RASD - Requirement Analysis and Specification Document

POLITECNICO
MILANO 1863

Edoardo Venir - 10570524 - Mat. 962566
Leonardo Ruzza - 10608001 - Mat. 963206

<Version 1.0 - 13/11/2020>

Repository Link

Contents

1 Introduction

1.1 Purpose

1.1.1 RASD Purpose

This document represents the Requirement Analysis and Specification Document (RASD). Goals of this document are to completely describe the system in terms of functional and non-functional requirements, analyse the real needs of the customer in order to model the system, show the constraints and the limit of the software and indicate the typical use cases that will occur after the release. This document is addressed to the developers who have to implement the requirements and could be used as a contractual basis.

1.1.2 Application Purpose

CLup - Customers Line Up - is born to help people in a crucial historic period, Covid-19 era, in which the daily life of all us changed drastically. One of these changes, and probably the most important one, is the social distancing to avoid the spread of pandemic.

Indeed, the specific goal of this project is to develop an easy-to-use application that, on the one side, allows store's managers to regulate the influx of people in the building and, on the other side, saves people from having to line up and stand outside of stores for hours on end, which are themselves a source of hazards.

The application would work as a digital counterpart to the common situation where people who are in line for a service retrieve a number that gives their position in the queue. This method of ticketing allows a person to approach the store in time and in a more safe mood, only when his/her number is going to be reached.

In particular, CLup will provide a real-time ticketing service which, on request, give to the customer the number associated to the current store's queue that the customer want to visit. The stores adopting CLup system, can also guarantee the possibility to hand out tickets on the spot; of course this two approaches are integrated in a unique queue, always managed by CLup. CLup will also allows customer to book a visit in advance to the store, specifying in many ways the times and modalities of his/her visit.

The application should be very simple to use, as the range of users include all demographics.

1.1.3 Goals

[G1] Allow a person to become a registered User providing some basic personal info (like Name, Surname etc.).

[G2] Allow a store's Visitor to pick up a Physical ticket from the store itself.

[G3] Allow a User to reserve in real-time a ticket for the current queue of a particular store:

- **[G3.1]** specifying the approximate expected duration of the visit.
- **[G3.2]** confirming the approximate expected duration of the visit suggested by the system (for User who is a long-term customer of a store).
- **[G3.3]** specifying the means of transport.

[G4] Allow a User to book a visit to a store:

- **[G4.1]** specifying the approximate expected duration of the visit.
- **[G4.2]** confirming the approximate expected duration of the visit suggested by the system (for User who is a long-term customer of a store).
- **[G4.3]** specifying date and time of the future visit.

[G5] Allow a User/Visitor to enter in the store within the use of QR code when it is his/her turn.

[G6] Allow a User to search and find available stores:

- **[G6.1]** within a certain distance from his/her current location for the real-time ticketing.
- **[G6.2]** within a Country constraint for the booking option.

[G7] The system should provide Users/Visitors with a reasonably precise estimation of the waiting time associated to their respectively instant and physical tickets.

[G8] The system should alert in time a User to reach the store, for which he took the instant ticket, taking into account the time he need to get to the store from the place he currently is.

[G9] Allow a System Manager to do operations on the system for updating and maintenance:

- **[G9.1]** Allow a System Manager to add a store to the system.
- **[G9.2]** Allow a System Manager to delete a store from the system.
- **[G9.3]** Allow a System Manager to edit timetable of a store.
- **[G9.4]** Allow a System Manager to edit maximum people capacity of a store.
- **[G9.5]** Allow a System Manager to see all the stores registered into the system, or search for a specific one using its commercial name and/or address.

- **[G9.6]** Allow a SM to edit the default permanence time of a store.

[G10] Allow a User/Visitor to see the current ticket number on the store's queue board.

1.2 Scope

CLup might be adopted by every type of stores all over the world. Of course, it will be mainly used by stores who sells necessities and who are subject to great turnover of people. A small shop will not benefit as much as a large one from the use of the application. Moreover, CLup is not intended for these activities, which already have a booking system, since it will need to be integrated with the existing system.

This project is strictly related with Covid-19 period, but nothing forbid that avoid a lot of queues and save a time would be very fine for people. So, a pandemic is not the only scope of action of CLup.

The application mainly aims to simplify the access to stores for customers, and at the same time it aims to help owners managing people's flow in their business. In addition, CLup can be useful in the people day's planning, to improve their time management; thanks to the possibility of book a visit to the store in advance, to the real-time knowledge of queues in shops and of the travel time a person can decide whereas today is a good day to go shopping or not.

The protagonist of the CLup's scope are mainly people and stores. CLup should offer the possibilities to improve their relations and let positively grow the approach between customers and shops.

Where is true CLup also might offer the chance for the users to indicate the categories of products/service they are going to purchase, in order to allow the system to better manage Maximum People Capacity, the improvement and management of the act of buy itself – such as for example indicate to the user the max quantity of a product available in the store - is completely outside from CLup's scope.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Ticket: represents the reservation of a place in a queue for an available store and always includes a number and a QR Code.

Instant ticket: Ticket associated to the Real-time ticketing, which also includes the waiting time.

Booked ticket: Ticket associated to the booking of a visit, which also includes the reserved date and time.

Physical ticket: Ticket associated to a visitor (so, the physical version of the instant ticket of users), which also includes the waiting time.

Line-up: retrieve a ticket.

Available Stores: shops which adopt CLup, and so, available in the system to CLup's user/visitor who want to reserve a ticket or book a visit to them (for simplicity almost always referred to by writing only "stores", but not to be confused to the generic ones, and it would be clear by the context).

Queue: persons who are waiting to enter in the store.

Real-time ticketing: the supply service of tickets, which allow users to line up for a current queue ("real-time") of a shop.

Book a visit: the reservation of a ticket only in advance, which is different from a real-time reservation.

Current ticket number: the number to be served now (this is the number "calling").

Waiting time: the amount of time reasonably estimated that a user/visitor will wait before his turn.

Travel time: the amount of time reasonably estimated that a user will spend to reach the store.

Permanence time: estimated time the user/visitor will spend in the store. Also, the approximate expected duration of the visit.

DefaultPermanence-time: default estimated time the visitor will spend in the store. This is a system parameter set for every store.

Maximum people capacity: the maximum amount of people that may be present at the same time in a store. This is not computed dynamically by the system but predefined by stores' owners.

CLup store's devices: It include Queue Board, Ticket Printer and QR Reader.

System: the total software and hardware infrastructure which CLup is composed of.

Check-in: passing the QR code of a ticket on the QR code reader of the store to enter.

QR code Reader: a device connected to CLup's system which can read QR codes generated with the tickets in order to check-in users/visitors.

Queue Board: a particular screen connected to CLup's system which show the current ticket's number.

Ticket Printer: a particular printer connected to CLup's system which print the first available ticket for the current queue acting as a proxy for a visitor.

1.3.2 Acronyms

RASD: Requirement Analysis and Specification Document.

API: Application Programming Interface.

CLup: Customers Line-up.

S2B: SoftwareToBe

GPS: Global Positioning System

UI: User Inteface

1.3.3 Abbreviations

Gn: n-goal.

Dn: n-domain assumption.

Rn: n-functional requirement.

SM: System Manager

1.4 Revision History

1.5 Reference Documents

[1] R&DD Assignment AY 2020-2021

[2] M. Jackson, P. Zave, "Deriving Specifications from Requirements: An Example", Proceedings of ICSE 95, 1995

[3] Slides from lectures of "Software Engineering 2" course AY 2020-2021 at Politecnico of Milan

[4] Slides from lectures of "Ingegneria del Software" course AY 2019-2020 at Politecnico of Milan

1.6 Document Structure

The RASD is composed by 6 parts:

1. In the first part an introduction to the problem is given listing all the identified goals, describing the scope of the project and providing some basic information to better understand the other sections of the document.

2. The second part consists of an overall description of the system including a product perspective (with scenarios, further details on the shared phenomena and a domain model), the main product functions (requirements), user's characteristics (to clarify his necessities) and all the fundamental domain assumptions (to complete the customers' specification given and considered true in the project development).

3. The third part develops all the most important aspects of Section 2 which can be useful for the development team. Indeed, it highlights all specific requirements identified, both functional and non-functional.

4. The fourth part contains a brief presentation of the main objectives driving the formal modelling activity, as well as a description of the project's Alloy model itself, what can be proved with it, and why what is proved is important. To show the soundness and correctness of the model, this section shows some worlds obtained by running it, and the results of the checks performed on meaningful assertions.

5. The fifth part is accessory and summarizes in detail the hours spent in the document production.

6. The sixth part is the last and is composed by all the references of the tools used to redact this document and its contents.

2 Overall Description

2.1 Product Perspective

The system will be developed from scratch. It will needs support only for the travel-time estimation, like a specific API provided by Google Maps (it will be discuss more in the DD document), but all the rest will be done without using other external System.

As already said, CLup gives the user the estimated waiting time to enter in a certain store. By entering the system, first of all he/she can choose between the 2 types of ticket: instant one or booked one; every ticket category is associated to a group of available stores for each user, relying on user's GPS position and country. Then by inserting store's name and/or address in a search bar, user can select the desired store.

CLup can calculate how much time a person generally stays in the store, relying on average time (custom-fit to the person if they are usual ones) and info given by the user while booking or in real-time ticketing.

Thanks to the option to indicate the means of transport during a real-time ticketing, CLup will be able to give users alert when they need to leave to be in the store in time, relying on the GPS position provided by the user.

It's important to underline that from the system point of view the queues are managed like "static", thus when a ticket is created, the system decide "a priori" when it will be "called", and this does not change in future. It will be possible thanks to the precise estimation and manipulation of waiting times, permanence times, travel times and all the following assumptions/requirements in this document.

Instead, from the business owners' point of view, CLup give the possibility to edit timetables and maximum indicative amount of people at a given time, to better fit different situations (not by her/himself, only through a SM).

As is implicitly understood, the events catchable by CLup's system are the transmission of information from an user through the application, similarly the physical lining up by a visitor who interact through a ticket printer and the QR code scanning by an user/visitor using the QR code reader. Instead the main events throwable by the system to the real world are the showing of the current ticket number in the queue board, and the notifications (like the alert to reach the store, the response during registration etc.) and ticket provided to an user/visitor.

2.1.1 Scenarios

2.1.1.1 Scenario 1

Speaking with a friend, the elderly Mrs. Maria learns that her trusted bakery, always very crowded, has adopted CLup. Maria's friend Sofia advises her to download the handy app to avoid the endless queue to be served. However, Maria, who has always been reluctant to technology, is not equipped with a modern smartphone that supports CLup. But going to the bakery as usual, she finds that at the entrance there is a ticket printer integrated with the CLup virtual management system. Also, the shop is not at all crowded as usual. The lady then goes to the ticket printer and takes her ticket, where is printed a number, a QR code and the estimated waiting time. Maria sees that she has time to do some more shopping during the wait. Then, when her number is called, she enters via the QR code and she is promptly served.

2.1.1.2 Scenario 2

During the ongoing Covid-19 pandemic, Fabio finds out from his colleague Antonio that the next-door supermarket has finally renewed the queue management system at the way in to limit entrances and reduce crowds. So, Fabio decide to download the CLup app. He registers by indicating the necessary data, and after receiving a confirmation e-mail containing the password of his new account, he logs in. In the app's homepage, Fabio chooses to get an instant ticket, he look for the supermarket of interest among the various shops available near him, and clicks to get an instant ticket, indicates the permanence time, selects the item "on foot" as a means of transport and finally obtains his ticket within an estimated waiting time. After a few minutes, he receives a notification inviting him to reach the supermarket to arrive on time. Once arrived, he does the check-in with the QR code associate to the ticket shown on the app, and he enters the supermarket.

2.1.1.3 Scenario 3

Luigi has been using CLup for months to book his visits to the post office in the neighbourhood. Once the steps for accessing and searching for the office in the app have been completed, he decides to book a visit by selecting 5pm of the following day. At this point Luigi confirms the permanence time suggested by the app, thus he obtains the ticket. The next day Luigi goes to the post office, he sees on the office's board that the current ticket number it's his one, so he knows he can now do the check-in using the QR code on the ticket.

2.1.1.4 Scenario 4

Caterina is a very healthy girl. She decides next week she will go to one of the organic shops she frequents, "BioFood". So, she opens CLup on his smartphone and finds BioFood in the booking section among the stores available, she selects as desired date and time the next Saturday at 17 pm, finally she proceeds by indicating an approximate stay time of

25 minutes. She has now obtained her ticket. Unfortunately, on that Saturday Catherine is unexpectedly called to work for an emergency. Her number appears at the time of booking on the screen in the shop, and after a few minutes that she has not shown up, another customer's turn comes.

2.1.1.5 Scenario 5

It is Friday evening, and the new season of Francesco and Luca's favourite TV series is out. For this event, Francesco invited Luca to his house. They decide to eat burgers with fries, ordering them from an excellent fast-food. Luca, being the guest, wants to take care of buying the dinner. While he is preparing to reach Francesco, he opens his CLup application from the smartphone, he enters his credentials to log in and chooses the section of instant ticket. Then he searches for this famous fast-food restaurant and once found, selects it. Then he specifies the time of the visit of 10 minutes, he selects "car" as a means of transport and finally gets his ticket, discovering that it will take about half an hour before being served. Fifteen min. later he receives a notification from CLup inviting him to leave to arrive at the fast-food restaurant on time. Once there, after two minutes the number "54" appears on the board, it's his turn! So, he scans his QR code at the entrance, and finally he can order two Special Burger Menus with double fries.

2.1.1.6 Scenario 6

In the last period, Valerio and Giulia have been saving money to be finally able to buy a television for their new apartment in which they now live together. Therefore, they decide to not waste more time, they open CLup from Valerio's phone, click to get an instant ticket, and look for the "TechWorld" store in their city. At this point, they indicate the permanence time rather than confirming the one suggested by CLup based on their previous visits (which had been easier and faster than buying a TV). Finally, they select "car" as their means of transport, as public transport or walking would certainly not be appropriate choices! Once this last step is also completed, they notice a short waiting time, in fact only after a few minutes CLup notifies that it is time to leave. Once on site, their turn is immediately arrived, they check-in with the ticket, and after completing the purchase, they go out with their new TV in the cart.

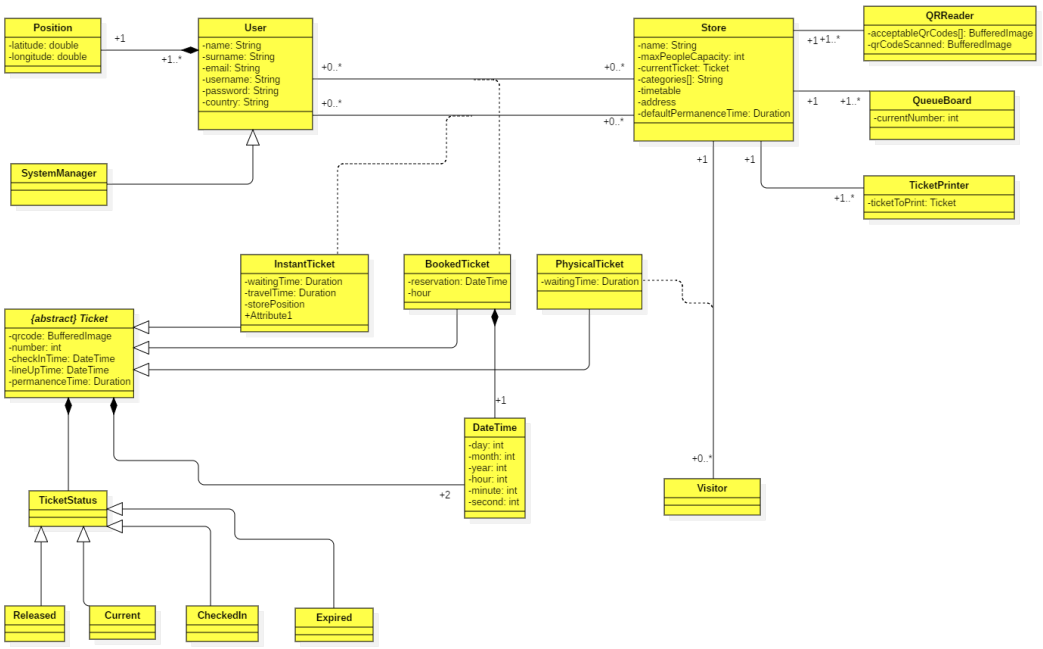
2.1.1.7 Scenario 7

Marco, a CLup's system manager, has been charged to add a new store and edit the Maximum People capacity of an available store to CLup. So, he opens the browser on his PC and goes to CLup System Managers access page to the insert his credentials to login. Once in the home page, he selects the button "Add new store" and then he inserts all the necessary information in the dedicated page's field. Then he confirms the operation, and he goes back to the home page. Now he selects the button "Search store" and insert the store's name and address he needs to edit in the search bar. In the list of

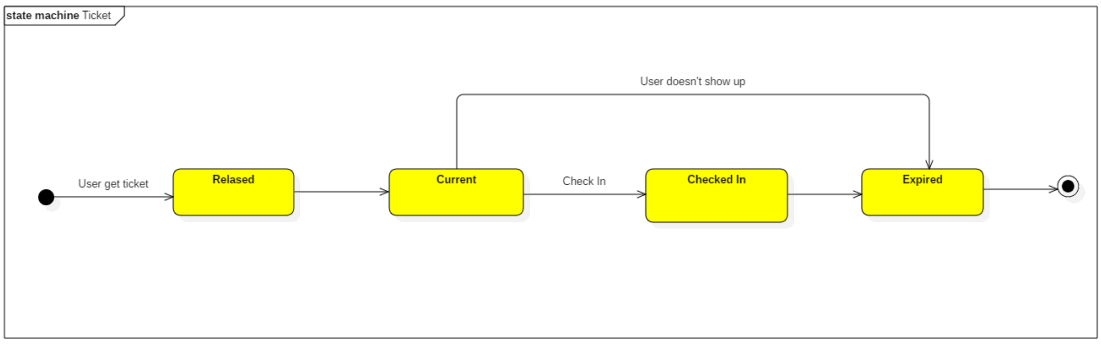
proposed fields, he chooses the Maximum People capacity, and he inserts the new value, finally confirming his choice.

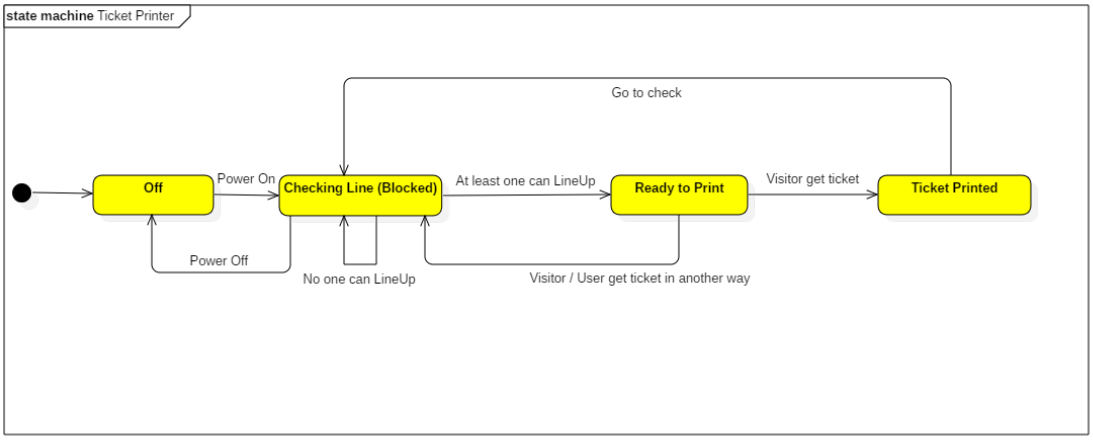
2.1.2 UML Description

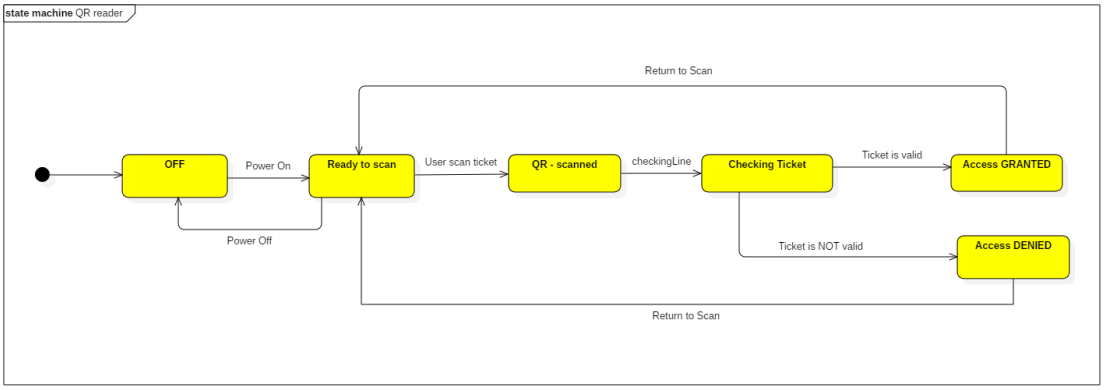
2.1.2.1 Class Diagram



2.1.2.2 State Chart




state machine Ticket Printer
Go to check
Off
Power On
Checking Line (Blocked)
At least one can LineUp
Ready to Print
Visitor get ticket
Ticket Printed
Power Off
No one can LineUp
Visitor / User get ticket in another way


state machine QR reader
Return to Scan
OFF
Power On
Ready to scan
User scan ticket
QR - scanned
checkingLine
Checking Ticket
Ticket is valid
Access GRANTED
Power Off
Ticket is NOT valid
Access DENIED
Return to Scan

2.2 Product functions

Here we provide a list of all the principal functions provided by CLup to better understand the system boundaries. Of course, all the goals identified in the previous part will be offered as functions of the system in development; here we want to give a high-level overview of the most important requirements, also focusing on what the system will not offer. For a complete specific list of all the application requirements, refer to the dedicated section.

2.2.1 Real Time Ticketing

The system must guarantee the possibility for every registered user to obtain a ticket (Line Up) in order to be able to enter in an available store via the use of the application (only for registered users). Moreover, the system must check that every person can only get an instant ticket for one store at a time; for the in-person situation (physical tickets) the system cannot ensure this property.

2.2.1.1 Estimate Travel Time

For the real time ticketing, the system will also give information and alerts about the estimated travel time from user's current position (while ticketing) to the store where he/her has Line Up. This is done thanks to the Google Maps API, so the precision and accuracy of this info is outside the S2B's control.

2.2.2 Ticket in place

The system must guarantee to people who does not want/cannot use the application to retrieve a physical ticket directly at the store from a Ticket Printer. The physical tickets are of course integrated in a unique queue for each store with the other ones (instant and booked). The reliability and maintenance of the hardware components of the Ticket Printer will not be of interest for the S2B, but only of the stores and a specific CLup's department.

2.2.3 Book a visit

The system must guarantee the possibility for every registered user to Book a visit to a store in advance, and that at the scheduled time (chosen by the user) the user will be able to enter in the store with no waits. As for the instant tickets, the system must check that every person can only get a booked ticket for one store at a time.

2.2.4 System Management

The system must guarantee that a SM will be able to edit store's "parameters" such as timetables, Maximum People Capacity, and default permanence time. In addition, the SM will be able to see all the store registered to CLup until that moment and add

or delete one of them. The interaction between store's owners/managers and CLup employees to take these decisions will not be support by the S2B and it's outside of the RASD's interest.

2.2.5 Check in with QR code

In every available store, users must be able to do the check-in. It must be tracked and considered by the S2B. The reliability and maintenance of the hardware components of the QR code reader will not be of interest for the S2B, but only of the stores and a specific CLup's department.

2.2.6 Queue Board

Every store's screen must be updated in real-time and show correctly the next called number, to let the called user know when they can enter or not. Similarly, to the QR-code reader, the reliability and maintenance of the hardware components of the Queue board will not be of interest for the S2B, but only of the stores and a specific CLup's department.

2.2.7 Others

- As anticipated in the Scope, the system might allow the user to indicate the categories of items/services he/she is going to purchase, but this function is not taken into account by the S2B and therefore not even in the managing of the Maximum People Capacity (which is considered static at the moment). But this system's feature can be added in future releases to compute dynamically the Maximum People Capacity.
- A possible future development of CLup can provide a new feature to allow a user to cancel a ticket. In this case it will have to be considered that the queue will probably need to become more "dynamic".
- Another future possibility is to allow a SM to reset manually a queue of a store, and so the associated tickets (alerting users). This can be useful in case of unpredictable problems reported maybe by the store's manager/owner.

2.3 User characteristics

2.3.1 Actors

User: a person passed through a successful registration process and now able to use all the CLup services. He/she can login to the system and, after that, use all the platform's functionalities.

Visitor: a person using CLup trough a ticket printer present in a store to line-up without having to be registered. He/she can be also a person using CLup to being registered.

System Manger: an employee of CLup able to maintain and update the system. Registration for this kind of users is not possible and they have to be added directly during system's installation process.

Store: shop which adopt CLup, and so, available in the system to CLup's user/visitor who want to reserve a ticket or book a visit to them.

2.4 Assumptions, constraints

2.4.1 Text Assumption

- The possibility to indicate the categories of products/services the user intends to buy, is not considered of primary importance for this first development of the S2B, but only like a possible future feature.
- There are "3 categories" of ticket: Instant, Physical and Booked ones. But they are merged in a unique queue for each store.
- Each user can Line Up only once a time per ticket category, not including the physical ones. This means he/she can only have 1 ticket associated per category, Instant or Booked one, but not for the same store.
- A generic person, to become a user, must provide the following information: name, surname, email, username, and country.
- The password is sent to the user's provided email during the registration process, to confirm user's identity.
- To access the system a system manager must provide the username and password associated to him/her.
- Stores should be added to the system by a SM. Since a store is added it immediately becomes available.
- Store's owner/manager should contact CLup's dedicated department, which will charge an SM, to add/delete the store or edit its features (like maximum people capacity, position etc..) from the system.
- Real-time ticketing function for a user is limited to the available stores nearby the user's actual position.
- User can take advantage of the booking function for all the available stores in his/her Country.
- Every ticket has associated a number and a QR code, in all the cases.While a QR code is unique in all the system, indeed the number is unique only during the daily store's opening.

- All the available stores must have at least one queue board, one QR code reader and one ticket printer.
- A user does not receive a notification by the system to inform him/her that is time to leave for a booked ticket in advance, because for example it would require a continuous GPS tracking of the user him/herself. Obviously, he/she does not specify a means of transport booking a ticket.
- To get a ticket in place, a visitor does not need to give the system any information (to speed up this physical service); but the physical tickets themself will be treated in the same identical way of the instant ones in the queue.
- During a real-time ticketing, there are three supported means of transport available: "on foot", "public transport", "car".
- Users and visitors rarely miss the arrival of their turn thanks to the information and notifications provided by CLup, therefore is not necessary design an appropriate product function to cancel a ticket (it could be a future development).
- The number of visitors is proportionally very smaller than the number of users, thanks to the simplicity of the application.
- A user is obviously considered a visitor when he/she get a ticket in place to line up.
- If a user/visitor does not show up and do the check-in when his/her turn come, the rest of the queue is not updated in order to respect the waiting times (which are "static") specified to all the next users/visitors in the queue. For example, a user cannot be instantly in a store, if he was expecting to wait other time and an alert to start the travel, because someone does not check-in.
- Users provide the approximated duration of a visit both in real-time ticketing and booking to allow the system to create a queue exploiting this information.
- Queue board is a necessary device to allow visitor knowing what's the current ticket (for example, it was possible to substitute the queue board with notifications of the current ticket to users, but the visitors can't be reached by them for definition).
- The travel time is statically calculated considering the position of the user while he/she is retrieving the ticket.

2.4.2 Domain Assumption

[D1] The Username must be unique.

[D2] When the system sends a password to a new registered User this will be surely received by him/her.

[D3] A store has at least 1 ticket printer available.

[D4] Every CLup store's device is able to communicate with the system.

[D5] Every physical ticket always contains correct information (no "printing error").

[D6] A visitor always retrieves his/her physical ticket.

[D7] A visitor only retrieves 1 physical ticket per visit.

[D8] The real visitor's permanence time in a store is always no longer then a maximum delay time over the default permanence time known for the store by the system (the maximum delay can be represented in percentage respect to the default permanence time).

[D9] The means of transport selected by a user is the one he/she will use for real.

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[D11] If GPS show the user in a certain position, it means he/she is there for real.

[D12] A user/visitor every time check-in within the QR code entering in the store.

[D13] A user/visitor every time enter in the store only if the check-in goes successful (QR code accepted).

[D14] Every QR code reader always read correctly and show correct feedbacks.

[D15] A store has at least 1 QR reader available.

[D16] Each store can be identified uniquely by its name and address.

[D17] A user start going to the store associated to the instant ticket when the specific notification is received (almost immediately).

[D18] There are no variations between the estimated travel time and what will be used by a user.

[D19] Notifications are certainly received by the user.

[D20] User position is retrieved by GPS.

[D21] Once added by an SM, a store become immediately available in the system.

[D22] SM perform operations on the system only when authorized.

[D23] SM always adds a new store or edit an available one inserting information that matches with the real ones.

[D24] A store has at least 1 queue board observable.

[D25] Every queue board always contains correct information (no "showing error").

2.4.3 Constraints

2.4.3.1 Regulatory policies

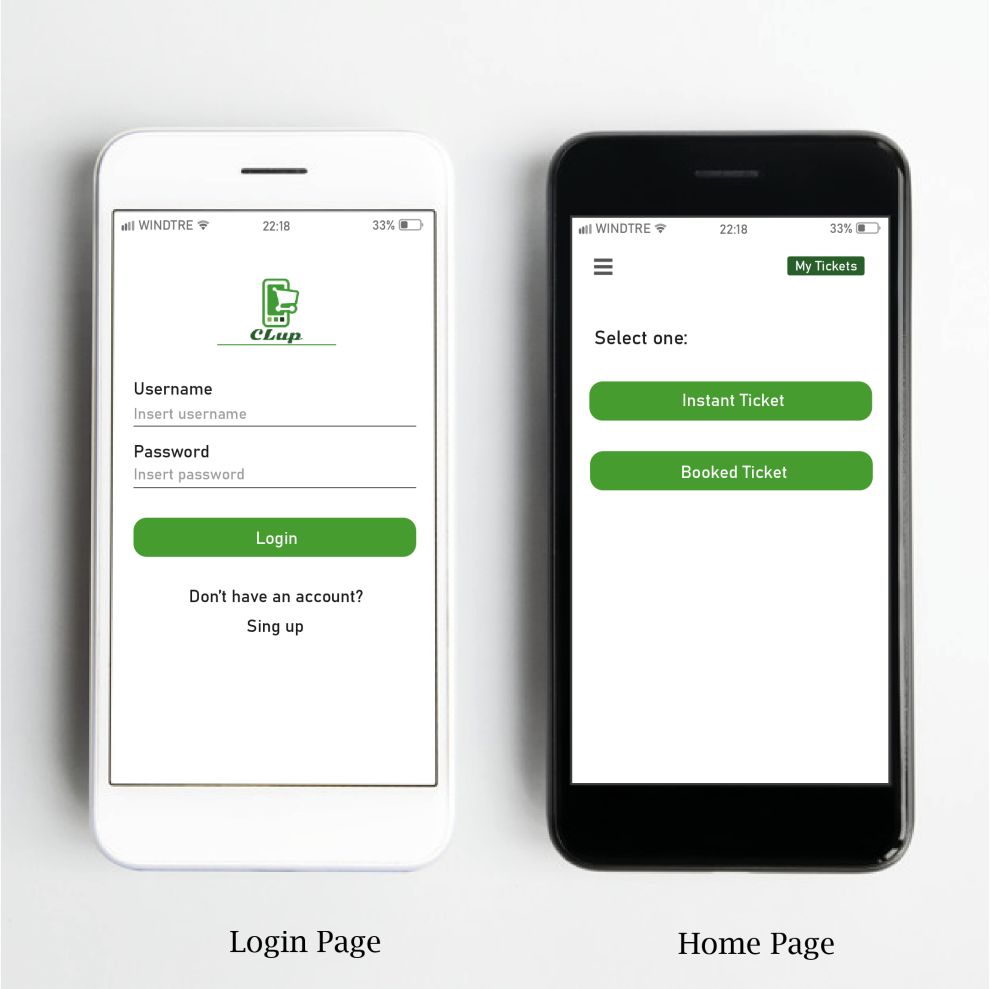
- The system will have to ask for users' permission to retrieve and use their GPS positions (without storing them).
- Email, addresses will not be used for commercial uses.

3 Specific requirements

3.1 External Interfaces Requirements

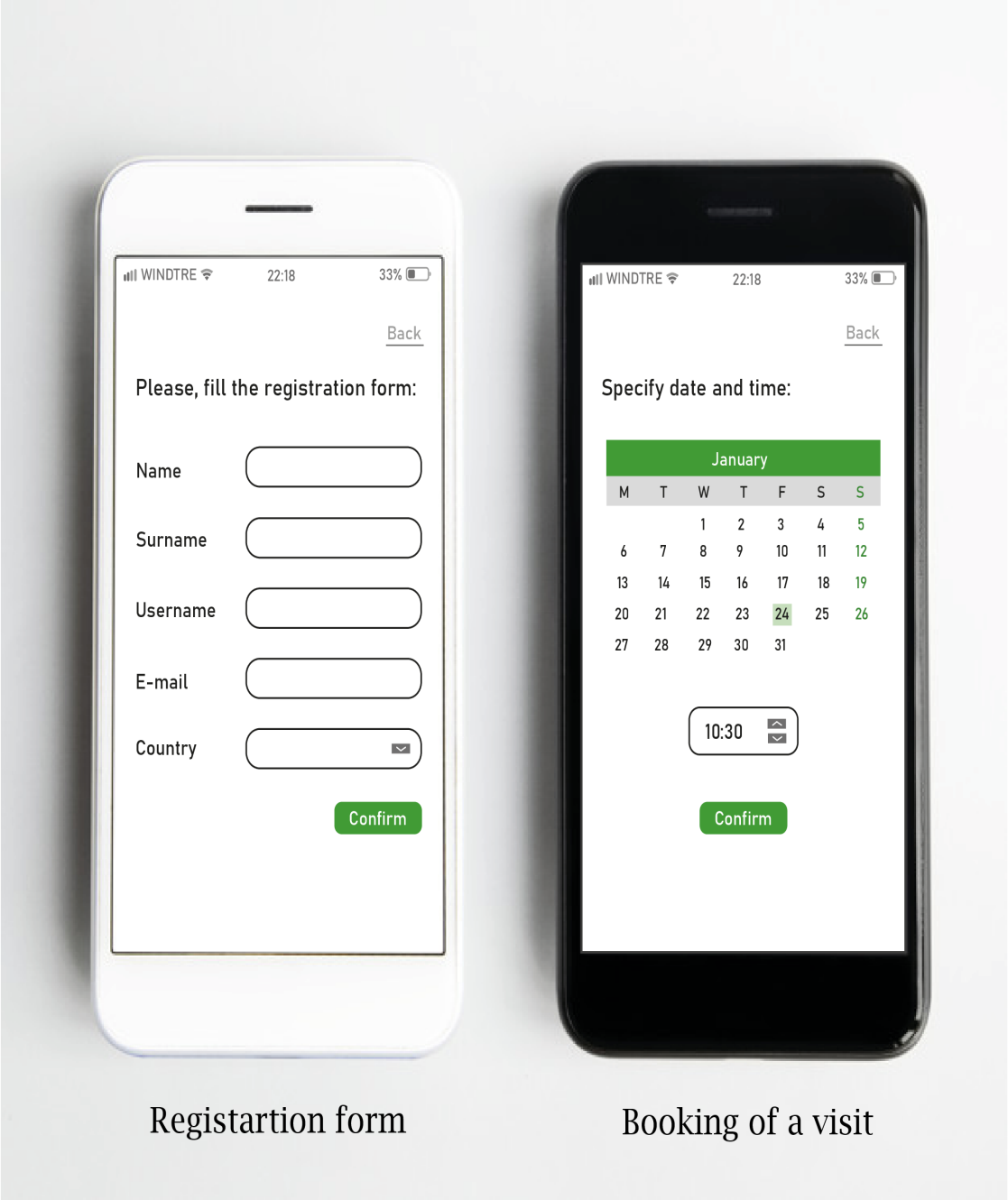
3.1.1 User Interface

Here we provide a list of mockups of the CLup mobile application:



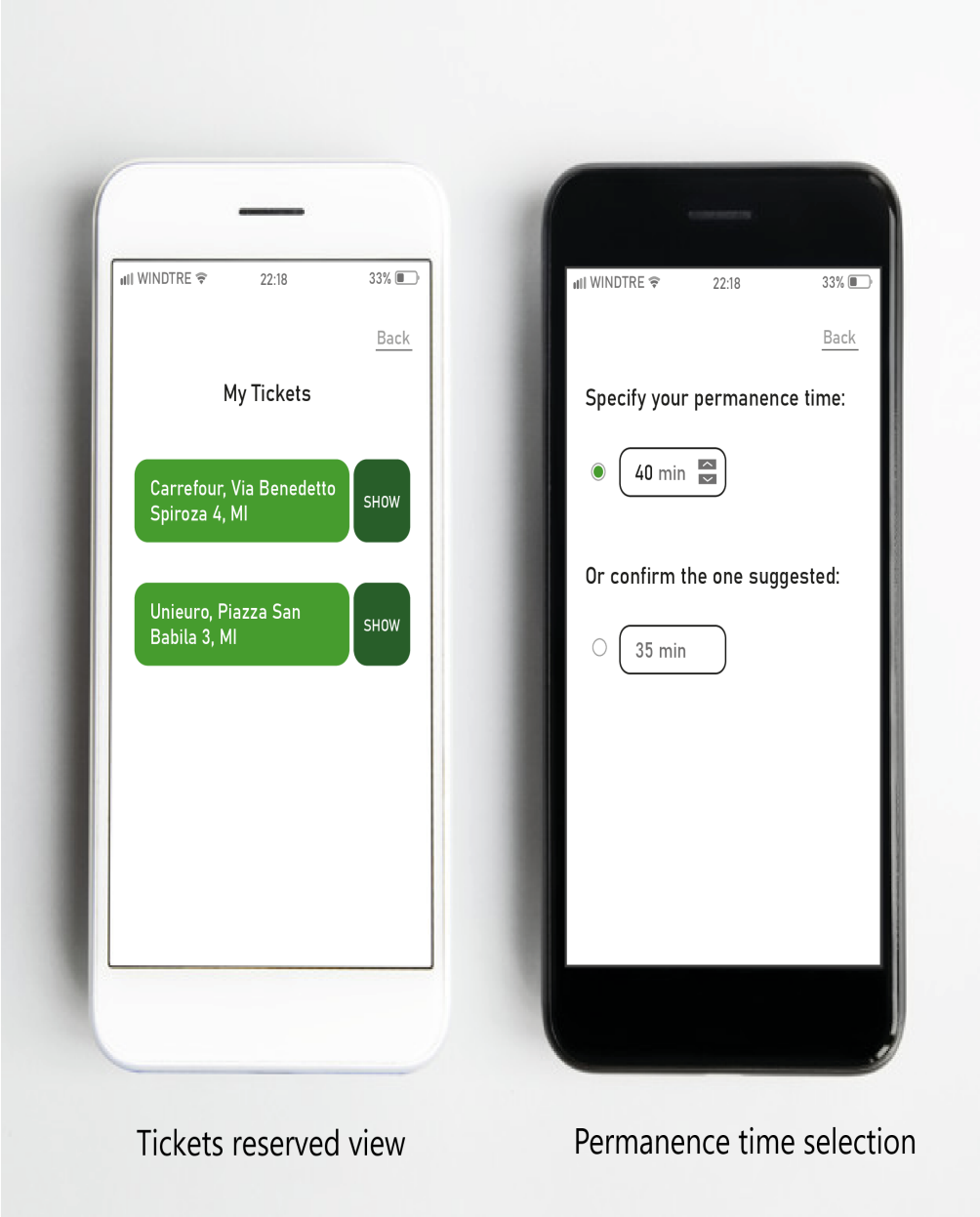
Login Page

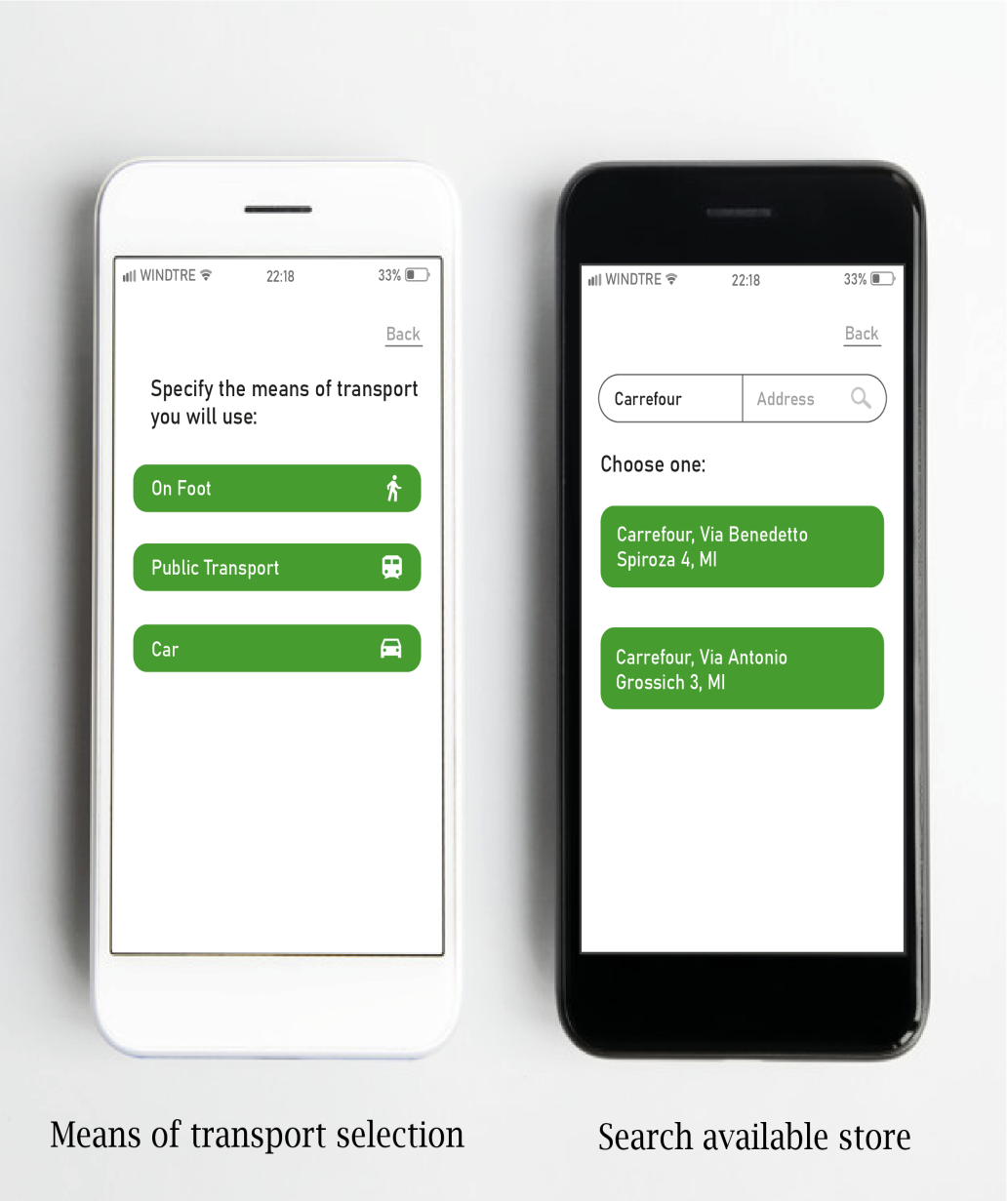
Home Page



Registartion form

Booking of a visit


WINDTRE
22:18
33%
Back
My Tickets
Carrefour, Via Benedetto Spiroza 4, MI
SHOW
Unieuro, Piazza San Babila 3, MI
SHOW
Tickets reserved view
WINDTRE
22:18
33%
Back
Specify your permanence time:
40 min
Or confirm the one suggested:
35 min
Permanence time selection



Means of transport selection

Search available store

Ticket's info view

3.1.2 Hardware Interfaces

- User: use a mobile device (like a smartphone) which support the Mobile App.
- System manager: use a generic device (like computer, smartphone etc.) which support the Web App.
- QR code reader: a device of CLup's system which can scan QR codes.
- Queue board: a device of CLup's system which can show images on a screen.
- Ticket printer: a device of CLup's system which can print a paper ticket and receive a simple input by the user.

3.1.3 Software Interfaces

- We assume that the system uses a public API (like Google Maps' API) to provide to the user his position in real time for the report compilation.

3.1.4 Communication Interfaces

The various devices (QR code reader, queue board, ticket printer, user's mobile device, SM's device) are connected to CLup's system via Internet connection.

3.1.5 Interfaces to other applications

In the first release no public interfaces will be opened, and third-party services won't be able to interoperate with CLup.

3.2 Functional Requirements

3.2.1 Requirements

[R1] A Visitor must be able to begin the registration process. During the process, the system will ask him/her to provide the following credentials: username, name, surname, email, and country.

[R2] The system must generate a new random password and send it to the User via email.

[R3] Ticket printer must be able to print a valid physical ticket with one number, one QR code and the estimated waiting time.

[R4] Physical ticket must be correctly integrated with the virtual ones in a unique queue for each store.

[R5] The system can release new instant/physical tickets if the current waiting time plus the permanence time does not exceed store's closing time.

[R6] The system can release an instant ticket or a physical one only if the store is open.

[R7] System must associate to each visitor as Permanence-time, the default one.

[R8] A user must be able to login to the system using his/her credentials.

[R9] A user must be able to get 1 ticket(Instant or Booked) for an available store, if he/she has not already another ticket of the chosen category (even for a different store) and he/she has not already a ticket of the other category for the chosen store.

[R10] A user must be able to: specify a permanence time (with reasonable min and max constraints) or confirm the suggested permanence time calculated by the system (this is available only if the user has already done some visit in the same store).

[R11] A user must be able to specify a means of transport between: "on foot", "public transport", "car".

[R12] User can get an instant ticket only if he/she is in a certain range near the store.

[R13] After the selection of the wanted ticket category (Instant or Booked), a user must be able to search an available store for that category specifying its name and/or its address (specifying only one of them, the system may provide more than one store).

[R14] The system can only release booked tickets if in all the period of the visit the maximum people capacity is not exceeded (this period starts at the arrival time specified and end over the permanence time).

[R15] The system can only release a booked ticket if its booked date and time, and the permanence time, respect store's timetable.

[R16] A user can get a booked ticket if chosen store's country is the same indicated by him/her in the registration.

[R17] A user can get a booked ticket only until the 23.59 of the previous day of the reservation, always respecting the store's timetable.

[R18] A QR code reader must accept a ticket's QR code if and only if it has become a current ticket for the store in question since no more than few minutes (2 minutes can be reasonable) and it has not already a check-in event associated; in both these two cases then a ticket expires.

[R19] The system must ensure that an instant ticket become the current ticket when the maximum between its associated waiting time and travel time is expired.

[R20] The system must ensure that a physical ticket become the current ticket when its associated waiting time is expired.

[R21] The system must ensure that a booked ticket become the current ticket when its associated date and time are reached.

[R22] The system must ensure that a ticket remain the current ticket for some seconds (20 seconds are reasonable).

[R23] A QR code reader show a positive/negative feedback on its integrated screen, if accept/reject the QR code just scanned.

[R24] The user/visitor must be able to see/show his/her ticket's information.

[R25] The system must be able to calculate the estimated waiting time reasonably and precisely, overestimating the permanence time of all the users/visitors already in the current queue to allow possible (little) delay of them (avoiding a cascade effect), respecting the maximum people capacity of the store, and considering that a current ticket must remain unchanged for some seconds.

[R26] The system must be able to calculate the travel time precisely, considering the current position of the user, while releasing the ticket (with GPS), and the position of the store.

[R27] If the difference time between the waiting time and the travel time is strictly positive (>0), the system alerts the user to reach the store when this time expires.

[R28] The system alerts the user to reach the store immediately when the difference between the waiting time and the travel time is not strictly positive (<=0).

[R29] A SM must be able to login using his/her credentials.

[R30] A SM must be able to add a new store in the system providing its commercial name, address, Maximum People Capacity, timetable, and the default permanence time.

[R31] A SM must be able to delete an available store from the system.

[R32] A SM must be able to edit the timetable of an available store.

[R33] A SM must be able to edit maximum people capacity of a store.

[R34] A SM must be able to edit the default permanence time of a store.

[R35] A SM must be able to see all the stores registered into the system, or search for a specific one using its commercial name and address.

[R36] The queue board must be able to show the current ticket number.

3.2.2 Mapping of Goal, Requirements and Domain assumptions

[G1] **Allow a person to become a registered User providing some basic personal info (like Name, Surname etc.).**

- [R1] A Visitor must be able to begin the registration process. During the process, the system will ask him/her to provide the following credentials: username, name, surname, email, and country.
- [R2] The system must generate a new random password and send it to the User via email.
- [D1] The Username must be unique.
- [D2] When the system sends a password to a new registered User this will be surely received by him/her.

[G2] **Allow a store's Visitor to pick up a Physical ticket from the store itself.**

- [R3] Ticket printer must be able to print a valid physical ticket with one number, one QR code and the estimated waiting time.
- [R4] Physical ticket must be correctly integrated with the virtual ones in a unique queue for each store.
- [R5] The system can release new instant/physical tickets if the current waiting time plus the permanence time does not exceed store's closing time.
- [R6] The system can release an instant ticket or a physical one only if the store is open.
- [R7] System must associate to each visitor as Permanence-time, the default one.
- [R25] The system must be able to calculate the estimated waiting time reasonably and precisely, overestimating the permanence time of all the users/visitors already in the current queue to allow possible (little) delay of them (avoiding a cascade effect), respecting the maximum people capacity of the store, and considering that a current ticket must remain unchanged for some seconds.
- [D3] A store has at least 1 ticket printer available.
- [D4] Every CLup store's device is able to communicate with the system.
- [D5] Every physical ticket always contains correct information (no "printing error").
- [D6] A visitor always retrieves his/her physical ticket.
- [D7] A visitor only retrieves 1 physical ticket per visit.

[D8] The real visitor's permanence time in a store is always no longer then a maximum delay time over the default permanence time known for the store by the system (the maximum delay can be represented in percentage respect to the default permanence time).

[G3] Allow a User to reserve in real-time a ticket for the current queue of a particular store.

[R5] The system can release new instant/physical tickets if the current waiting time plus the permanence time does not exceed store's closing time.

[R6] The system can release an instant ticket or a physical one only if the store is open.

[R8] A user must be able to login to the system using his/her credentials.

[R9] A user must be able to get 1 ticket(Instant or Booked) for an available store, if he/she has not already another ticket of the chosen category (even for a different store) and he/she has not already a ticket of the other category for the chosen store.

[R10] A user must be able to: specify a permanence time (with reasonable min and max constraints) or confirm the suggested permanence time calculated by the system (this is available only if the user has already done some visit in the same store).

[R11] A user must be able to specify a means of transport between: "on foot", "public transport", "car".

[R12] User can get an instant ticket only if he/she is in a certain range near the store.

[R13] After the selection of the wanted ticket category (Instant or Booked), a user must be able to search an available store for that category specifying its name and/or its address (specifying only one of them, the system may provide more than one store).

[R25] The system must be able to calculate the estimated waiting time reasonably and precisely, overestimating the permanence time of all the users/visitors already in the current queue to allow possible (little) delay of them (avoiding a cascade effect), respecting the maximum people capacity of the store, and considering that a current ticket must remain unchanged for some seconds.

[D9] The means of transport selected by a user is the one he/she will use for real.

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[D11] If GPS show the user in a certain position, it means he/she is there for real.

[D16] Each store can be identified uniquely by its name and address.

[G4] Allow a User to book a visit to a store.

[R8] A user must be able to login to the system using his/her credentials.

[R9] A user must be able to get 1 ticket(Instant or Booked) for an available store, if he/she has not already another ticket of the chosen category (even for a different store) and he/she has not already a ticket of the other category for the chosen store.

[R10] A user must be able to: specify a permanence time (with reasonable min and max constraints) or confirm the suggested permanence time calculated by the system (this is available only if the user has already done some visit in the same store).

[R13] After the selection of the wanted ticket category (Instant or Booked), a user must be able to search an available store for that category specifying its name and/or its address (specifying only one of them, the system may provide more than one store).

[R14] The system can only release booked tickets if in all the period of the visit the maximum people capacity is not exceeded (this period starts at the arrival time specified and end over the permanence time).

[R15] The system can only release a booked ticket if its booked date and time, and the permanence time, respect store's timetable.

[R16] A user can get a booked ticket if chosen store's country is the same indicated by him/her in the registration.

[R17] A user can get a booked ticket only until the 23.59 of the previous day of the reservation, always respecting the store's timetable.

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[D16] Each store can be identified uniquely by its name and address.

[G5] Allow a User/Visitor to enter in the store within the use of QR code when it is his/her turn.

[R8] A user must be able to login to the system using his/her credentials.

[R18] A QR code reader must accept a ticket's QR code if and only if it has become a current ticket for the store in question since no more than few minutes (2 minutes can be reasonable) and it has

not already a check-in event associated; in both these two cases then a ticket expires.

[R19] The system must ensure that an instant ticket become the current ticket when the maximum between its associated waiting time and travel time is expired.

[R20] The system must ensure that a physical ticket become the current ticket when its associated waiting time is expired.

[R21] The system must ensure that a booked ticket become the current ticket when its associated date and time are reached.

[R22] The system must ensure that a ticket remain the current ticket for some seconds (20 seconds are reasonable).

[R23] A QR code reader show a positive/negative feedback on its integrated screen, if accept/reject the QR code just scanned.

[R24] The user/visitor must be able to see/show his/her ticket's information.

[R36] The queue board must be able to show the current ticket number.

[D4] Every CLup store's device is able to communicate with the system.

[D8] The real visitor's permanence time in a store is always no longer then a maximum delay time over the default permanence time known for the store by the system (the maximum delay can be represented in percentage respect to the default permanence time).

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[D12] A user/visitor every time check-in within the QR code entering in the store.

[D13] A user/visitor every time enter in the store only if the check-in goes successful (QR code accepted).

[D14] Every QR code reader always read correctly and show correct feedbacks.

[D15] A store has at least 1 QR reader available.

[G6] Allow a User to search and find available stores

[R8] A user must be able to login to the system using his/her credentials.

[R12] User can get an instant ticket only if he/she is in a certain range near the store.

[R13] After the selection of the wanted ticket category (Instant or Booked), a user must be able to search an available store for that category specifying its name and/or its address (specifying only one of them, the system may provide more than one store).

[R16] A user can get a booked ticket if chosen store's country is the same indicated by him/her in the registration.

[D16] Each store can be identified uniquely by its name and address.

[G7] The system should provide Users/Visitors with a reasonably precise estimation of the waiting time associated to their respectively instant and physical tickets.

[R22] The system must ensure that a ticket remain the current ticket for some seconds (20 seconds are reasonable).

[R24] The user/visitor must be able to see/show his/her ticket's information.

[R25] The system must be able to calculate the estimated waiting time reasonably and precisely, overestimating the permanence time of all the users/visitors already in the current queue to allow possible (little) delay of them (avoiding a cascade effect), respecting the maximum people capacity of the store, and considering that a current ticket must remain unchanged for some seconds.

[D5] Every physical ticket always contains correct information (no "printing error").

[D8] The real visitor's permanence time in a store is always no longer then a maximum delay time over the default permanence time known for the store by the system (the maximum delay can be represented in percentage respect to the default permanence time).

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[G8] The system should alert in time a User to reach the store, for which he took the instant ticket, taking into account the time he need to get to the store from the place he currently is.

[R12] User can get an instant ticket only if he/she is in a certain range near the store.

[R26] The system must be able to calculate the travel time precisely, considering the current position of the user, while releasing the ticket (with GPS), and the position of the store.

[R27] If the difference time between the waiting time and the travel time is strictly positive (>0), the system alerts the user to reach the store when this time expires.

[R28] The system alerts the user to reach the store immediately when the difference between the waiting time and the travel time is not strictly positive (<=0).

[D9] The means of transport selected by a user is the one he/she will use for real.

[D10] The permanence time specified or confirmed by a user will be respected (with very little variations).

[D11] If GPS show the user in a certain position, it means he/she is there for real.

[D17] A user start going to the store associated to the instant ticket when the specific notification is received (almost immediately).

[D18] There are no variations between the estimated travel time and what will be used by a user.

[D19] Notifications are certainly received by the user.

[D20] User position is retrieved by GPS.

[G9] Allow a System Manager to do operations on the system for updating and maintenance.

[R29] A SM must be able to login using his/her credentials.

[R30] A SM must be able to add a new store in the system providing its commercial name, address, Maximum People Capacity, timetable, and the default permanence time.

[R31] A SM must be able to delete an available store from the system.

[R32] A SM must be able to edit the timetable of an available store.

[R33] A SM must be able to edit maximum people capacity of a store.

[R34] A SM must be able to edit the default permanence time of a store.

[R35] A SM must be able to see all the stores registered into the system, or search for a specific one using its commercial name and address.

[D16] Each store can be identified uniquely by its name and address.

[D21] Each store can be identified uniquely by its name and address.

[D22] Once added by an SM, a store become immediately available in the system.

[D23] SM perform operations on the system only when authorized.

[G10] Allow a User/Visitor to see the current ticket number on the store's queue board.

[R19] The system must ensure that an instant ticket become the current ticket when the maximum between its associated waiting time and travel time is expired.

[R20] The system must ensure that a physical ticket become the current ticket when its associated waiting time is expired.

[R21] The system must ensure that a booked ticket become the current ticket when its associated date and time are reached.

[R22] The system must ensure that a ticket remain the current ticket for some seconds (20 seconds are reasonable).

[R36] The queue board must be able to show the current ticket number.

[D4] Every CLup store's device is able to communicate with the system.

[D24] A store has at least 1 queue board observable.

[D25] Every queue board always contains correct information (no "showing error").

3.2.3 Use cases

3.2.3.1 Visitor Registration

Actors	Visitor
Goals	G1
Input Conditions	No entry condition
Events flow	1 The Visitor on the access page clicks on the "Sign In" button to start the registration process.
	2. The Visitor fills all the mandatory fields in the form.
	3. The Visitor click on the "Submit" button.
	4. The system saves the inserted data..
	5. The system sends an email to the new User with the password
Output conditions	The Visitor successfully ends the registration process and become a new User. From now on he/she can log in to the application providing his/her credentials and start using CLup.
Exceptions	1. The Visitor is already a User.
	2. The Visitor inserts not valid information in one or more mandatory fields.
	3. The Visitor chooses a username that has already been taken by another user.
	4. The Visitor chooses an email that has been associated with another user.
	All exceptions are handled notifying the issue to the Person and taking back the Event Flow to the point 2.

3.2.3.2 User Login

Actors	User
Goals	G3, G4, G6
Input Conditions	The User is already on the access page.
Events flow	1. The User inserts his/her credentials into the "username" and "password" fields.
	2. The User clicks on the "Log In" button in order to access.
	3. The system redirects the user to the home page
Output conditions	The User is successfully redirected to the Home page.
Exceptions	1. The User inserts a not valid username.
	2. The User inserts a not valid password.
	All exceptions are handled notifying the issue to the Visitor and taking back the Event Flow to the point 2.

3.2.3.3 Visitor line-up at the store

Actors	Visitor
Goals	G2
Input Conditions	The Visitor has to be already at the entrance of the store.
Events flow	1. The User inserts his/her credentials into the "username" and "password" fields.
	2. The User clicks on the "Log In" button in order to access.
	3. The system redirects the user to the home page
Output conditions	The visitor has successfully line-up retrieving a physical ticket.
Exceptions	1. The store cannot admit a complete visit before closing.
	This exception is handled printing a message which notifies the Visitor that there is not a place in the current queue.

3.2.3.4 User reservation of an instant ticket

Actors	User
Goals	G3
Input Conditions	The User has to be already logged into the system.
	1. The User click on the "Instant Ticket" button.

Events flow

	2. The User insert the name and/or the address of the store in the search bar.
	3. The system shows a list of corresponding stores near the User's position.
	4. The User chooses a store clicking it.
	5. The system shows the suggested permanence time considering the previous visits.
	6. The User confirm the suggested permanence time (if showed) or write a permanence time in the appropriate box.
	7. The User chooses one of the possible means of transport.
	8. The system verifies that it can complete the reservation.
	9. The system creates the Instant Ticket adding the User to the store's queue.
	10. The system redirects the User to a ticket confirmation page and starts the timer for the alert to reach the store.
Output conditions	The User has successfully reserved an Instant Ticket.
Exceptions	1. The list of corresponding stores is empty.
	This exception is handled notifying the issue to the User and taking back the Event Flow to the point 2.
	2. The system cannot calculate the suggested permanence time.
	This exception is handled skipping directly to the point 6 of the Event Flow.
	3. The User specify a permanence time not respecting the min/max constraint.
	This exception is handled notifying the issue to the User and taking back the Event Flow to the point 6.
	4. The system cannot admit the reservation because the visit exceeds the store's closing time, or the store is not open.
	This exception is handled redirecting the User to an error page in which will be explained the issue occurred.

3.2.3.5 User reservation of a booked ticket

Actors	User
Goals	G4
Input Conditions	The User has to be already logged into the system.
	1. The User click on the "Booked Ticket" button.

Events flow

	2. The User insert the name and/or the address of the store in the search bar.
	3. The system shows a list of corresponding stores which are in the User's country.
	4. The User chooses a store clicking it.
	5. The User select the date and the hour wanted from a sort of calendar.
	6. The system shows the suggested permanence time considering the previous visits.
	7. The User confirm the suggested permanence time (if showed) or write a permanence time in the appropriate box.
	8. The system verifies that it can complete the reservation.
	9. The system creates the Booked Ticket adding the User to the store's queue.
	10. The system redirects the User to a ticket confirmation page.
Output conditions	The User has successfully reserved a Booked Ticket.
Exceptions	1. The list of corresponding stores is empty.
	This exception is handled notifying the issue to the User and taking back the Event Flow to the point 2.
	2. The system cannot calculate the suggested permanence time.
	This exception is handled skipping directly to the point 6 of the Event Flow.
	3. The User specify a permanence time not respecting the min/max constraint.
	This exception is handled notifying the issue to the User and taking back the Event Flow to the point 6.
	4. The system cannot admit the reservation because the User is not booking until the 23.59 of the previous day of the visit.
	5. The system cannot admit the reservation because the visit exceeds the store's closing time or more in general do not respect the store's timetable.
	These two exceptions are handled redirecting the User to an error page in which will be explained the issue occurred.

3.2.3.6 User check-in

Actors	User
Goals	G5
Input Conditions	The User has already reserved an instant/booked ticket and he/she has to be already at the entrance of the store.
Events flow	1. The User select the section "My Tickets".
	2. The User select the correct ticket.
	3. The User show the ticket's QR code to the QR code reader.
	4. The system verifies that the ticket is acceptable to check-in.
	5. The system accepts the ticket's QR code scanned, giving a positive feedback.
	6. The User see the positive feedback on the QR code reader's screen.
Output conditions	The user has successfully completed the check-in.
Exception	1. The system does not accept the ticket because it has not yet become the current ticket or has not been so for too long time.
	This exception is handled showing the negative feedback on the QR code reader's screen.

3.2.3.7 Visitor check-in

Actors	Visitor
Goals	G5
Input Conditions	The Visitor has to have already reserved a physical ticket and he/she has to be already at the entrance of the store.
Events flow	1. The Visitor show the physical ticket's QR code to the QR code reader.
	2. The system verifies that the ticket is acceptable to check-in.
	3. The system accepts the ticket's QR code scanned, giving a positive feedback.
	4. The Visitor see the positive feedback on the QR code reader's screen.
Output conditions	The Visitor has successfully completed the check-in.
Exception	1. The system does not accept the ticket because it has not yet become the current ticket or has not been so for too long time.

	This exception is handled showing the negative feedback on the QR code reader's screen.

3.2.3.8 User receive alert to reach the store

Actors	Visitor
Goals	G8
Input Conditions	The User has to have just reserved an instant ticket.
Events flow	1. The system sends the alert to the User immediately when the difference between the waiting time and the travel time is not strictly positive, otherwise when the difference between the waiting time and the travel time expires.
	2. The User receives the alert on his/her mobile device.
Output conditions	The User successfully receive the alert to reach the store associated to his/her instant ticket.
Exception	

3.2.3.9 User/Visitor see the current number on the queue board

Actors	User/Visitor
Goals	G10
Input Conditions	The User/Visitor has already reserved a ticket and he/she has to be already at the entrance of the store.
Events flow	1. The system updates the current number of a store because the maximum between waiting time and travel time of an instant ticket is expired, or date and time of a booked ticket are reached, or the waiting time of a physical ticket is expired.
	2. The new current ticket's number for the store is showed on the queue board of the same store.
	3. User/Visitor see the current ticket's number showed on the store's queue board.
Output conditions	The User/Visitor has successfully seen the current ticket's number on the queue board.
Exception	1. There is not a new current ticket for the store.

	This exception is handled by the system by showing nothing on the store's queue board.

3.2.3.10 System manager login

Actors	System manager
Goals	G9
Input Conditions	There are no entry conditions.
Events flow	1. The SM opens the access page of his private interface.
	2. The SM inserts his/her credentials into the "username" and "password" fields.
	3. The SM clicks on the "Log In" button in order to access.
	4. The system redirects him to the SM's home page.
Output conditions	The SM manager is correctly redirected to the SM's home page.
Exception	1. The SM inserts a not valid username..
	2. The SM inserts a not valid password.
	All exceptions are handled notifying the issue to the System Manager and taking back the Event Flow to the point 2.

3.2.3.11 System manager searches for an available store

Actors	System manager
Goals	G9
Input Conditions	The SM has to be already logged into the system.
Events flow	1. The SM click on the "Search store" button.
	2. The SM insert store's name (and address if needed) in the search bar.
	3. The system shows the SM all the information (commercial name, address, Maximum People Capacity, timetable, default permanence time) about the founded store, plus the options "Delete and "Edit".
Output conditions	The SM successfully finds the store.
Exception	1. The SM do not insert the real correct store's name/address.
	2. The SM insert a name/address of a not available store.

	All exceptions are handled notifying the issue to the SM and taking back the Event Flow to the point 2.

3.2.3.12 System manager adds a new store

Actors	System manager
Goals	G9
Input Conditions	The SM has to be already logged into the system.
Events flow	1. The SM click on the "Add new store" button.
	2. The SM fill out all form's fields with new store's information (commercial name, address, Maximum People Capacity, timetable, default permanence time).
	3. The SM click on the "Confirm" button.
	4. The system adds the store to the list of the available one.
	5- The system redirects the SM to the home page
Output conditions	The new store is correctly added to the system.
Exception	1. SM insert a commercial name (and address) of an already available store.
	This exception is handled taking back the Event Flow to the point 2).

3.2.3.13 System manager deletes an available store

Actors	System manager
Goals	G9
Input Conditions	The SM has already found the store in the system with the search option.
Events flow	1. The SM click on the "Delete" button.
	2. The system shows a dialogue box with a "Confirm" and "Cancel" button.
	3. The SM click on the "Confirm" button.
	4. The system deletes the store from the list of the available one.
	5. The system redirects the SM to the home page.
Output conditions	The store is correctly deleted from the system.
Exception	1. The SM clicks on the "Cancel" button.

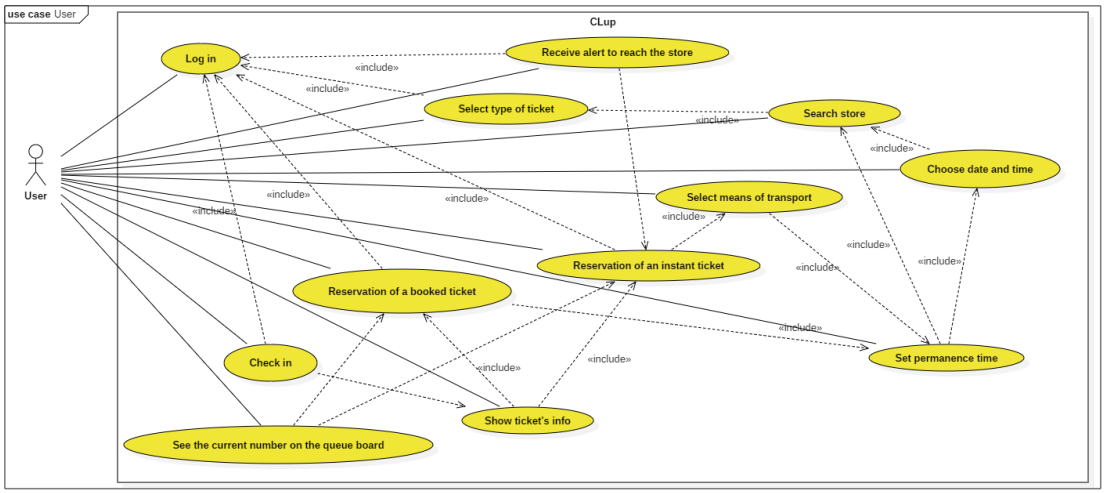
	This exception is handled taking back the Event Flow to the point 1.

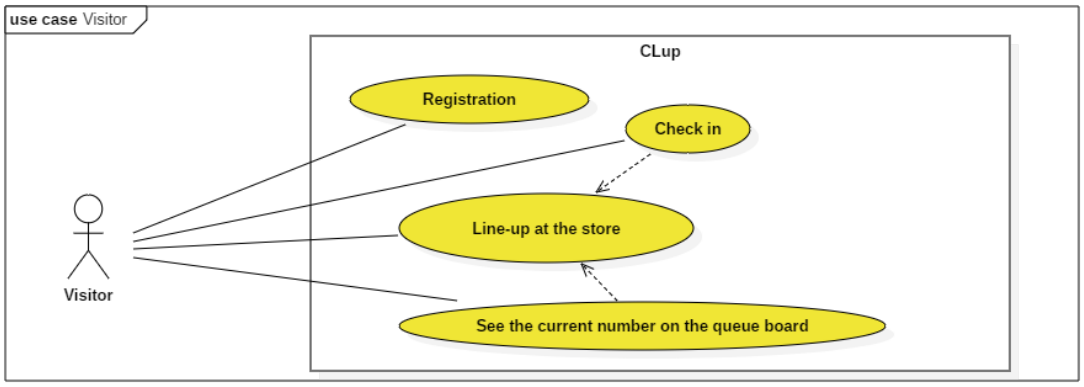
3.2.3.14 System manager edits some store's field.

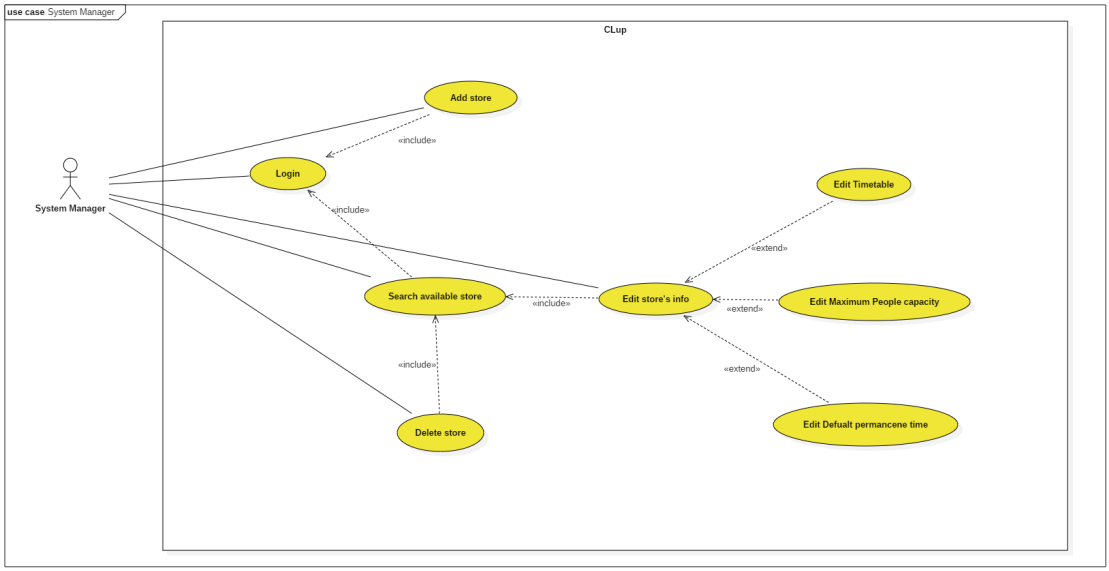
Actors	System manager
Goals	G9
Input Conditions	The SM has already found the store in the system with the search option.
Events flow	1. The SM click on the "Edit" button.
	2. The system shows the SM a list of fields of store's information he can edit (Maximum People Capacity, timetable, default permanence time).
	3. The SM clicks on the one he wants to modify.
	4. The SM insert the new value for the selected fields.
	5. The SM click on the confirm button.
	6. The system updates the fields.
	7. The system redirects the SM to the home page.
Output conditions	The system has correctly updated the information for the chosen store.
Exception	

3.2.4 UML Description associated to use cases

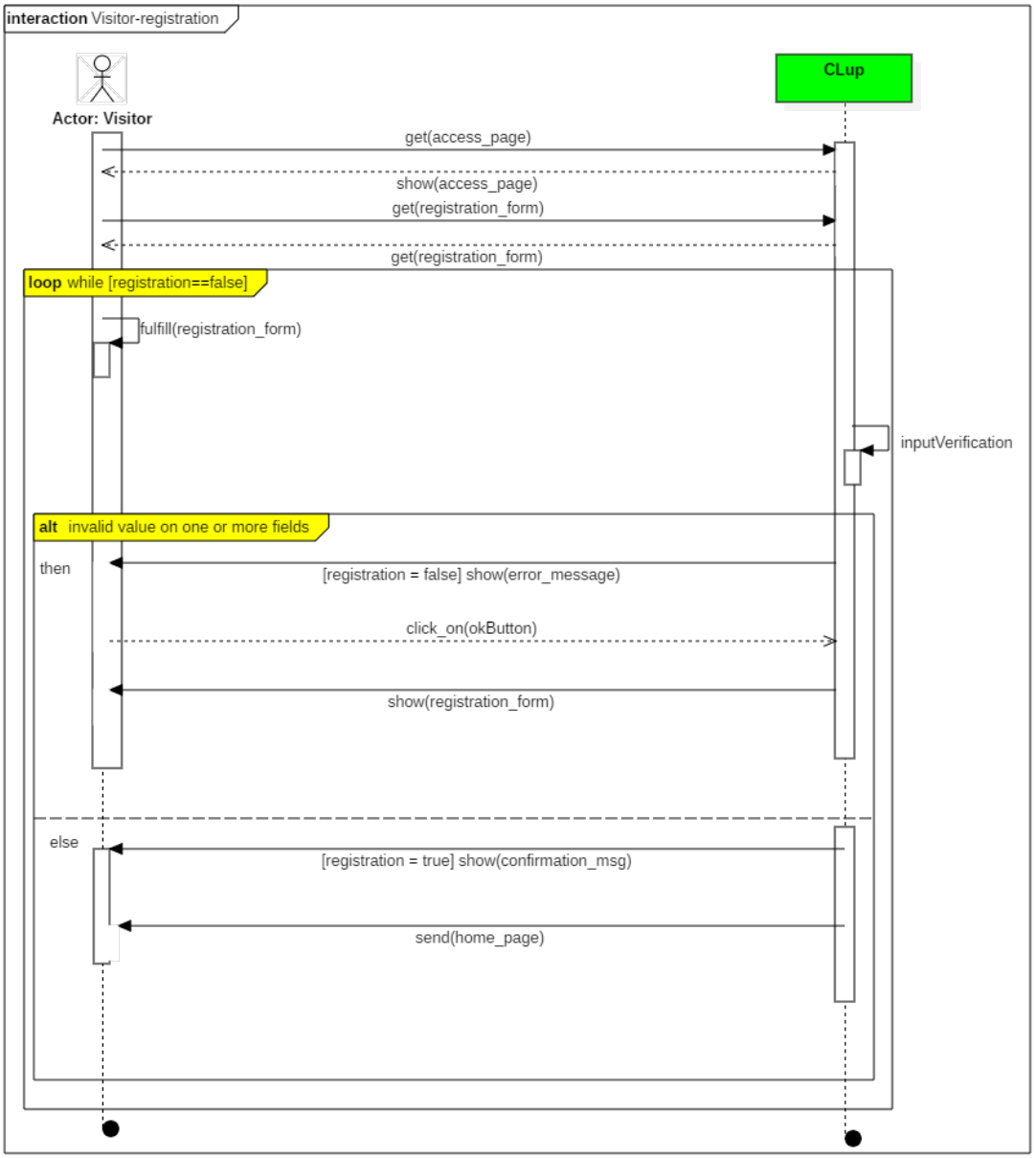
3.2.4.1 Use case Diagram

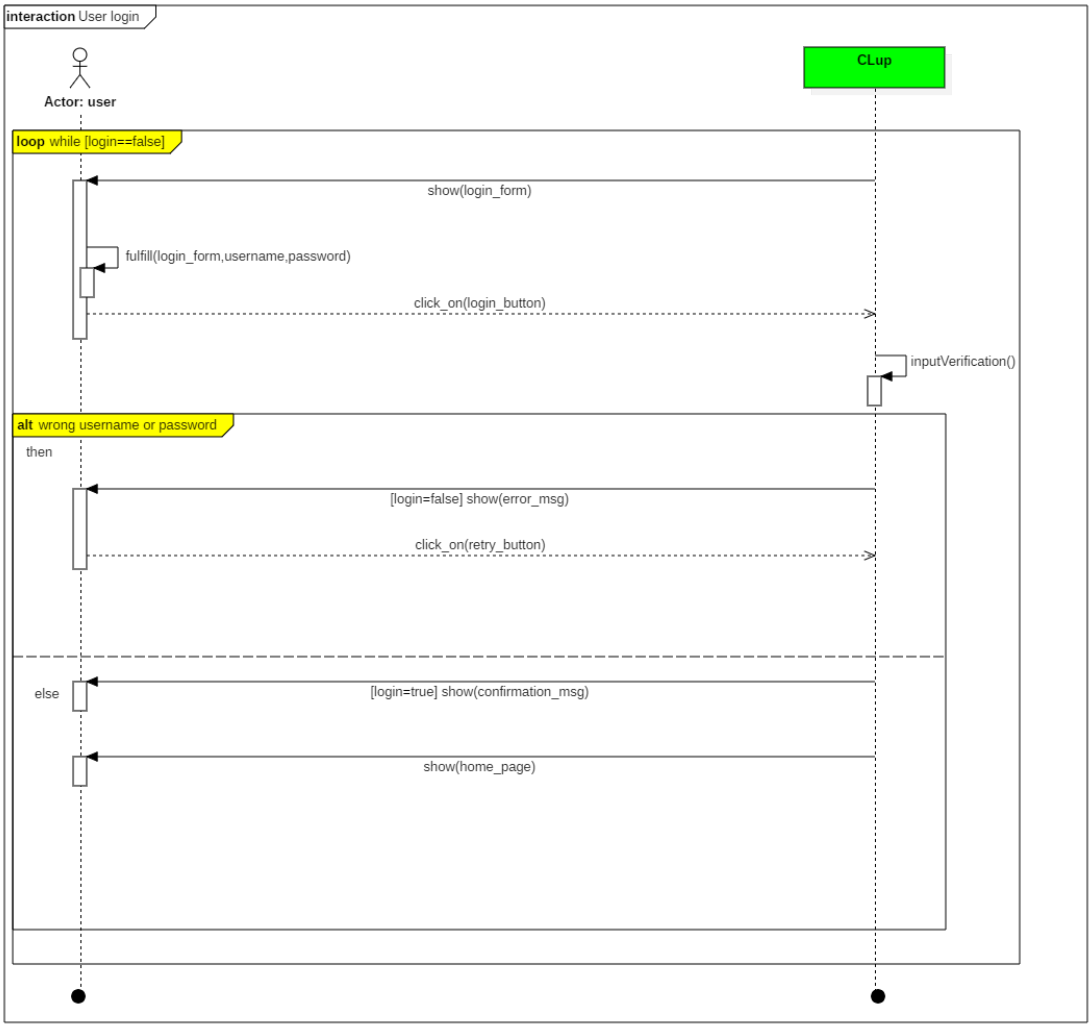


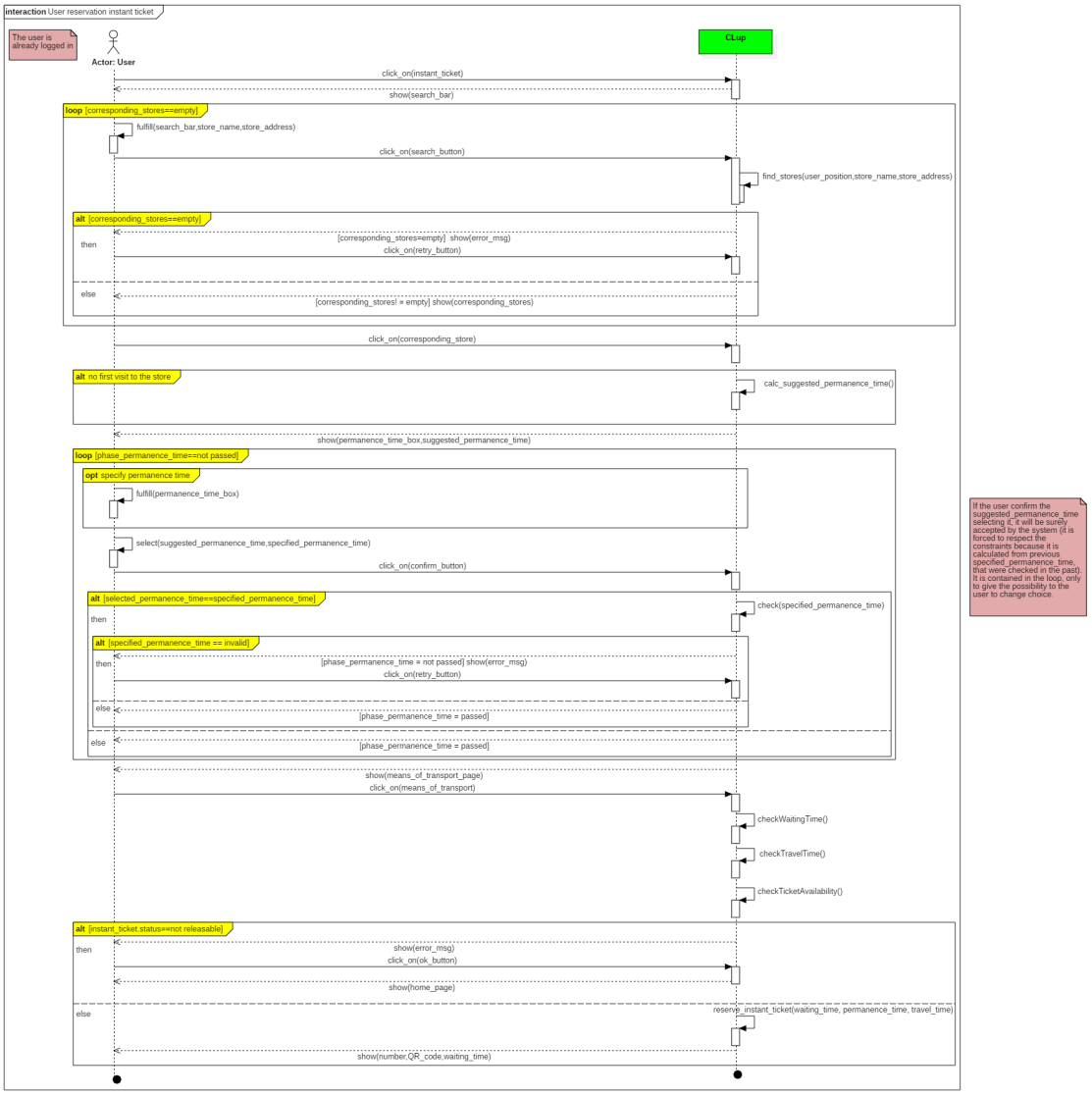


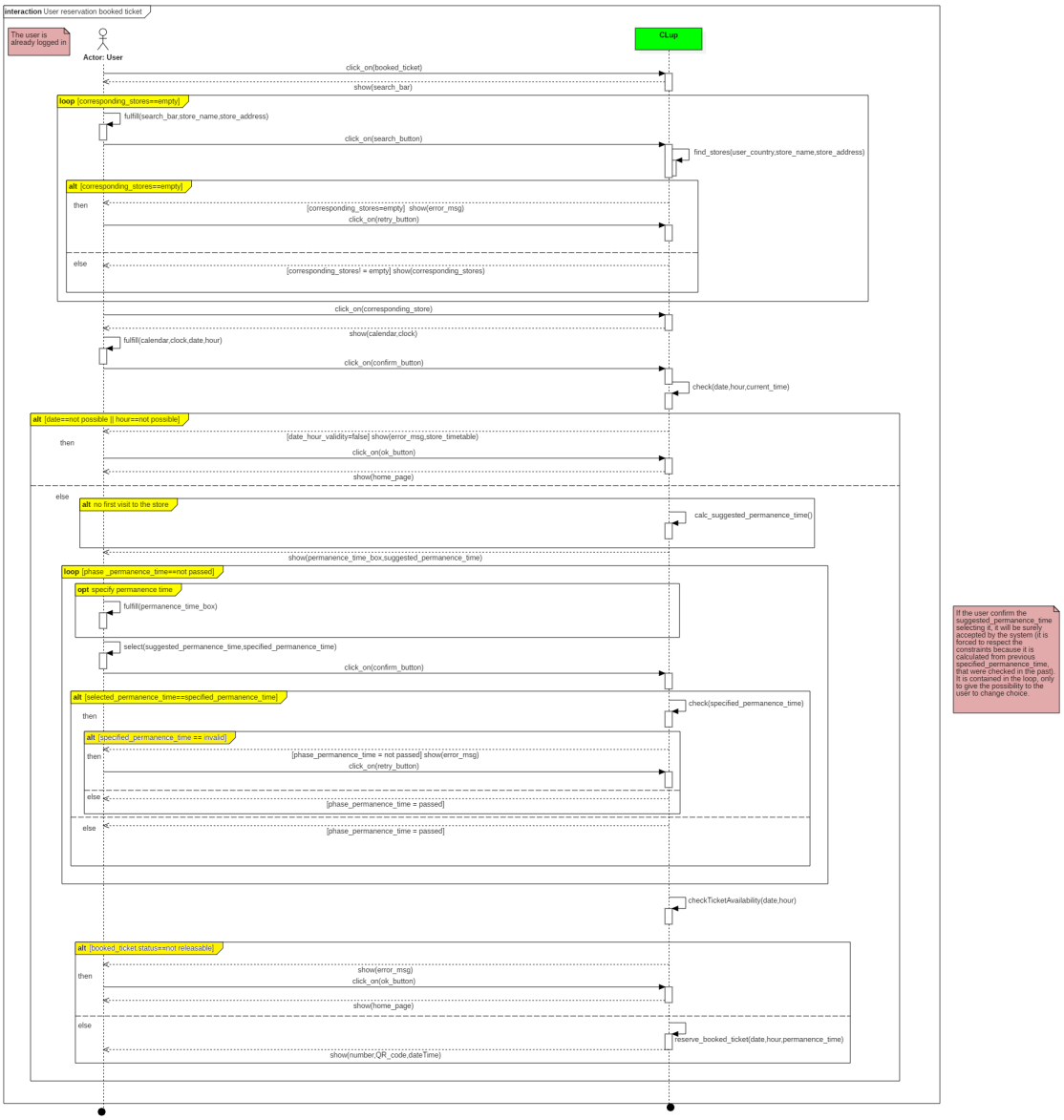

use case System Manager
CLup
Add store
«include»
Login
System Manager
«include»
Edit Timetable
«extend»
Search available store
«include»
Edit store's info
«extend»
Edit Maximum People capacity
«include»
«extend»
Delete store
Edit Defualt permancene time

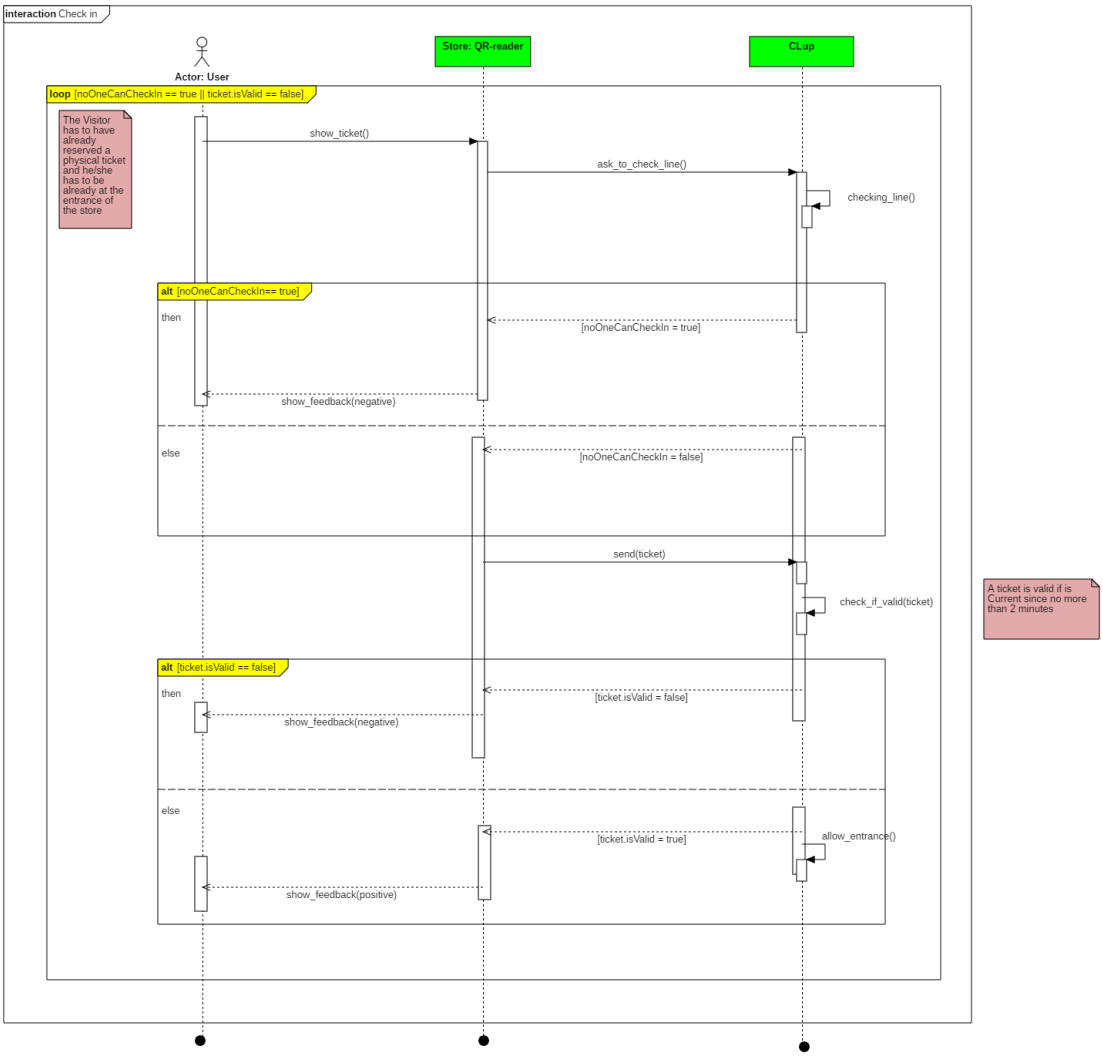
3.2.4.2 Sequence Diagram

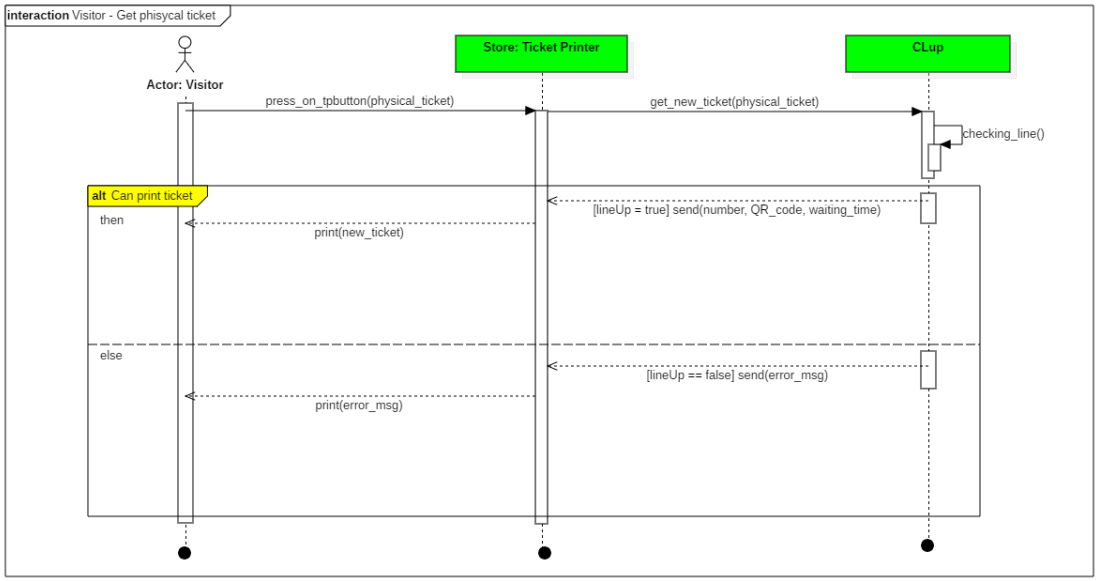


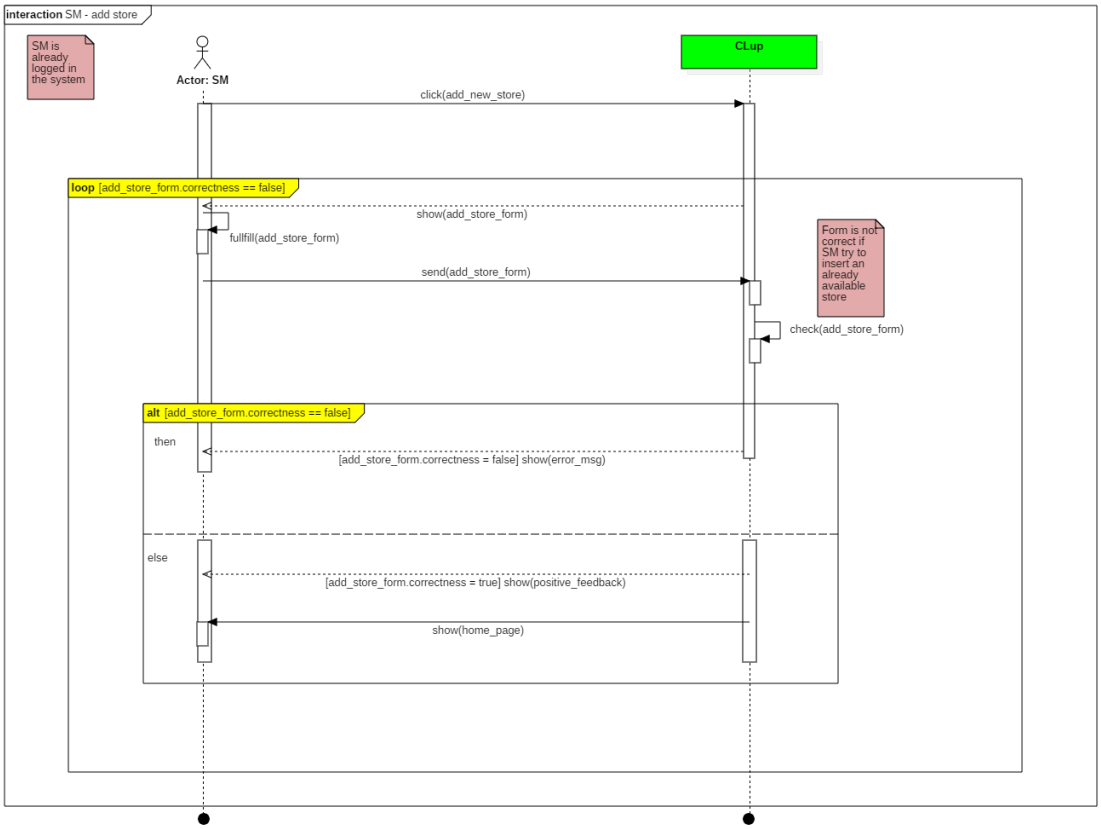

interaction User login
Actor: user
CLup
loop while [login==false]
show(login_form)
fulfill(login_form,username,password)
click_on(login_button)
inputVerification()
alt wrong username or password
then
[login=false] show(error_msg)
click_on(retry_button)
else
[login=true] show(confirmation_msg)
show(home_page)

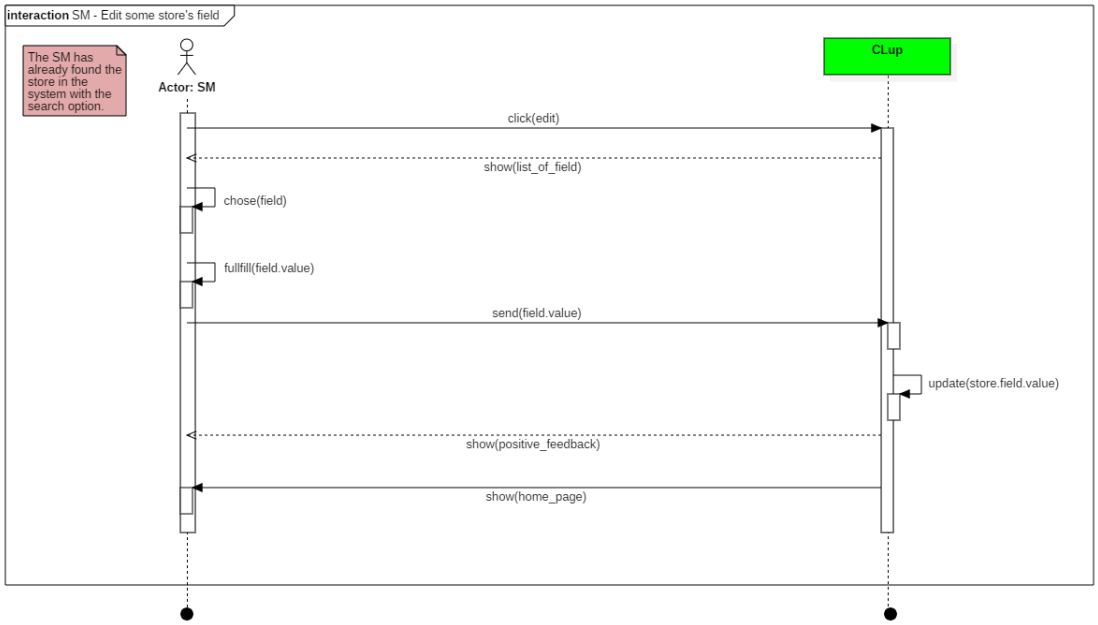

interaction User reservation instant ticket
The user is already logged in
Actor: User
CLup
click_on(instant_ticket)
show(search_bar)
loop [corresponding_stores==empty]
fulfill(search_bar,store_name,store_address)
click_on(search_button)
find_stores(user_position,store_name,store_address)
alt [corresponding_stores==empty]
then
[corresponding_stores=empty] show(error_msg)
click_on(retry_button)
else
[corresponding_stores! = empty] show(corresponding_stores)
click_on(corresponding_store)
alt no first visit to the store
calc_suggested_permanence_time()
show(permanence_time_box,suggested_permanence_time)
loop [phase_permanence_time==not passed]
opt specify permanence time
fulfill(permanence_time_box)
select(suggested_permanence_time,specified_permanence_time)
click_on(confirm_button)
alt [selected_permanence_time==specified_permanence_time]
check(specified_permanence_time)
then
alt [specified_permanence_time == invalid]
then
[phase_permanence_time = not passed] show(error_msg)
click_on(retry_button)
else
[phase_permanence_time = passed]
else
[phase_permanence_time = passed]
If the user confirm the suggested_permanence_time selecting it, it will be surely accepted by the system (it is forced to respect the constraints because it is calculated from previous specified_permanence_time, that were checked in the past). It is contained in the loop, only to give the possibility to the user to change choice.
show(means_of_transport_page)
click_on(means_of_transport)
checkWaitingTime()
checkTravelTime()
checkTicketAvailability()
alt [instant_ticket.status==not releasable]
then
show(error_msg)
click_on(ok_button)
show(home_page)
else
reserve_instant_ticket(waiting_time, permanence_time, travel_time)
show(number,QR_code,waiting_time)


interaction User reservation booked ticket
The user is already logged in
Actor: User
CLup
click_on(booked_ticket)
show(search_bar)
loop [corresponding_stores==empty]
fulfill(search_bar,store_name,store_address)
click_on(search_button)
find_stores(user_country,store_name,store_address)
alt [corresponding_stores==empty]
then
[corresponding_stores==empty] show(error_msg)
click_on(retry_button)
else
[corresponding_stores! = empty] show(corresponding_stores)
click_on(corresponding_store)
show(calendar,clock)
fulfill(calendar,clock,date,hour)
click_on(confirm_button)
check(date,hour,current_time)
alt [date==not possible || hour==not possible]
then
[date_hour_validity=false] show(error_msg,store_timetable)
click_on(ok_button)
show(home_page)
else
alt no first visit to the store
calc_suggested_permanence_time()
show(permanence_time_box,suggested_permanence_time)
loop [phase _permanence_time==not passed]
opt specify permanence time
fulfill(permanence_time_box)
select(suggested_permanence_time,specified_permanence_time)
click_on(confirm_button)
alt [selected_permanence_time==specified_permanence_time]
check(specified_permanence_time)
then
alt [specified_permanence_time == invalid]
then
[phase_permanence_time = not passed] show(error_msg)
click_on(retry_button)
else
[phase_permanence_time = passed]
else
[phase_permanence_time = passed]
checkTicketAvailability(date,hour)
alt [booked_ticket status==not releasable]
then
show(error_msg)
click_on(ok_button)
show(home_page)
else
reserve_booked_ticket(date,hour,permanence_time)
show(number,QR_code,dateTime)
If the user confirm the suggested_permanence_time selecting it, it will be surely accepted by the system (it is forced to respect the constraints because it is calculated from previous specified_permanence_time, that were checked in the past). It is contained in the loop, only to give the possibility to the user to change choice.


interaction Check in
Actor: User
Store: QR-reader
CLup
loop [noOneCanCheckIn == true || ticket.isValid == false]
The Visitor has to have already reserved a physical ticket and he/she has to be already at the entrance of the store
show_ticket()
ask_to_check_line()
checking_line()
alt [noOneCanCheckIn== true]
then
[noOneCanCheckIn = true]
show_feedback(negative)
else
[noOneCanCheckIn = false]
send(ticket)
check_if_valid(ticket)
A ticket is valid if is Current since no more than 2 minutes
alt [ticket.isValid == false]
then
[ticket.isValid = false]
show_feedback(negative)
else
[ticket.isValid = true]
allow_entrance()
show_feedback(positive)


interaction Visitor - Get phisycal ticket
Actor: Visitor
Store: Ticket Printer
CLup
press_on_tpbutton(physical_ticket)
get_new_ticket(physical_ticket)
checking_line()
alt Can print ticket
then
[lineUp = true] send(number, QR_code, waiting_time)
print(new_ticket)
else
[lineUp == false] send(error_msg)
print(error_msg)


interaction SM - add store
SM is already logged in the system
Actor: SM
CLup
click(add_new_store)
loop [add_store_form.correctness == false]
show(add_store_form)
fullfill(add_store_form)
send(add_store_form)
Form is not correct if SM try to insert an already available store
check(add_store_form)
alt [add_store_form.correctness == false]
then
[add_store_form.correctness = false] show(error_msg)
else
[add_store_form.correctness = true] show(positive_feedback)
show(home_page)


interaction SM - Edit some store's field
The SM has already found the store in the system with the search option.
Actor: SM
CLup
click(edit)
show(list_of_field)
chose(field)
fullfill(field.value)
send(field.value)
update(store.field.value)
show(positive_feedback)
show(home_page)

3.3 Performance Requirements

The system has to be able to respond to a possibly great number of simultaneous requests and more generally to a great number of requests throughout the day. Basing the analysis on people necessities, an average of 5M subscribed users and 8M reservations per day may have to be expected and so correctly managed.

3.4 Design constraints

3.4.1 Standards compliance

3.4.2 Hardware limitations

Mobile App

- iOS or Android smartphone
- 2G/3G/4G connection
- GPS

Web App

- Internet connection

3.4.3 Design Constraint

3.4.3.1 QR code and ticket number

- The QR code unicity can be guaranteed creating a QR code from the concatenation of the lineup time, the name and the address of the store associated to the ticket.
- The number unicity of a ticket during the store's opening hours of a day, is necessary to let the user/visitor recognize thanks to the queue board his/her turn.

3.4.3.2 Any other constraint

- Apache as Web server.
- TomcatEE as Application server.
- Oracle as DBMSto store data persistently.
- RESTFull and JSON for API communication over HTTP(S).
- Javascript, CSS and HTML to create responsive site that communicate to server using REST API. These files are got via HTTP(S).
- Modern browser with javascript and ajax support.
- Xamarin or Flutter respectively for both android and iOS apps.

- External maps API, such as Google Maps for Android users and Apple Maps for iOS users.

3.5 Software System Attributes

3.5.1 Reliability

All the components of the system work properly almost all the time, and the few periods of failure, respecting the constraints showed for the Availability, expire almost immediately. This is possible thanks to the preventive maintenance that can be operated on the various system's parts and CLup store devices.

The last are the components more subject to failures , so more bigger is an available store, more large should be the number of CLup store's devices (in order to serve a greater number of customers), and if one of them goes down, it does not affect all the reliability in the store thanks to the same type of devices which still work in parallel.

Guarantee the correct storage of user's and store's data is of primary importance for all the system functionalities, so specific backup, replicas and recovery measures about these data must be taken in consideration.

3.5.2 Availability

The system must guarantee a 24/7 service. But, more in detail it should be up for the 99% of time and it means that the downtime will be 3.65 days/year approximately. This downtime includes all the possible time's period of failure and maintenance.

3.5.3 Security

User's credentials need to be stored and encrypted to guarantee a high level of privacy. Generated password must be hashed/encrypted while send to the user and stored in safe databases. Possible thefts and attacks to user's credentials and passwords must be prevented.

Finally, security of the communications user-system and CLup store's devices-central system is also of primary concern.

3.5.4 Maintainability

The system must guarantee a high level of maintainability. Code must be written with good standard with high level of abstractions without hard code as well, and with large use of documentations that cover all the main aspect of code itself.

3.5.5 Portability

The software must run in different platforms like Windows operating system, Linux operating system and Mac operating system. Also, must support Android and iOS operating systems for mobile devices.

3.5.6 Scalability

System must be designed to allow and not prevent easy introduction of new functionalities. Moreover, it must be possible to enlarge the system's infrastructure at any time to support a greater number of users.

4 Fromal analysis using Alloy

4.1 Signatures

```
open util/time
open util/integer

sig Duration{
	start: one Time,
	end: one Time
}{
	 gt[end,start]
}

sig Position{}

sig  User{
	username: one String,
	position: Position lone -> lone Time
}

sig Visitor{}

sig QRcode{}

sig QRreader{
	 scanned: QRcode lone-> lone Time,
	 acceptableQRcodes:  QRcode -> Time
}

sig QueueBoard{
	 show:  Int lone -> lone Time
}

sig TicketPrinter{
	toPrint: PhysicalTicket lone -> lone Time
}

abstract sig Ticket{
	store: one Store,
	qrcode: one QRcode,
	number: one Int,
	permanenceTime: one Duration,
	checkInTime: lone Time,
	lineUpTime: one Time,
	ticketStatus: TicketStatus lone -> lone Time
}{
	number > 0
	 gt[checkInTime, lineUpTime]
	 #ticketStatus > 0 and #ticketStatus < 5
}
```

```
abstract sig TicketStatus{}
        one sig Released extends TicketStatus{}
        one sig Current extends TicketStatus{}
        one sig CheckedIn extends TicketStatus{}
        one sig Expired extends TicketStatus{}

sig InstantTicket extends Ticket{
       user: one User,
       waitingTime: one Duration,
       travelTime: one Duration
}{
        gte[travelTime.start,waitingTime.start]
        gte[travelTime.end,waitingTime.end] <=> travelTime.start = waitingTime.start
        waitingTime.start = lineUpTime
}

sig BookedTicket extends Ticket{
       reservation: one Time,
       user: one User
}{
        gt[reservation, lineUpTime]
}

sig PhysicalTicket extends Ticket{
       visitor: one Visitor,
       waitingTime: one Duration
}{
        waitingTime.start = lineUpTime
        gt[permanenceTime.start,waitingTime.end]
}

sig Store{
        name: one String,
        address: one String,
        defaultPermanenceTime: set Duration,
        maximumPeopleCapacity: one Int,
        qrReader: some QRreader,
        queueBoard: some QueueBoard,
        ticketPrinter: some TicketPrinter
}{
        maximumPeopleCapacity > 0
        #defaultPermanenceTime>2
}
```

4.2 Static Model

```
fact defaultPermanenceTimeSameLenghtInAStore{
        all s:Store |
                (all pt: s.defaultPermanenceTime| pt.end = pt.start.next)
                or
                (all pt: s.defaultPermanenceTime| pt.end = pt.start.next.next)
                or
                (all pt: s.defaultPermanenceTime| pt.end = pt.start.next.next.next)
}

fact usernameUnique{
        no disjoint u1,u2: User | u1.username=u2.username
}

fact storeNameAddressUnique{
        no disjoint s1,s2: Store | s1.name=s2.name and s1.address=s2.address
}

fact QRcodeIsUnique{
        no disjoint t1,t2: Ticket | t1.qrcode=t2.qrcode
}

fact oneInstantTicketPerUser{
        no disjoint t1,t2: InstantTicket | t1.checkInTime=none and  t2.checkInTime=none and t1.user=t2.user
}

fact oneBookedTicketPerUser{
        no disjoint t1,t2: BookedTicket | t1.checkInTime=none and  t2.checkInTime=none and t1.user=t2.user
}

fact noQRcodeWithoutATicket{
        all qc: QRcode| one tck: Ticket | qc=tck.qrcode
}

fact noVisitorWithoutTicket{
        all v: Visitor | some pt:PhysicalTicket| pt.visitor=v
}

fact noPositionWithoutUser{
        all p: Position | some u: User, t:Time | u.position.t=p
}
```

```
fact ticketStateChart{
	--These first 3 lines are only to generate a world in which is possible to see all the possible states
	some tck: Ticket | one t': Time | tck.ticketStatus.t' = Current
	some tck: Ticket | one t': Time | tck.ticketStatus.t' = CheckedIn
	some tck: Ticket | one t': Time | tck.ticketStatus.t' = Expired
	all tck: Ticket | all t: TicketStatus.(tck.ticketStatus) |
		--Once a Ticket is "Expired"...
		(tck.ticketStatus.t = Expired =>
			((all t':TicketStatus.(tck.ticketStatus) | t!=t' and gte[t',t] => tck.ticketStatus.t' != CheckedIn and tck.ticketStatus.t' != Current)
			and
			(one t'': TicketStatus.(tck.ticketStatus) | tck.ticketStatus.t'' = Current and t''=t.prev.prev.prev)))
		and
		--Once a Ticket is "Current"...
		(tck.ticketStatus.t = Current =>
			((all t': TicketStatus.(tck.ticketStatus) | t!=t' and gte[t',t] => tck.ticketStatus.t' != Released)
			and
			(one t'': TicketStatus.(tck.ticketStatus) | tck.ticketStatus.t'' = Released)))
		and
		--Once a Ticket is "CheckedIn"...
		(tck.ticketStatus.t = CheckedIn =>
			((all t': TicketStatus.(tck.ticketStatus) | t!=t' and gte[t',t] => tck.ticketStatus.t' != Current and tck.ticketStatus.t' != Released)
			and
			(one t'': TicketStatus.(tck.ticketStatus) | tck.ticketStatus.t'' = Current)))
}

fact lineUpTimeIsReleasedStatusTime{
	all tck: Ticket|
		tck.lineUpTime = Released.(tck.ticketStatus)
}

fact checkInTimeIsCheckedInStatusTime{
	all tck: Ticket|
		tck.checkInTime = CheckedIn.(tck.ticketStatus)
}

fact noDevicewithoutStore{
	(all qr:QRreader | one s:Store | qr in s.qrReader)
	and
	(all qb:QueueBoard | one s:Store | qb in s.queueBoard)
	and
	(all tp:TicketPrinter |  one s:Store | tp in s.ticketPrinter)
}

fact permancenceTimeIsDefaultPermanenceTimeForVisitor{
	all pt: PhysicalTicket | one d: Duration|
		pt.permanenceTime = d and (d in pt.store.defaultPermanenceTime)
}
```

```
fact oneCurrentTicketForStoreAtTime{
	no disjoint tck1,tck2: Ticket|one t:Time | tck1.ticketStatus.t=Current and  tck2.ticketStatus.t=Current and tck1.store=tck2.store
}

fact oneUserPositionAtLiningUpInstantTicketTime{
	all tck:InstantTicket | one p:Position|
		tck.lineUpTime in p.(tck.user.position)
}

fact ticketPermanenceTimeCorrespondencesIfCheckedIn{
	all tck: Ticket |
		CheckedIn.(tck.ticketStatus)!=none =>
			tck.checkInTime = tck.permanenceTime.start
}

fact ticketScannedBeforeCheckedIn{
	all tck: Ticket |
		CheckedIn.(tck.ticketStatus) != none =>
			(one qr: QRreader | (qr in tck.store.qrReader) and (tck.qrcode)->(CheckedIn.(tck.ticketStatus).prev) in qr.scanned)
}

fact ticketWaitingTimeTravelTimeCorrsepondences{
	all tck: PhysicalTicket |
		Current.(tck.ticketStatus)!=none =>
			(tck.waitingTime.end=Current.(tck.ticketStatus))
		else
			(gt[tck.waitingTime.end, Released.(tck.ticketStatus)])
	all tck: InstantTicket |
			Current.(tck.ticketStatus)!=none =>
				(gte[tck.waitingTime.end, tck.travelTime.end] =>
					(tck.waitingTime.end=Current.(tck.ticketStatus))
				else
					(tck.travelTime.end=Current.(tck.ticketStatus)))
			else
				(gt[tck.waitingTime.end, Released.(tck.ticketStatus)] and gt[tck.travelTime.end, Released.(tck.ticketStatus)])
}

fact bookedTicketBecomeCurrent{
	all tck: BookedTicket|
		tck.reservation = Current.(tck.ticketStatus)
}

fact physicalTicketStoreEqualTicketPrinterStore{
	all tp: TicketPrinter, tck: tp.toPrint.Time | tp in tck.store.ticketPrinter
}

fact queueBoardShowTheCurrentTicketNumberOfStore{
	(all tck: Ticket, t:Time |
		tck.ticketStatus.t = Current => (all qb: tck.store.queueBoard| qb.show.t = tck.number))
	and
	(all t:Time, s: Store | (all tck: store.s| tck.ticketStatus.t != Current) <=> s.queueBoard.show.t  = none)
}

fact oneTicketPrinterPrintPhysicalTicketAtReleasingTime{
	(all tck: PhysicalTicket|
		one tp: TicketPrinter|
			(tp in tck.store.ticketPrinter) and (tck->(Released.(tck.ticketStatus)) in tp.toPrint) and (all tp':TicketPrinter| tp!=tp' => tck.(tp'.toPrint) = none))
	(all tp: TicketPrinter |
		all t: PhysicalTicket.(tp.toPrint) | tp.toPrint.t.ticketStatus.t = Released)
}

fact whenQRcodeAreAcceptableQRcodes{
	all tck: Ticket, t:Time | all qr: QRreader|
		(gte[t,Current.(tck.ticketStatus)] and (qr in tck.store.qrReader) and (CheckedIn.(tck.ticketStatus)!= none => gt[CheckedIn.(tck.ticketStatus),t])
		 and (Expired.(tck.ticketStatus)!=none => gt[Expired.(tck.ticketStatus),t]))
			<=> (tck.qrcode->t in qr.acceptableQRcodes)
}
```

4.3 Dynamic Model

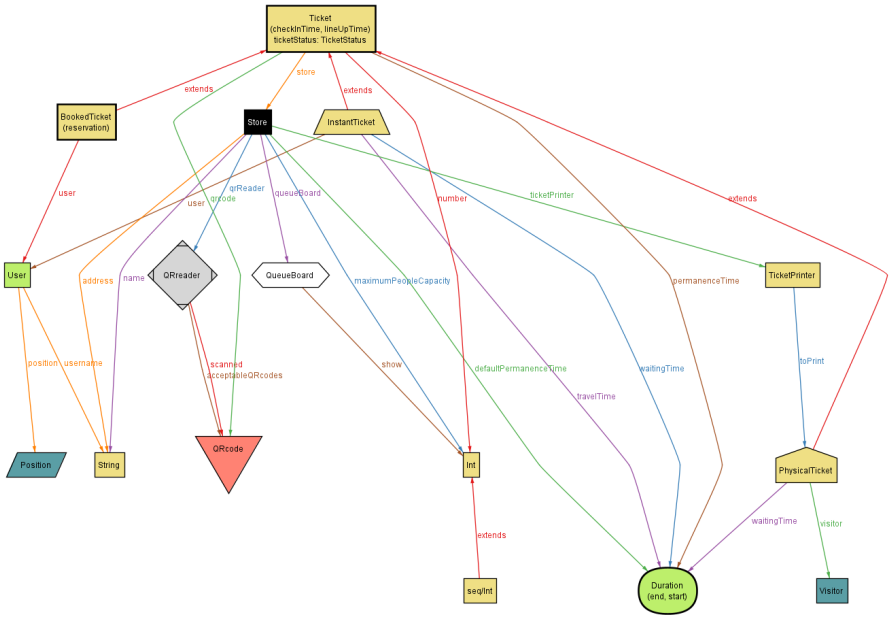
```
pred hasUserAValidInstantTicket[u:User,t:Time]{
        one tck: InstantTicket | tck.user=u and tck.ticketStatus.t != Expired
}

pred hasUserAValidBookedTicket[u:User,t:Time]{
        one tck: BookedTicket | tck.user=u and tck.ticketStatus.t != Expired
}

pred hasVisitorAValidPhysicalTicket[v:Visitor,t:Time]{
        one tck: PhysicalTicket | tck.visitor=v and tck.ticketStatus.t != Expired
}

pred hasUserAValidInstantTicketForTheStore[u:User,t:Time,s:Store]{
        one tck: InstantTicket| tck.user=u and tck.ticketStatus.t != Expired and tck.store=s
}

pred hasUserAValidBookedTicketForTheStore[u:User,t:Time,s:Store]{
        one tck: BookedTicket| tck.user=u and tck.ticketStatus.t != Expired and tck.store=s
}

pred userMakeAInstantTicketReservation[u:User,s:Store,t:Time,tck: InstantTicket]{
        //preconditions
        not hasUserAValidInstantTicket[u,t]
        not hasUserAValidBookedTicketForTheStore[u,t,s]
        //postconditions
        tck.user=u
        tck.store=s
        hasUserAValidInstantTicket[u,t.next]
        not hasUserAValidBookedTicketForTheStore[u,t.next,s]
        tck.ticketStatus.(t.next) = Released
}

pred userMakeABookedTicketReservation[u:User,s:Store,t:Time,tck: BookedTicket]{
        //preconditions
        not hasUserAValidBookedTicket[u,t]
        not hasUserAValidInstantTicketForTheStore[u,t,s]
        //postconditions
        tck.user=u
        tck.store=s
        hasUserAValidBookedTicket[u,t.next]
        not hasUserAValidInstantTicketForTheStore[u,t.next,s]
        tck.ticketStatus.(t.next) = Released
}
```

```
pred visitorMakeAPhysicalTicketReservation[v:Visitor,s:Store,t:Time,tck:PhysicalTicket]{
        //postconditions
        tck.visitor=v
        tck.store=s
        tck.ticketStatus.(t.next) = Released
}

pred userChekInWithInstantTicket[u:User,s:Store,t:Time,tck:InstantTicket]{
        //preconditions
        hasUserAValidInstantTicketForTheStore[u,t,s]
        not hasUserAValidBookedTicketForTheStore[u,t,s]
        tck.store=s
        tck.user=u
        gt[t,Current.(tck.ticketStatus)]
        //postconditions
        CheckedIn.(tck.ticketStatus)=t.next
}

pred userChekInWithBookedTicket[u:User,s:Store,t:Time,tck:BookedTicket]{
        //preconditions
        hasUserAValidBookedTicketForTheStore[u,t,s]
        not hasUserAValidInstantTicketForTheStore[u,t,s]
        tck.store=s
        tck.user=u
        gt[t,Current.(tck.ticketStatus)]
        //postconditions
        CheckedIn.(tck.ticketStatus)=t.next
}

pred visitorChekInWithPhysicalTicket[v:Visitor,s:Store,t:Time,tck:PhysicalTicket]{
        //preconditions
        hasVisitorAValidPhysicalTicket[v,t]
        tck.store=s
        tck.visitor=v
        gt[t,Current.(tck.ticketStatus)]
        //postconditions
        CheckedIn.(tck.ticketStatus)=t.next
}

pred show{
        #BookedTicket>0
        #InstantTicket>0
        #PhysicalTicket>0
}
```

4.4 Results

run show for 6 but 10 int, exactly 2 String
run userMakeAInstantTicketReservation for 6 but 10 int, exactly 2 String
run userMakeABookedTicketReservation for 6 but 10 int, exactly 2 String
run visitorMakeAPhysicalTicketReservation for 6 but 10 int, exactly 2 String
run userChekInWithInstantTicket for 6 but 10 int, exactly 2 String
run userChekInWithBookedTicket for 6 but 10 int, exactly 2 String
run visitorChekInWithPhysicalTicket for 6 but 10 int, exactly 2 String

4.4.1 Generated World



4.4.2 Proof of consistency

```
Executing "Run show for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1349359 vars. 51042 primary vars. 3550577 clauses. 4110ms.
  Instance     found. Predicate is consistent. 3328ms.

Executing "Run userMakeAInstantTicketReservation for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1350421 vars. 51066 primary vars. 3553550 clauses. 4366ms.
  Instance     found. Predicate is consistent. 8995ms.

Executing "Run userMakeABookedTicketReservation for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1350421 vars. 51066 primary vars. 3553550 clauses. 3825ms.
  Instance     found. Predicate is consistent. 3450ms.

Executing "Run visitorMakeAPhysicalTicketReservation for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1349737 vars. 51066 primary vars. 3551191 clauses. 3972ms.
  Instance     found. Predicate is consistent. 5650ms.

Executing "Run userChekInWithInstantTicket for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1350215 vars. 51066 primary vars. 3552829 clauses. 4013ms.
  Instance     found. Predicate is consistent. 4726ms.

Executing "Run userChekInWithBookedTicket for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1350215 vars. 51066 primary vars. 3552829 clauses. 4060ms.
  Instance     found. Predicate is consistent. 7519ms.

Executing "Run visitorChekInWithPhysicalTicket for 6 but 10 int, exactly 2 String"
  Solver=sat4j Bitwidth=10 MaxSeq=6 SkolemDepth=1 Symmetry=20
  1350029 vars. 51066 primary vars. 3552164 clauses. 3990ms.
  Instance     found. Predicate is consistent. 7070ms.

7 commands were executed. The results are:
  #1: Instance found.     show is consistent.
  #2: Instance found.     userMakeAInstantTicketReservation is consistent.
  #3: Instance found.     userMakeABookedTicketReservation is consistent.
  #4: Instance found.     visitorMakeAPhysicalTicketReservation is consistent.
  #5: Instance found.     userChekInWithInstantTicket is consistent.
  #6: Instance found.     userChekInWithBookedTicket is consistent.
  #7: Instance found.     visitorChekInWithPhysicalTicket is consistent.
```

5 Effort Spent

5.1 Table of the work days

Day	People	Time	Rasd Section	Topic
14/10/20	Edo	2h	-	Build RASD with LyX(Frontispiece and skeleton of sections)
15/10/20	Group	2h.30m	1.Introduction	Purpose and Goals
16/10/20	Edo	30m	1.Introduction	Frontispiece
	Group	3h.30m	1.Introduction	Scope and Goals
	Leo	2h.45m	1.Introduction	Goals, Defin/Acron/Abbrev
17/10/20	Group	1h	2.Overall Description	Product Perspective
18/10/20	Group	2h	2.Overall Description	Product Functions, User characteristics
19/10/20	Edo	2h	-	1. and 2. General Review
20/10/20	Leo	1h.30m	2.Overall Description	Class Diagram
21/10/20	Group	1h.30m	2.Overall Description	Text assumptions
22/10/20	Group	1h	3.Specific Requirements	Functional Requirements
	Edo	2h.30m	2.Overall Description	State Diagram
23/10/20	Group	3h	3.Specific Requirements	Mapping goal, domain assumptions and requirements
24/10/20	Leo	4h.30m	3.Specific Requirements	Mapping goal, domain assumptions and requirements
25/10/20	Group	1h.30m	-	General Review
	Edo	3h.30m	-	General Review
27/10/20	Group	3h	3.Specific Requirements	External Interface Requirements, Software system attributes

	Edo	2h	3.Specific Requirements	Mockups
28/10/20	Group	1h.30m	3.Specific Requirements	Use cases
29/10/20	Group	3h	3.Specific Requirements	Use cases, Scenarios
30/10/20	Group	2h	3.Specific Requirements	Use case diagram
01/11/20	Group	2h	3.Specific Requirements	Sequence Diagrams
02/11/20	Leo	1.30h	3.Specific Requirements	Sequence Diagrams
03/11/20	Edo	2h	3.Specific Requirements	Sequence Diagrams
	Leo	2h	3.Specific Requirements	Sequence Diagrams
5/11/20	Edo	3h	4.Alloy	Signatures
	Leo	3h	4.Alloy	Signatures
6/11/20	Group	6h	4.Alloy	Facts
7/11/20	Leo	2h	4.Alloy	Facts
	Edo	2h	4.Alloy	Facts
09/11/20	Group	5h	-	General Review
10/11/20	Leo	8h	4.Alloy	Facts
11/11/20	Edo	3h	-	Build RASD with LyX
	Edo	2h	3.Specific Requirements	UML review
	Leo	5h	4.Alloy	Facts,Pred
12/11/20	Edo	2h	-	Build RASD with LyX

5.2 Total work hour per RASD section

- **1.Introduction** = 9.15h
- **2.Overall Description** = 8.30h
- **3. Specific Requirements** = 29.30h
- **4.Alloy** = 29h

5.3 Total work hour per person

- **Edo** = 26.30m
- **Leo** = 30h.15m

- **Group** =33h.30m

6 References

6.1 Used tool

- LyX - v2.3.5.2
- Microsoft Word
- Github
- Adobe Reader PDF
- StarUML - v3.2.2
- Adobe XD
- Alloy analyzer - v4.2_2015-02-22