></p> <ul style="list-style-type: none"> 5.b.1) OnlineUser chooses a time slot 5.b.2) If the time slot is not available, OnlineUser <ul style="list-style-type: none"> a. chooses a different slot b. chooses a different store from the suggested ones <p>6) OnlineUser confirms the recap</p> <p>7) If OnlineUser already has an active reservation, the system asks him if he wants to delete it and saves the new one</p> <p>8) System receives the request and generates the relative QR Code</p> <p>9) OnlineUser sees the QR Code</p>
Exit Conditions	OnlineUser booked successfully
Exceptions	<p>1) The internet connection is not available</p> <p>The above exception is handled by notifying the user with an error message</p> <p>2) In point 7), OnlineUser does not accept to overwrite the current active reservation</p> <p>The above exception is handled by bringing back the OnlineUser to the Home</p>

4. Load map

Name	Load map
Actors	OnlineUser
Entry Condition	1) OnlineUser has already logged in the application
Event Flow	<ul style="list-style-type: none"> 1) OnlineUser opens the app 2) The system loads the map of the last known position of the OnlineUser 3) OnlineUser inserts an address or enables the GPS position 4) The system loads the map of the given position
Exit Conditions	OnlineUser can see the map
Exceptions	<ul style="list-style-type: none"> 1) The internet connection is not available <p>This exception is handled by notifying the user with an error message</p>

5. Cancellation of a reservation

Name	Cancellation of a reservation
Actors	OnlineUser
Entry Condition	<ul style="list-style-type: none"> 1) OnlineUser has already logged in the application 2) OnlineUser has an active reservation
Event Flow	<ul style="list-style-type: none"> 1) OnlineUser clicks on “My Reservations” section 2) OnlineUser clicks on the active reservation 3) OnlineUser clicks on “Cancel Reservation” button 4) OnlineUser confirms the action
Exit Conditions	OnlineUser cancelled the reservation
Exceptions	<ul style="list-style-type: none"> 1) The internet connection is not available <p>This exception is handled by notifying the user with an error message</p>

6. Approach to the store Notification

Name	Approach to the store Notification
Actors	System
Entry Condition	1) The system detects the waiting time is almost equal to the time required to reach the store
Event Flow	1) The system sends a reminder notification to the OnlineUser to inform him that it is time to approach the store from its current position
Exit Conditions	OnlineUser receives the notification
Exceptions	<ul style="list-style-type: none"> 1) The internet connection is not available <p>This exception is handled showing the notification when the internet connection will be available (managed by the OS)</p>

7. Scan of the QR Code at the entrance of the store

Name	Entrance scan of the QR code
Actors	OnlineUser
Entry Condition	<ul style="list-style-type: none"> 1) OnlineUser has logged in 2) OnlineUser has correctly completed the booking procedure 3) OnlineUser must have his mobile device with him
Event Flow	<ul style="list-style-type: none"> 1) OnlineUser opens the “My Reservations” section and clicks on the active reservation 2) OnlineUser positions the device in front of the scanning machine 3) QR Code is recognized 4) The system checks the validity of the code and shows the user that the code was accepted with an OK message
Exit Conditions	OnlineUser successfully validated its reservation and can now enter the store
Exceptions	<ul style="list-style-type: none"> 1) QR code is not recognized 2) QR code is recognized but the time slot is incorrect <p>All the exceptions are handled by notifying the user with an error message</p>

8. Reminder Notification

Name	Reminder Notification
Actors	System
Entry Condition	<ul style="list-style-type: none"> 1) The system detects the reservation time is near
Event Flow	<ul style="list-style-type: none"> 1) The system sends a reminder notification to the OnlineUser <p>Three possibilities:</p> <ul style="list-style-type: none"> 2.a) OnlineUser confirms the reservation within 1h 2.b) OnlineUser choose, within 1h, to do not confirm the reservation 2.c) OnlineUser does not reply within 1h. Therefore, the system deletes the reservation
Exit Conditions	OnlineUser receives the notification
Exceptions	<ul style="list-style-type: none"> 1) The internet connection is not available <p>This exception is handled showing the notification when the internet connection will be available (managed by the OS)</p>

9. Scan of the QR Code at the exit of the store

Name	Exit scan of the QR code
Actors	OnlineUser
Entry Condition	<ul style="list-style-type: none"> 1) OnlineUser has logged in 2) OnlineUser must have his mobile device with him 3) OnlineUser must be inside the store 4) OnlineUser must have finished grocery shopping
Event Flow	<ul style="list-style-type: none"> 1) OnlineUser opens the app which automatically shows the QR Code of the active reservation (no other action is allowed) 2) OnlineUser positions the device in front of the scanning machine 3) QR code is recognized 4) The system checks the validity of the code and shows the user that the code was accepted with an OK message 5) OnlineUser clicks on the end button 6) The reservation ends successfully
Exit Conditions	OnlineUser ended its reservation and can now leave the store
Exceptions	<ul style="list-style-type: none"> 1) QR Code is not recognized <p>This exception is handled by notifying the user with an error message and the assistance of a clerk is required</p>

10. Choice of means of transport

Name	Choice of means of transport
Actors	OnlineUser
Entry Condition	<ul style="list-style-type: none"> 1) OnlineUser has logged in 2) OnlineUser is not currently inside the store with an active reservation
Event Flow	<ul style="list-style-type: none"> 1) OnlineUser opens the app 2) OnlineUser sees the home page with the map of the stores and the icons of the types of transports 3) OnlineUser clicks on the chosen type 4) The system computes the travel times to the stores 5) The new chosen means of transport becomes highlighted
Exit Conditions	OnlineUser has successfully changed his transportation
Exceptions	<ul style="list-style-type: none"> 1) OnlineUser clicks on the already selected means of transport <p>This exception does not lead to any change in the means of transport</p>

11.Check the state of an active reservation

Name	Check the state of an active reservation
Actors	OnlineUser
Entry Condition	<ul style="list-style-type: none">1) OnlineUser has already logged in the application2) OnlineUser has an active reservation
Event Flow	<ul style="list-style-type: none">1) OnlineUser clicks on “My Reservations” section2) OnlineUser clicks on the active reservation3) OnlineUser can see how many people are currently in front of him in the queue and the approximate time he will be able to enter the store
Exit Conditions	OnlineUser has read all the requested information
Exceptions	<ul style="list-style-type: none">1) The internet connection is not available <p>This exception is handled by notifying the user with an error message</p>

3.2.3.2 Use cases diagram

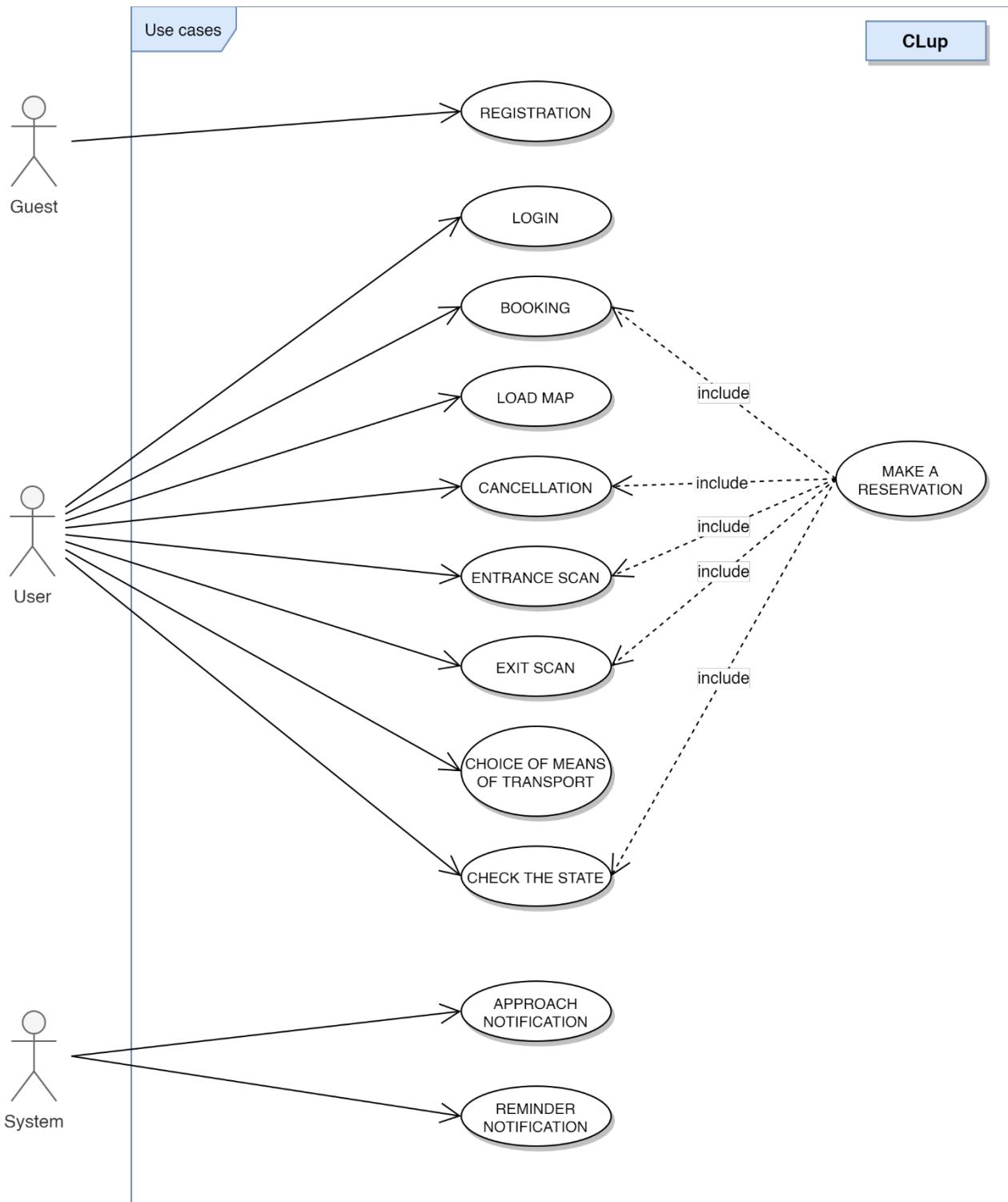


Figure 4 - Use cases diagram

3.2.4 Sequence diagrams

Note: to perform any operation, apart from Registration and Login, the user must be logged in. In the following sequences diagrams this requirement is implicit and so not represented.

1. Registration of an OnlineUser

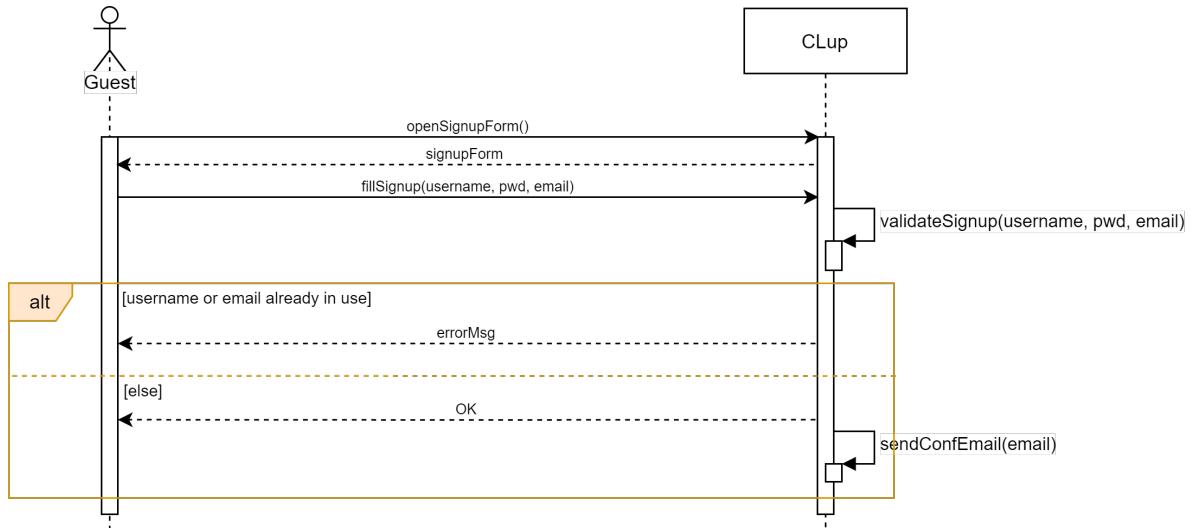


Figure 5 - Sequence Diagram: registration of an OnlineUser

2. Login of an OnlineUser

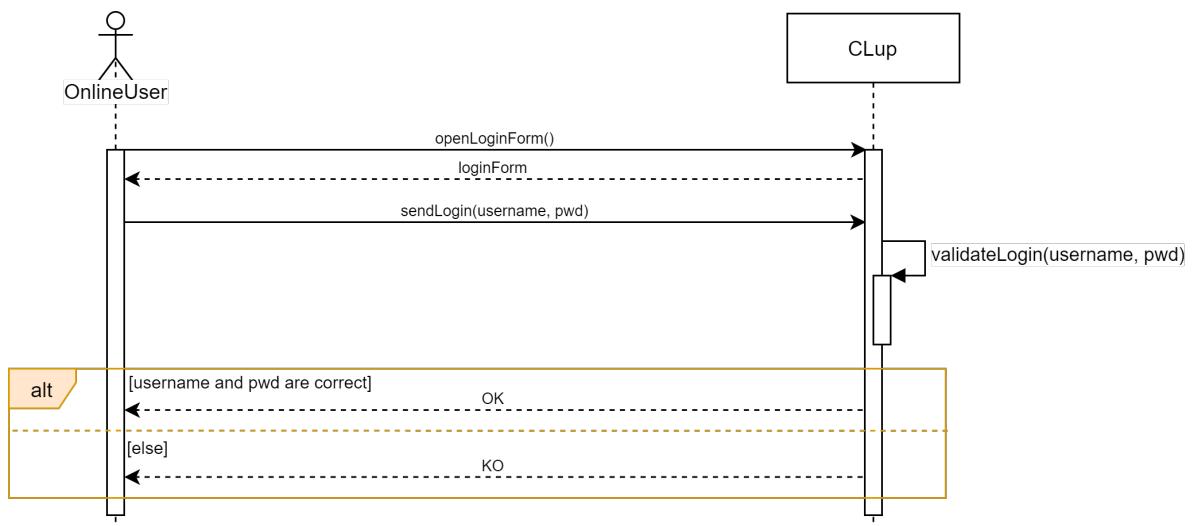


Figure 6 - Sequence Diagram: login of an OnlineUser

3. Booking



Figure 7 - Sequence Diagram: booking

4. Load map

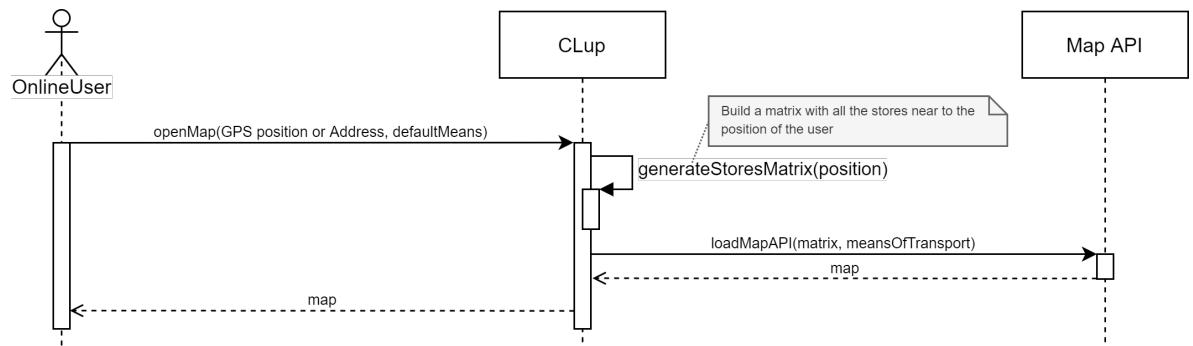


Figure 8 - Sequence Diagram: load map

5. Cancellation of a reservation

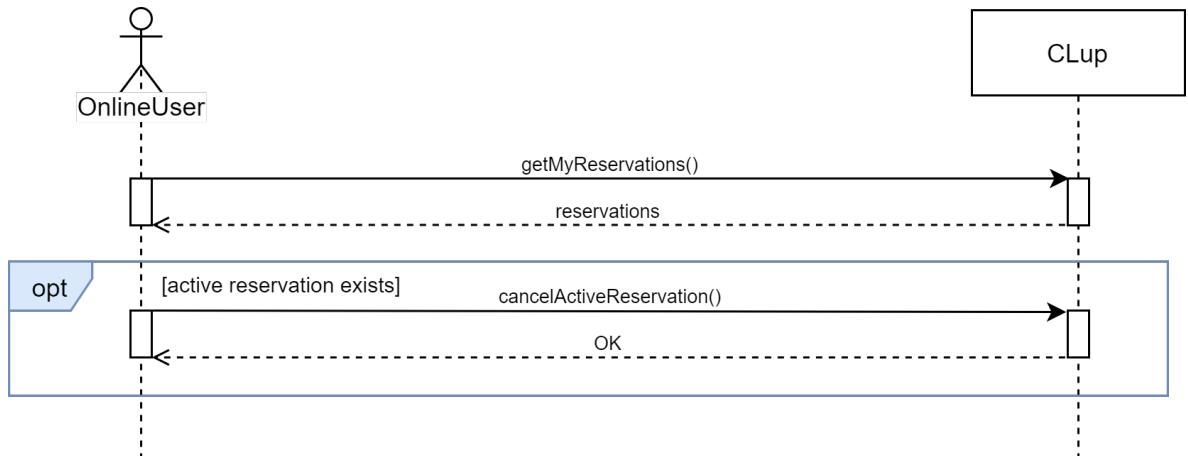


Figure 9 - Sequence Diagram: cancellation of a reservation

6. Approach to the store Notification

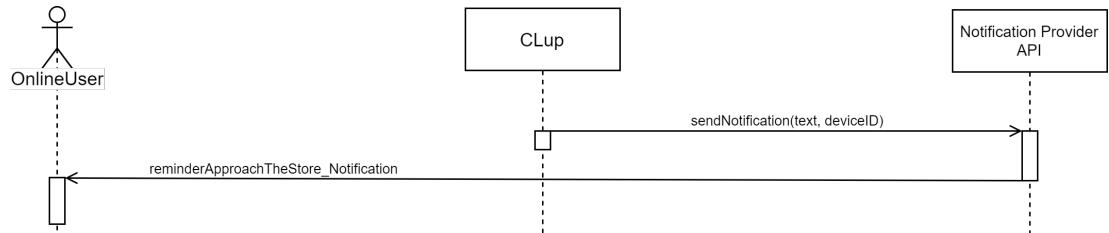


Figure 10 - Sequence Diagram: approach to the store Notification

7. Scan of the QR Code at the entrance of the store

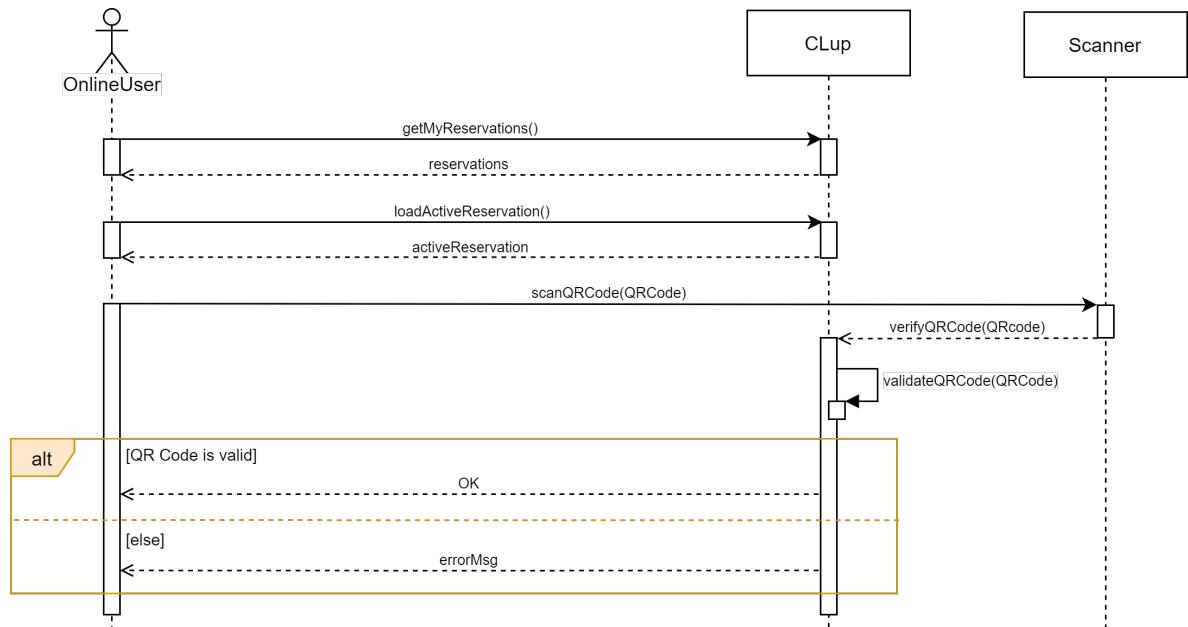


Figure 11 - Sequence Diagram: scan of the QR Code at the entrance of the store

8. Reminder Notification

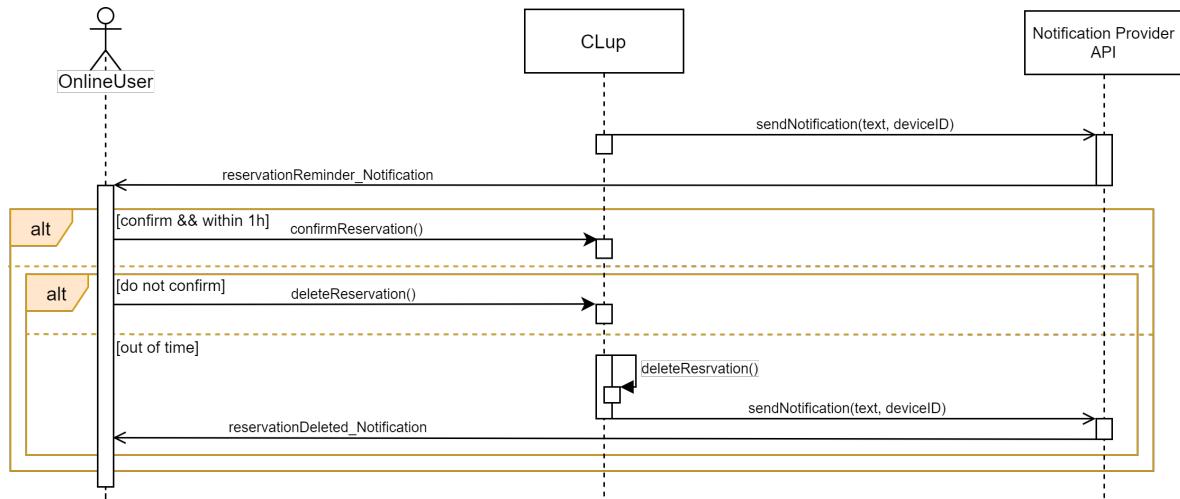


Figure 12 - Sequence Diagram: reminder Notification

9. Scan of the QR Code at the exit of the store

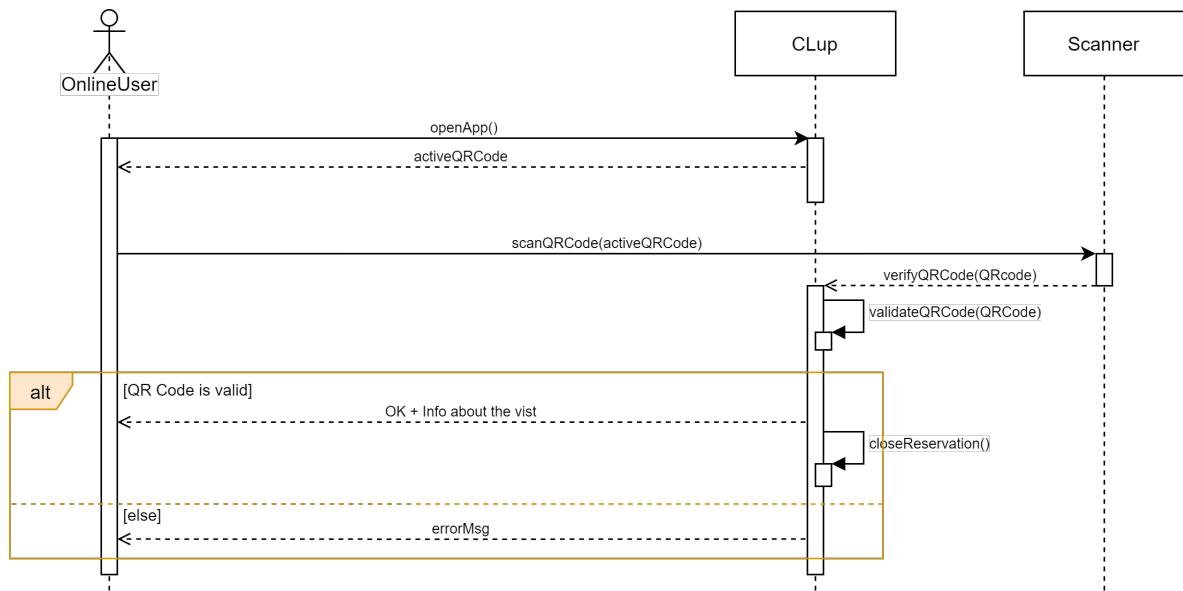


Figure 13 - Sequence Diagram: scan of the QR Code at the exit of the store

10. Choice of means of transport

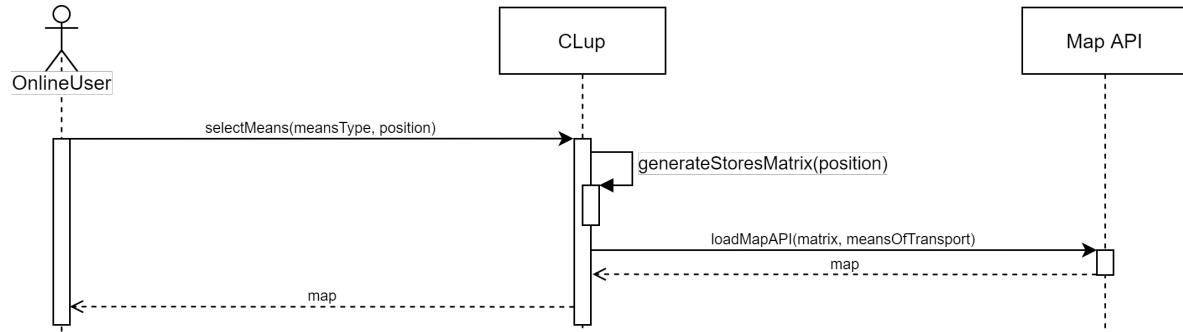


Figure 14 - Sequence Diagram: choice of means of transport

11. Check the state of an active reservation

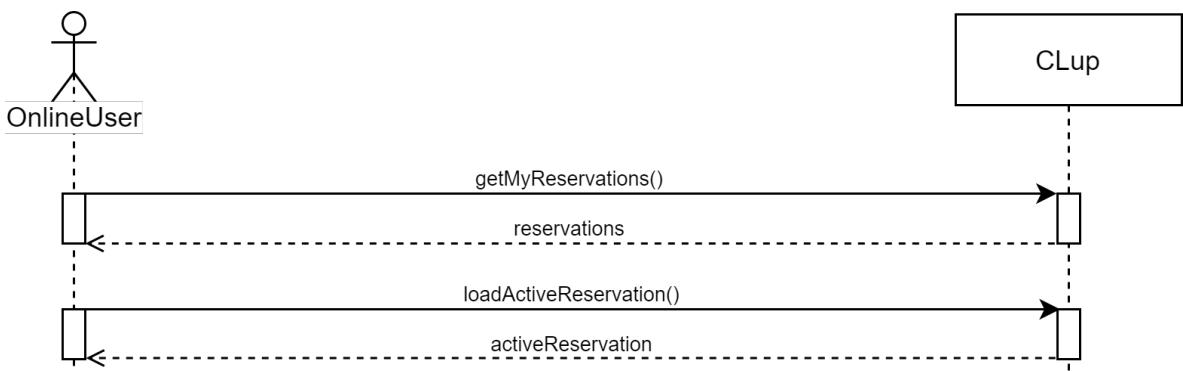


Figure 15 - Sequence Diagram: check the state of active reservation

3.3 Performance Requirements

The system must be able to work with simultaneous requests, also referred to the same sales point.

Performance is certainly a key aspect of the system, as it must be able to update in real-time the waiting time and the number of people for each store. Moreover, each query must be processed in a few seconds.

3.4 Design constraints

3.4.1 Standards compliance

The app should be available for the two main operating systems of smartphones: Android OS and Apple iOS.

3.4.2 Hardware limitations

As said in section 3.4.1, CLup app will be developer for the two main mobile OS (Android OS and iOS), thus the smartphone must run one of them.

Also, the smartphone requires an integrated GPS and a working internet connectivity.

3.5 Software system attributes

3.5.1 Reliability and Availability

The software should be available and functioning 24/7, so that the onlineUser can make a reservation at any given time.

However, since the booking process is a non-critical service, short downtime can be tolerated. On the other hand, the ability of the system to accept QR Code scans (i.e.: when a user enters or leaves a store) must have an availability in the order of 99.9% (corresponding to 1.44 minutes of downtime per day) during the usual opening time of the stores.

The system could accept long downtime (~ hours) during the night due to maintenance operations, which in any case should be scheduled and users should be notified.

In any case, if the server is not reachable, a warning must be displayed on the mobile app.

3.5.2 Security

The data stored in the system has to be encrypted in order to ensure that the onlineUsers' privacy is protected. The statistics provided to the onlineUsers about the stores are completely anonymous and no information about users can be retrieved from an external entity. However, some statistics shown to the store manager are not anonymous, such as the list of the most usual customer, useful to provided discount to them (users can choose to not take part in this initiative).

Also, CLup uses HTTPS for a secure communication between OnlineUsers and the Server.

3.5.3 Maintainability

The System will follow good software engineering practices to allow maintainability. There are some parameters, as the percentage of customers that the app should manage, which must be updated referring to the previous statistics. Through their progressive tuning the software will improve itself and will be able to manage in a finer way the users' flow.

3.5.4 Portability

The software must support both Android and iOS operating systems for mobile devices. The smartphone needs to allow the use of an internet connection. Portability from a device to another is possible by entering personal login data.

3.5.5 Scalability

As due to the Coronavirus emergency, people owning a smartphone are strongly recommended to use CLup to avoid very long queue and the number of users will constantly grow. For this reason, the system must be scalable without the necessity of reformulating core part of the software for a rising number of individuals.

4. Formal analysis using Alloy

This section contains the Alloy model which explains in more detail the features of the system, with a particular attention to some constraints. The code sequentially contains Signatures, Facts, Asserts, and Predicates.

```
open util/integer

/**** SIGNATURES ****/

sig Float {
    left: one Int,
    right: one Int
}

sig Position {
    latitude: one Float,
    longitude: one Float
}

abstract sig User {
    id: one Int,
    reservations: set Reservation
}{

    id > 0
}

sig PhysicalUser extends User {}

sig OnlineUser extends User {
    firstname: one Int,
    lastname: one Int,
    email: one Int,
    password: one Int,
    livePosition: one Position,
    meansOfTransport: one Int // 0 = car, 1 = walk, 2 = publicTransport
}{

    // in order to impose only 3 means of transport, we enumerated the three choices
    and imposed the range with a fact
    meansOfTransport >= 0 and meansOfTransport <= 2
}

sig Department {
    name: one Int,
    visitingTime: one Int
}{

    visitingTime > 0
}

sig Slot {
    startTime: one Time
}
```

```

sig QRCode {
    base64: one Int,
    entranceScanTime: one Time,
    exitScanTime: one Time
}

sig Suggestion {
    suggested: some Store
}

sig Queue {
    reservations: set Reservation
}

sig Store {
    name: one Int,
    id: one Int,
    position: one Position,
    capacity: one Int,
    queue: one Queue,
    currentOccupation: one Int,
    onlineUsersOccupation: one Int,
    departments: some Department
}{

    currentOccupation <= capacity // the current occupation of a store cannot exceed
    its capacity
    and currentOccupation >= 0 and capacity > 0
    and onlineUsersOccupation <= currentOccupation // the number of online users
    inside of a store has to be less than the total current occupation
    and id > 0
    // and onlineUsersOccupation <= div[mul[capacity, 80], 100] // 80% condition
    omitted to avoid long computation due to big integers
}

sig Time {
    year: one Int,
    month: one Int,
    day: one Int,
    h: one Int,
    m: one Int
}{

    // correct constraints (i.e.: year > 2019, month <= 12, etc.) not inserted to avoid
    long computation
    year > 0 and
    month > 0 and
    day > 0 and
    h > 0 and
    m >= 0
}

sig Map {}

```

```

abstract sig Reservation {
    id: one Int,
    store: one Store,
    estimatedResidenceTime: one Int,
    numberOfWorkers: one Int,
    qrCode: one QRCode,
    suggestions : lone Suggestion,
    status: one Int // 0 = active, 1 = pending, 2 = finished
}{

    id > 0 and estimatedResidenceTime > 0
    and numberOfWorkers <= 2 and numberOfWorkers > 0 // the users can visit the store
    with at most another companion
    and status >= 0 and status <= 2 // in order to impose only 3 status we enumerated
    the three choices and imposed the range with a fact
}

sig Booking extends Reservation {
    slot: one Slot
}

sig LineUp extends Reservation {
    waitingTime: one Int
}{

    waitingTime >=0
}

/**** FACTS ****/

// The times in all the QRCodes have to be plausible, in particular the
entranceScanTime has to anticipate the exitScanTime
fact checkQRCodeTime {
    all q: QRCode |
        q.entranceScanTime.year = q.exitScanTime.year and q.entranceScanTime.month =
        q.exitScanTime.month
        and q.entranceScanTime.day = q.exitScanTime.day
        and ((q.entranceScanTime.h = q.exitScanTime.h) implies q.entranceScanTime.m <
        q.exitScanTime.m else q.entranceScanTime.h < q.exitScanTime.h)
}
// Auxiliary function which returns -1 if t1<t2, 0 if t1=t2, 1 if t1>t2
fun compareTime(t1 : Time, t2 : Time) : one Int {
    t1.year < t2.year implies -1 else (
    t1.year > t2.year implies 1 else (
    t1.month < t2.month implies -1 else (
    t1.month > t2.month implies 1 else (
    t1.day < t2.day implies -1 else (
    t1.day > t2.day implies 1 else (
    t1.h < t2.h implies -1 else (
    t1.h > t2.h implies 1 else (
    t1.m < t2.m implies -1 else (
    t1.m > t2.m implies 1 else 0
))))))))))
}

```

```

// The entrance and exit time in all the reservations of a user cannot overlap
fact noOverlappingReservationsFromSameUser {
    all u : User, disj r1, r2 : Reservation |
        r1 in u.reservations and r2 in u.reservations implies (
            compareTime[r1.qrCode.entranceScanTime, r2.qrCode.exitScanTime] = 1
        or
            compareTime[r2.qrCode.entranceScanTime, r1.qrCode.exitScanTime] = 1
        )
}
}

// Two different reservations cannot have the same QRCode
fact twoDifferentReservationsNoSameQRCode {
    no disjoint r1, r2 : Reservation |
        r1.qrCode.base64 = r2.qrCode.base64
}

// Two different users cannot have the same email address
fact twoDifferentUsersNoSameEmail {
    no disjoint u1, u2 : User |
        u1.email = u2.email
}

// In the following three facts the unicity of the id for the Users, Stores and
// Reservations is imposed
fact twoDifferentUsersNoSameID {
    no disjoint u1, u2 : User |
        u1.id = u2.id
}

fact twoDifferentStoresNoSameID {
    no disjoint s1, s2 : Store |
        s1.id = s2.id
}

fact twoDifferentReservationsNoSameID {
    no disjoint r1, r2 : Reservation |
        r1.id = r2.id
}

// All the reservations in the queue of a store must be related to that specific store
fact sameStoreInAllReservationsInAQueue {
    all s : Store | all r : Reservation |
        r in s.queue.reservations implies r.store = s
}

fact physicalUsersCantDoBookings {
    all b : Booking | all p : PhysicalUser |
        b not in p.reservations
}

```

```

fact physicalUserCanHaveOnlyOneReservation {
    no p : PhysicalUser |
        #p.reservations > 1
}

// A User cannot book twice on the same slot of time
fact noMultipleBookingsForSameSlotByOneUser {
    all u : User | no disj b1, b2 : Booking |
        b1 in u.reservations and b2 in u.reservations and b1.slot = b2.slot
}

// All users cannot have more than one pending and/or active reservation simultaneously
fact oneUserOnlyOnePendingReservation {
    all u : User |
        no disj r1, r2 : Reservation |
            (r1.status = 1 and r2.status = 1) or (r1.status = 0 and r2.status = 1) or
            (r1.status = 0 and r2.status = 0)
                and r1 in u.reservations and r2 in u.reservations
}

// In a list of suggestions there cannot be the same store more than once
fact allSuggestedStoresAreDifferent {
    all s : Suggestion |
        no s1, s2 : Store |
            s1 = s2 and s1 in s.suggested and s2 in s.suggested
}

// A store cannot be a suggested alternative store for itself
fact storeNotPresentInItsSuggested {
    all r : Reservation | no s : Store |
        s in r.suggestions.suggested and s = r.store
}

// A department is to only one store
fact departmentUniqueForEachStore {
    all d : Department | all s1, s2 : Store |
        (d in s1.departments and d in s2.departments) implies s1=s2
}

// All stores have departments with different names
fact noTwoDepartmentsWithSameNameInSameStore {
    all s : Store | no disj d1, d2 : Department |
        d1 in s.departments and d2 in s.departments and d1.name = d2.name
}

// The next 3 facts define that a Position (intended as an object) can't be associated
// to more than an object between OnlineUsers and Stores
fact _1_aStoreAndAUserNoSamePosition {
    all u : OnlineUser | no s : Store |
        u.livePosition = s.position
}

```

```

fact _2_twoOnlineUsersNoSamePosition {
    no disj u1, u2 : OnlineUser |
        u1.livePosition = u2.livePosition
}

// Two different stores can't have the same location (which is different to say the
// same Position)
fact _3_twoStoresNoSamePositionAndLocation {
    no disj s1, s2 : Store |
        s1.position = s2.position
        or
        (s1.position.latitude.left = s2.position.latitude.left and
        s1.position.latitude.right = s2.position.latitude.right and
        s1.position.longitude.left = s2.position.longitude.left and
        s1.position.longitude.right = s2.position.longitude.right)
}

// In order to exist, a Position must be associated to a Store or to an OnlineUser
fact eachPositionMustBeAssociated {
    all p : Position |
        (one s : Store | s.position = p) or (one u : OnlineUser | u.livePosition = p)
}

// Two different stores cannot have the same queue
fact queueUniqueForEachStore {
    no disj s1, s2 : Store |
        s1.queue = s2.queue
}

// In order to exist, a department has to be associated to a store
fact eachDepartmentMustBeInAStore {
    all d : Department |
        one s : Store |
            d in s.departments
}

// In order to exist, a queue has to be associated to a store
fact eachQueueMustBeInAStore {
    all q : Queue |
        one s : Store |
            s.queue = q
}

// In order to exist, each QRCode has to be associated to a reservation
fact eachQRCodeMustBeAssociatedToReservation {
    all qr : QRCode |
        one r : Reservation |
            r.qrCode = qr
}

```

```

fact QRCodesMustBeUnique {
    no disj qr1, qr2 : QRCode |
        qr1.base64 = qr2.base64
}

// A reservation should always be in a queue
fact eachReservationMustBeInAQueue {
    all r : Reservation |
        one q : Queue |
            r in q.reservations
}

// A reservation has to be associated to a user
fact eachReservationMustBeAssociatedToAUser {
    all r : Reservation |
        one u : User |
            r in u.reservations
}

<*/**** ASSERTS ****/>

// Two different reservations that belong to a same queue cannot be for two different
stores
assert allReservationInAQueueAreForTheSameStore {
    all q : Queue |
        no disj r1, r2 : Reservation |
            r1 in q.reservations and r2 in q.reservations and r1.store != r2.store
}

// Number of positions should be equal to the number of Stores+OnlineUsers
assert PositionsEqualToOnlineUsersAndStores {
    #Position = add[#OnlineUser, #Store]
}

// Number of QRCodes should be equal to the number of reservations
assert EqualQRCodesAndReservations {
    #QRCode = #Reservation
}

<*/**** PREDICATES ****/>

Pred buildBasicWorld {
    all s : Store | #s.departments = 1
}

pred buildComplexWorld {
    all s : Store | #s.departments >= 2
}

pred error {
    one u : OnlineUser, disj r1, r2 : Reservation |
        r1 in u.reservations and r2 in u.reservations and r1.status = 1 and r2.status = 1
}

```

```

***** CHECKS ****/

check allReservationInAQueueAreForTheSameStore for 2
check PositionsEqualToOnlineUsersAndStores
check EqualQRCodesAndReservations

***** RUN *****

run buildBasicWorld for 2 but exactly 1 OnlineUser, exactly 1 Store, exactly 1 Booking
run buildComplexWorld for 10 but exactly 2 OnlineUser, exactly 1 PhysicalUser, exactly 2 Store, exactly 2 LineUp, exactly 3 Booking
run error for 8 but exactly 1 OnlineUser, exactly 1 Store, exactly 0 LineUp, exactly 0 PhysicalUser, exactly 3 Booking

```

```

Executing "Check allReservationInAQueueAreForTheSameStore for 2"
Solver=sat4j Bitwidth=4 MaxSeq=2 SkolemDepth=1 Symmetry=20
17091 vars. 930 primary vars. 51185 clauses. 139ms.
No counterexample found. Assertion may be valid. 90ms.

Executing "Check PositionsEqualToOnlineUsersAndStores"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
29187 vars. 1434 primary vars. 866667 clauses. 237ms.
No counterexample found. Assertion may be valid. 131ms.

Executing "Check EqualQRCodesAndReservations"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
29168 vars. 1434 primary vars. 86587 clauses. 337ms.
No counterexample found. Assertion may be valid. 323ms.

Executing "Run buildBasicWorld for 2 but exactly 1 OnlineUser, exactly 1 Store, exactly 1 Booking"
Solver=sat4j Bitwidth=4 MaxSeq=2 SkolemDepth=1 Symmetry=20
14423 vars. 726 primary vars. 43673 clauses. 89ms.
Instance found. Predicate is consistent. 282ms.

Executing "Run buildComplexWorld for 10 but exactly 2 OnlineUser, exactly 1 PhysicalUser, exactly 2 Store, exactly 2 LineUp, exactly 3 Booking"
Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
86794 vars. 3215 primary vars. 268948 clauses. 593ms.
Instance found. Predicate is consistent. 38470ms.

5 commands were executed. The results are:
#1: No counterexample found. allReservationInAQueueAreForTheSameStore may be valid.
#2: No counterexample found. PositionsEqualToOnlineUsersAndStores may be valid.
#3: No counterexample found. EqualQRCodesAndReservations may be valid.
#4: Instance found. buildBasicWorld is consistent.
#5: Instance found. buildComplexWorld is consistent.

```

Figure 16 – Alloy result

One additional predicate was implemented to check the correct behavior of the model. As a matter of fact, no instances were found when running the predicate error which had one OnlineUser with two reservations both pending, which should not be allowed in our model.

```

Executing "Run error for 8 but exactly 1 OnlineUser, exactly 1 Store, exactly 0 LineUp, exactly 0 PhysicalUser, exactly 3 Booking"
Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
53035 vars. 2182 primary vars. 161450 clauses. 585ms.
No instance found. Predicate may be inconsistent. 13ms.

```

Figure 17 – Alloy: “error” predicate result

The next two illustrations show two possible representations of the implemented model. In the first one, there is just one instance for each of the main entities in order to have a basic visualization of the design and correct functioning. In the second, the world is more complete, and it is possible to see the correct behavior of the model in a more complex situation.

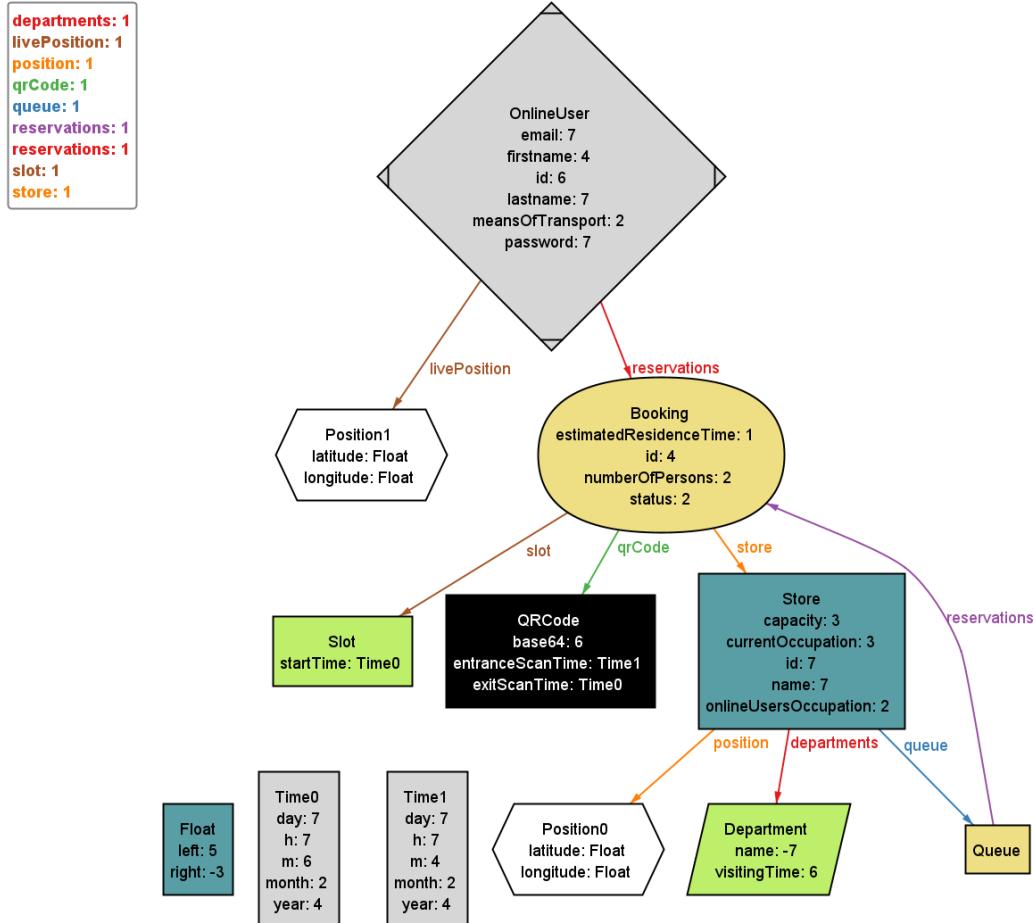


Figure 18 – Alloy: basic world

In the basic world represented above there is just one OnlineUser, one of the main actors described in Section 2.3.1.

The OnlineUser is situated in a position and owns a reservation of type Booking. The Booking is related to a single Store, situated in a different position from the one of the OnlineUser, and it owns one department and a queue which contains the Booking. To the booking is also associated a unique QRCode and a Slot time.

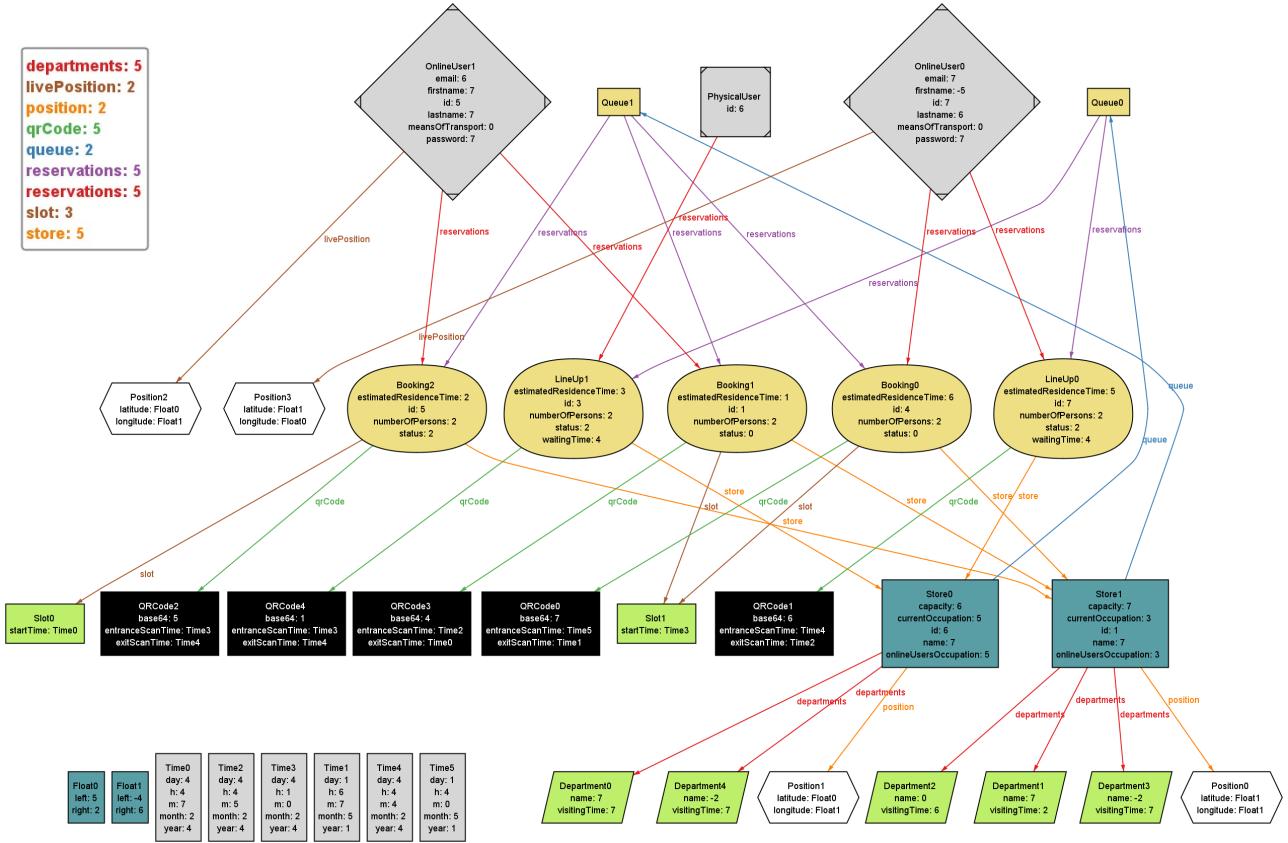


Figure 19 – Alloy: complex world

In Figure 19, a possible world generated using the Alloy verification software is illustrated. In it, the unique PhysicalUser is linked only to a LineUp type of reservation, since it cannot access to the Booking functionalities, while the two OnlineUsers respectively possess two disjoint groups of Booking and LineUp reservations. There is no more than one active or pending reservation associated to a user, visible through the status attribute.

Moreover, the number of QRCodes is equal to the total number of reservations since each of them is unique, and this can be verified checking the uniqueness of the base64 attribute. By looking at the two queues, it is possible to see that the groups of reservations they contain are disjoint. Regarding the time, Slot1 is linked to two reservations which are done by two different OnlineUsers, since a singular user could not have booked twice the same slot.

All the constraints concerning the validity and correct order of the times are respected.

About the two stores, they are in two different positions and are linked respectively to two and three departments, which are all different between each other. Additionally, the occupation of the stores caused by the OnlineUsers is correctly lower than the total currentOccupation to always give space also to some PhysicalUsers.

Some of the numerical values used in the illustration are different from the real world to allow a correct execution and avoid long computation in the tool without any loss of general meaning (e.g.: plausible year).

5. Effort spent

5.1 Galzerano Arianna

HOURS	TASK
2	Initial discussion on first part
1.30	Use cases
1	Sequence diagrams
0.30	Goals
1.30	Brainstorming
1.30	Product functions and user characteristics
1.30	Scenarios
2.30	List of requirements and mapping
4.30	UML
7.30	Alloy
2.30	Final revision

5.2 Lampis Andrea

HOURS	TASK
2	Initial discussion on first part
2	State chart
2	Use cases
2.30	Sequence diagram
1.30	Brainstorming
2.30	List of requirements and mapping
3	UML
12	Mockup
7.30	Alloy
0.30	Final revision

5.3 Leone Monica

HOURS	TASK
2	Initial discussion on first part
3	Introduction
1	Use cases and domain assumptions
1.30	Brainstorming
0.30	Sequence diagrams
1	Product functions
0.30	Scenarios
2.30	List of requirements and mapping
2.30	Mapping tables, Performance requirements, Design constraints, Software system attributes
5.30	UML
6.30	Alloy
1	Hardware interfaces, Software interfaces, Communication interfaces
2.30	Final revision