POLITECNICO
MILANO

Politecnico di Milano

Department of Computer Science and Engineering

Software Engineering 2

CLup – Customers Line-up Requirements Analysis and Specification Document

December 14, 2020

Student
Damiano Derin

Student
Jas Valencic

Contents

Chapter 1

Introduction

1.1 Purpose

1.1.1 Description of the Given Problem

Our application is relative a situation of global pandemic that requires stores to help prevent the diffusion of the virus. And so, to guarantee their use in total safety. One of means to contribute to the success of this goal is to guarantee the absence of crowds. And that is the focus of our application. Considered one store and established the maximum number of people that is allowed to be inside, without having a crowd, our application is an instrument to keep the influx of people below that threshold. It works mainly managing automatically the influx of people inside the store, staggering the entrances in the store by using some parameters to keep them safe. This is done by offering to customers a basic service that consist in a digital lining up system. It mainly works remotely but in case it also works from a physical spot inside the store. In this way it should also help avoid having crowds outside the store. To reach the largest number of people this app should be easy to use, and its functions should work consistently in time. Instead to the manager it is given a dashboard where are shown significant data like the number of people currently in the store, the status of the queue, the influx of people during different intervals of time, etc. It also allows, when it is necessary, the store manager to make decision that influences the flux of people (like for example blocking the entrances for a period or modifying some parameters value which the algorithm uses to manage the flux). Furthermore, it is given to the customers a function that allow them to book a spot in the store by choosing the day and the time. This is thought to help people with limited availability during the day. This option requires optionally at people to point out which product categories they are going to purchase or the departments they are going to visit. This is used to optimize the algorithm of the system to maximize the number of people inside the store during a certain time, by knowing their distribution. This can also be made automatically by using their data for long-time customers.

1.1.2 Goals

We identified the following goals:

- **[G1]**: Keep customers in safe condition w.r.t the *"decreto del Presidente del Consiglio dei ministri"* (d.P.C.m) in force inside the store.
- **[G2]**: Limit the physical line situation in the proximity of the store.
- **[G3]**: Allow customers to line up from a remote device.
- **[G4]**: Allow store manager to monitor entrances.
- **[G5]**: Allow customers to line up from a physical spot.
- **[G6]**: Allow customers to book a visit from a remote device.

1.2 Scope

This document will refer mainly on a single supermarket chain with some already shared services, even though it could be naturally extended to a bigger set of different supermarkets chains. Inside a store we will encounter three possible users of our application, the store manager that will manly work with statistics, the physical spot that will act as the user for the people without the app, and the customers that are people using the app. Our application will mainly focus on the last ones as they are the ones that will give us more information to work with and so consequently the ones that we will be more able to control and manage to respect our goals. In fact, the application will start, when they line up, to monitor and track their position and it will estimate their time of arrival, in order to have the least amount of people close to the store. The application will track the order of people in the digital queue by the time they line up. They will know about their position in the queue from a ticket that will be handed to them, in a digital format if the line up is done by the app or as a paper if this is done using the physical spot. Inside queue we will also consider the people that used the advance function of booking, even though it will be done in a different manner. The people that use this last function will give us additional information to enhance the precision of our model and so to better predict their behaviours and timings.

1.2.1 Phenomena table

Phenomenon	**Shared**	**Who Controls it**
A person wants to do groceries	N	World
A person finish to do groceries	N	World
A person vist a department of the store	N	World
A person sings up	Y	World
A person looks at their position in the queue	Y	World
The store is full	Y	World
A person shows a QR code to enter	Y	World
A person enters in the store	Y	World
A person books a visit in the store	Y	World
A person point out a department that they will visit	Y	World
A person is notified that their turn is coming	Y	Machine
A person is not allowed to enter	Y	Machine
A person is allowed to enter	Y	Machine
The system tracks the number of people inside the store	Y	Machine

Table 1.1: Phenomena table

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Customer	a person who buys goods from the stores. We will use the term *customers* to refer to natural persons, instead the term *users* will be used to specify the virtual entity served by the application.
Store Manager	a person who is in charge of the store. In our context, we assume that the *store manager* controls the entrances to the store with the help of CLup service. In the real world scenario this activity can be delegated, without loss of generality.
Physical Spot	a digital device positioned outside the store that allows customers to obtain tickets to line up.
User	a virtual entity that interacts with the virtual service offered by CLup. The user can be a customers, a store manager and a physical spot (when it is acting as proxy). In case of ambiguous interpretations, we will specify the real entity name.
Proxy	an intermediary entity that exchanges information between two other entities. In our system, the physical spot can be seen as a proxy, since it allows customers to line up without the necessity to create an user account. From the point of view of the server, the physical spot is seen as an user.
Virtual Queue	a queue of users allocated in the memory of the server. When a user asks for a lining up operation, or a booking a visit operation, it is allocated in this queue.
Physical Queue	a queue of customers outside the store.
Ticket	a piece of paper or a virtual card given to customers to show that they have performed a lining up or a booking a visit operation.
QR code	a matrix composed by white and black squares encoding a string. It is reported on the ticket.
System	we use this term to represent the entire service, composed by smartphone application and servers.
Application	program executable on smartphone.

1.3.2 Acronyms and Abbreviations

API	Application Programming Interface
CLup	Customers Line-up

d.P.C.m	*"decreto del Presidente del Consiglio dei ministri"*
FIFO	First In First Out
GPS	Global Positioning System

1.4 Revision History

1.5 Reference Documents

1.6 Documents Structure

Chapter 2

Overall Description

2.1 Product Perspective

The next image shows the mock-up of the system made in UML. The UML is not definitive model, but it shows the starting point of the Clup system. It is made to help visualize the main elements of our system and their interactions. The scheme wants to highlight the following interactions:

- The ticket represents the person inside the queue
- The physical spot is treated like a user in the system, the default one.
- Since the booking ticket may point to different days, they are not immediately saved in the queue but instead in a booking list that will interact with the queue to schedule at the right time.
- The dashboard interacts with the system elements to show relevant data, and the complete picture.

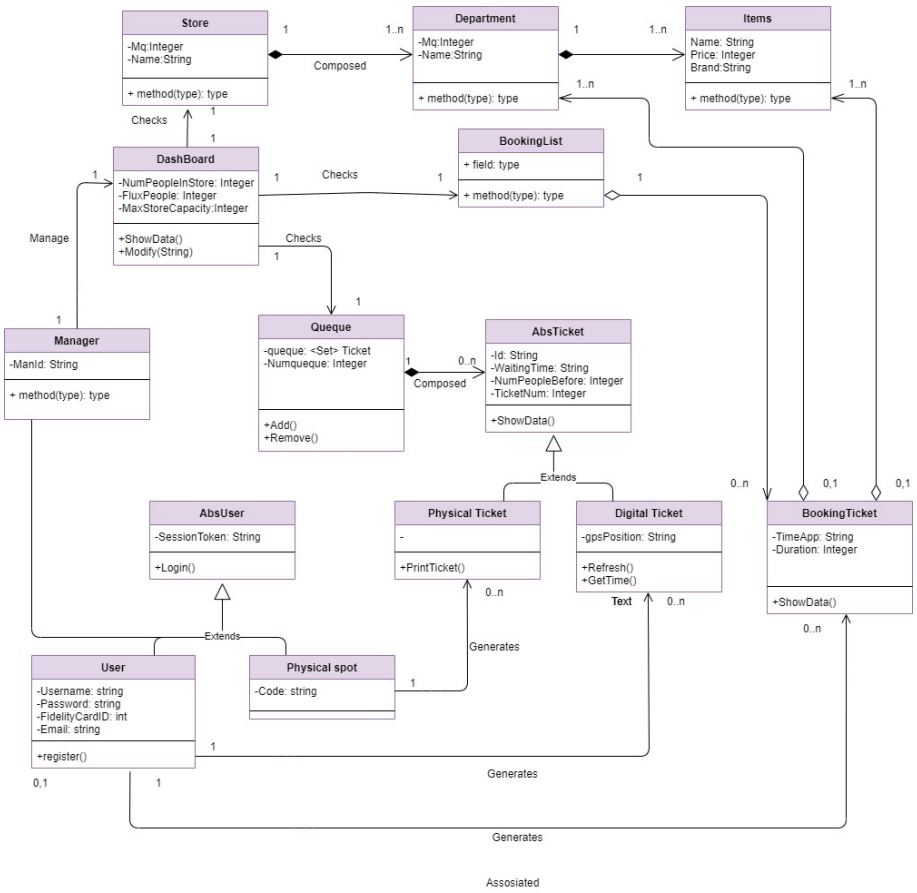


Figure 2.1: Class diagram.

In the next state diagrams, it will be shown how the lining up feature works. It will also highlight the difference in the handling between the physical ticket and the digital ticket.

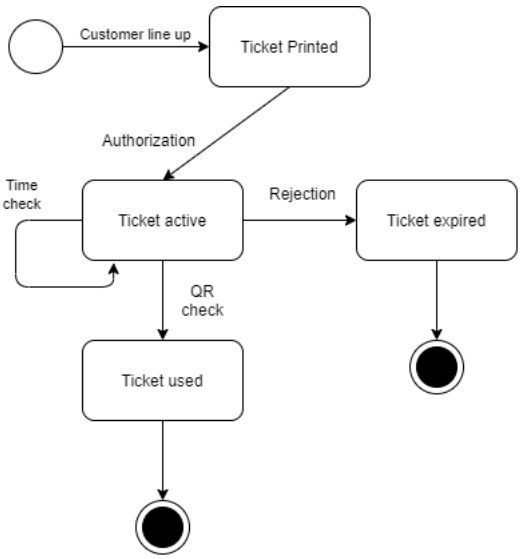


Figure 2.2: State diagram physical ticket.

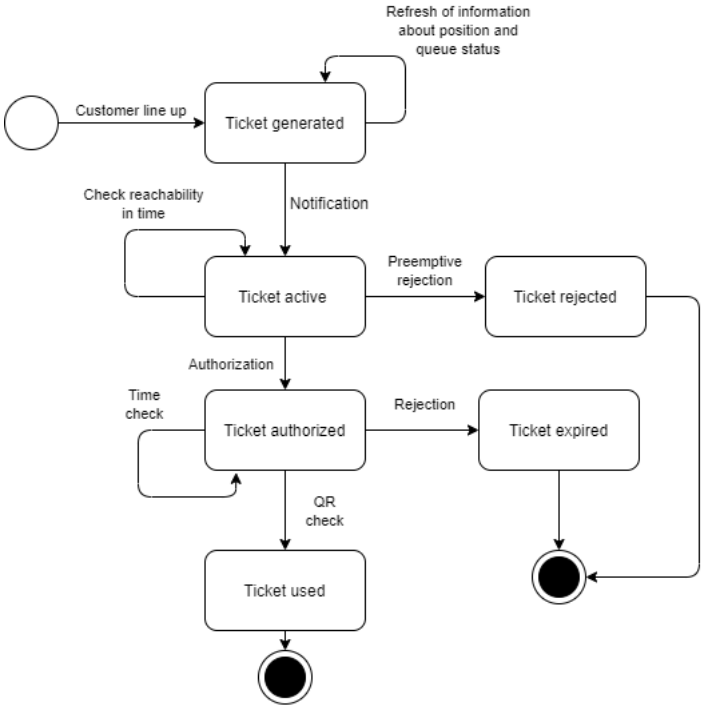


Figure 2.3: State diagram digital ticket

The main difference is that in the digital lining up, since it can be done from a distant position, Clup must first calculate a feasible time to reach the store and then schedule it in the queue. Instead, the physical lining up it is immediately

scheduled since the position is the store one. The second difference is that for the digital lining up a notification is sent to the customer, that suggest going to the store when it is the right time. But it is possible that other notifications are sent, if the reachable time check is violated, and if it exceeds a certain value the ticket is pre-emptively discarded. Instead for the physical one, since they are close to the store and that is not possible to notify them, they will able only to wait. And the end is the same for both, the ticket will authorize them to enter for a period then if not used it will expire.

2.2 Product Functions

In this section are described the main functionalities offered by the service.

2.2.1 Lining Up

In light of the motivations described in the previous sections, the main purpose of the application is to allow customers to line up from remote.
To achieve this result, the application provides the possibility to line up from the smartphone. The users have to log in the application, select the lining up operation, choose the store (in which they want to go) and confirm. Once the operation has been completed successfully, users are able to check the status of the queue and watch the QR code associated to the lining up. Moreover, users will receive live notifications about the status and the remaining time to be authorized to enter in the store. When the countdown is ending, customers have to approach to the store and wait outside for the call of their ticket number (showed with the QR code). At this time, they have to show the QR code to be scanned by the system and to be authorized to enter.

From the point of view of the server, when it receives the request, it has to check if the user can be allocated in the virtual queue and in which position. If it can, the user will be allocated, otherwise it will reply with an error message. The application sends to the server information about the global position of the customer. These information are used to estimate the time necessary by the customer to arrive to the store. All the collected information are used to schedule the entrances to the store. More precisely, the algorithm takes into account the position of the customer, to infer the cruise speed and the time needed to arrive, the number of customers already inside the store, the number of customers previously allocated to the virtual queue and the number of customers in the physical queue. The server can infer the residence time in the store looking at previous purchases of the same user or by computing the average residence time of the customers. Based on these data, the order in which customers ask for a lining up operation can be different by the allocation order in the virtual queue. In case of huge delays (parameter that can be controlled by the store manager) by the customers, the virtual queue can be reorganized dynamically.

2.2.2 Booking a Visit

This functionality is an extension of the previous one, in particular it allows to specify the date and time to visit the store.

To book a visit, customers should select the corresponding button from the menu of the application, insert the requested data, such as the store in which they want to go, the date, the slot time, the category of grocery they want to buy (it is not mandatory), and confirm the operation. As for the lining up operation, customers can check the status and the obtained QR code.

From the point of view of the server, the behavior is similar to the lining up, but in this case it can infer more information from the category of grocery specified, such as the section in the store that will be visited by the customer. If not specified, the server can infer information from previous purchases. In any case, the server can allocate users in a finer way in the virtual queue exploiting these data: knowing the maximum capacity of the store and the section with higher density of customers, it can decide if an user has to be allocated in one or another slot of time.

2.2.3 Lining Up from Physical Spot

If a customer hasn't an user account, or if he doesn't want to use (or can't use) the smartphone application, he can line up from a physical spot installed outside the store. The physical spot is a digital device that runs the same smartphone application used by other users, but with less functionalities. From the physical spot, a customer can line up clicking on a button to confirm the operation and the spot will print the ticket showing the QR code and the ticket number.

The physical spot acts as proxy. The physical spot is logged in the application with a custom account. In this way customers haven't to insert the credentials when they perform a lining up operation. The server retrieves the missing information (destination store, position of the customer, etc.), about the lining up, by the association between the physical spot account and the associated store (It can be best appreciated in 3.11). In this way the physical spot can be treated as a common user.

2.2.4 Monitoring and Controlling the Queue

These are functionalities dedicated to the store manager. Since the system performs different estimations, it can occur that the real situation differs from the theorized sequence of events. To handle this possibility, the store manager can monitor the status of the queue and the number of customers inside and outside the store from his device and decide if the server has to schedule in a different way the users arrivals. To do that, the application provides a different home page, if you are logged in with a store manger account, that provides buttons and interfaces to get the status of the situation and to set scheduling parameters that the server will use. In extreme cases he can stop issuing tickets.

2.3 User Characteristics

The actors in our application are the following:

- Customer: The customer is a person who has downloaded the application and regularly uses it to interact with the store. The first time they use it, they must register themself. The following times they must log in and authenticate in order to use the functionality of the app. Then they can

either line up remotely or book a visit. If they uses the line-up function, then, for some time they can view the relative ticket with the associated data. Same happens when using the function of booking a visit, after compiling the form and sending it, they can look at their appointments data for a period.

- Physical Spot: The physical spot act as representant of the users that do not have the application and gives them the possibility to use the line-up function. He does that by handing to them a paper ticket which represent a position in the digital queue.
- Manager: The manager is the person responsible of the store, and the one who can log as manager in the application.They are given a dashboard where they can check relevant data regarding the store, the influx of the people in it and the status of the queue. They can modify parameters important to the elaborations of the data and take decisions that influence the influx of people, like stopping the release of new tickets for a time period.

2.4 Assumptions, Dependencies and Constraints

Below has been reported the list of our domain assumptions.

- **[D1]**: There is a d.P.C.m in force.
- **[D2]**: Customers follow the rules impsed by the d.P.C.m in force.
- **[D3]**: Customers enter in the store only if the system authorizes them.
- **[D4]**: Customers don't stay in the shop longer than necessary and they go away from the sotre after they have done their shopping.
- **[D5]**: Customers line up physically only if they have a valid (non expired) QR code.
- **[D6]**: Outside the store there is space to queue.
- **[D7]**: Customers have a smartphone.
- **[D8]**: Customers have installed the CLup application.
- **[D9]**: Customers allow the permissions requested by the application.
- **[D10]**: Customers keep Internet connection active.
- **[D11]**: There is a store manager present in the store.
- **[D12]**: Store managers allow the permissions requested by the application.
- **[D13]**: Store managers keep Internet connection active.
- **[D14]**: Physical spots are powered on every working day.
- **[D15]**: Physical spots are refilled when asked by the system.

Chapter 3

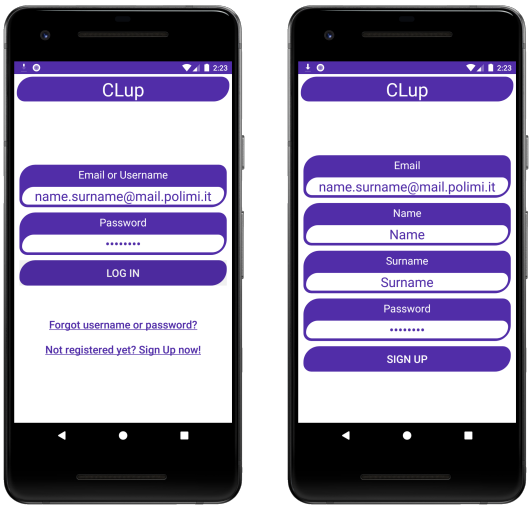
Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

In this section we describe and present a possible mock-up for the application, following the order of events proposed in the sequence diagrams.

At the launch of the application the user has to insert the credentials in the Log In page 3.1(a). In case of new user, he can click on a button to open the Sign Up page 3.1(b), to create a new account.



(a) Log In page. (b) Sign Up page.

Figure 3.1: Example of Log In and Sign Up pages.

Once the login has been completed successfully, the user is redirected to the Home page 3.2. In particular, the reported figure shows the Home page for the customer account with all the functionalities. The application, knowing the type

of account, is able to show different buttons. For example, in case of physical spot account, the application will display only the Lining Up button, hiding the others. In case of store manager account, there will be displayed only buttons to control the queue and to analyze the statistics.

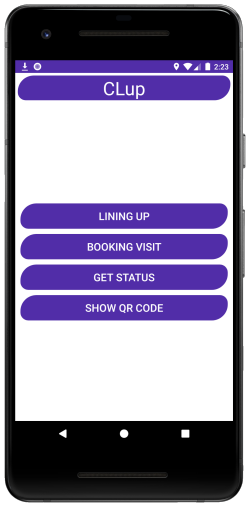
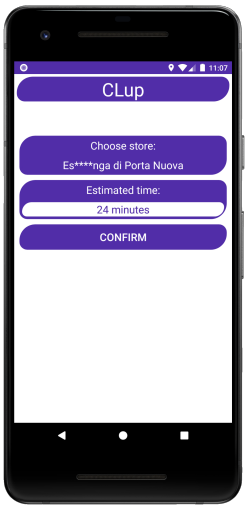
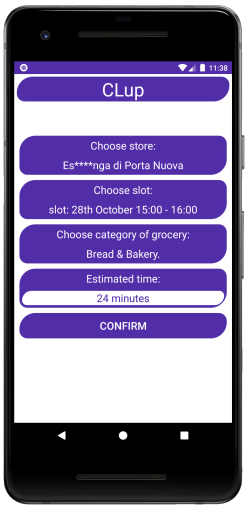


Figure 3.2: Home page.

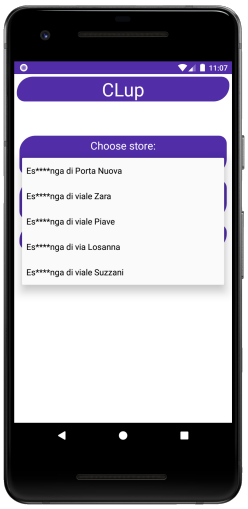
If a customer wants to line up, or book a visit, he can click on the corresponding button, in this way the application will show the form to insert the requested parameters 3.3. In the mock-up we show how the user can choose the store, the time slot and, in case of booking, the category of grocery, by expanding the drop down menu.



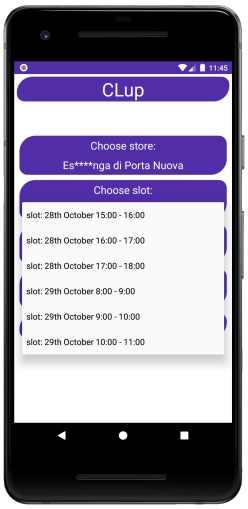
(a) Lining Up page.



(b) Booking Visit page.



(c) Lining Up page with expanded spinner.



(d) Booking Visit page with expanded spinner.

Figure 3.3: Example of Lining Up and Booking Visit pages.

The application provides the possibility to check the queue status by clicking on the Get Status button and to watch the QR code by clicking on the relative button. These buttons aren't visible from the Home page until the user performs a lining up, or booking a visit, operation. If visible, the users will be able to see the interfaces: 3.4(a), 3.4(b).

(a) Get Status page. (b) Show QR code page.

Figure 3.4: Example of Get Status and Show QR code pages.

In figure 3.5(a) has been reported a mockup showing a possible interface for the store manager to control the statistics about the status of the queue, instead, in figure 3.5(b) a page to control the parameters of the algorithm that schedules users and releases QR codes.

(a) Show Stats page. (b) Control Queue page.

Figure 3.5: Example of Show Stats and Control Queue pages.

3.1.2 Hardware Interfaces

The system is distributed over three main hardware resources.

- **Smartphone**: used by the customers that wants to line up or book a visit from remote. The smartphone has to have a Global Positioning System (GPS) module to send the position and an Internet connection active to receive live updates by the server.
- **Physical spot**: is a totem composed by a tablet and a printer. The printer is a peripheral of the tablet. The tablet must have an Internet connection active.
- **Turnstile**: is used to control the entries. The tablet of the store manager, the QR code scanner and the display, that shows the next ticket numbers, are parts of the turnstile. More precisely, turnstile, scanner and display are peripherals of the tablet. We can imagine the turnstile as the metal detector in the airports, in which the guards are seated on the opposite side of the entrance. The store manager controls everything by the tablet and the tablet controls the peripherals under the constraints imposed by the store manager and the remote server. Indeed the tablet has to have an Internet connection active.

3.1.3 Software interfaces

The system, to work properly, needs external Application Programming Interface (API).

- **Store APIs**: are well-liked if they there are. Additional data provided by the store, such as the history of the purchases of the customers, can be used to improve the quality of the estimations of the residence time; the sections of the store that will be visited by customers and so on.
- **Maps APIs**: can be used to improve the estimation of the needed time to arrive to the store by the global position of the customer.

3.1.4 Communications Interfaces

In this service we aren't not scheduling to use external communications interfaces. Only the standard Internet protocols.

3.2 Functional Requirements

3.2.1 Scenarios

First scenario:
Mario reads the newspaper and realize that the Covid-19 is a serious problem. He thinks that we should try to have behaviours that helps contain the problem, like take appointments on the spots we will go. He learns from internet that the supermarket where he usually goes to, is using an app to take appointments, Clup. He registers himself, and from day on when he needs to do grocery shopping, he uses the app to digitally line-up. When the app notifies him that his turn is coming, he goes out in the directions of the supermarket. And only in few cases he found himself waiting few minutes to enter inside the store.

Second scenario:
Francesca usually goes on the same supermarket to do groceries, but for one reason or another she did not go there for two weeks. When she returned to that store, she learnt that she needs to use an app to line up. She was told to download the app and to use it to line up, she did that and after she sign in. On the app she founds that there are shown some data like the number of people before her, the time she needs to wait before entering the store, etc. Seeing that the waiting time is large she decided to go for a walk around the neighbourhood and to return when the waiting time is of five minutes. She returned from her walk and saw that there was only a couple of people outside, she decided to wait and after few minutes she entered.

Third Scenario:
Maria is an old lady, she does not possess a smartphone but only an old phone that does not have internet connection. She decides to go to the supermarket in which a clerk tells her that she needs to use an app to line up. She explains to him that she does not have the means to do it and so, she is brought to the totem where she retrieves a ticket. She is recommended to stay close to the supermarket and to wait it there. During the chat with the clerk she also found that this mechanism is always guaranteed. Thanks to the fact that most people uses the app to line up, outside the store where the old lady is waiting, there are no crowds. Because all people either enter right away or at most they wait for few minutes, and so it reduces the risk for all.

Fourth scenario:
Andrea is a worker with a fixed schedule, every day he finishes to work at eight p.m., and so he has little time to do the groceries. He has tried to use the app Clup to line up, but he found himself, sometimes, with a waiting time of twenty minutes, without knowing what to do except to go to the store and wait outside. To try to solve this problem he used the "book a visit" function and took an appointment for the following day, choosing a schedule that worked for him, allowing him to arrive without hurry to the store. The next day he went to the store right on time and with a hint of amazement he entered right away. Impressed by the pleasant experience from that day on he started using regularly that function since it allows him to adjust it to his needs.

Fifth scenario:
Alfredo always uses the app to line up at the supermarket. One day he notices that when he lines up from his house the waiting time is always of twenty minutes even on the days where he goes there and see few people. With some reasoning he comes to the conclusion that the app gives him always a waiting time that is similar at the time he usually needs to get there. In fact, he really needs little less than twenty minutes to go there.

Sixth scenario 6:
Claudia is the manager of a supermarket that uses Clup and she checks the app to manage the flux of people inside the store. She always checks that the flux of people never reaches the eighty-five percent of the maximum capacity that the D.P.C.M. imposed to the store, when there are not bookings and not more than ninety percent of the maximum capacity if there are some bookings that specify

the department. One day even if the store respects these constrains, she notices that on one department, next to the entrance, a crowd is forming. Maybe it is happening because a recent sale on a category of products. She thinks that it would be appropriate to do not let other people enter until the problem is solved. So, she decides to use the app to temporarily block the number of new entrances. Leaving like that for few minutes the situation solve itself and she can allow again people to enter.

3.2.2 Requirements

Bla bla bla...

Goal	**G1: Keep customers in safe condition w.r.t the d.P.C.m in force inside the store.**
Requirements	• **[R1]**: The system has to schedule entrances to the store. • **[R2]**: The system has to compute the maximum capacity of the store w.r.t. the social distances imposed by the d.P.C.m in force. • **[R3]**: The system has to monitor the customers residence time in the store. • **[R4]**: The system has to allow authorized customers to enter in the store. • **[R5]**: The system has to deny unauthorized customers to enter in the store. • **[R6]**: The system has to know when a customer enters in the store. • **[R7]**: The system has to know when a customer has left the store.
Domain Assumptions	• **[D1]**: There is a d.P.C.m in force. • **[D2]**: Customers follow the rules imposed by the d.P.C.m in force. • **[D3]**: Customers enter in the store only if the system authorizes them. • **[D4]**: Customers don't stay in the shop longer than necessary and they go away from the store after they have done their shopping.

Goal	**G2: Limit the physical line situation in the proximity of the store**
Requirements	• **[R8]**: The system has to estimate the residence time, of a customer, in the store. • **[R9]**: The system has to infer the residence time of the customers based on past purchases. • **[R10]**: The system has to estimate the time needed to arrive, to the store, from the position of the customer. • **[R11]**: The system has to track the global position of the customers. • **[R12]**: The system has to release QR codes to the customers. • **[R13]**: The system has to limit the number of releasable QR codes if imposed by the store manager. • **[R14]**: The system has to allow the store manager to monitor the status of the queue. • **[R15]**: The system has to notify customers about the remaining time to be authorized to enter in the store. • **[R16]**: The system has to communicate which is the next served QR code number.
Domain Assumptions	• **[D5]**: Customers line up physically only if they have a valid (non expired) QR code. • **[D4]**: Customers don't stay in the shop longer than necessary and they go away from the store after they have done their shopping. • **[D6]**: Outside the store there is space to queue.

Goal	**G3: Allow customers to line up from a remote device.**
Requirements	• **[R17]**: The system has to allow customers to register to the application. • **[R18]**: The system has to allow customers to login to the application. • **[R19]**: The system has to allow customers to get a QR code from the application. • **[R20]**: The system has to release QR codes to the customers through the application. • **[R21]**: The system has to alert customers if the queue is full. • **[R22]**: The system has to encode the lining up number in the QR code. • **[R23]**: The system has to allow customers to watch the QR code from the application. • **[R24]**: The system has to allow customers to watch the lining up number encoded in the QR code. • **[R25]**: The system has to allow customers to watch the remaining time to be authorized to enter in the store. • **[R26]**: The system has to update the remaining time showed to the customers. • **[R27]**: The system has to allow customers to delete a lining up operation. • **[R28]**: The system has to notify customers about the validation status of the QR code. • **[R29]**: The system has to check if customers have Internet connection active. • **[R30]**: The system has to check if customers have allowed the permissions requested by the application.
Domain Assumptions	• **[D7]**: Customers have a smartphone. • **[D8]**: Customers have installed the CLup application. • **[D9]**: Customers allow the permissions requested by the application. • **[D10]**: Customers keep Internet connection active.

Goal	**G4: Allow store manager to monitor entrances.**
Requirements	• **[R31]**: The system has to register store managers in the application. • **[R32]**: The system has to allow store managers to login to the application. • **[R33]**: The system has to allow store managers to monitor the status of the queue. • **[R34]**: The system has to allow store managers to limit the number of QR codes released. • **[R35]**: The system has to allow store managers to monitor the number of customers inside the store. • **[R36]**: The system has to scan the QR codes of the customers. • **[R37]**: The system has to allow store managers to modify the timing parameters of the scheduler. • **[R38]**: The system has to check if store managers have Internet connection active. • **[R39]**: The system has to check if store managers have allowed the permissions requested by the application.
Domain Assumptions	• **[D11]**: There is a store manager present in the store. • **[D12]**: Store managers allow the permissions requested by the application. • **[D13]**: Store managers keep Internet connection active.

Goal	**G5: Allow customers to line up from a physical spot.**
Requirements	• **[R40]**: The system has to allow unregistered customers to line up. • **[R41]**: The system has to allow customers to get a QR code from a physical spot. • **[R42]**: The system has to release QR codes to the customers through the physical spot. • **[R43]**: The system has to encode the lining up number in the QR code. • **[R44]**: The system has to print QR codes on a paper tickets. • **[R45]**: The system has to alert customers if the queue is full. • **[R46]**: The system has to alert when the paper and toner of the physical spot is going to finish.
Domain Assumptions	• **[D14]**: Physical spots are powered on every working day. • **[D15]**: Physical spots are refilled when asked by the system.

Goal	**G6: Allow customers to book a visit from a remote device.**
Requirements	• **[R17]**: The system has to allow customers to register to the application. • **[R18]**: The system has to allow customers to login to the application. • **[R19]**: The system has to allow customers to get a QR code from the application. • **[R47]**: The system has to allow customers to specify the date and time for a visit to the store. • **[R48]**: The system has to allow customers to specify the category of grocery they want to buy. • **[R20]**: The system has to release QR codes to the customers through the application. • **[R21]**: The system has to alert customers if the queue is full. • **[R49]**: The system has to encode the book-a-visit number in the QR code. • **[R23]**: The system has to allow customers to watch the QR code from the application. • **[R50]**: The system has to allow customers to watch the book-a-visit number encoded in the QR code. • **[R25]**: The system has to allow customers to watch the remaining time to be authorized to enter in the store. • **[R51]**: The system has to allow customers to delete a book-a-visit operation. • **[R28]**: The system has to notify customers about the validation status of the QR code. • **[R29]**: The system has to check if customers have Internet connection active. • **[R30]**: The system has to check if customers have allowed the permissions requested by the application.
Domain Assumptions	• **[D7]**: Customers have a smartphone. • **[D8]**: Customers have installed the CLup application. • **[D9]**: Customers allow the permissions requested by the application. • **[D10]**: Customers keep Internet connection active.

3.2.3 Definition of Use Case Diagrams

Bla bla bla...

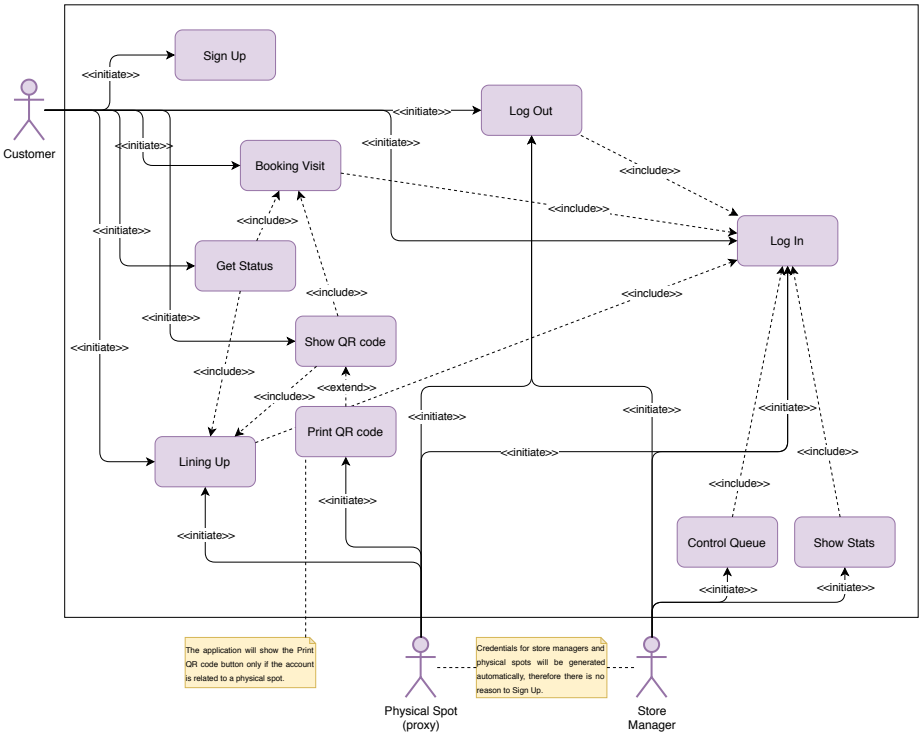


Figure 3.6: Use cases diagram.

Name	Sign Up
Actor	Customer
Entry Conditions	Customer is on the Sign Up page.
Event Flows	• Customer inserts the requested information in the form. • Customer clicks on the Sign Up button.
Exit Conditions	Sign Up completed successfully and customer is logged in, then the application shows the Home page.
Exceptions	• Customer's username already in use. • Empty form field. • Policy agreement rejected. • Lost Internet connection.

Table 3.1: Use case: **Sign Up**.

Name	Log In
Actor	Customer - Physical Spot - Store Manager
Entry Conditions	Actor is on the Log In page.
Event Flows	• Actor inserts the requested information in the form. • Actor clicks on the Log In button.
Exit Conditions	Log In completed successfully and actor is redirected to the Home page.
Exceptions	• Actor's username or password incorrect. • Empty form field. • Lost Internet connection.

Table 3.2: Use case: **Log In**.

Name	Log Out
Actor	Customer - Physical Spot - Store Manager
Entry Conditions	Actor is on the Log Out page.
Event Flows	• Actor clicks on the Log Out button.
Exit Conditions	Log Out completed successfully and actor is redirected to the Log In page.
Exceptions	• Actor already logged out. • Lost Internet connection.

Table 3.3: Use case: **Log Out**.

Name	Lining Up
Actor	Customer - Physical Spot
Entry Conditions	Actor is on the Home page.
Event Flows	• Actor clicks on the Lining Up button. • Actor inserts the requested data in the form. • Actor clicks on the confirmation button.
Exit Conditions	Lining Up completed successfully, the application returns the Status page.
Exceptions	• Previous Lining Up action was not expired (only in case of remote customer). • Previous Booking Visit action was not expired (only in case of remote customer). • Actor wasn't logged. • Lost Internet connection.

Table 3.4: Use case: **Lining Up**.

Name	Booking Visit
Actor	Customer
Entry Conditions	Customer is on the Home page.
Event Flows	• Customer clicks on the Booking Visit button. • Customer fills the form with the requested data. • Customer clicks on the Submit button.
Exit Conditions	Booking Visit completed successfully and the application returns, to the customer, the Status page.
Exceptions	• Previous Lining Up action was not expired. • Previous Booking Visit action was not expired. • Customer wasn't logged. • Lost Internet connection.

Table 3.5: Customer - use case: **Booking Visit**.

Name	Show QR code - Print QR code
Actor	Customer - Physical Spot
Entry Conditions	Actor is on the Home page.
Event Flows	• Actor clicks on the Show QR (Print QR) code button.
Exit Conditions	The application shows (print) the QR code.
Exceptions	• QR code wasn't saved on the application correctly (only in case of remote customer). • No Lining Up, or Booking Visit, action previously performed (only in case of remote customer). • Actor wasn't logged. • Spot finished the paper. • Spot finished the ink.

Table 3.6: Use case: **Show QR code - Print QR code**.

Name	Get Status
Actor	Customer
Entry Conditions	Customer is on the Home page.
Event Flows	• Customer clicks on the Get Status button.
Exit Conditions	The application returns the Get Status page showing information about the last Lining Up, or Booking Visit, operation.
Exceptions	• No operation previously performed, therefore there is no data to show. • Customer wasn't logged. • Lost Internet connection.

Table 3.7: Customer - use case: **Get Status**.

Name	Control Queue
Actor	Store Manager
Entry Conditions	Store Manager is on the Home page.
Event Flows	• Store Manager clicks on the Control Queue button.
Exit Conditions	The application returns the Control Queue page showing options to manage the queue.
Exceptions	• Store Manager wasn't logged. • Lost Internet connection.

Table 3.8: Store Manager - use case: **Control Queue**.

Name	Show Stats
Actor	Store Manager
Entry Conditions	Store Manager is on the Home page.
Event Flows	• Store Manager clicks on the Show Stats button.
Exit Conditions	The application returns the Show Stats page showing information about the number of customers inside the store, the length of the queue and other information about the waiting time in queue.
Exceptions	• Store Manager wasn't logged. • Lost Internet connection.

Table 3.9: Store Manager - use case: **Show Stats**.

3.2.4 Use Cases and Sequence/Activity Diagrams

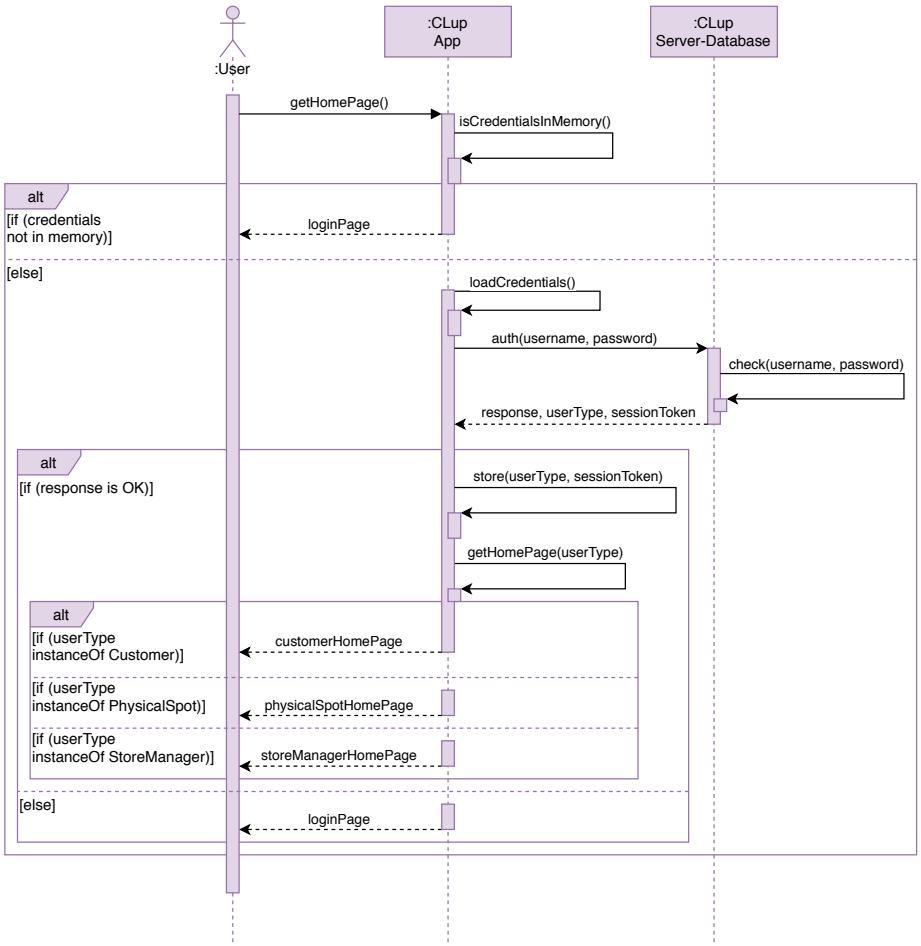


Figure 3.7: Home page sequence diagram.

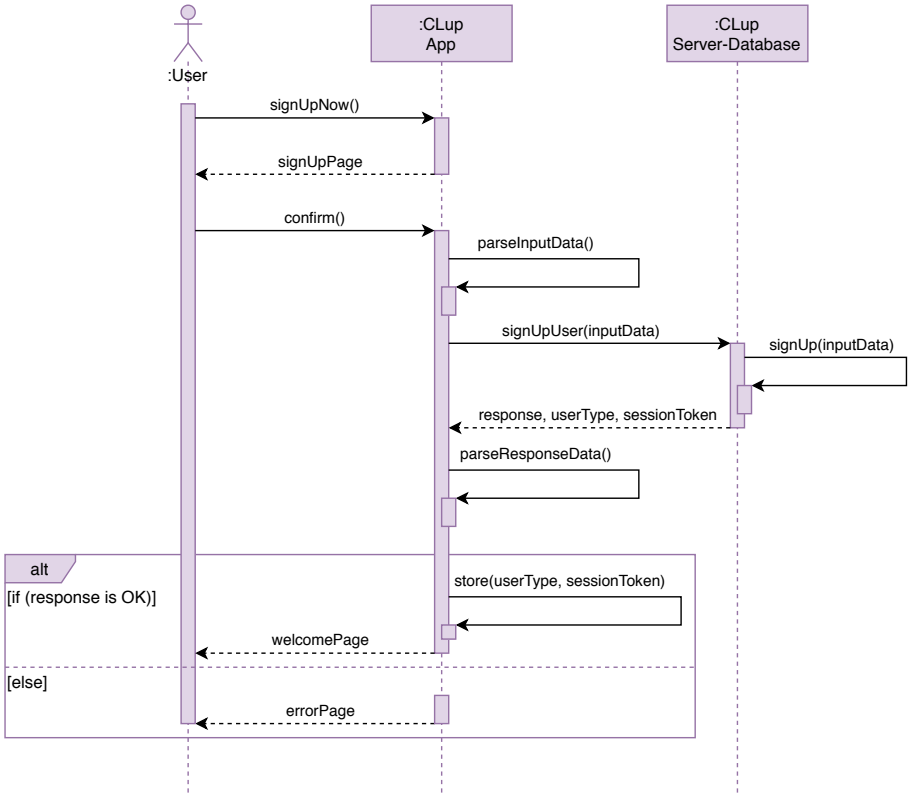


Figure 3.8: Sign Up sequence diagram.

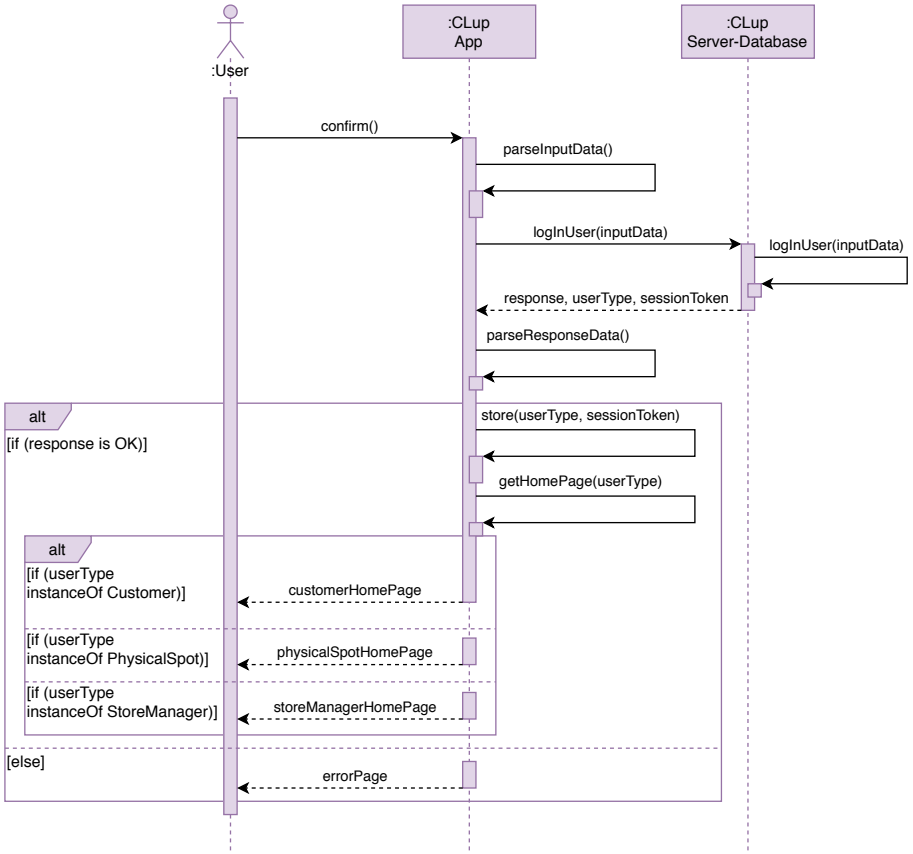


Figure 3.9: Log In sequence diagram.

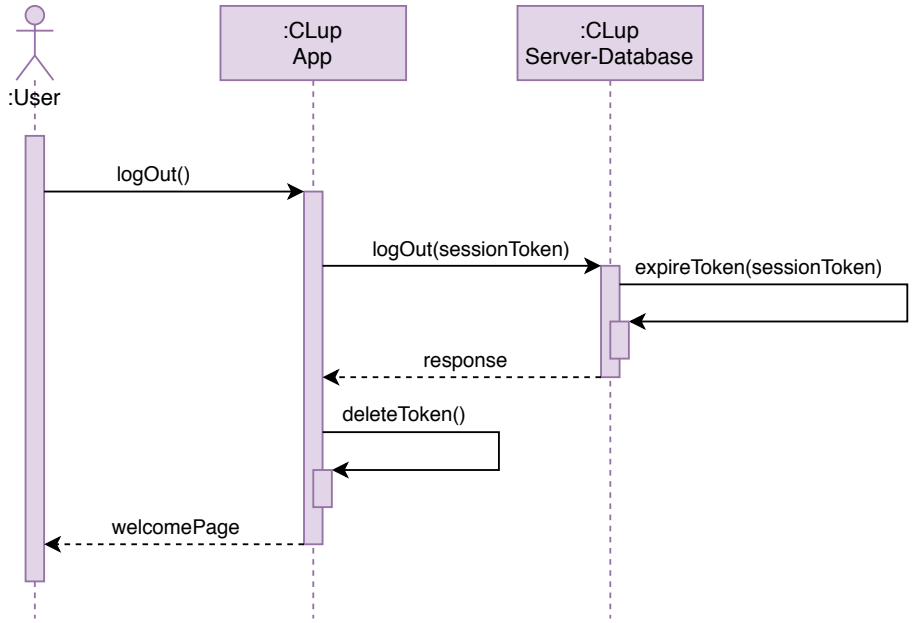


Figure 3.10: Log Out sequence diagram.

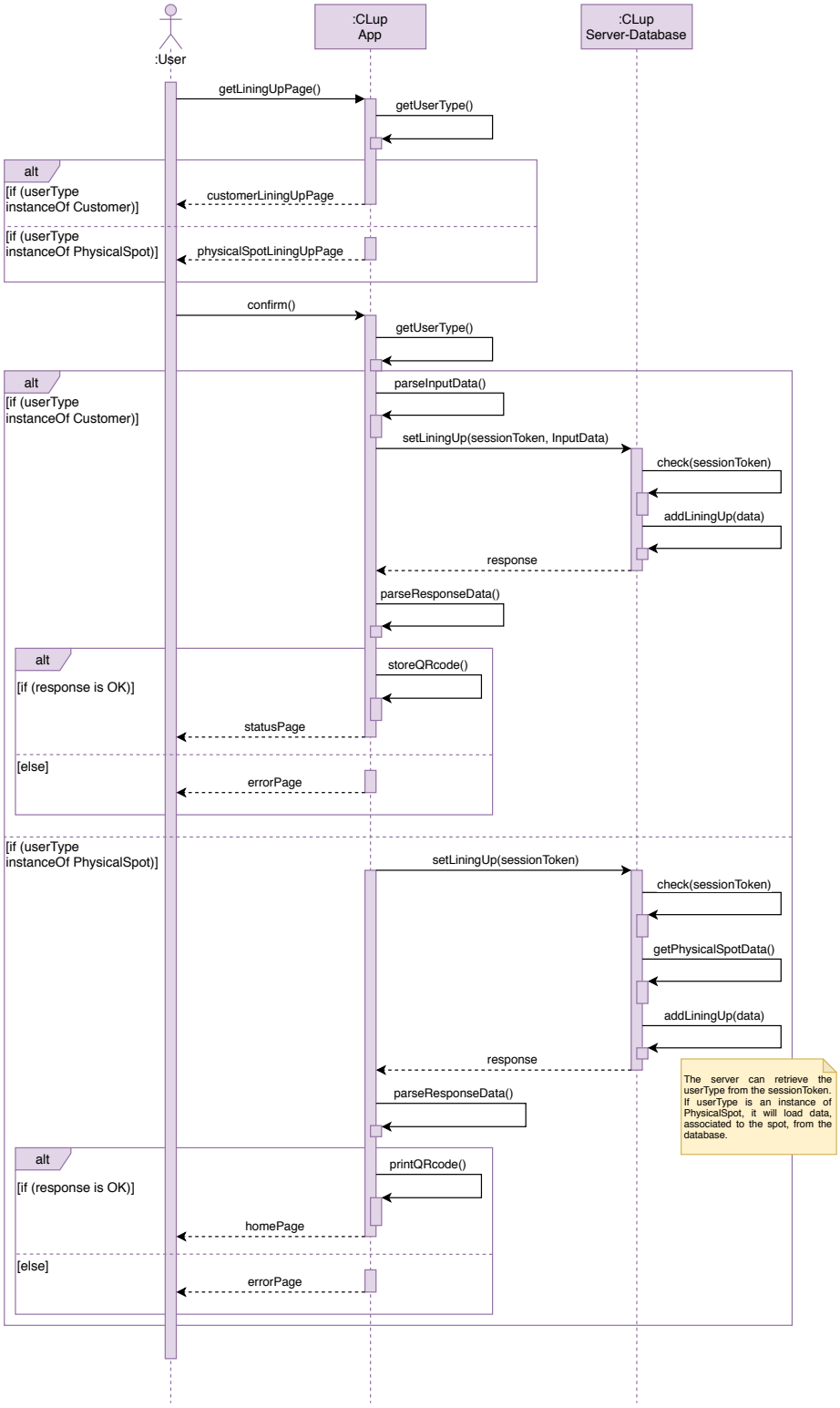


Figure 3.11: Lining Up sequence diagram.

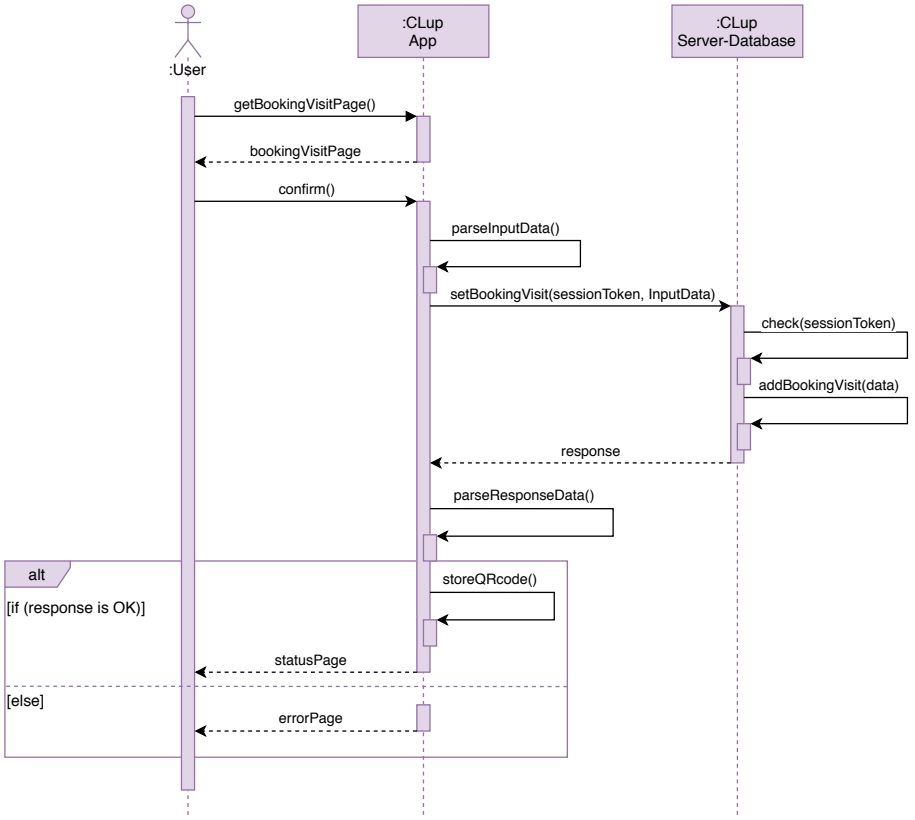


Figure 3.12: Booking a Visit sequence diagram.

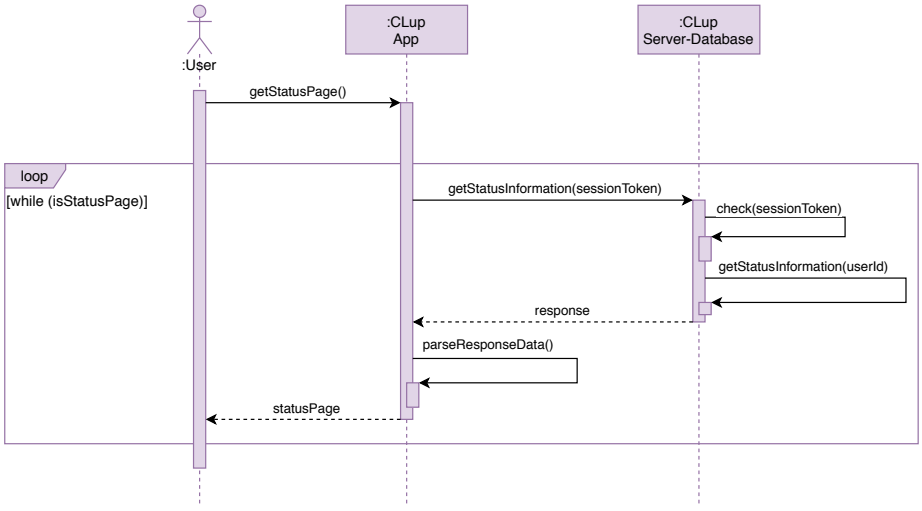


Figure 3.13: Get Status sequence diagram.

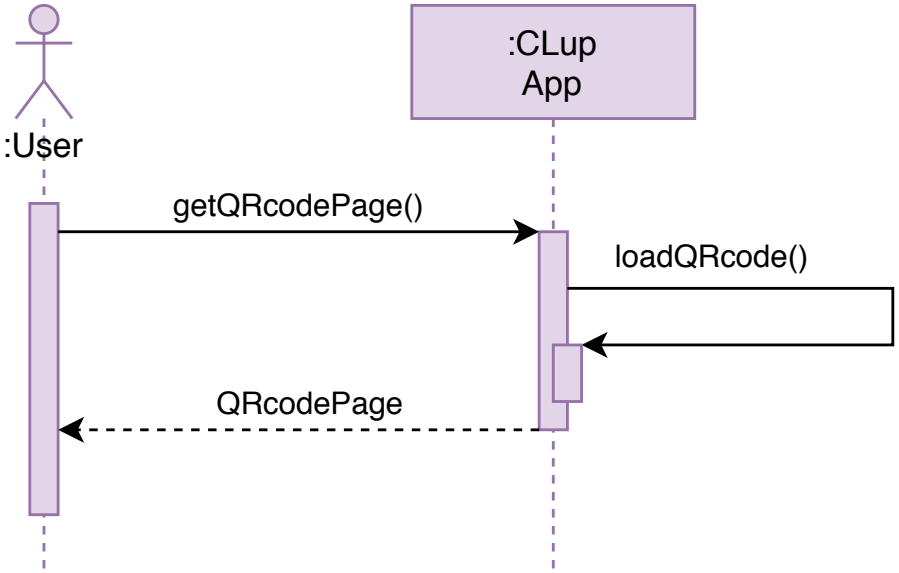


Figure 3.14: QR code page sequence diagram.

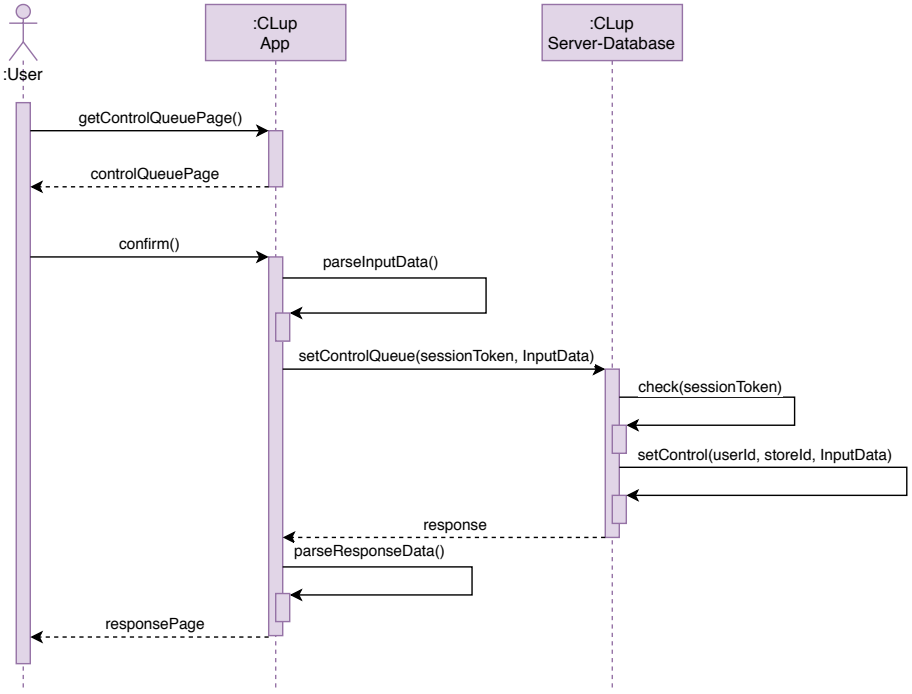


Figure 3.15: Control Queue sequence diagram.

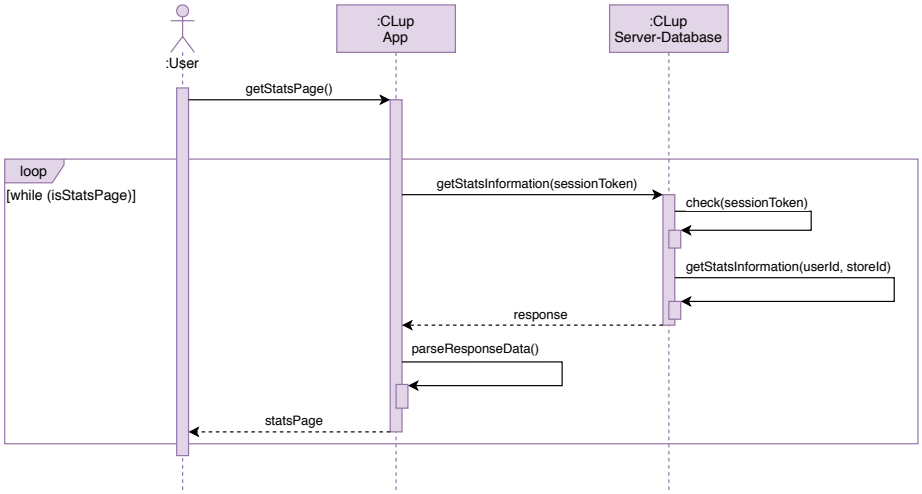


Figure 3.16: Show Stats sequence diagram.

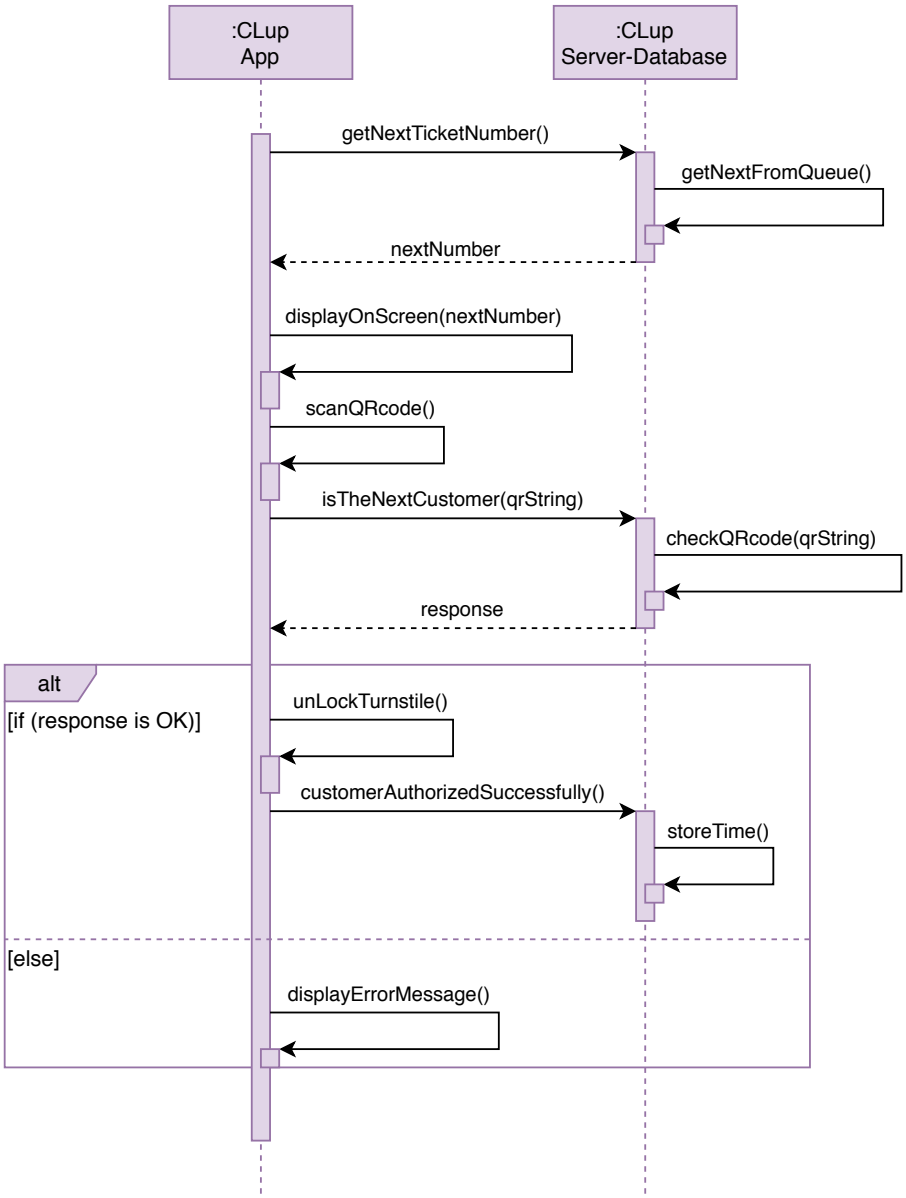


Figure 3.17: Sequence diagram reporting the activity periodically performed in background, by the application of the store manager, to authorize customers to enter in the store. For simplicity, the session authentication has been omitted.

3.2.5 Mapping on Requirements

3.3 Performance Requirements

We can differentiate our performance requirements in 2 categories:

- The first ones are necessary to respect a goal, in our case are relative to the goal to limit the physical line situation in the proximity of the store.

In order to achieve that, our application must precisely estimate the right time of arrival time of every single customer with an absolute error less than five minutes. That depends on the precise estimation of the position of the customer after lining up and a correct estimation on the duration of the customers time staying. Them also need to be under a certain value of error, for the first one the error must be less than ten meters instead the estimation of the time of the customer staying must be less than three minutes. Also, these estimations must not require more than twenty seconds to compute in order to be always relevant when it is shown to the customer.

- The second ones are required to guarantee a good level of QoS (Quality of Service). Such as being able to manage at least a five hundred simultaneous user connections for each store, in order to let the all customer being able to line up, re-elaborate periodically the stats for the manager with a period less than one minute.

3.4 Design Constraints

3.4.1 Standard Compliance

All measurements are made by using the SI standard of measurements:

- Distance [m]
- Time [s]

All information regarding the users and the data collected by the application will be kept safe, more information in that regard will be stated on an agreement the user will have to accept. Clup pledge to respect what it will be written in that agreement.

3.4.2 Hardware limitations

The most important hardware requirements are the ones used by the store, necessaries for the basic function of the service. It must have two different hardware components:

- One that has to acts as physical spot and so it is preferably a totem but in more general terms a tablet connected to a printer should suffice, and it is implied that the tablet or the totem should be also connected to internet to communicate with the Clup server. The tablet is needed for letting people choose to get a paper ticket and the printer should print and hand it to them.
- The second component must be able to scan the QR code in order to let people enter when It is their turn and reject them when it is not their turn. So, it is preferably to have a turnstile with a scanner implemented on it for this purpose. Also is suggested to have a screen that shows the current

position of the queue in order to reduce interactions. But the minimum requirement is to have at least a mobile scanner, that is required to be used by an operator that so needs to stay close to the entrance. Also, in this case, the scanner has to be connected to the internet or It has to be connected to another device in the store network that is connected to the internet in order to verify the correctness of the QR code.

There are also hardware requirements for the manager and the customers, the first one needs a device with a screen like a computer or a tablet, that is connected to internet for the entirety opening hours of the store in order to let the manager control and manage the influx of people. The customers need a mobile device with internet access and a GPS sensor, so a smartphone is preferred, in order to use both line-up and booking functionalities.

3.4.3 Any Other Constraint

The user cannot line-up if he is already in line The user must give to the app permission to GPS during all the time he is in line, without this permission the user will be not able to look at his ticket and show the QR code.

3.5 Software System Attributes

3.5.1 Reliability

The Clup system should be operating continuously as much as possible, so it should be up for at least ninety nine percent of time. The faults and problem in the system should not occur during times where there is a high influx of people as they could cause the most dangerous situations. So, the system must contain some mechanism to increase its fault tolerance especially in critical periods of the day like peak hours of stores. And have, extra mechanisms during high load periods like holidays. An alternative of the required hardware of a store should always be available in case of failures, and they both should regularly checked. The collection of data is not a particularly important aspect of the application, but it should be still robust, so the data should be backed-up regularly in physical storage devices.

3.5.2 Availability

As stated above, the operating time of the system, and so its availability should be at least ninety nine percent of time. Scheduled maintenance should be programmed during the period between midnight and five a.m. since it is the period when most stores are closed, and the working ones have small load.

3.5.3 Security

The data collected by the application is not particularly sensible, but even so the applications should focus on keep it safe, by mainly encrypting the data before storing it and by using an encrypted channel to transmit it. Furthermore, the development of all components of the applications should be done in a way that considers the latest possible exploit and the possible vulnerability of the system and aiming to cover them all.

3.5.4 Maintainability

The application should be designed by keeping in mind the design patterns of coding. Such as, dividing the application in functional components that are the least dependent on each other. So that their maintenance is done independently and does not spawn irregular behaviours and requires less downtime for the whole system to implement it.

3.5.5 Portability

The applications should made available for: Linux, Windows, macOS, Android and iOS for the manager device and physical spot Android and iOS for the customers device. It should also be available for some older versions of these operative systems without it being a constrain.

Chapter 4

Formal Analysis Using Alloy

4.1 Alloy Code

In this section is shown the alloy code that represent some parts of Clup model. The representation focus mainly on how the queue works with different kind of users, from the moment the line-up and so they appear inside this model to the moment they enter inside the store. Their place in the queue is represented by their ticket that encodes inside the QR code the information of lining-up and so of the user. That it is used to recognize the place that a person has in the queue and so to allow only the correct one to enter the store. It is also shown by the Alloy's graphic tool some worlds that evidence how the different components of the model and their properties work with each other in different context. And some properties are checked by using the assertion feature.

```
//USERS

abstract sig User {
}

sig Username{}{
        one this.~username
}
sig Password{}{
        one this.~password
}

// CUSTOMER: people who uses the app to line up
sig Customer extends User{
        username: one Username,
        password: one Password,
        digitalticket: lone DigitalTicket,
        bookingticket: lone BookingTicket
}

fact{
        no disj a,a': Customer | a.username = a'. username
}

//PHYSICAL SPOT: The totem located in the store
sig PhysicalSpot extends User{
        physicaltickets: PhysicalSpot one -> PhysicalTicket,
        printer: one Printer
}

sig Printer{}{
        one this.~printer
}
```

```
one sig StoreManager{
        manages: one Dashboard
}

//TICKETS

abstract sig Ticket {
        qrCode: one QRCode,
        ticketNumber: one Int,
        state: one State,
        waitingtime: one Int
}{
        waitingtime > 0
        ticketNumber > 0
}

sig DigitalTicket extends Ticket{
        gpsposition : one Position
}{
        one this.~digitalticket
}

sig PhysicalTicket extends Ticket{}

//RESERVATION TICKET
sig BookingTicket{
        qrCode: lone QRCodeB
}

fact {
        no disj t,t': BookingTicket | t.qrCode = t'.qrCode
}


//QRCode

sig QRCode {
        user: one User,
        time: one Int
}{
 this in ( DigitalTicket.qrCode + PhysicalTicket.qrCode)
 time > 0
}

sig QRCodeB {
        user: one Customer,
        time: one Int
}{
 time > 0
}


// POSITION

sig Position{
 }{
this in (DigitalTicket.gpsposition)
}

//QUEUE: List of people that are in line or rather that have a ticket

one sig Queue{
        contains: Int one -> Ticket,
}
//DASHBOARD: The interface that the store manager uses to control the people
    ↪ influx

one sig Dashboard{
        checks :one Queue,
        refers: one Store,
        controls: one Enters,
        manage: one BookingList,
        controls2:one Turnstile
```

```
}

sig QRCodeScanner {}{
        one this.~scanner
}
sig Turnstile {
        scanner: one QRCodeScanner
}{
        one this.~controls2
}


//STORE

one sig Store{
        capacity: one Int,
        peoplein: one Int,
        peopleinq: one Int,
        location: one Position,
        clock: one Int,
        departments: set Department
}{
        clock > 0
        peoplein + peopleinq < capacity
}

// DEPARTMENT & ITEMS

sig Department{
 items: set Item
}{
        one this.~departments
}

sig Item{}{
        one this.~items
}

//STATE: Contains the states that a ticket can have

abstract sig State{}
one sig AUTHORIZED extends State{}
one sig UNAUTHORIZED extends State{}

//ENTERS: People who are in the shop

one sig Enters{
        enters: Ticket set  -> lone Store,
        entersP: BookingTicket set -> lone Store
}

// BOOKING LIST : List of reservaion
one sig BookingList{
        containsb: set BookingTicket
}{
        one this.~manage
}

//FACTS

// Associates to two attributes of store(peoplein, peopleinq)  the numbers of
    ↪   istances 'enters' respectively from the Queue and from BookingList

fact NumInsideNumEnters { all s: Store | all e: Enters | s.peoplein = #e.
    ↪ entersP and s.peopleinq = #e.enters}

//Only people who have a ticket authorized can enter

fact Autorizedonlyifbeforeentered{ all t: Ticket | one s: Store | one a:
    ↪ AUTHORIZED | all e: Enters |  t -> s in e.enters implies t.state = a}

//One can enters only if he has the smallest ticket number and of the people
    ↪ that are not in the store and he is right on time
```

```
fact Autorizedonlyifbeforeentered2{all disj t,t': Ticket | one a: AUTHORIZED|
    ↪  one s: Store |   (t.ticketNumber < t'.ticketNumber and t.qrCode.time
    ↪  = s.clock) or ( t.ticketNumber > t'.ticketNumber  and t'.state = a
    ↪ and t'.qrCode.time < s.clock ) implies t.state = a }

//If One is authorized to enter, it means that all the people with a smaller
    ↪ ticket number is already entered

fact OrderedEntering { all disj t,t': Ticket | one a: AUTHORIZED  |  all e:
    ↪ Enters | one s: Store | t.ticketNumber >  t'.ticketNumber and t.state
    ↪  = a  implies  t' -> s in e.enters}

// Two tickets cannot have the same qr code

fact NoSharedQR{ no disj t,t': Ticket | t.qrCode = t'.qrCode}

//Two digital tickets cannot refer to the same user

fact NoSharedUser{ no disj t,t': DigitalTicket | t.qrCode.user = t'.qrCode.
    ↪ user}

//Having a smaller ticket number of an another person means that the entering
    ↪  time is also smaller

fact TimeandNum {all disj t,t': Ticket | t.ticketNumber < t'.ticketNumber
    ↪ implies t.qrCode.time < t'.qrCode.time}

//Two tickets cannot have the same ticket number

fact UniqueNumberPerDay{ no disj t,t': Ticket | t.ticketNumber = t'.
    ↪ ticketNumber}

// The user in a digital ticket has to be a customer

fact Circle{ all d:DigitalTicket| some u:Customer |  ((u.digitalticket + d).
    ↪ qrCode).user = u}

// The user in a physical ticket has to be a physical spot

fact SecondCircle{ all p:PhysicalSpot | all p2:PhysicalTicket | some q:
    ↪ QRCode | ((p + p2).qrCode + q).user = p}

// The user in a booking ticket has to be a customer

fact ThirdCircle{ all b:BookingTicket | some u:Customer  |  ((u.bookingticket
    ↪  + b).qrCode).user = u}

//Il numero del ticket viene considerato dalla queue

fact RightNumbers { all q: Queue | all t: Ticket | q.contains.t = t.
    ↪ ticketNumber }

//If a customer is in the store it means that has the same gps location

fact InsidePosition{all e:Enters | some d: DigitalTicket | some s: Store| d -
    ↪ > s in e.enters implies d.gpsposition = s.location}

//Having a smaller ticket number of an another person means that the waiting
    ↪ time is also smaller

fact waitingTime{ all t,t': Ticket | t.ticketNumber > t'.ticketNumber implies
    ↪  t.waitingtime ≥ t'.waitingtime }

//One items can be found in only one department

fact ItemInOneDep { all d,d': Department | all i: Item | i in d.items implies
    ↪  i not in d'.items}

//Every person who has a booking ticket is considered inside the store if
    ↪ their entering time is passed

fact BTEnters {one e: Enters | some b: BookingTicket | one s: Store | b.
    ↪ qrCode.time < s.clock implies b -> s in e.entersP}
```

```
//Booking tickets are always in a booking list and associated with an user

fact NoBTalone { all b: BookingTicket | some u:Customer | some bl:
    ↪ BookingList | b in BookingTicket implies u.bookingticket = b and b in
    ↪  bl.containsb}

//QRCodeB are always associated to some booking ticket

fact{  all b: BookingTicket | some q:QRCodeB | q in QRCodeB implies b.qrCode
    ↪ = q}

//Assertions

//We check that if the store capacity is not reached, one person can enter,
    ↪ it also to show the enter predicate

assert  EntersifNotFull { all e, e': Enters | some t,t': Ticket| one s: Store
    ↪ |  s.capacity > (s.peoplein + s.peopleinq) implies Entra[e, e',t,t',s
    ↪ ]
}

// We check that if the store capacity is reached, one person cannot enter

assert  NoEntersifFull { all e, e': Enters | some t,t': Ticket| one s: Store|
    ↪   s.capacity < (s.peoplein + s.peopleinq) or s.capacity = (s.peoplein
    ↪  + s.peopleinq)  implies Entra[e, e',t,t',s]
}

//We check that at most one person at a time is allowed to enter

assert  NumPersoneAutorizzateFuori1 {one e: Enters | all t,t': Ticket | one a
    ↪ : AUTHORIZED | one s:Store | t.state = a and t'.state = a and t -> s
    ↪ not in e.enters and t' -> s not in e.enters  iff t = t' and t.state=
    ↪ a and t'.state = a  and t -> s not in e.enters and t' -> s not in e.
    ↪ enters}

//PREDS

// Eliminates one relationship 'enters' to Enters

pred Esce( e, e': Enters, t, t': Ticket, s: Store ){e'.enters = e.enters - t
    ↪ -> s and t'.state = UNAUTHORIZED }

// Adds one relationship 'enters' to Enters

pred Entra[e , e': Enters, t,t': Ticket, s: Store]{
         t'-> s not in e.enters implies e'.enters = e.enters + t'-> s and t'.
             ↪ state = AUTHORIZED and t'.ticketNumber = t.ticketNumber
}

pred test{}

//SOME WORLDS

//Shows clearly how the entities work with each other and that the costrains
    ↪ holds.
pred SmallNumber {
#DigitalTicket = 1
#Customer = 1
#QRCodeB = 1
#BookingTicket = 1
#PhysicalTicket = 0
#PhysicalSpot = 0
#Queue = 1
#Enters.enters = 0
#Enters.entersP = 0
#Department = 0
}

//Shows that the entities work with each other and that the costrains holds
    ↪ also for large numbers
pred BigNumbers {
#DigitalTicket = 4
#Customer = 4
```

```
#QRCodeB = 2
#BookingTicket = 2
#PhysicalTicket = 3
#PhysicalSpot = 1
#Queue = 1
#Enters.enters = 2
#Enters.entersP = 2
#Department = 2
}

//START PROGRAM


//check  EntersifNotFull

//check  NoEntersifFull

//check  NumPersoneAutorizzateFuori1

//run SmallNumber for 3

//run BigNumbers for 8
```

```
Executing "Check NoEntersifFull"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  8668 vars. 558 primary vars. 22543 clauses. 281ms.
  No counterexample found. Assertion may be valid. 31ms.

Executing "Check NumPersoneAutorizzateFuori1"
  Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
  8557 vars. 556 primary vars. 22267 clauses. 101ms.
  No counterexample found. Assertion may be valid. 173ms.
```

Figure 4.1: Assertions verification

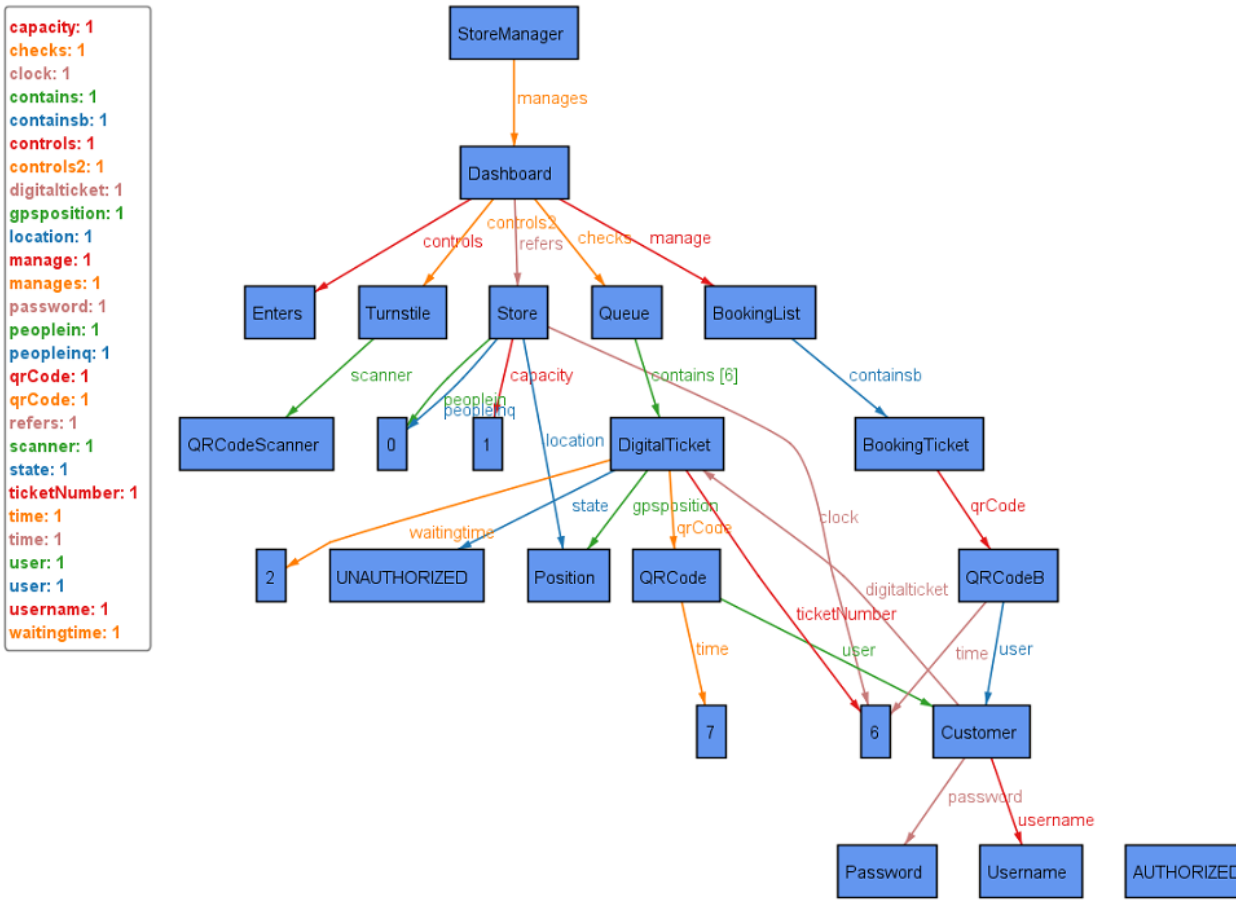


Figure 4.2: World1

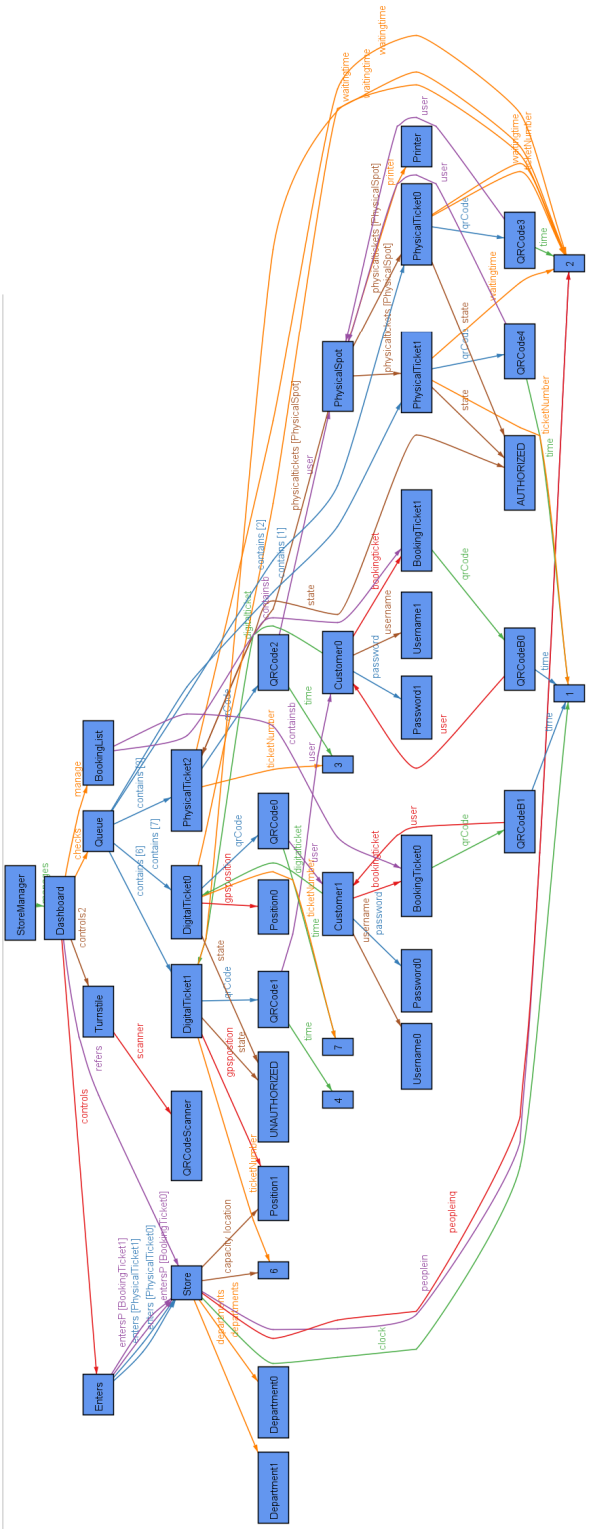


Figure 4.3: World2

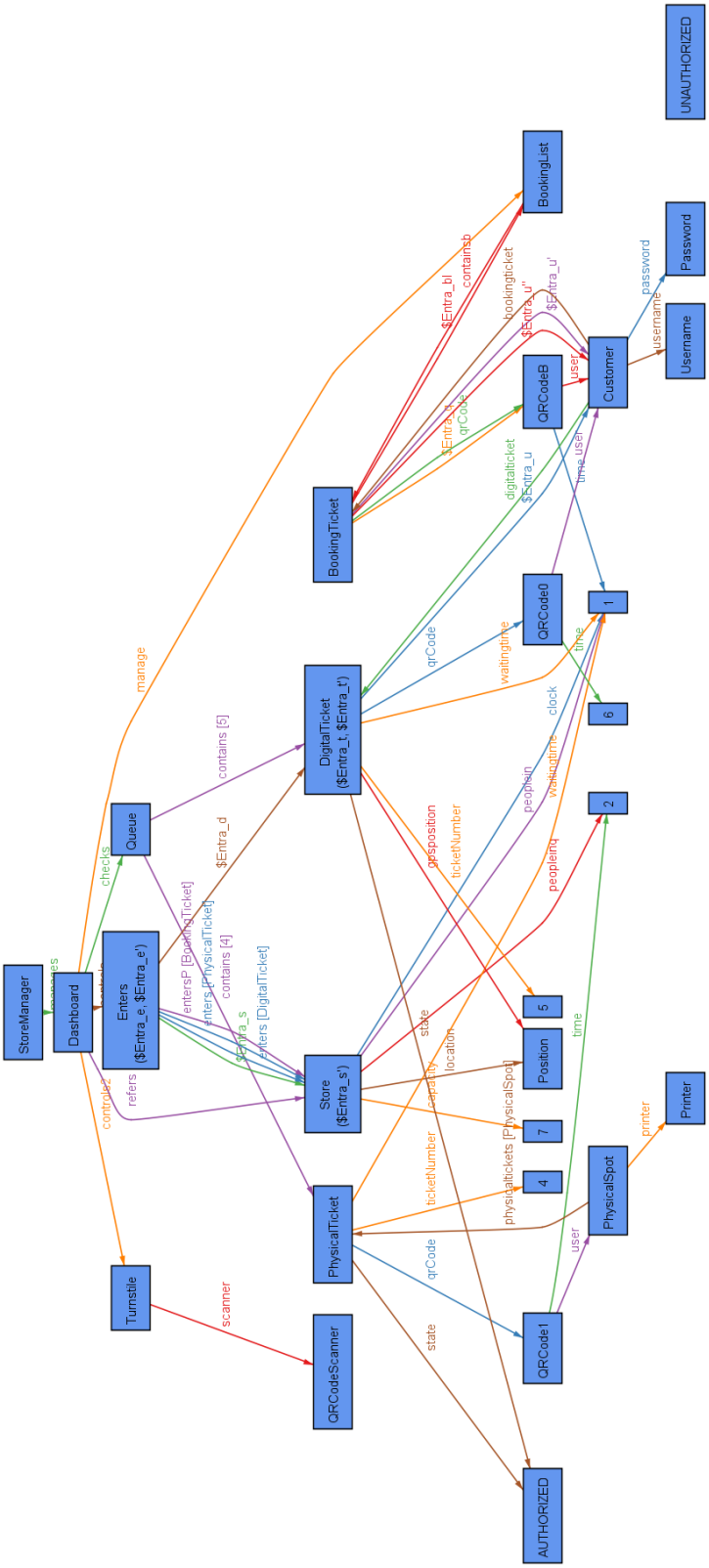


Figure 4.4: Entra function

Chapter 5

Effort Spent

Chapter 6

References

Bibliography

[1] Athena-fidus. `https://www.asi.it/it/flash/telecomunicazioni-e-navigazione/athena-fidus`, 2018.

[2] Dirk Gómez Depoorter, Omar Raissouni, Elisenda Temprado Garriga, and Oliver Lücke. The across testbed for the future aeronautical data communications. Technical report, IEEE, 2016.

[3] SESAR, Joint Undertaking. *Single Programming Document*, 2017-2019.

API	Application Programming Interface
CLup	Customers Line-up
d.P.C.m	*"decreto del Presidente del Consiglio dei ministri"*
FIFO	First In First Out
GPS	Global Positioning System