

SOFTWARE ENGINEERING 2 PROJECT

Requirement Analysis and Specification Document

**CLup - Customer Line-up**

**Version 1**

*Authors Supervisor*

Galzerano Arianna Dr Di Nitto Elisabetta  
Lampis Andrea  
Leone Monica

**Contents**

1. **Introduction**
   1. Purpose
      1. Description of the given problem
      2. Goals
   2. Scope
      1. World phenomena
      2. Machine phenomena
      3. Shared phenomena
   3. Definitions, Acronyms, Abbreviations
      1. Definitions
      2. Acronyms
      3. Abbreviations
   4. Revision History
   5. Reference Documents
   6. Document Structure
2. **Overall Description**
   1. Product perspective
      1. UML Description
      2. State Charts
   2. Product functions
      1. Line up
      2. Booking a visit
      3. Suggest alternatives when the chosen store is not available
   3. User characteristics
   4. Assumptions, dependencies, and constraints
      1. Domain assumptions
      2. Dependencies
3. **Specific Requirements**
   1. External Interface Requirements
      1. User Interfaces
      2. Hardware Interfaces
      3. Software Interfaces
      4. Communication Interfaces
   2. Functional Requirements
      1. List of Requirements
      2. Mapping
      3. Use Cases
         1. Use Cases Description
         2. Use Cases Diagram
      4. Sequence Diagrams
      5. Scenarios
   3. Performance Requirements
   4. Design Constraints
      1. Standards compliance
      2. Hardware limitations
      3. Any other constraint
   5. Software System Attributes
      1. Reliability
      2. Availability
      3. Security
      4. Maintainability
      5. Portability
      6. Scalability
4. **Formal Analysis Using Alloy**
5. **Effort Spent**
6. **References**
7. **Introduction**
   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. **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 a few 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 store where the chosen slot is still available.

In both cases 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 booking has been successfully completed, the user receives a QR Code which identifies the reservation (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.

* + 1. **Goals**

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

|  |  |
| --- | --- |
| NAME | BRIEF DESCRIPTION |
|  | Allows OnlineUser to line up |
|  | Allows OnlineUser to book a slot of time |
|  | Allows OnlineUser to visualize the map of the city with the stores |
|  | Allows OnlineUser to see statistics about the stores |
|  | The system gives suggestions about alternative stores when the chosen one is not available |
|  | The system notifies the onlineUser when he should start coming closer to the supermarket |
|  | The system notifies the onlineUser to remember him about his slot reservation and asks for a confirmation |

* 1. **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. **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 booking |
| WP3.2 | PhysicalUser goes to the store without a booking |
| WP4 | Each store has a certain capacity to contain people |
| WP5 | Costumer line up out of the store |

* + 1. **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 |

* + 1. **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 make a booking |
| SP6 | OnlineUser delete a booking |

* 1. **Definitions, Acronyms, Abbreviations**
     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 *subject*, maintains a list of its dependents, called *observers*, and notifies them automatically of any state changes, usually by calling one of their methods |

* + 1. **Acronyms**

|  |  |
| --- | --- |
| NAME | BRIEF DESCRIPTION |
| API | Application Programming Interface |
| GPS | Global Positioning System |
| ToComplete | To complete |
| ToComplete | To complete |
| ToComplete | To complete |

* + 1. **Abbreviations**

|  |  |
| --- | --- |
| NAME | BRIEF DESCRIPTION |
| S2B | Software to be |
| ToComplete | To complete |
| ToComplete | To complete |
| ToComplete | To complete |
| ToComplete | To complete |

* 1. **Revision history**

To complete

* 1. **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>
   1. **Document structure**

The document is divided in six parts.

* **INTRODUCTION**: It gives an overlooking 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
* **REFERENCES**: It shows the reference documents.

1. **Overall Description**
   1. **Product perspective**
      1. **UML Description**

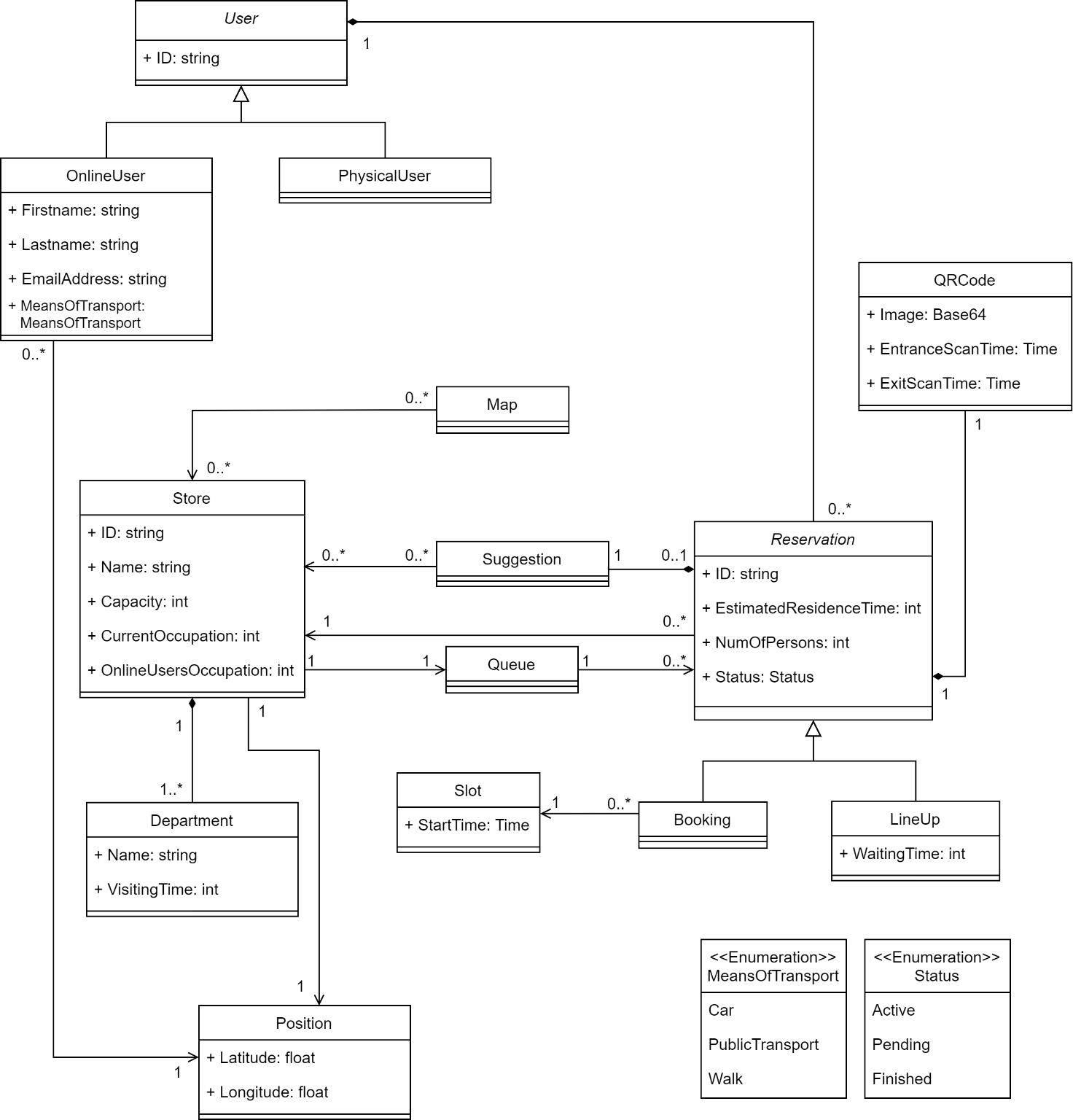


Figure 1 - UML 1: High-level Class Diagram

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.

* **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.
* **Store**: this class contains all the information about a store. First of all, there is its position, which is used to compute the distance from each onlineUser. Moreover, the capacity (which is a fixed value) and currentOccupation (which must be update real-time) are used to calculate how many customers are able to enter the store and to build statistics. Finally, it contains the list of departments which are present in the store.
* **Department**: this class represents a department of the store, with its name (ex: “fruit and vegetables”, “frozen food section”) and the estimated 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 ID of the user, the chosen store, the residence time and the number of people, which are all the data acquired during the reservation process. It also contains the status which can be active, pending, 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 chosen slot of time, which is associated to a startTime and endTime (represented with an int in order to specify only the hour of visit), in the second one there is the waiting time as attribute, which is updated by real-time checking the state of the queue.
* **Slot**: this class represent a slot of time which can be reserved. It contains the starting time and the ending 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 one is not available.
* **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.

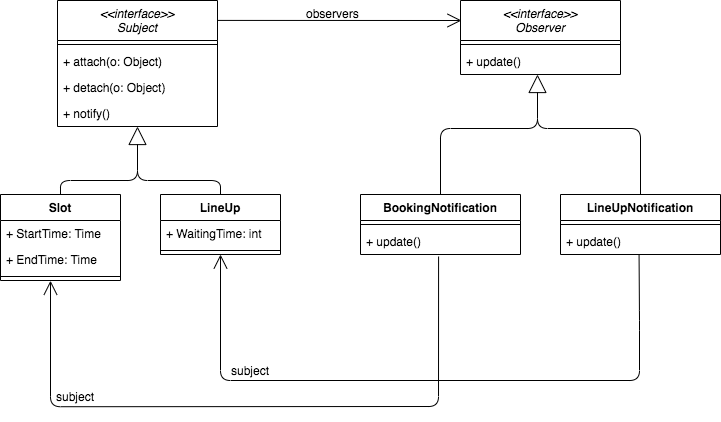


Figure 2 - UML 2: Observer pattern

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 is equal to one hour 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.

* + 1. **State Charts**

Now we are going to analyze some critical aspects of the application, modelling their behaviors, and showing the evolution over time of their states through appropriate state diagrams, which are reported below.

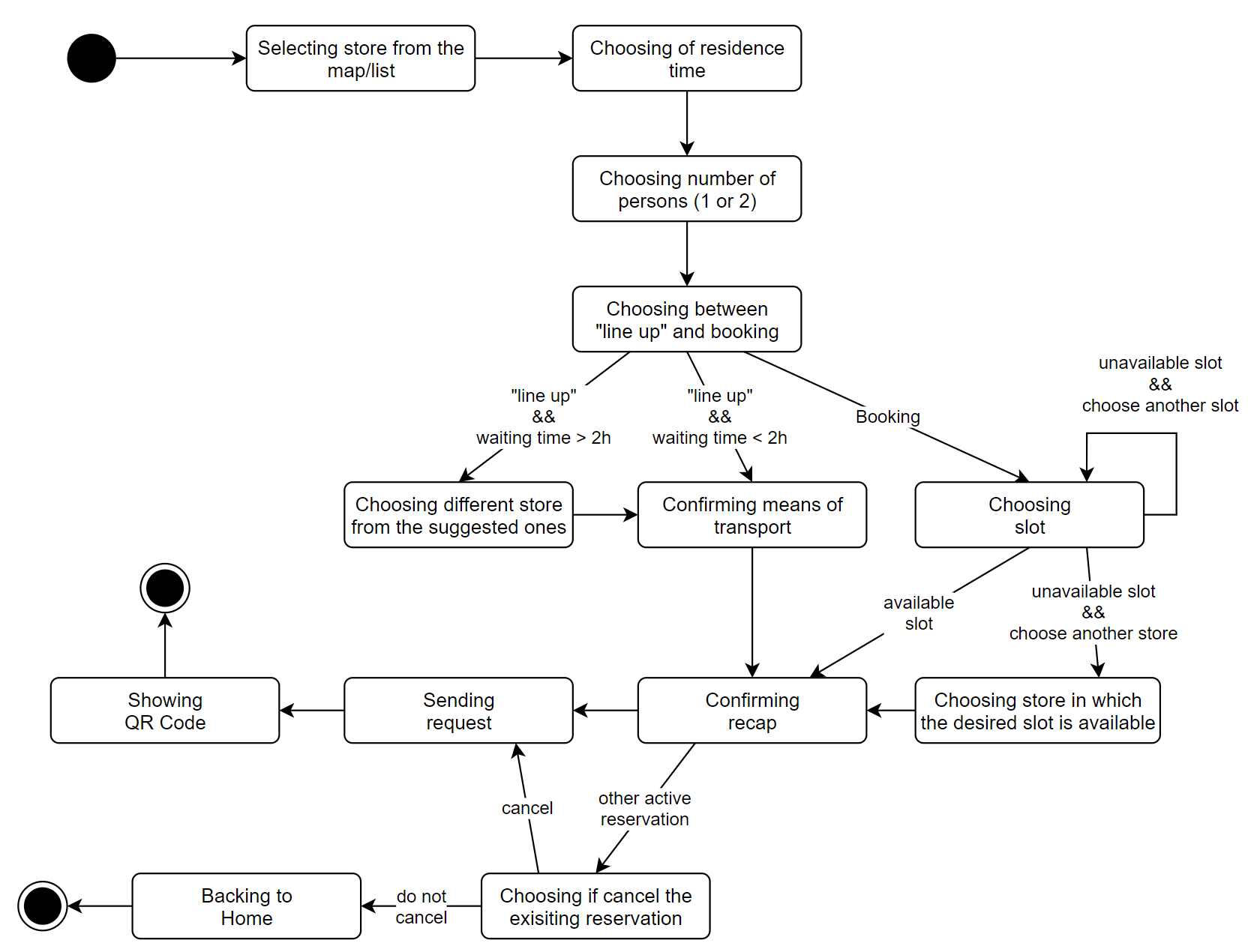


Figure 3 - State Diagram 1: creating a reservation

In the first state diagram (Figure 3), we model the creation of a reservation by the OnlineUser, showing the main actions that he has to perform.

* 1. **Product functions**

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

* + 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.

* + 1. **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, doesn’t 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 to ask him to confirm his presence.

* + 1. **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.

* + 1. **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 (decide how much) waiting time before entering the store; red for unavailability 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.

* + 1. **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 and also which shopping sections were visited. 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.

* 1. **User characteristics**

CLup is an application suitable for every person that has s 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.

* + 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

[ADD SYSTEM]

* 1. **Assumption, dependencies, and constraints**
     1. **Domain assumptions**

|  |  |
| --- | --- |
| NAME | BRIEF DESCRIPTION |
|  | 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 |
|  | 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 |
|  | 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 |
|  | About 80% of the customers goes to the supermarket through the use of CLup |
|  | OnlineUser comes to the supermarket with the means of transportation that he selected on the app |
|  | When OnlineUser selects the types of items that he will buy, he sticks to this indication |
|  | A high percentage of OnlineUsers confirm their reservation when it is asked |
|  | A high percentage of users enter in the store with a delay of up to five minutes with regards to his reservation |

* + 1. **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 needing 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. **Specific Requirements**
   1. **External Interface Requirements**
      1. **User Interfaces**

**[TODO]**

* + 1. **Hardware Interfaces**

**[TODO]**

* + 1. **Software Interfaces**

**[TODO]**

* + 1. **Communication Interfaces**

**[TODO]**

* 1. **Functional Requirements**
     1. **List of Requirements**

|  |  |
| --- | --- |
|  | The onlineUser must be able to point out the duration of his visit by choosing a defined period of time |
|  | The onlineUser must be able to point out the duration of his visit by selecting which categories of items he will buy |
|  | The system must be able to calculate the duration of an onlineUser’s visit by statistics on his previous visits |
|  | The onlineUser must be able to see in real-time an estimated waiting time to enter for each store |
|  | The onlineUser must be able to see in real-time the waiting time for his booking |
|  | The onlineUser must be able to delete a booking within an hour from its start |
|  |  |
|  |  |
|  | The onlineUser must be able to select the means of transportation |
|  | The onlineUser must be able to change the means of transformation at any moment |
|  | The onlineUser is not allowed to line up to a store where the current queue has a waiting time longer than two hours |
|  | Information must be used to build statistics about waiting time, duration of visits and most busy times of the day |
|  | The system must be able to count the number of people in each store in real-time |
|  | The system must be able to compute the amount of time of each visit inside the store |
|  | The internet connection of the OnlineUser must work properly |
|  | The smartphone of the OnlineUser must have an integrated GPS sensor |
|  | The system must be able to correctly compute the amount of time required to reach the store from the current position of the onlineUser |
|  | The onlineUser is not allowed to book a new reservation if he already has an active one |
|  | The onlineUser must be able to choose the number of people who will come to the store (1 or 2) |
|  | The system must be able to retrieve all the information about each past reservation |
|  | The onlineUser is not allowed to book a slot of time which is already full |
|  | The onlineUser must be able to confirm his reservation when he receives the related notification |
|  | 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 |
|  | 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 |

* + 1. **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 | 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 |
| G6 | D7 | R10, R17 |

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 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 |

|  |  |
| --- | --- |
| 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 |

* + 1. **Use Cases**
       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 connectivity 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. |

1. **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 can now 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. |

1. **Booking**

|  |  |
| --- | --- |
|  |  |
| Name | Booking |
| Actors | OnlineUser |
| Entry Condition | OnlineUser has already logged in the application |
| Event Flow | 1. OnlineUser clicks on a store on the interactive map 2. OnlineUser:    1. inserts the desired residence time between predefined intervals of time    2. inserts the categories of the items he needs to buy    3. accepts the suggested residence time or goes to 2.1) or 2.2) (only for regular customers) 3. OnlineUser inserts the number of person (max 2) 4. The system calculates an estimated waiting time 5. OnlineUser chooses between “Line up” and Booking:   Line up:   1. if waiting time < 2h, OnlineUser confirms the means of transport 2. if waiting time > 2h, OnlineUser chooses a different store from the suggested ones   Booking:   1. OnlineUser chooses a time slot 2. If the time slot is not available, OnlineUser    1. chooses a different slot    2. chooses a different store from the suggested ones 3. OnlineUser confirms the recap 4. If OnlineUser already has an active reservation, the system asks him if he wants to delete it and saves the new one 5. System receives the request and generates the relative QR Code 6. OnlineUser sees the QR Code |
| Exit Conditions | OnlineUser booked successfully |
| Exceptions | 1. The internet connection is not available   The above exception is handled by notifying the user with an error message   1. In point 7), OnlineUser does not accept to overwrite the current active reservation   The above exception is handled by bringing back the OnlineUser to the Home |

1. **Load map**

|  |  |
| --- | --- |
|  |  |
| Name | Load map |
| Actors | OnlineUser |
| Entry Condition | 1. OnlineUser has already logged in the application |
| Event Flow | 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 | 1. The internet connection is not available   This exception is handled by notifying the user with an error message |

1. **Cancellation of a reservation**

|  |  |
| --- | --- |
|  |  |
| Name | Cancellation of a reservation |
| Actors | OnlineUser |
| Entry Condition | 1. OnlineUser has already logged in the application 2. OnlineUser has an active reservation |
| Event Flow | 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 | 1. The internet connection is not available   This exception is handled by notifying the user with an error message |

1. **Approach to the store Notification**

|  |  |
| --- | --- |
|  |  |
| Name | Approach to the store Notification |
| Actors | System |
| Entry Condition | 1. The system detects the lineup time is about to expire |
| 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 | 1. The internet connection is not available   This exception is handled showing the notification when the internet connection will be available (managed by the OS) |

1. **Scan of the QR Code at the entrance of the store**

|  |  |
| --- | --- |
|  |  |
| Name | Entrance scan of the QR code |
| Actors | OnlineUser |
| Entry Condition | 1. OnlineUser has logged in 2. OnlineUser has correctly completed the booking procedure 3. OnlineUser must have his mobile device with him |
| Event Flow | 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 | 1. QR code is not recognized 2. QR code is recognized but the time slot is incorrect   All the exceptions are handled by notifying the user with an error message |

1. **Reminder Notification**

|  |  |
| --- | --- |
|  |  |
| Name | Reminder Notification |
| Actors | System |
| Entry Condition | 1. The system detects the reservation time is near |
| Event Flow | 1. The system sends a reminder notification to the OnlineUser   Three possibilities:   * 1. OnlineUser confirms the reservation within 1h   2. OnlineUser choose, within 1h, to do not confirm the reservation   3. OnlineUser does not reply within 1h. Therefore, the system deletes the reservation |
| Exit Conditions | OnlineUser receives the notification |
| Exceptions | 1. The internet connection is not available   This exception is handled showing the notification when the internet connection will be available (managed by the OS) |

1. **Scan of the QR Code at the exit of the store**

|  |  |
| --- | --- |
|  |  |
| Name | Exit scan of the QR code |
| Actors | OnlineUser |
| Entry Condition | 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 to grocery shopping |
| Event Flow | 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 screen to go on 6. The system shows to the OnlineUser the real amount of time he spent inside the store 7. OnlineUser clicks on the end button 8. The reservation ends successfully |
| Exit Conditions | OnlineUser ended its reservation and can now leave the store |
| Exceptions | 1. QR Code is not recognized   This exception is handled by notifying the user with an error message and the assistance of a clerk is required |

1. **Choice of means of transport**

|  |  |
| --- | --- |
|  |  |
| Name | Choice of means of transport |
| Actors | OnlineUser |
| Entry Condition | 1. OnlineUser has logged in 2. OnlineUser is not currently inside the store with an active reservation |
| Event Flow | 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 | 1. OnlineUser clicks on the already selected means of transport   This exception does not lead to any change in the means of transport |

1. **Check the state of an active reservation**

|  |  |
| --- | --- |
|  |  |
| Name | Check the state of an active reservation |
| Actors | OnlineUser |
| Entry Condition | 1. OnlineUser has already logged in the application 2. OnlineUser has an active reservation |
| Event Flow | 1. OnlineUser clicks on “My Reservations” section 2. OnlineUser clicks on the active reservation 3. 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 | 1. The internet connection is not available   This exception is handled by notifying the user with an error message |

* + - 1. **Use cases diagram**

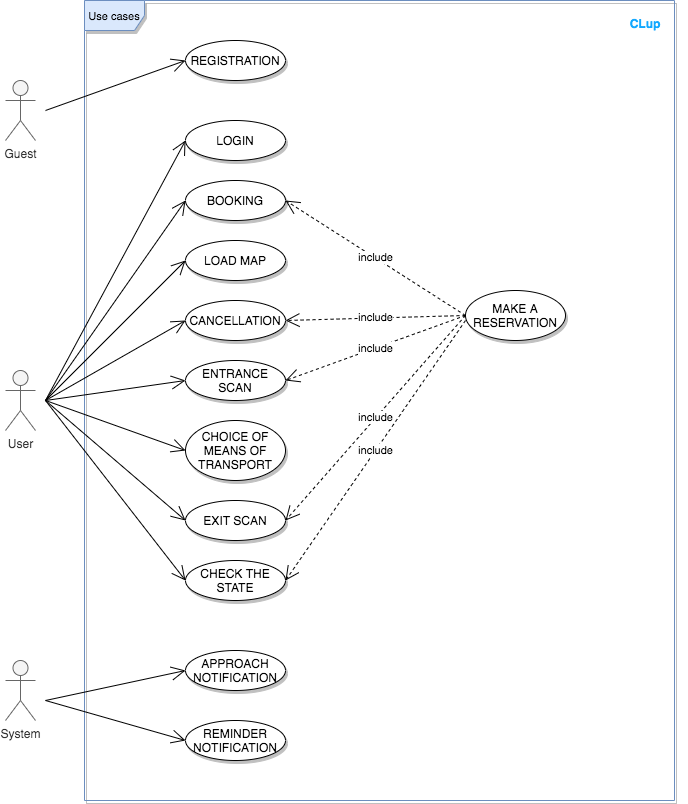


Figure 4 - Use cases diagram

* + 1. **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**

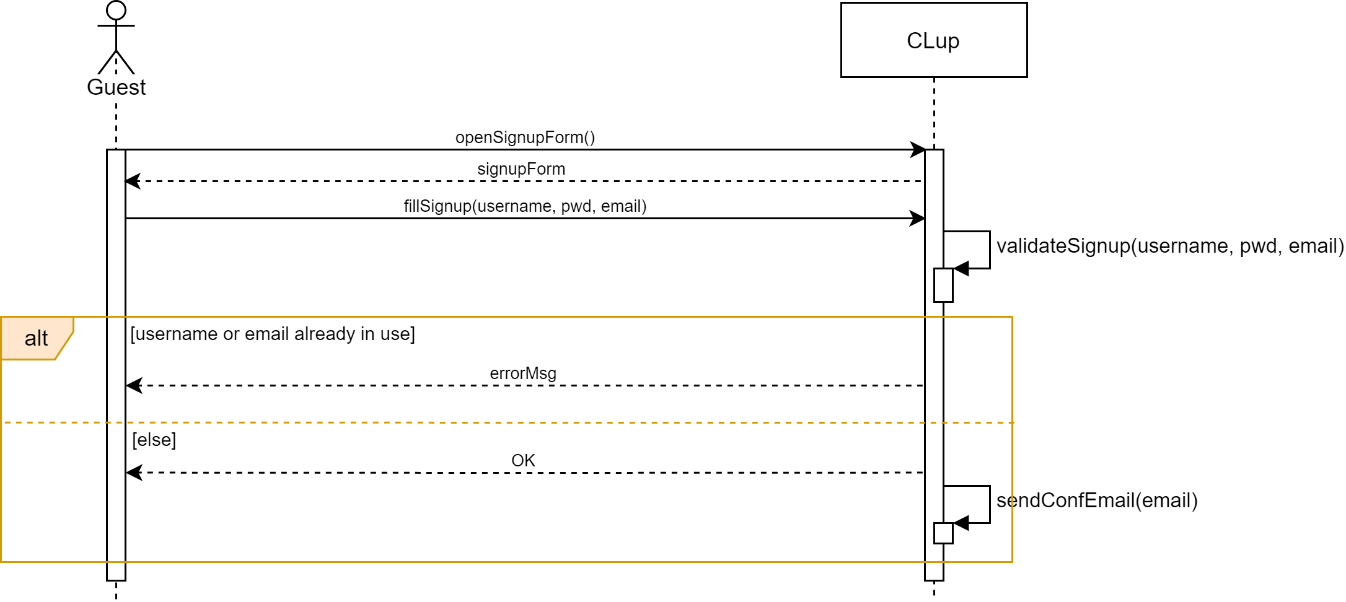
****

Figure 5 - Sequence Diagram: registration of an OnlineUser

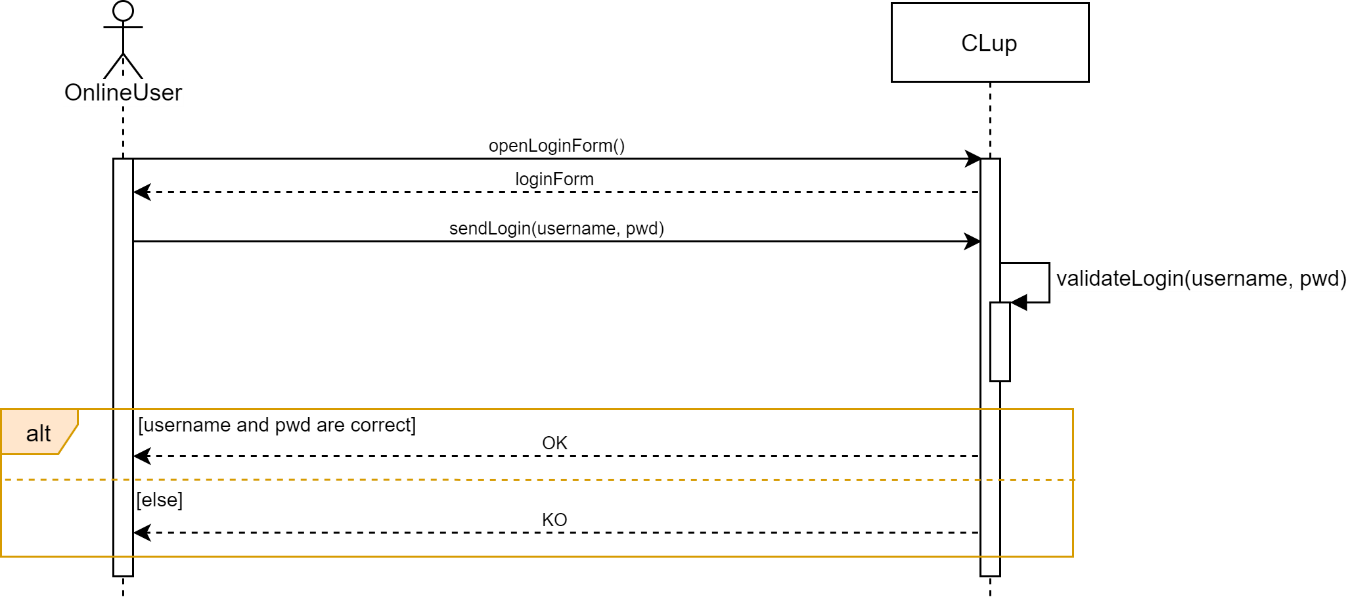
1. **Login of an OnlineUser**

Figure 6 - Sequence Diagram: login of an OnlineUser

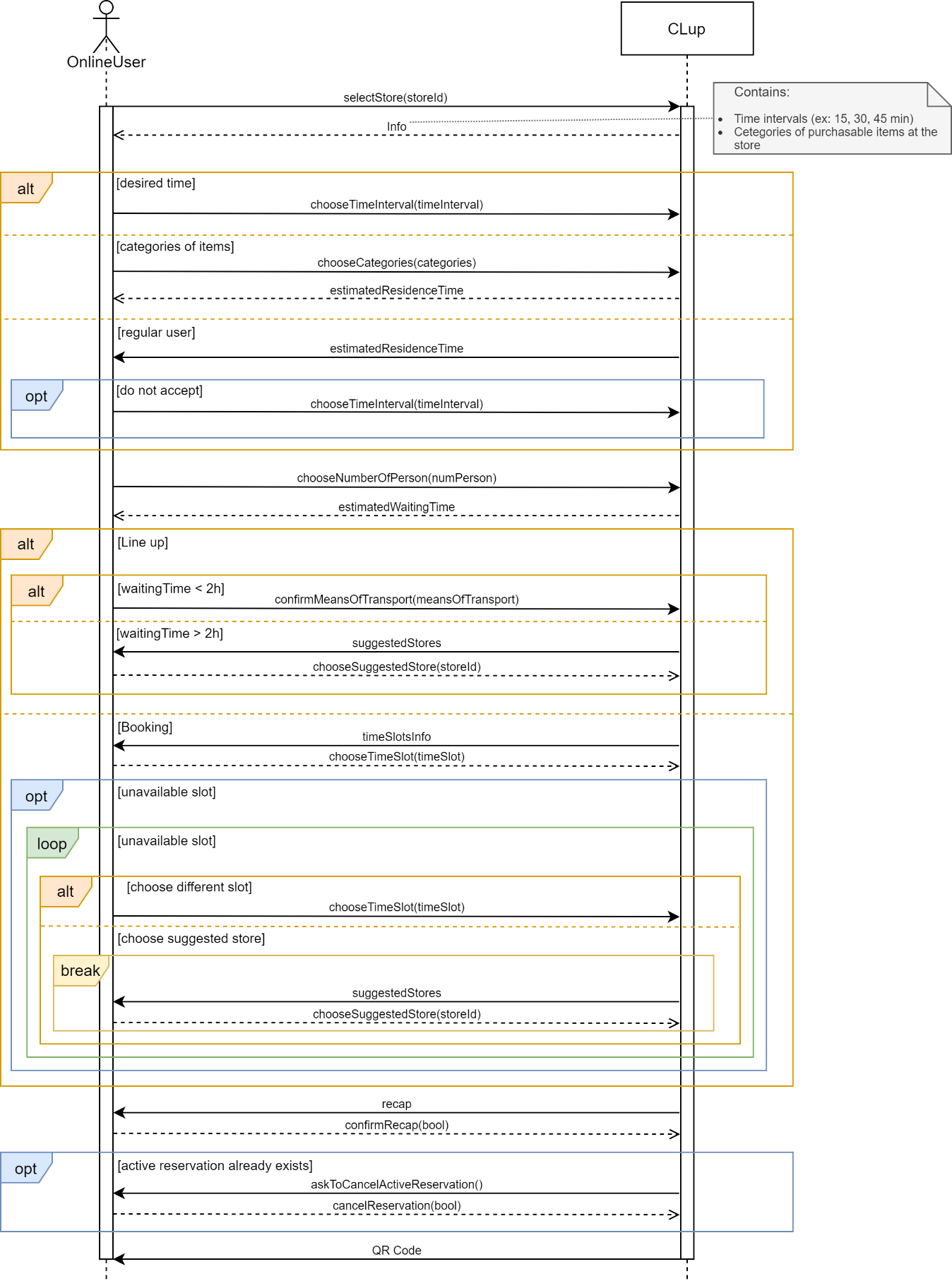
1. **Booking**

Figure 7 - Sequence Diagram: booking

1. **Load map**

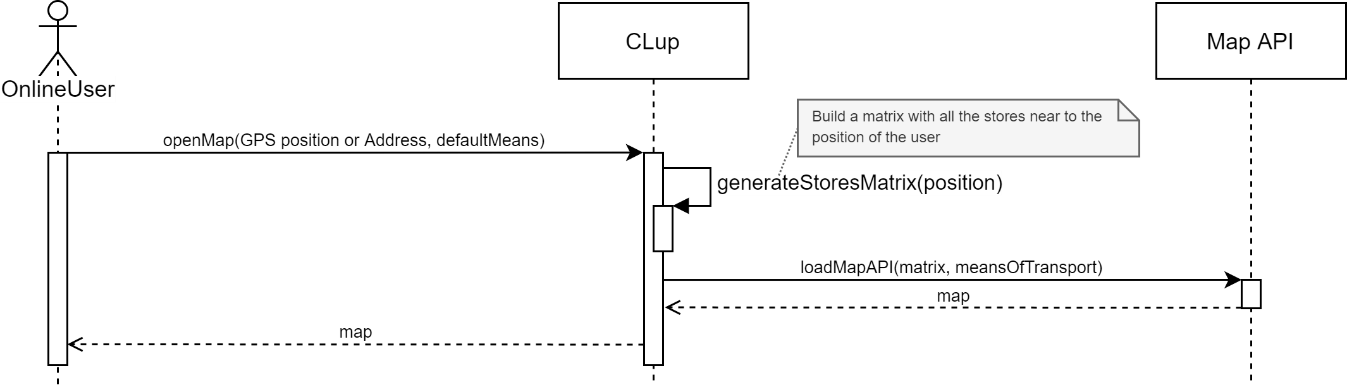
****

Figure 8 - Sequence Diagram: load map

1. **Cancellation of a reservation**

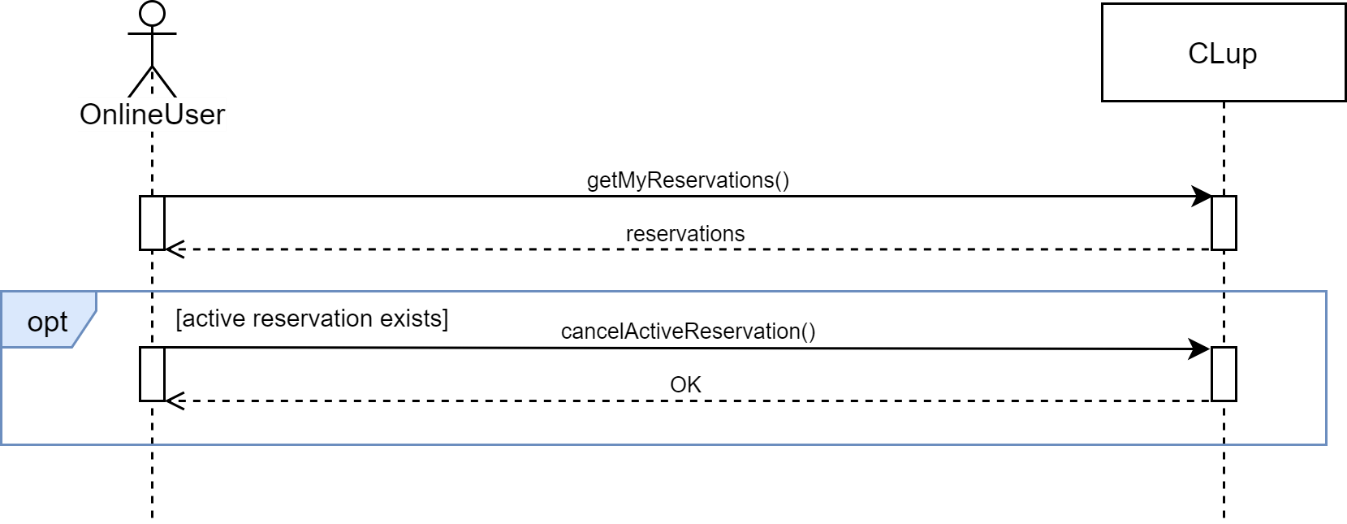
****

Figure 9 - Sequence Diagram: cancellation of a reservation

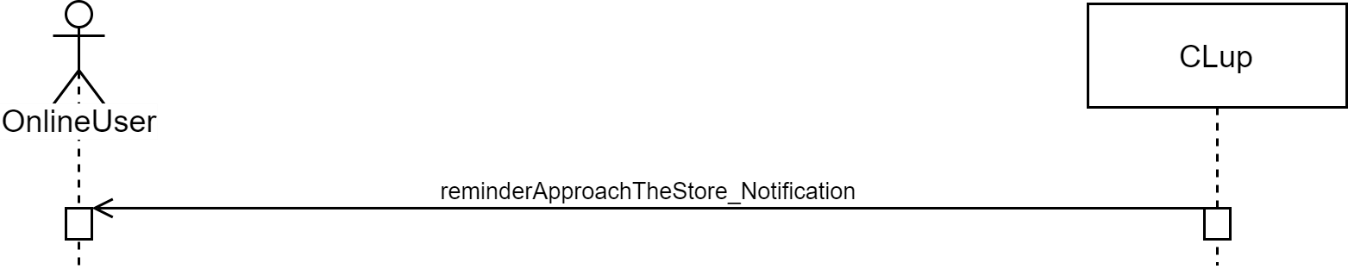
1. **Approach to the store Notification**

Figure 10 - Sequence Diagram: approach to the store Notification

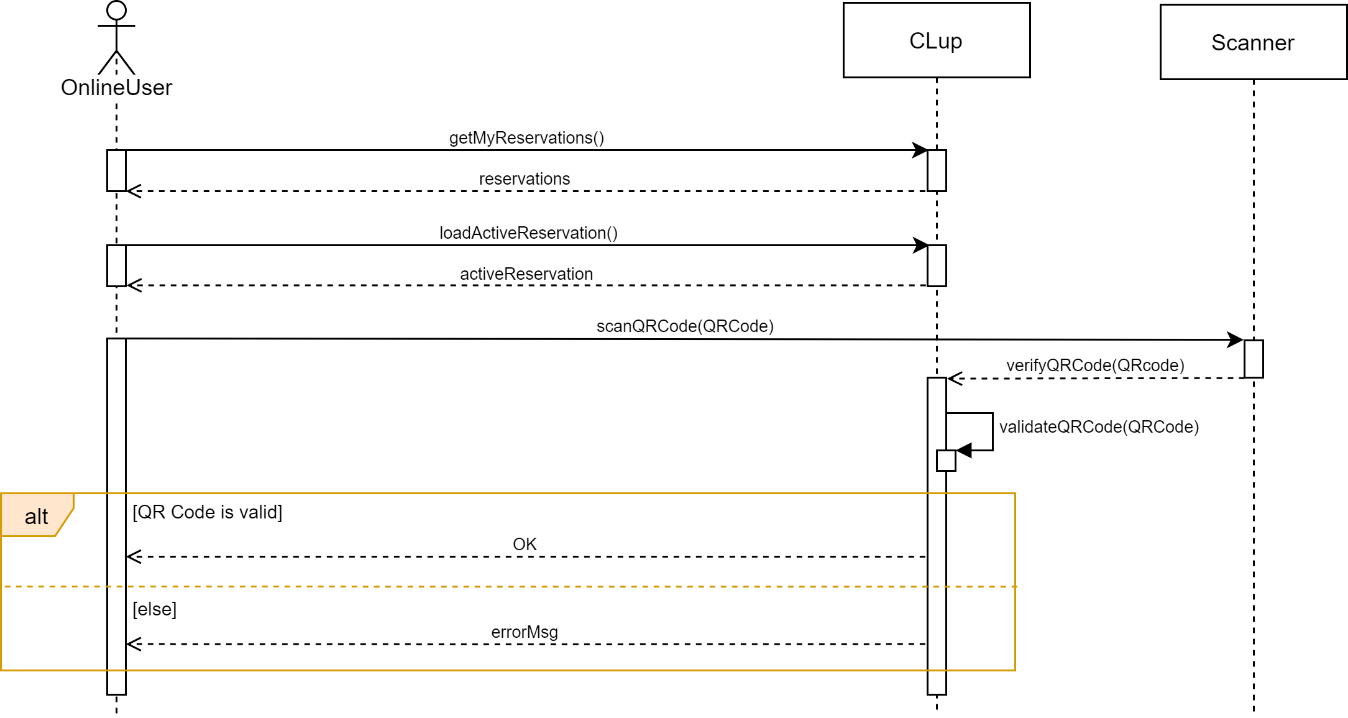
1. **Scan of the QR Code at the entrance of the store**

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

1. **Reminder Notification**

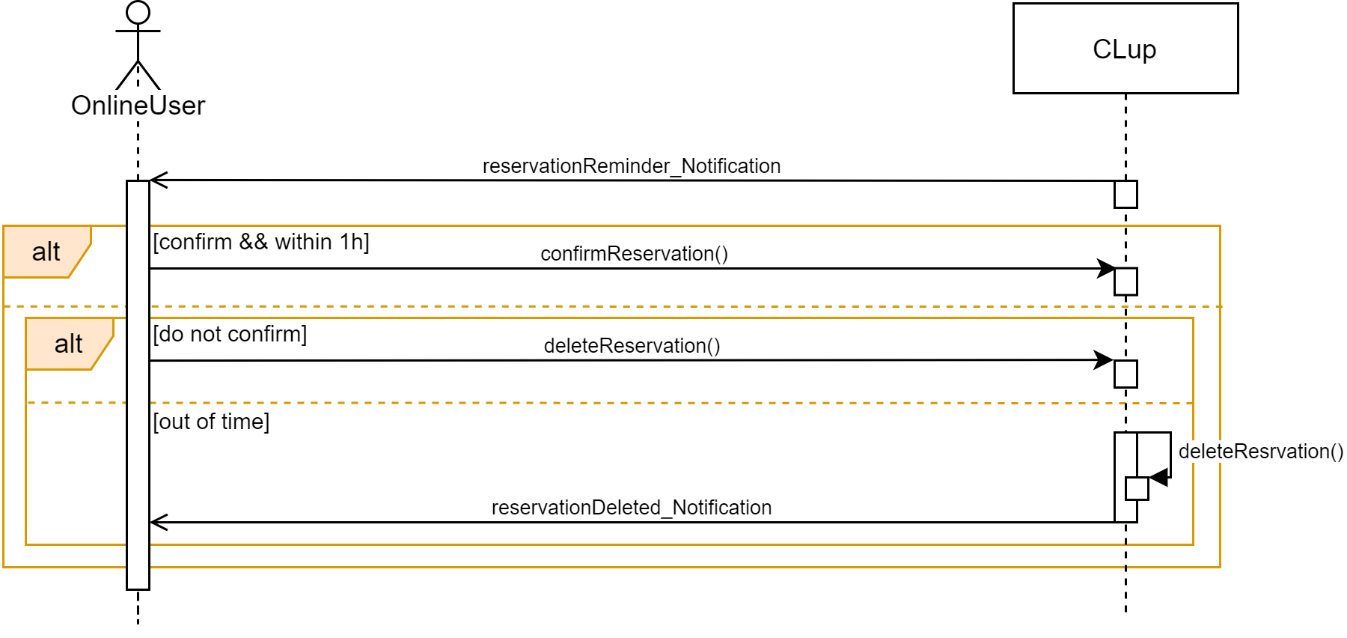


Figure 12 - Sequence Diagram: reminder Notification

1. **Scan of the QR Code at the exit of the store**

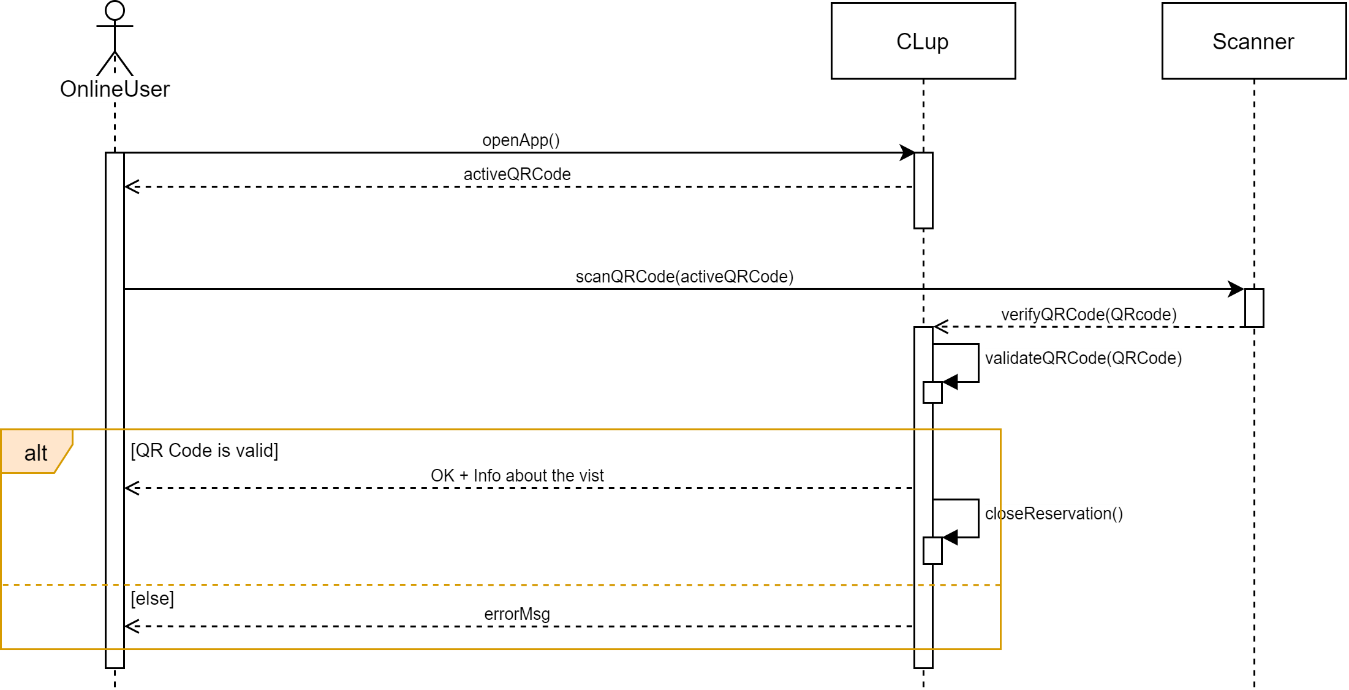
****

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

1. **Choice of means of transport**

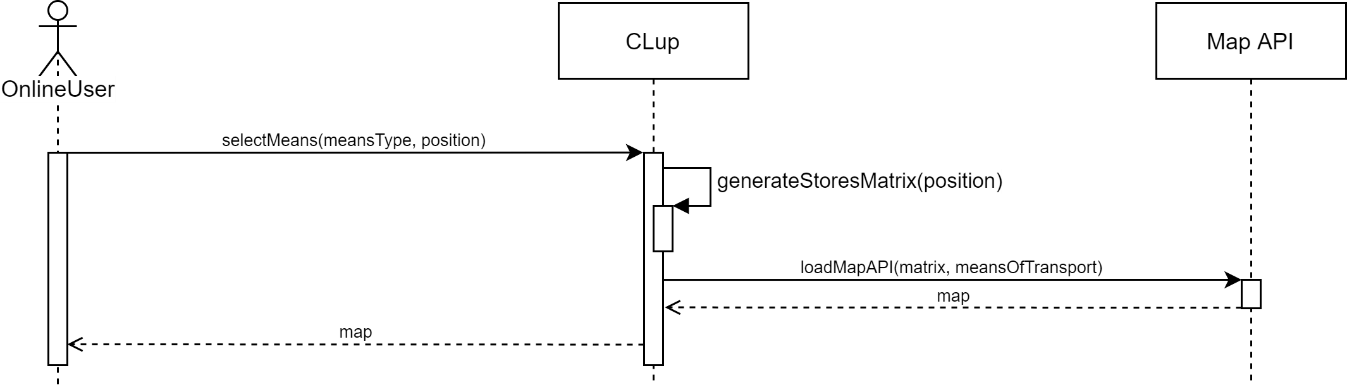


Figure 14 - Sequence Diagram: choice of means of transport

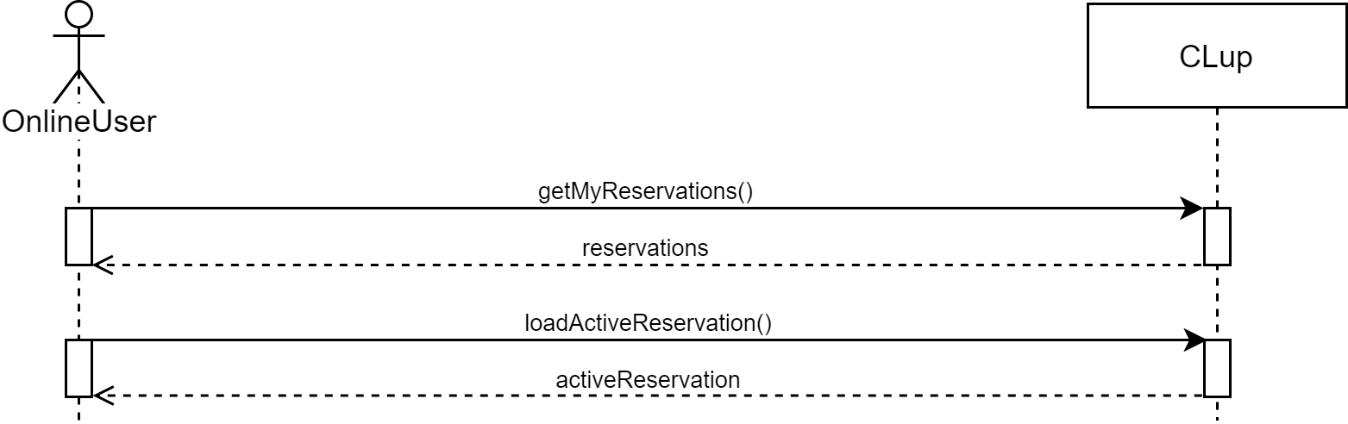
1. **Check the state of an active reservation**

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

* + 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 don’t 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 shooping 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.

* 1. **Performance Requirements**

This section contains some numerical requirements of the system, relative to the interaction between Users and System and to the performances.

The system must be able to work with simultaneous requests, also referred to the same sales point. For this reason, the onlineUser has five minutes to fill his booking request and to send it to CLup. His place in line will be reserved only during this time. Once expired, if the request has not already been sent, the onlineUser will have to restart it losing the previous position.

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.

* 1. **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**

The app will have a client-server architecture, so we have to analyze both sides. On the server side the main limitation is about the size of available storage, while on the client side it is the network connectivity (3G/4G/5G and/or WIFI connection).

* 1. **Software system attributes**
     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.

* + 1. **Security**

The data stored in the system has to be encrypted in order to ensure that the onlineUsers’ privacy is protected. The provided statistics about the stores are completely anonymous and no information about users can be retrieved from an external entity.

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

* + 1. **Maintainability**

The System will follow good software engineering practices to allow maintainability. There are some parameters, as the percentage of costumers 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.

* + 1. **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.

* + 1. **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.

1. **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 and Asserts and Predicates

open util/integer

open util/boolean

//Signatures

sig Float {

left: one Int,

right: one Int

}

sig Position {

latitude: one Float

longitude: one Float

}

enum MeansOfTransport {

Car, Walk, PublicTransport

}

abstract sig User {

id: one int,

reservations: set Reservations

}

sig PhysicalUser extends User {}

sig OnlineUser extends User {

firstname: some String,

lastname: some String,

email: one String,

password: one String,

livePosition: one Position,

meansOfTransport: one MeansOfTransport

}

sig Map {}

sig Department {

name: one String,

visitingTime: one Int

}{

visitingTime > 0 and visitingTime < 30 // in min

}

sig Store {

name: one String,

id: one Int,

capacity: one Int,

currentOccupation: one Int,

onlineUsersOccupation: one Int

departments: some Department

}{

currentOccupation <= capacity and currentOccupation >= 0 and capacity > 0

and onlineUsersOccupation <= currentOccupation

and onlineUsersOccupation <= capacity \* 0.8

}

sig Time {

year: one Int,

month: one Int,

day: one Int,

h: one Int,

m: one Int

}{

year > 2019,

month > 0 && month <= 12,

day > 0 && day <= 31,

h > 0 && h <= 24,

m > 0 && <= 60

}

sig Slot {

startTime: Time,

}

sig Suggestion {

suggested: some Store

}

sig Queue {

store: one Store,

reservation: set Reservation

}

sig QRCode {

base64: one String,

entranceScanTime: one Time,

exitScanTime: one Time

}

enum ReservationStatus {

Active, Pending, Finished

}

abstract sig Reservation {

id: one Int,

store: one Store,

estimatedResidenceTime: one Int,

numberOfPersons: one Int,

qrCode: one QRcode

status: one ReservationStatus

}{

estimatedResidenceTime > 0 estimatedResidenceTime <= 120 and numberOfPersons <= 2 and numberOfPersons > 0

}

sig Booking extends Reservation {

slot: one Slot

}

sig LineUp extends Reservation {

waitingTime: one Int

}{

waitingTime >=0 and waitingTime < 120

}

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

}

fact twoDifferentUsersNoSameEmail {

no disjoint u1, u2 : User |

u1.email = u2.email

}

fact twoDifferentStoresNoSameID {

no disjoint s1, s2 : Store |

s1.id = s2.id

}

fact twoDifferentReservationsNoSameID {

no disjoint r1, r2 : Reservation |

r1.id = r2.id

}

fact twoDifferentReservationsNoSameQRCode {

no disjoint r1, r2 : Reservation |

r1.qrCode.base64 = r2.qrCode.base64

}

fact sameStoreInAllReservationsInAQueue {

all q: Queue | all r: Reservation |

r in q.reservations implies q.store.id = r.store.id

}

fact oneUserOnlyOneActiveReservation {

all u : User |

no disj r1, r2 : Reservation |

r1.status = ReservationStatus.Active and r2.status = ReservationStatus.Active

and r1 in u.reservations and r2 in u.reservations

}

1. **Effort spent**
   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 |
| 3:00 | UML |
| 3:30 | Alloy model and UML diagram descriptions |

* 1. **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 |
| 3:00 | UML |
| ToComplete | To complete |

* 1. **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 |
| 3:00 | UML |
| 1.30 | UML of the observer pattern and brief explanation of the main classes |