POLITECNICO DI MILANO

Computer Science and Engineering

Requirements Analysis and Specifications Document

Customers Line-up

Software Engineering 2 Project

Academic year 2020 - 2021

20 December 2020

Version 1.1

Authors:
Samuele NEGRINI
Giorgio PIAZZA

Professor:
Matteo Giovanni ROSSI

Revision history

Date	Revision	Notes
06/12/2020	v.1.0	First release.
20/12/2020	v.1.1	– General document revision. – Add diagrams descriptions. – Improve readability.

Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of Customers Line-up (CLup). It provides functional and non-functional requirements for the development of the system, including use cases, features, user interaction and system constraints.

This document is addressed to the developers who have to implement the requirements and could be used as an agreement between the customer and the contractors.

1.1.1 Goals

The ambition is that the adoption of these requirements will avoid having crowds inside facilities such as supermarkets and will relieve the consequent long lines forming outside. This matter is particularly relevant during a pandemic emergency, like the one that is going on in the present day.
Below are presented the goals of CLup. Further description will be discussed in section 2.2.

G.1 Avoid the creation of physical queues outside stores.

- **G.1.1** Allow customers to avoid the creation of hazardous situations.
- **G.1.2** Allow supermarkets to avoid the creation of hazardous situations.
- **G.1.3** Shorten the amount of time a customer is in queue.
- **G.1.4** Allow customers to arrive at stores right on time.

G.2 Allow customers to take advantage of the time they are queued up to do other things.

G.3 Grant customers an overall experience as easy as possible.

G.4 Allow customers, even the ones who don't have access to technology, to enjoy the service.

G.5 Allow supermarkets to monitor access to stores in a better way.

G.6 Allow supermarkets to know in advance how many people are coming to stores.

G.7 Allow supermarkets to limit the number of access to stores.

1.2 Scope

Customer Line-up (CLup) is an **easy-to-use** application which aims to settle for various queuing problems faced by supermarkets and their customers.
On the one side, it allows store managers to regulate the influx of people in the building and, on the other side, it saves people from having to line up and stand outside of stores for hours on end.

Customers can enter a queue in *real-time* by **taking a ticket** via different channels such as Self Service Ticketing Kiosk and Mobile App. During this process, the user is given an estimation of the waiting time and the **leave-at-time** (i.e. the time they need to depart from their current position to reach the store). This ticket comprehends a queue number, which identify user's position in the queue, and a QR code, which is used for the ticket validation.
The validation process will be performed by a store employee using a dedicated application.

The system offers store managers a way to monitor the customers flow inside the store at any given time.

The platform also support an *advanced functionality* where a customer can **book a visit** to the store by indicating the approximate expected duration of the trip and the main categories of items they intend to buy. For long-term customers, it suggests a time inferred by the system based on an analysis of the previous visits.

This application works as a digital counterpart to the common situation where people who are in line for a service retrieve a number that gives their position in the queue. The *legacy system* will be completely superseded by the application. Indeed, its effectiveness is strictly bound to the number of users who use it.

1.2.1 World Phenomena

WP.1 Customers choose which store to go to.

WP.2 Customers approach the chosen store.

WP.3 Supermarkets restrict access to their stores.

WP.4 Supermarkets monitor influx of people in the building.

WP.5 Customers line up outside the store.

WP.6 Customers wait their turn.

1.2.2 Shared Phenomena

SP.1 Ticket is retrieved from the customer remotely.

SP.2 Ticket is retrieved from the customer locally at the store.

SP.3 Customers can book a time-slot at the store.

SP.4 Customers check the queue status of the store.

SP.5 Ticket are validated by the system.

SP.6 Supermarkets monitor store data using the system.

1.3 Glossary

1.3.1 Definitions

Term	Definition
Customers	Identifies the store customers.
Employees	Used in this document to mean both entrance-staff and cashiers.
Hazardous Situation	Identifies a situation of danger due to overcrowding.
Store Pass	General term that comprehends both tickets and bookings.
Ticket	Pass generated from the system which is comprehensive of the queue number and QR code.
Booking, Reservation	Pass generated from the system as a result of the reservation process.
Queue number	Identifies user's position in the queue.
QR code	Type of matrix barcode, used by the system for the ticket validation.
System	Totality of the hardware/software applications that contribute to provide the service concerned. Also referred as CLup, Application, Platform.
Legacy System	Used in this document to mean the physical ticketing system where you retrieve a ticket from a stand.

1.3.2 Acronyms

Acronyms	Term
CLup	Customers Line-up
QR	Quick Response
GPS	Global Positioning System
UI	User Interface

1.3.3 Abbreviations

Abbreviations	Term
e.g.	Exempli gratia
i.e.	Id est
w.r.t.	With reference to
G	Goal
WP	World Phenomena
SP	Shared Phenomena
R	Requirement

1.4 Reference documents

- Project assignment specification document.
- ISO/IEC/IEEE 29148 - Systems and software engineering.
- Course slides on BeeP.

1.5 Document Structure

This document is presented as it follows:

1. **Introduction**: contains a preamble of the given problem and proposes, in a simple way, the system and its goals as solution.
2. **Overall Description**: gives a general description of the system, focusing on its functions and constraints. Moreover, it provides the domain assumptions of the analysed world.
3. **Specific Requirements**: explains in detail the functional and non functional requirements. It lists the possible interactions with the system in the form of scenarios, use cases and sequence diagrams.
4. **Formal Analysis Using Alloy**: contains the Alloy model of some critical aspects of the system and an example of the generated world.
5. **Effort Spent**: keeps track of the time spent to complete this document. The first table defines the hours spent as a team for taking the most important decisions, the seconds contain the individual hours.

Chapter 2

Overall Description

2.1 Product perspective

The system will be developed from scratch and it will completely replace the legacy system.

2.1.1 Class diagram

The class diagram in Figure 2.1 is a high-level representation of the system as a whole.
The main elements in the class diagram are:

- **User**: identifies three categories of users which can access the system with credentials: **Manager**, **Employee** and **Admin**. This distinction is necessary to assign different levels of permissions.
- **Store**: identifies a store with all the relevant information about it.
- **TimeSlot**: identifies the time slots a store can have and it comprehends a list of the bookings that are assigned to them.
- **OpeningHours**: identifies the opening hours of a store.
- **Queue**: identifies the virtual queue of tickets in a store.
- **StorePass**: identifies the parent class of both **Ticket** and **Booking**. It contains information shared with both.
- **PassStatus**: identifies the status of a store pass which can be: **Valid**, **Active** or **Expired**.
- **BookingStatus**: identifies the confirmation status of a booking: **Pending** or **Confirmed**.
- **ItemCategory**: identifies the categories of items that are present in a store.

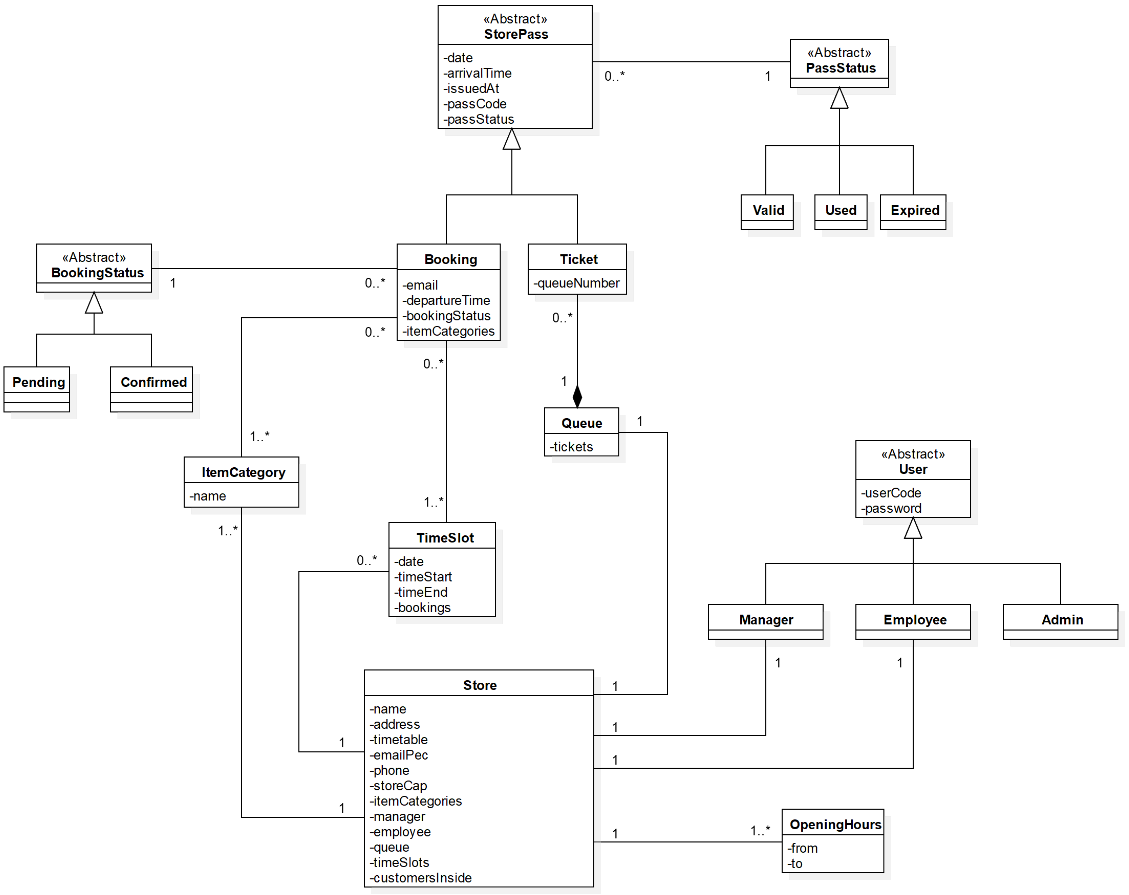


Figure 2.1: Class diagram.

2.1.2 State diagrams

State diagrams describe the behaviour of the system while considering all possible states the objects can have when an event occurs. This analysis helps to clarify the most critical aspects of the system.

The state diagram in Figure 2.2 represents the life cycle of a *Booking*, starting from the creation to the expiration.
After filling a form, the data is sent to the system where the availability of time slots is checked and also that the customer has not already booked for that day. A failure in any of these checks will abort the booking creation process.
Then the booking is created, its status is set to *Pending* and a confirmation link is sent via email. If the time for the confirmation expires the booking is deleted.
Otherwise, its status is updated to *Confirmed* and is ready for validation by the store (see Figure 2.3 for further details).

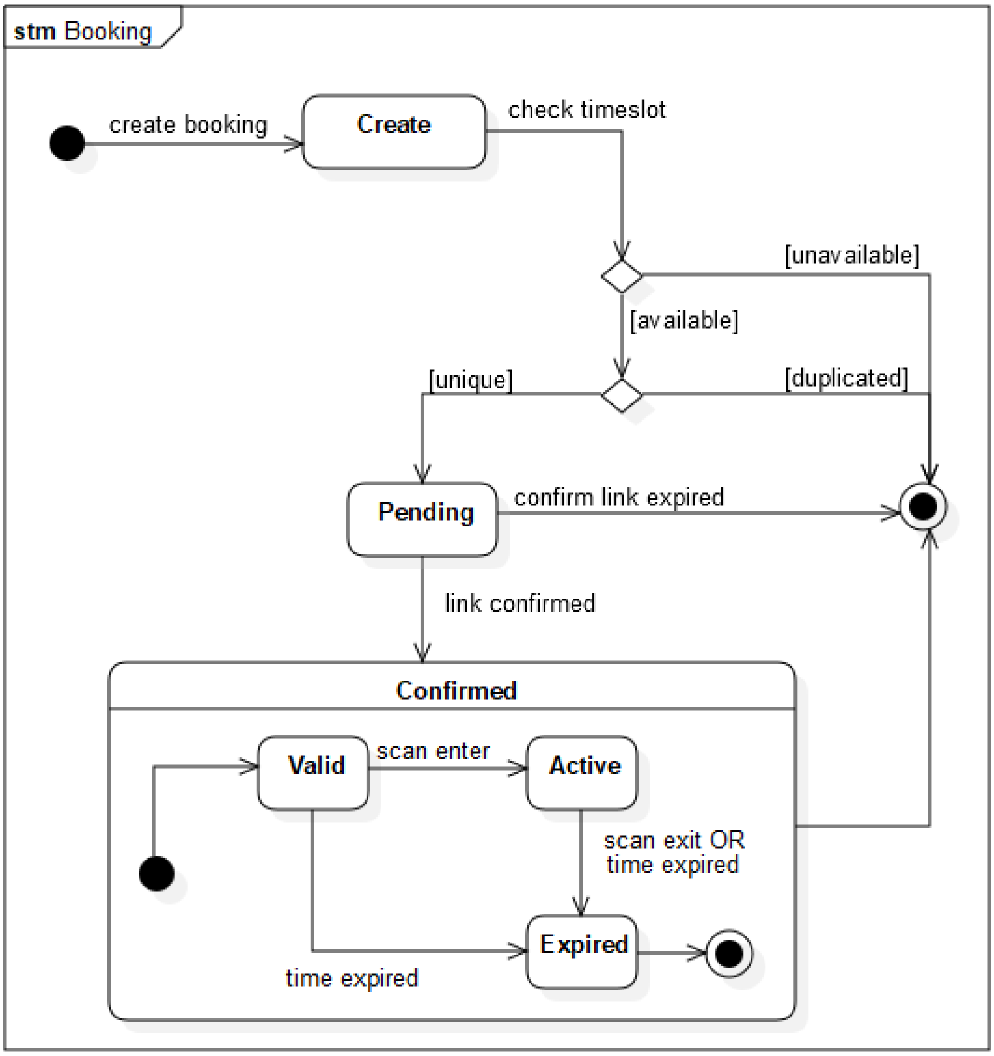


Figure 2.2: State diagram: Booking.

The state diagram in Figure 2.3 represents the validation process of a store pass.
Upon scanning a QR code, a validation request is sent to the system. Then the code existence is checked, if it does not exist the validation process fails. Otherwise, the process flow is inferred by the current status of the store pass.

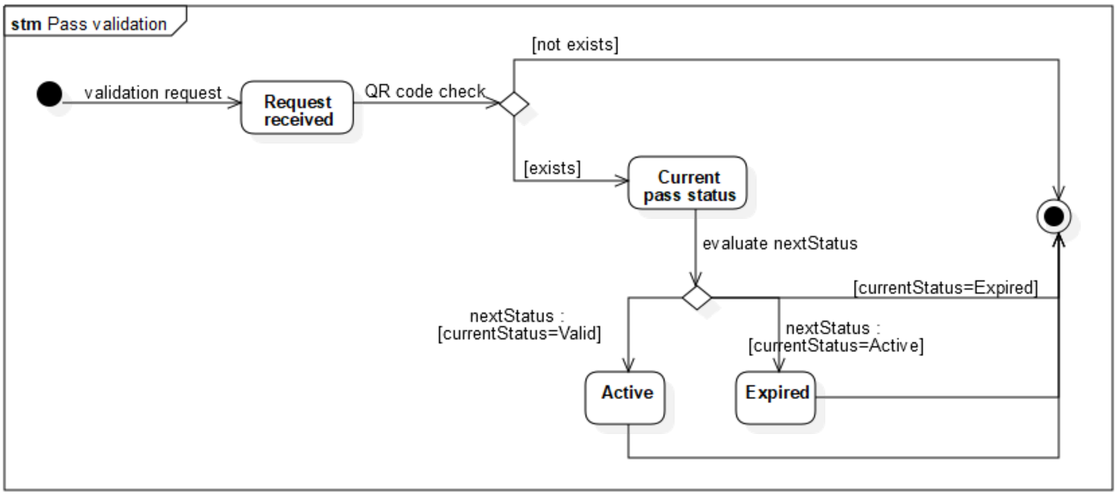


Figure 2.3: State diagram: Pass validation.

2.2 Product functions

This section provides a summary of the major functions that the software will perform w.r.t. the goals already described in section 1.1.1.

2.2.1 Store pass function

A Store pass consists of a QR code and a pass status. The latter identifies the current status inside the pass life-cycle:

- VALID: pass generated but not used yet.
- ACTIVE: pass in use.
- EXPIRED: pass no longer acceptable. The period of time for which it could be used has ended or the pass has been successfully used.

The system also takes care of managing delays of customers: a delay greater than 15 minutes will result in the expiration of the pass. This implies that the customer will not be able to enter the store with that pass, instead they shall create a new one.

Note that to benefit from these functions, the GPS service on customers' smartphones must be enabled, otherwise the application will not work.

Queue function

The main function of *CLup* is to manage queues. A user who wants to do grocery-shopping will join the queue for a specific store through the system.

Firstly, the user will be asked to select the store in which he desires to line-up.
Then the system will prompt the current size of the queue and an estimate of waiting time. If the user is satisfied with his choice it can subscribe to the queue, operation that will be confirmed by the emission of a ticket. The ticket comprehends a queue number, which identify user's position in the queue, and a QR code, which is used for the ticket validation.

At any time the user will be able to check his position in the queue and the *leave-at-time* (i.e. the time they need to depart from their current position to reach the store). When the user has to leave from where he is, the system sends a notification. The current position will be retrieved by the GPS of the user device.

Reserve function

This advanced function allows customers to "book a visit" to the store. A booking has an additional status attribute (i.e. booking status) which can be:

- PENDING: the booking has not been confirmed via email yet.
- CONFIRMED: the booking has been confirmed via email.

Customers are asked to select the store they want to visit. Then they need to fill in a form by indicating:

- an **email address**, which will be needed to send a receipt with the confirmation and deletion links;
- the **date** of the visit;
- the approximate **expected duration** of the trip;
- the main **categories of items** they intend to buy;
- a **time slot** chosen from the ones suggested by the system.

Every field of the form is mandatory, this means that the user must complete all of them in order to submit the reservation.

Customers will have to make reservations at least one day in advance.
The system will send an email with a *confirmation link* to the email address provided. To complete the booking process, the user must click the confirmation link within 24 hours from the time of subscription and in any case at least 1 hour before the chosen time slot. Users who fail to do so will have their booking expired.

Among the available store seats, about the 15% of them are reserved only for the bookings.
The time slots will be suggested according to the saturation of the grocery shelves.
The system will try to balance the number of people in each section of the store. In case of a particularly crowded shelve, the system will disable the booking for that time slot.

Customers are also allowed to delete their booking at any time by going to the right section of the application or by clicking the *deletion link* sent via email after the reservation. Also this function support the *leave-at-time* features, as described above in the Queue Function.

Both the *confirmation* and *deletion links* will land the user to a page of CLup where they will receive a response of the requested operation.

In addition, for long-term customers, the app suggests a time inferred by the system based on an analysis of the previous visits.

2.2.2 Validation function

The system provides an interface that, **upon authentication**, allows to scan QR codes by using the smartphone camera. This process is required in order to validate store passes.
Ticket validation allows to speed up the check-in process and to increase the number of people in the store. At the same time it helps staff to detect the authenticity of tickets brought to them.
The scan of the QR code will be executed:

- at the **entrance** by a dedicated store employee to control the influx at the store entrance.
- at the **exit** by the cashiers in order to notify the exit of customers from the store. In case any customer loses the QR code inside the store, the cashiers will use a backup QR code.

2.2.3 Monitor function

Customers Line-up platform grants supermarkets a way to manage queues and customers inside their stores. In particular, **upon authentication**, it offers two levels of access: *manager level* and *entrance-staff level*.

- **Manager level**
  With the **dashboard**, store managers have access to tables and data visualizations of their customers visits and behaviours. In particular, they can monitor the number of people in the queue and the ones inside the store. Then, based on those data they can setup the maximum cap of people inside the store.
  Store managers can view, edit and delete the list of reservations made by the customer. Last but not least, they can inspect information about the booked visits at any given time.
- **Entrance-staff level**
  A store employee is able to view data about the number of people in the building and the store passes which will be admitted inside. This is needed during the validation of tickets when the check-in staff scans QR and allows customers inside the store.

2.2.4 Register function

The system also grants CLup administrators an *admin level* access to register and manage stores. Specifically, an administrator of CLup is able to register new supermarkets and generate the respective credentials to access *manager-level* and *staff-level*. These credentials will be sent automatically via PEC to the respective stores.
In the event that the supermarkets credentials get lost or there is a need to change them, the administrator will generate new ones from the dashboard.

2.2.5 Scenarios

Scenario 1

John has to go to the supermarket to buy some groceries. Since there is a pandemic going on, he would like to go there as safely as possible.
So he opens CLup, chooses his trusted supermarket and takes a queue ticket. The application tells him to go to the supermarket at 12:30 and since John is about ten minutes away from the store it notifies him to leave at 12:15. Once in the store, an employee checks the ticket by scanning his QR code and John can enter the supermarket. At the exit the cashier will scan again the QR to notify the exit of a customer from the store.

Scenario 2

Tomorrow John is going to go for work near the largest Essecorta in the province. There is no better opportunity to go shopping in his favourite supermarket.
However, since he will be tight on time, he needs to book a visit to the store so he won't lose even a minute. Thus, he takes his phone and opens CLup, he chooses the store from the list and enters some information, like the category of items he would like to buy. He adds the desired time slot and finally opens the email that just arrived and confirms his booking: he is now ready to go.

Scenario 3

John's business appointment near the famous Essecorta has been cancelled. For this reason, he can no longer go to the store.
Since John is a model citizen, he decides to delete the booking that he did with CLup. So he opens the app, he goes to the store pass section and deletes the booking.

Scenario 4

The CLup admin Bob was contacted from the manager of Superal store near the Cathedral to provide CLup to his customers. After asking him some information about the store, like name and PEC address, the admin fills the form to register a new store.

Scenario 5

The Superal store has recently expanded due to some renovation works. The manager Alice wants to increase the number of maximum customer inside the store and monitor them.
For doing so, she logs into the system and increases the maximum cap. From that page she also watches the number of customers inside the store.

2.2.6 Use cases description

Use cases capture functional requirements of a system from the users' perspective.

Name	Retrieve ticket
ID	UC.1
Actors	Customer
Entry conditions	• The customer has access to the application on their device; • The application is running.
Flow of events	i. The customer selects a store from the home page; ii. The customer presses the "Retrieve Ticket" button. iii. The application notifies the customer when it's time to leave.
Exit conditions	The customer has retrieved a ticket for the selected store.
Exceptions	• If the customer has already retrieved a ticket for a store, the system displays an error message telling the customer they cannot get more than one ticket simultaneously. • If the system fails to retrieve the GPS location, no notification will be shown.

Table 2.1: *Retrieve ticket* use case description.

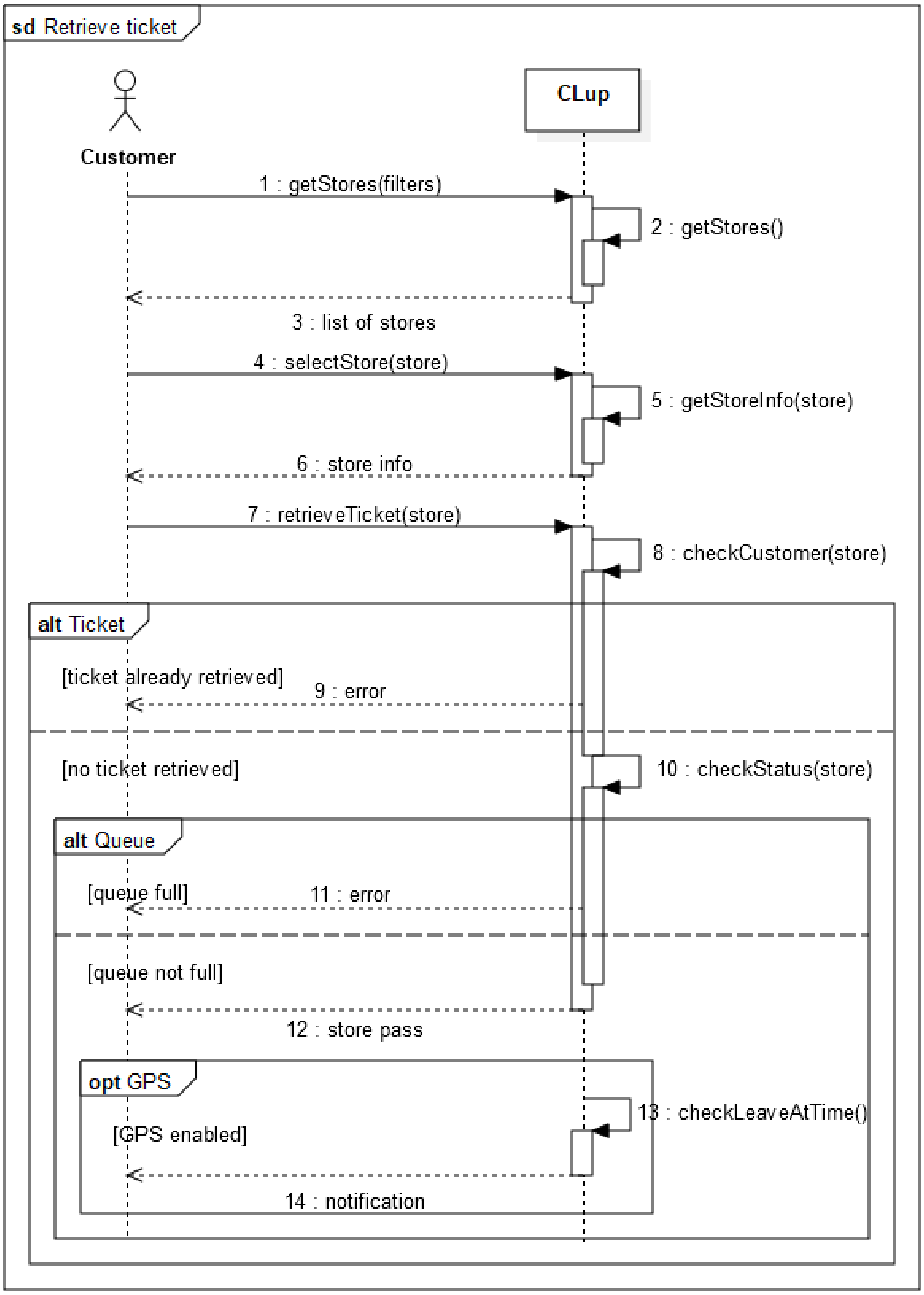


Figure 2.4: *Retrieve ticket* sequence diagram.

Name	Book a visit
ID	UC.2
Actors	Customer
Entry conditions	• The customer has access to the application on their device; • The application is running.
Flow of events	i. The customer selects a store from the home page; ii. The customer presses the "Book a visit" button; iii. The customer inserts their email address, the date, the approximate duration of the visit and the main categories of items they intend to buy; iv. The customer selects the time slot; v. The customer submits the form; vi. The system shows to the customer the booked visit. vii. The application notifies the customer when it's time to leave.
Exit conditions	The customer has booked a visit for the selected store.
Exceptions	• If the customer has already booked a visit to a store in a certain date, the system displays an error message telling the customer it cannot book again for the same day at the same store; • If the customer selects a date so that there are no more available time slots, the system displays an error message telling the customer to select a different date. • If the system fails to retrieve the GPS location, no notification will be shown.

Table 2.2: *Book a visit* use case description.

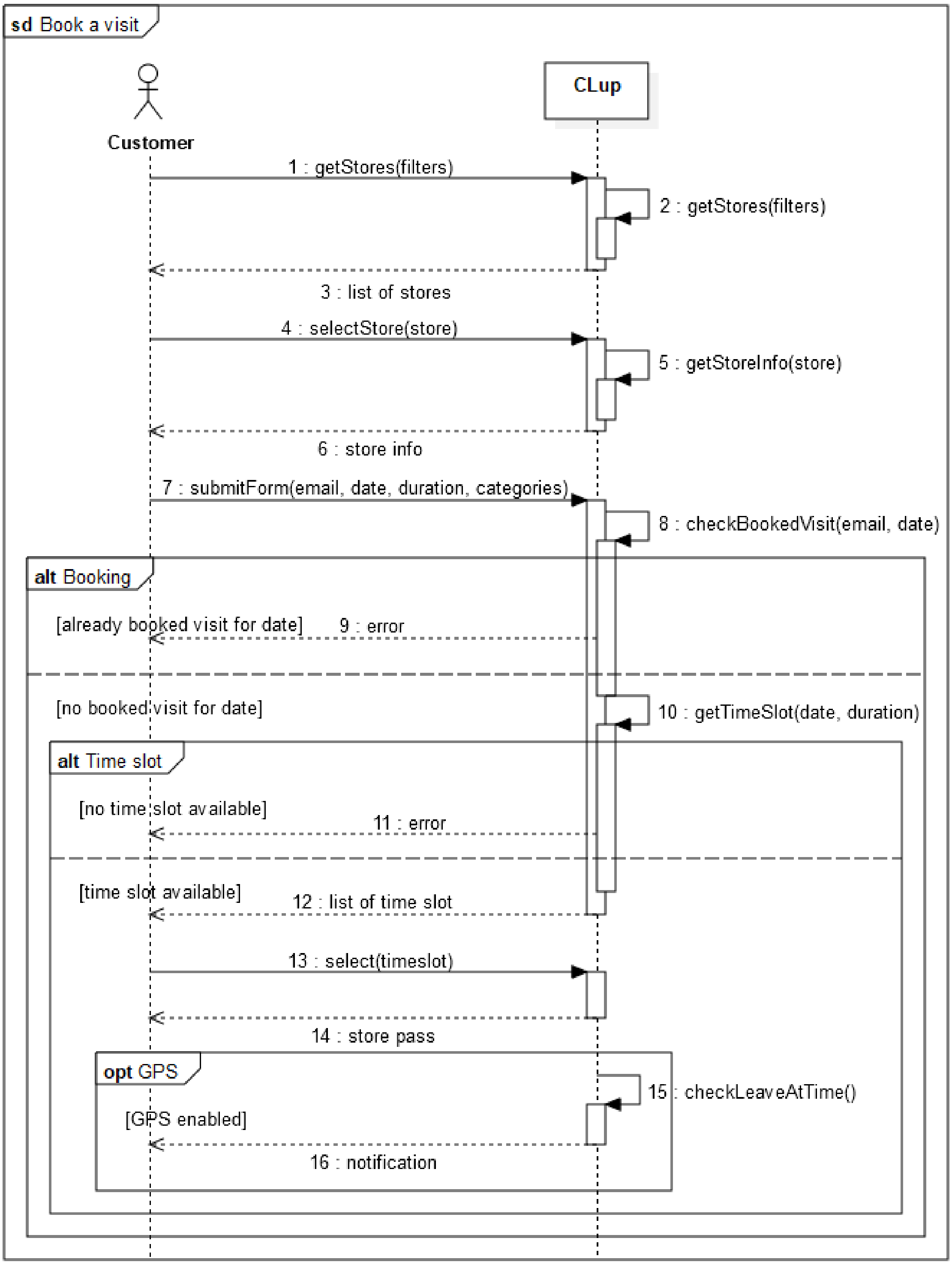


Figure 2.5: *Book a visit* sequence diagram.

Name	Delete a store pass
ID	UC.3
Actors	Customer
Entry conditions	• The customer has access to the application on their device; • The application is running.
Flow of events	i. The customer goes to the "My Store Pass" page; ii. The customer selects one of the available store passes; iii. The customer presses the "Delete Store Pass" button; iv. The customer selects "Yes" on the confirmation message; v. The customer is notified by the system that the store pass has been deleted successfully.
Exit conditions	The customer has deleted successfully a store pass.
Exceptions	If the customer has no store pass, the system displays a label: "No store pass available".

Table 2.3: *Delete a store pass* use case description.

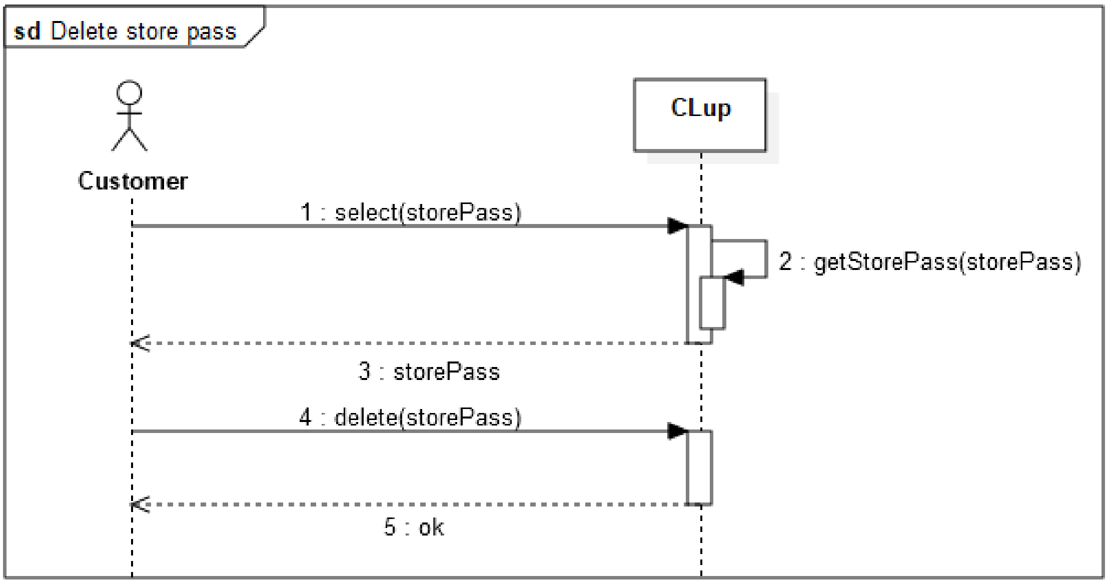


Figure 2.6: *Delete a store pass* sequence diagram.

Name	Platform login
ID	UC.4
Actors	Store manager, Store employee, CLup admin
Entry conditions	• The platform is running.
Flow of events	i. The actor goes to the login page; ii. The actor inserts their usercode and password; iii. The actor submits the form.
Exit conditions	The actor has successfully logged into the web platform.
Exceptions	• If the username is not recognized by the system, the credentials are not registered or the usercode is incorrect. The system notifies the actor and the procedure is aborted. • If the inserted password is wrong, the system notifies the actor and the procedure is aborted.

Table 2.4: *Platform login* use case description.

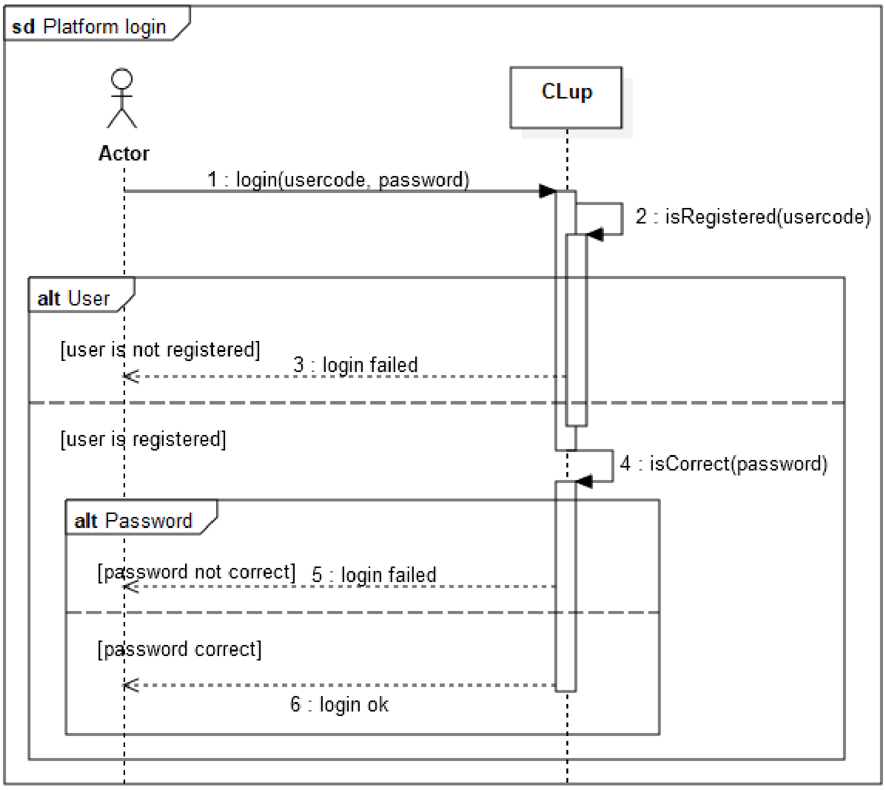


Figure 2.7: *Platform login* sequence diagram.

Name	Register new store
ID	UC.5
Actors	CLup admin
Entry conditions	• The platform is running. • The CLup admin is logged in. • The CLup admin has been contacted by a store manager.
Flow of events	i. The CLup admin goes to the "Add supermarket" page; ii. The CLup admin fills in the form providing store information: name, address, PEC and timetables. iii. The CLup admin submits the form. iv. The system generates credentials for the store managers and store employee and send them via email to the PEC address of the store.
Exit conditions	The CLup admin has added a store to the platform.
Exceptions	• If the store name is already used by another store, the system displays an error message asking the CLup admin to insert a different one.

Table 2.5: *Register new store* use case description.

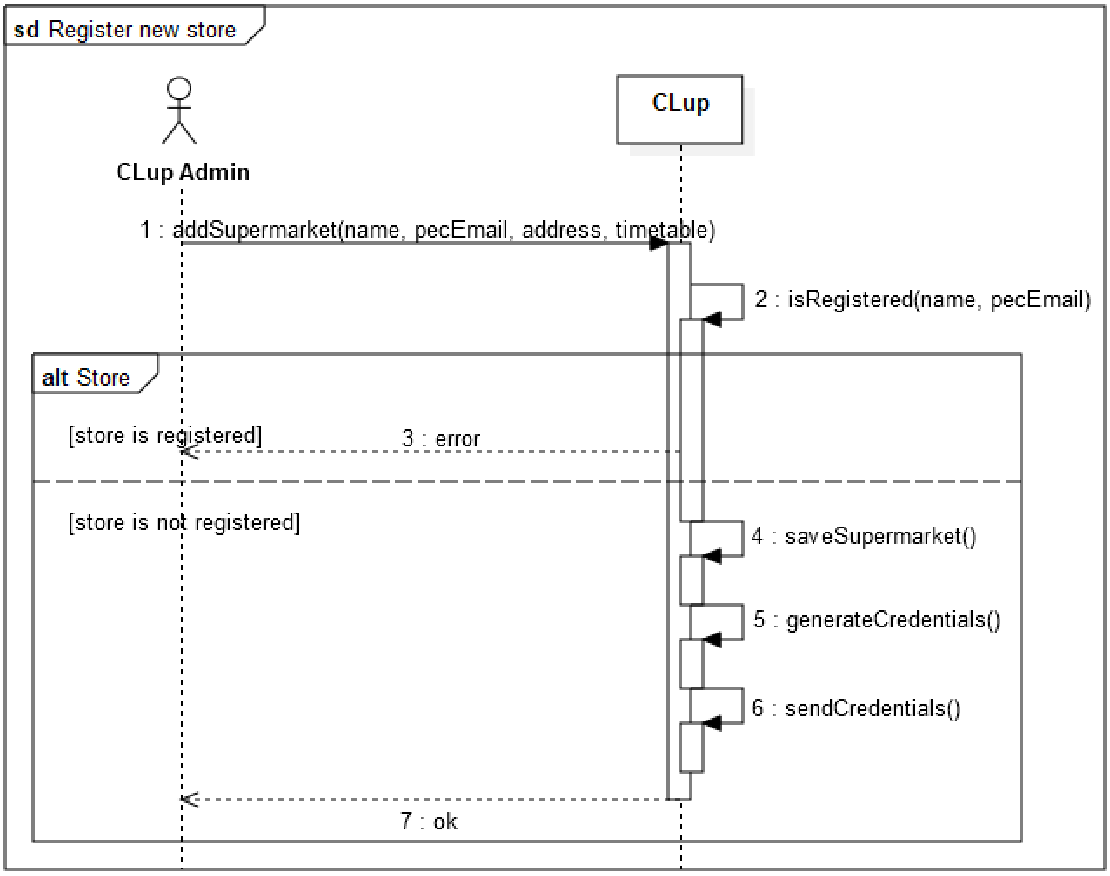

sd Register new store
CLup Admin
CLup
1 : addSupermarket(name, pecEmail, address, timetable)
2 : isRegistered(name, pecEmail)
alt Store
[store is registered]
3 : error
[store is not registered]
4 : saveSupermarket()
5 : generateCredentials()
6 : sendCredentials()
7 : ok

Figure 2.8: *Register new store* sequence diagram.

Name	Monitor bookings
ID	UC.6
Actors	Store manager
Entry conditions	• The platform is running. • The store manager is logged in.
Flow of events	i. The Store manager go to the "Manage bookings list" page; ii. The Store manager can filter bookings by date, time slot and categories; iii. The system provides to the store manager a list with all the bookings that match the filter. If no filter are provided, a full list of all the bookings is shown.
Exit conditions	The Store manager views the bookings.
Exceptions	• If no bookings are available, an error is prompted to the store manager.

Table 2.6: *Monitor bookings* use case description.

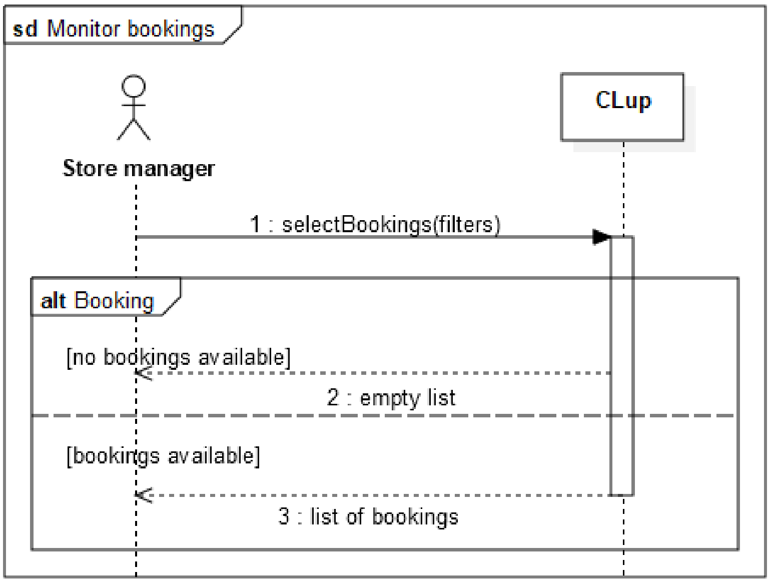


Figure 2.9: *Monitor bookings* sequence diagram.

Name	Validate store pass
ID	UC.7
Actors	Store employee
Entry conditions	• The employee has access to the application on a device; • The store employee is logged in.
Flow of events	i. The store employee clicks on "Scan QR"; ii. The store employee points the camera to the customer's QR code; iii. The system prompts a response about the validity of the pass;
Exit conditions	The store pass is accepted.
Exceptions	• If the store pass is invalid, an error is prompted to the store employee.

Table 2.7: *Validate store pass* use case description.

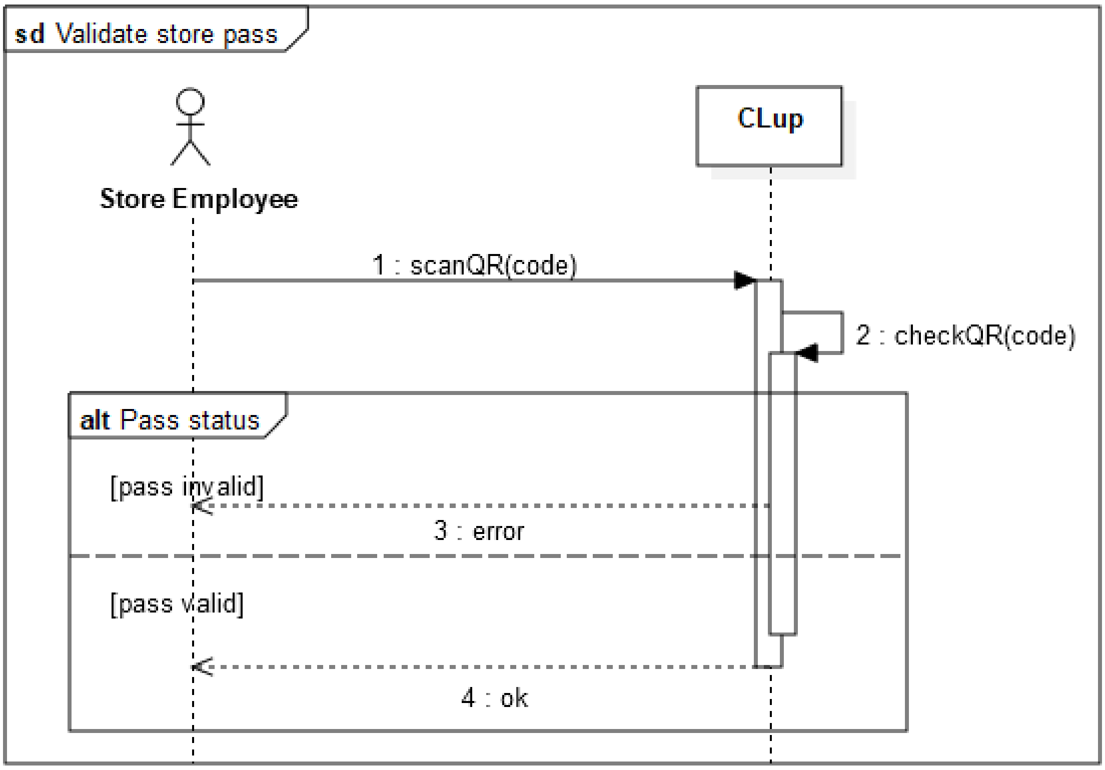


Figure 2.10: *Validate store pass* sequence diagram.

2.3 User characteristics

CLup has four different groups of users:

- **Customers**: they are customers of the supermarket. They can be of all ages and don't necessarily have experience with technology. They can also be elderly people who don't have a smartphone at all. For this reason the system should be as easy to use as possible and should provide an alternative way to retrieve a ticket in addition to the mobile app.
- **Store managers**: they are the managers of the store. They manage the number of people who can access the store and can monitor the ones that are in the supermarket. It is reasonable to assume that they have at least a minimum experience with technology and the use of computer.
- **Store employee**: they are the employees of the store. They validate the tickets at the entrance/exit of the store. They could not have experience with technology. For this reason the platform has a simple interface and the interaction that they should have with it is reduced to the bare minimum.
- **CLup admin**: an operator of CLup able to login to the platform by using his special credentials. It can register supermarkets and generate their credentials. It can also maintain and update the system. Registration for this kind of users is forbidden and it has to be added directly during the system's installation process.

2.4 Constraints

In this topic it is put on paper a general description about considerations, boundaries and items that will limit the system's options.

2.4.1 Regulatory policies

The application requests the user's permission in order to retrieve and use their position at runtime.
The email addresses provided when booking through CLup will not be used for commercial purposes or given to third parties.
The email addresses provided by the stores during the registration process must be PEC addresses identified by a Certification Authority.

2.4.2 Hardware limitations

Here is listed where CLup is available depending on the devices. Please note that not all devices are supported.

- iPhones with iOS version 13.5 or above, phones and tablets running Android 6 (Marshmallow) or above;
- 2G/3G/4G connection or Wi-Fi available;

- GPS service;
- camera;
- modern web browser like Firefox, Chrome or Safari;
- internet connection available.

2.4.3 Interfaces to other applications

The proper functioning of the app is strictly subordinated to an external map service. This is required to compute travel distance and time for a matrix of origins and destinations (i.e. customer and store position).

A failure in the above described service will translate into the inability to use CLup.

2.5 Assumptions and Dependencies

The properties that hold in the analysed world will be listed below.

2.5.1 Domain assumptions

D.1 The majority of the customers has a smartphone with internet available.

D.2 Customers attains to the declared categories of items.

D.3 Stores have an internet contract.

D.4 Stores have a PEC address.

D.5 GPS modules of customers' smartphones are working properly.

D.6 The precision of the GPS modules of customers' smartphones is greater than twenty meters.

D.7 Customers bring with themselves the smartphone that they used to retrieve a ticket or reserve a time slot.

D.8 A ticket or a booked visit is associated with exactly one person.

D.9 Customers who want to book-a-visit have an email address.

D.10 Stores have unique names.

D.11 Each store has an employee at the entrance which checks-in people.

D.12 Store employees have a supplied smartphone with camera.

D.13 The cameras of employees' smartphone are working properly.

D.14 Customers comply with the arrival time assigned by the mobile app.

D.15 Store managers know the maximum capacity of customers in the building.

D.16 Each store has a Self Service Ticketing Kiosk accessible.

Chapter 3

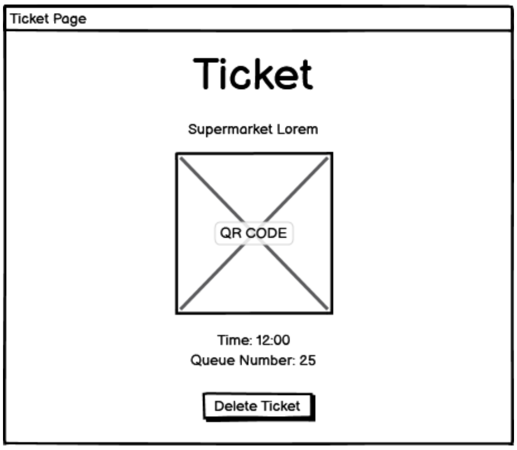
Specific Requirements

This section is devoted to a specific description of every kind of requirement the system has to deal with in order to achieve all the functionalities described.

3.1 External interface Requirements

3.1.1 User Interfaces

Figures below will present the hypothetical early phase user interfaces of the core functions of the system.



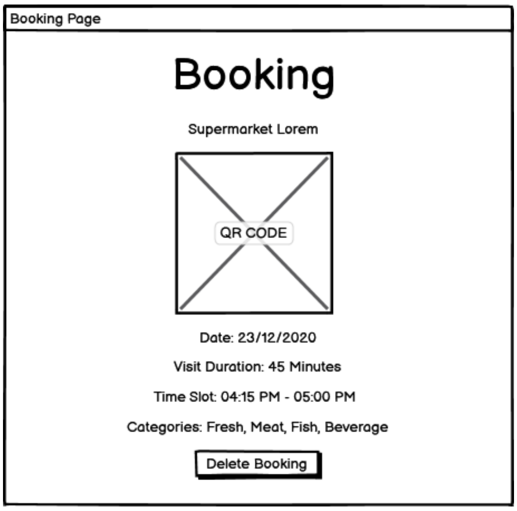
Ticket

Supermarket Lorem

Time: 12:00

Queue Number: 25

Figure 3.1: Ticket page.



Booking

Supermarket Lorem

Date: 23/12/2020

Visit Duration: 45 Minutes

Time Slot: 04:15 PM - 05:00 PM

Categories: Fresh, Meat, Fish, Beverage

Figure 3.2: Booking page.

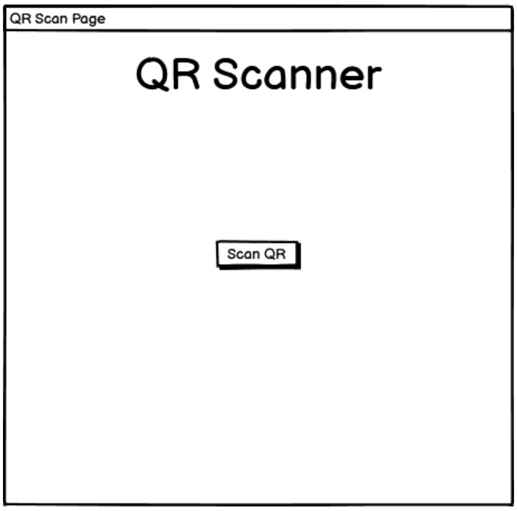


Figure 3.3: QR Scan page.

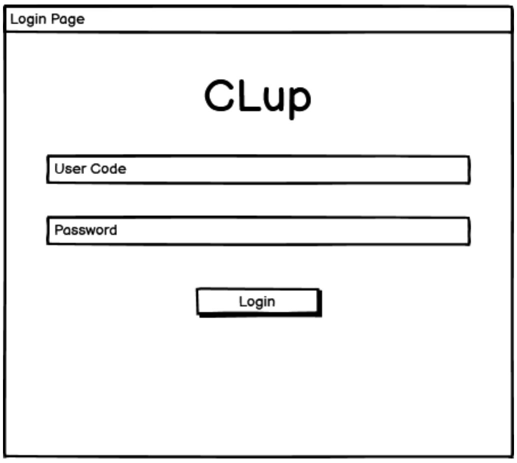


Figure 3.4: Login Page.

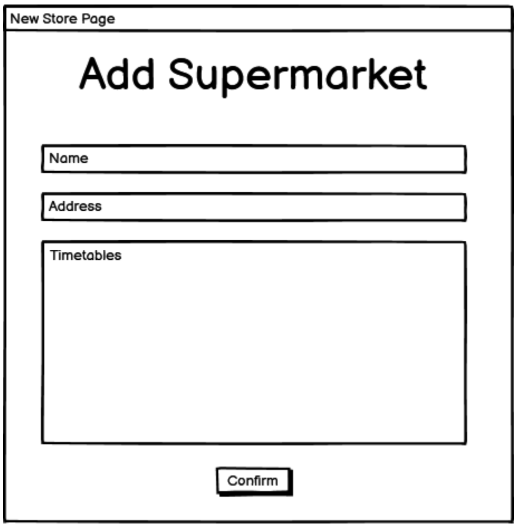
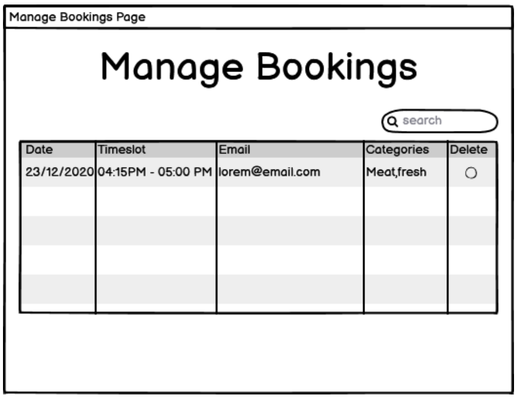


Figure 3.5: New Store page.



Date	Timeslot	Email	Categories	Delete
23/12/2020	04:15PM - 05:00 PM	lorem@email.com	Meat,fresh	○

Figure 3.6: Manage Bookings page.

3.1.2 Hardware Interfaces

In order to operate correctly, the system needs several devices which supermarkets shall acquire:

- smartphones equipped with camera module for employees (one for the store entrance staff and one for each cash desk);
- a device with an attached display to show the store pass calls at the entrance;
- a tablet connected with a printer to setup the local Ticketing Kiosk.

For the normal usage, customers enjoy CLup using smartphones and their GPS module. Store managers, store employees and CLup admins will access CLup services through web browser of their devices.

3.1.3 Software Interfaces

The system interfaces with a external map service for computing the distance between current place and store.

3.1.4 Communication Interfaces

All the communications from and to CLup are made via HTTPS.

3.2 Functional Requirements

In this section, it is given a complete description of the functional requirements of the system.

3.2.1 Requirements

Customer

R.1 The system shall allow customers to line-up remotely in a store queue.

R.2 The system shall generate a new ticket when a customer enters a queue.

R.3 The system shall allow customers which do not have a smartphone to get a ticket in place.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.6 The system shall fetch the GPS position while the user has retrieved a store pass.

R.7 The system shall allow customers to leave a queue.

R.8 The system shall allow customers to filter stores by name.

R.9 The system shall notify customers when it’s time to leave for the store.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

R.11 The system shall allow book-a-visit customers to specify the main categories of item they intend to buy.

R.12 The system shall allow customers to delete a booked visit.

R.13 The system shall notify customers when a ticket or booked visit is deleted.

R.14 The system shall accept bookings based onto the already booked category items.

Store manager

R.15 The system shall allow a registered store manager to login by using their credentials.

R.16 The system shall allow store managers to view the current status of people inside the store.

R.17 The system shall allow store managers to view the current status of people in the queue.

R.18 The system shall allow store managers to view the booked visits to the store.

R.19 The system shall allow store managers to set a maximum cap of people inside the store.

R.20 The system shall allow store managers to delete tickets and booked visits.

Store employee

R.21 The system shall allow a registered store employee to login by using their credentials.

R.22 The system shall allow store employee to view the current status of people inside the store.

R.23 The system shall allow store employee to view the current status of people in the queue.

R.24 The system shall allow store employee to scan QR codes.

R.25 The system shall allow store employee to validate store passes.

CLup admin

R.26 The system shall allow CLup admins to register new supermarkets.

R.27 The system shall generate new manager and staff credentials for each supermarket registered.

3.2.2 Goal mapping on requirements

G.1 Avoid the creation of physical queues outside stores.

G.1.1 Allow customers to avoid the creation of hazardous situations.

R.1 The system shall allow customers to line-up remotely in a store queue.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.9 The system shall notify customers when it's time to leave for the store.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

R.11 The system shall allow book-a-visit customers to specify the main categories of item they intend to buy.

R.14 The system shall accept bookings based onto the already booked category items.

D.1 The majority of the customers has a smartphone with internet available.

D.2 Customers attains to the declared categories of items.

D.9 Customers who want to book-a-visit have an email address.

D.8 A ticket or a booked visit is associated with exactly one person.

D.14 Customers comply with the arrival time assigned by the mobile app.

G.1.2 Allow supermarkets to avoid the creation of hazardous situations.

R.14 The system shall accept bookings based onto the already booked category items.

R.16 The system shall allow store managers to view the current status of people inside the store.

R.17 The system shall allow store managers to view the current status of people in the queue.

R.18 The system shall allow store managers to view the booked visits to the store.

R.19 The system shall allow store managers to set a maximum cap of people inside the store.

R.20 The system shall allow store managers to delete tickets and booked visits.

R.22 The system shall allow store employee to view the current status of people inside the store.

R.23 The system shall allow store employee to view the current status of people in the queue.

D.2 Customers attains to the declared categories of items.

D.8 A ticket or a booked visit is associated with exactly one person.

D.3 Stores have an internet contract.

D.4 Stores have a PEC address.

D.11 Each store has an employee at the entrance which checks-in people.

D.14 Customers comply with the arrival time assigned by the mobile app.

D.15 Store managers know the maximum capacity of customers in the building.

G.1.3 Shorten the amount of time a customer is in queue.

R.1 The system shall allow customers to line-up remotely in a store queue.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.9 The system shall notify customers when it's time to leave for the store.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

R.24 The system shall allow store employee to scan QR codes.

R.25 The system shall allow store employee to validate store passes.

D.1 The majority of the customers has a smartphone with internet available.

D.5 GPS modules of customers' smartphones are working properly.

D.6 The precision of the GPS modules of customers' smartphones is greater than twenty meters.

D.7 Customers bring with themselves the smartphone that they used to retrieve a ticket or reserve a time slot.

D.11 Each store has an employee at the entrance which checks-in people.

D.12 Store employees have a supplied smartphone with camera.

D.13 The cameras of employees' smartphone are working properly.

D.14 Customers comply with the arrival time assigned by the mobile app.

G.1.4 Allow customers to arrive at stores right on time.

R.1 The system shall allow customers to line-up remotely in a store queue.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.6 The system shall fetch the GPS position while the user has retrieved a store pass.

R.9 The system shall notify customers when it's time to leave for the store.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

D.1 The majority of the customers has a smartphone with internet available.

D.5 GPS modules of customers' smartphones are working properly.

D.6 The precision of the GPS modules of customers' smartphones is greater than twenty meters.

D.7 Customers bring with themselves the smartphone that they used to retrieve a ticket or reserve a time slot.

D.14 Customers comply with the arrival time assigned by the mobile app.

G.2 Allow customers to take advantage of the time they are queued up to do other things.

R.1 The system shall allow customers to line-up remotely in a store queue.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.9 The system shall notify customers when it's time to leave for the store.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

D.1 The majority of the customers has a smartphone with internet available.

D.5 GPS modules of customers' smartphones are working properly.

D.6 The precision of the GPS modules of customers' smartphones is greater than twenty meters.

D.9 Customers who want to book-a-visit have an email address.

D.7 Customers bring with themselves the smartphone that they used to retrieve a ticket or reserve a time slot.

D.14 Customers comply with the arrival time assigned by the mobile app.

G.3 Grant customers an overall experience as easy as possible.

R.2 The system shall generate a new ticket when a customer enters a queue.

R.4 The system shall allow customers to view the number of people lined up in a queue.

R.5 The system shall give customers an estimated waiting time.

R.7 The system shall allow customers to leave a queue.

R.8 The system shall allow customers to filter stores by name.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

R.12 The system shall allow customers to delete a booked visit.

R.13 The system shall notify customers when a ticket or booked visit is deleted.

R.24 The system shall allow store employee to scan QR codes.

R.25 The system shall allow store employee to validate store passes.

D.1 The majority of the customers has a smartphone with internet available.

D.5 GPS modules of customers' smartphones are working properly.

D.10 Stores have unique names.

D.12 Store employees have a supplied smartphone with camera.

D.16 Each store has a Self Service Ticketing Kiosk accessible.

G.4 Allow customers, even the ones who don't have access to technology, to enjoy the service.

R.1 The system shall allow customers to line-up remotely in a store queue.

R.3 The system shall allow customers which do not have a smartphone to get a ticket in place.

R.10 The system shall allow customers to book-a-visit to the store and send them the confirmation link and receipt via email.

D.1 The majority of the customers has a smartphone with internet available.

- **D.3** Stores have an internet contract.
- **D.4** Stores have a PEC address.
- **D.16** Each store has a Self Service Ticketing Kiosk accessible.

G.5 Allow supermarkets to monitor access to stores in a better way.

- **R.14** The system shall accept bookings based onto the already booked category items.
- **R.15** The system shall allow a registered store manager to login by using their credentials.
- **R.16** The system shall allow store managers to view the current status of people inside the store.
- **R.17** The system shall allow store managers to view the current status of people in the queue.
- **R.18** The system shall allow store managers to view the booked visits to the store.
- **R.21** The system shall allow a registered store employee to login by using their credentials.
- **R.22** The system shall allow store employee to view the current status of people inside the store.
- **R.23** The system shall allow store employee to view the current status of people in the queue.
- **R.24** The system shall allow store employee to scan QR codes.
- **R.25** The system shall allow store employee to validate store passes.
- **R.26** The system shall allow CLup admins to register new supermarkets.
- **R.27** The system shall generate new manager and staff credentials for each supermarket registered.
- **D.3** Stores have an internet contract.
- **D.4** Stores have a PEC address.
- **D.11** Each store has an employee at the entrance which checks-in people.
- **D.15** Store managers know the maximum capacity of customers in the building.

G.6 Allow supermarkets to know in advance how many people are coming to stores.

- **R.16** The system shall allow store managers to view the current status of people inside the store.
- **R.17** The system shall allow store managers to view the current status of people in the queue.
- **R.18** The system shall allow store managers to view the booked visits to the store.
- **R.22** The system shall allow store employee to view the current status of people inside the store.
- **R.23** The system shall allow store employee to view the current status of people in the queue.
- **D.1** The majority of the customers has a smartphone with internet available.

D.2 Customers attains to the declared categories of items.

D.3 Stores have an internet contract.

D.4 Stores have a PEC address.

G.7 Allow supermarkets to limit the number of access to stores.

R.14 The system shall accept bookings based onto the already booked category items.

R.19 The system shall allow store managers to set a maximum cap of people inside the store.

R.20 The system shall allow store managers to delete tickets and booked visits.

R.24 The system shall allow store employee to scan QR codes.

R.25 The system shall allow store employee to validate store passes.

D.1 The majority of the customers has a smartphone with internet available.

D.2 Customers attains to the declared categories of items.

D.3 Stores have an internet contract.

D.4 Stores have a PEC address.

D.12 Store employees have a supplied smartphone with camera.

D.13 The cameras of employees' smartphone are working properly.

D.14 Customers comply with the arrival time assigned by the mobile app.

D.15 Store managers know the maximum capacity of customers in the building.

3.2.3 Traceability matrix

Item	R.1	R.2	R.3	R.4	R.5	R.6	R.7	R.8	R.9	R.10	R.11	R.12	R.13	R.14
UC.1	✓	✓		✓	✓	✓		✓	✓					
UC.2	✓					✓		✓	✓	✓	✓			✓
UC.3							✓	✓				✓	✓	
UC.4														
UC.5														
UC.6														
UC.7														
G.1.1	✓			✓	✓				✓	✓	✓			✓
G.1.2														✓
G.1.3	✓			✓	✓				✓	✓				
G.1.4	✓			✓	✓	✓			✓	✓				
G.2	✓			✓	✓				✓	✓				
G.3		✓		✓	✓		✓	✓		✓		✓	✓	
G.4	✓		✓							✓				
G.5														✓
G.6														
G.7														✓

Table 3.1: *Traceability matrix* for requirements R.1 to R.14.

Item	R.15	R.16	R.17	R.18	R.19	R.20	R.21	R.22	R.23	R.24	R.25	R.26	R.27
UC.1													
UC.2													
UC.3													
UC.4	✓						✓		✓			✓	
UC.5												✓	✓
UC.6	✓			✓		✓							
UC.7							✓	✓	✓	✓	✓		
G.1.1													
G.1.2		✓	✓	✓	✓	✓		✓	✓				
G.1.3										✓	✓		
G.1.4													
G.2													
G.3										✓	✓		
G.4													
G.5	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
G.6		✓	✓	✓				✓					
G.7					✓	✓				✓	✓		

Table 3.2: *Traceability matrix* for requirements R.15 to R.27.

Item	UC.1	UC.2	UC.3	UC.4	UC.5	UC.6	UC.7
SC.1	✓						✓
SC.2		✓					
SC.3			✓				
SC.4				✓	✓		
SC.5				✓		✓	

Table 3.3: *Traceability matrix* for scenarios and use cases.

3.3 Performance Requirements

This section specifies numerical requirements placed on the software or on human interaction with the software as a whole.
All the computation will take place on the servers of the system, therefore the mobile apps shall be lightweight and occupy little memory on the smartphones. Since the majority of stores usually open only during daytime, the load in the night is expected to decrease considerably.
Regarding the scan process to let people enter or exit the store, at least 95% of the passes scan shall be processed in less than 1 second: this is required in order to fulfil the goals of CLup.

3.4 Design Constraints

3.4.1 Standards Compliance

The system will store all the data submitted to it in a standardized form. In this way, it will be easier to catalog, retrieve, and run queries on the data.

CLup system shall use stateless protocols and standard operations to allow components to be managed and updated without affecting the system as a whole.
It's crucial to design modules properly so that ease of use, security and performance will remain the core factors of the system.

3.4.2 Other Constraints

Regulatory policies have to be considered for the interaction between CLup and customers. The application, in fact, will ask the position of the customer while retrieving a store pass and while checking the pass status. Further information will be provided in the section 3.6.1.

3.5 Software System Attributes

3.5.1 Reliability

In order to guarantee continuity, services are required to be fault tolerant. Errors handling and fault containment mechanisms to prevent error propagation and data loss are to be arranged.

3.5.2 Availability

It is essential to have the lowest downtime possible, specially for the plain queue function (see section 2.2). The ticketing components shall guarantee 99.9% (*three-nines*) of availability, so that only 8.76 hours of downtime per year are allowed. Meanwhile the book-a-visit components shall guarantee 99% (*two-nines*) of availability, so that 3.65 days of downtime per year are allowed.

3.5.3 Security

The system shall perform role based access control (RBAC): an authorization scheme that grants access rights based on the role of the use. In particular, such components shall grant user authentication and authorization:

- **Authentication**: request and verify the identity of CLup admin, store managers and employees attempting to login using a usercode and password.
- **Authorization**: verify the permissions of the logged user to perform any requested action (e.g. adding or removing a store, creating a new item category, etc.) before performing it.

3.5.4 Maintainability

The system shall be characterized by scalable and reusable modules which will be easier to maintain and replace in case of failure. Ordinary maintenance, for bug fixes and improvements, will be scheduled during night time, when the user traffic is minimal.
The core aspects of maintainability and modularity will be addressed in the design document.

3.5.5 Portability

The web platform for store manager, staff and admin will be accessible by any web browser.
The mobile application for customers must be accessible by as many users as possible, hence it must be developed for the major mobile OSes. Since the mobile apps do not demand special functions, a non native approach can be adopted to fasten the development process. The server side has no major requirements for portability.

3.5.6 Usability

The mobile app and the web platform of the system will be designed to be concise and user-friendly, with a graphical interface to help users identify the proper choice on the screen. It is expected that the 95% of the users will be able to complete tasks without requiring assistance.

3.6 Other Requirements

3.6.1 Privacy Requirements

The system shall ensure that the collection and transmission of personal data is handled in accordance with user's expectation and regulations.
In fact, to protect customers privacy, only the strictly necessary user data are requested to enjoy the app. Ensuring users privacy is protected positively influences user's experience, acceptance and continuous use of the platform. For instance the system shall block unauthorized access to implicit information (e.g. location) and encrypt data transmission.

3.6.2 Installation requirements

The complete solution shall be installed in the specified environment (i.e. a store) within 4 working days.
Smartphones to be supplied to store employees shall be configured and CLup shall be installed onto them.
No specific installation requirements are necessary from the customer point of view. Indeed a customer may install the app right before starting to use it.

Chapter 4

Formal Analysis Using Alloy

In this section will be provided a formal model of the problem achieved using Alloy.
The model represent only the most important part of the problem, few things have been simplified leaving the more relevant constraints.
For the sake of readability the generated world is split into three sub-worlds which will be explained below.

4.1 Alloy Model

```
---------------------------------
-- Signatures
---------------------------------
sig Date {}
sig Time {}
sig DateTime {
  date: Date,
  time: Time
}

sig StoreName {}
sig Address {}
sig PecEmail {}
sig Email {}
sig CategoryName {}
sig Password {}
sig PhoneNumber {}
sig UserCode {}
sig PassCode {}

sig Store {
  name: StoreName,
  address: Address,
  timetable: some OpeningHours,
  pecEmail: PecEmail,
  phone: PhoneNumber,
  storeCap: Int,
  itemCategories: some ItemCategory,
  manager: Manager,
  employee: Employee,
  queue: Queue,
  timeSlots: set TimeSlot,
  customersInside: Int
}
{
  storeCap > 0
  customersInside >= 0
}

sig OpeningHours {
  from: Time,
  to: Time
} {from != to}

sig ItemCategory {
  name: CategoryName
}

sig Queue {
  tickets: set Ticket
}

abstract sig StorePass {
  date: Date,
  arrivalTime: Time,
  issuedAt: DateTime,
  passCode: PassCode,
  passStatus: PassStatus
}

sig Booking extends StorePass {
  email: Email,
  departureTime: Time,
  bookingStatus: BookingStatus,
  itemCategories: some ItemCategory
}

sig Ticket extends StorePass {
  queueNumber: Int
} {
  queueNumber > 0
}

abstract sig PassStatus {}
one sig Valid extends PassStatus {}
one sig Active extends PassStatus {}
one sig Expired extends PassStatus {}

abstract sig BookingStatus {}
one sig Pending extends BookingStatus {}
one sig Confirmed extends BookingStatus {}

sig TimeSlot {
  date: Date,
  timeStart: Time,
  timeEnd: Time,
  bookings: set Booking
} {timeStart != timeEnd}

abstract sig User {
  userCode: UserCode,
  password: Password
}

sig Manager extends User {}
sig Employee extends User {}
sig Admin extends User {}

---------------------------------
-- Facts
---------------------------------
-- Store
fact storeNameAreUnique {
  no disj s1, s2: Store | s1.name = s2.name
}

fact noStoreNameWithoutStore {
  all sn: StoreName | one s: Store |  s.name = sn
}

fact storeEmailAreUnique {
  no disj s1, s2: Store | s1.pecEmail = s2.pecEmail
}

fact noStoreEmailWithoutStore {
  all pe: PecEmail | one s: Store |  s.pecEmail = pe
}

fact phoneNumberAreUnique {
  no disj s1, s2: Store | s1.phone = s2.phone
}

fact noPhoneWithoutStore {
  all p: PhoneNumber | one s: Store |  s.phone = p
}

fact addressAreUnique {
  no disj s1, s2: Store | s1.address = s2.address
}

fact noAddressWithoutStore {
  all a: Address | one s: Store |  s.address = a
}

fact maxCustomerInside {
  all s: Store | s.customersInside <= s.storeCap
}

```

```
-- Manager
fact oneManagerBelongToOneStore {
  no disj s1, s2: Store | s1.manager = s2.manager
}

fact  noManagerWithoutStore {
  all m: Manager | one s: Store |  s.manager = m
}

-- Employee
fact oneEmployeeBelongToOneStore {
  no disj s1, s2: Store | s1.employee = s2.employee
}

fact noEmployeeWithoutStore {
  all e: Employee | one s: Store |  s.employee = e
}

-- Queue
fact oneQueueBelongToOneStore {
  no disj s1, s2: Store | s1.queue = s2.queue
}

fact noQueueWithoutStore {
  all q: Queue | one s: Store |  q in s.queue
}

-- TimeSlot
fact oneTimeSlotsBelongToOneStore {
  all t: TimeSlot | no disj s1, s2: Store | t in s1.timeSlots and t in s2.timeSlots
}

fact noTimeSlotWithoutStore {
  all ts: TimeSlot | one s: Store |  ts in s.timeSlots
}

-- ItemCategory
fact oneItemCategoryBelongToOneStore {
  all i: ItemCategory | no disj s1, s2: Store | i in s1.itemCategories and i in s2.itemCategories
}

fact noItemCategoryWithoutStore {
  all i: ItemCategory | one s: Store |  i in s.itemCategories
}

-- Category Name
fact itemCatergoryNameAreUnique {
  no disj i1, i2: ItemCategory | i1.name = i2.name
}

fact noCategoryNameWithoutItemCategory {
  all cn: CategoryName | one ic: ItemCategory |  ic.name = cn
}

-- OpeningHours
fact oneOpeningHoursBelongToOneStore {
  all o: OpeningHours | no disj s1, s2: Store | o in s1.timetable and o in s2.timetable
}

fact noOpeningHoursWithoutStore {
  all o: OpeningHours | one s: Store |  o in s.timetable
}

-- UserCode
fact userCodeAreUnique {
  no disj u1, u2: User | u1.userCode = u2.userCode
}

fact noUserCodeWithoutUser {
  all uc: UserCode | one u: User |  u.userCode = uc
}

-- Password
fact noPasswordWithoutUser {
  all p: Password | one u: User |  u.password = p
}

-- PassCode
fact passCodeAreUnique {
  no disj sp1, sp2: StorePass | sp1.passCode = sp2.passCode
}

fact noPassCodeWithoutStorePass {
  all pc: PassCode | one sp: StorePass |  sp.passCode = pc
}

-- Ticket
fact oneTicketBelongToOneQueue {
  all t: Ticket | no disj q1, q2: Queue | t in q1.tickets and t in q2.tickets
}

fact onlyValidTicketInQueue {
  all t: Ticket | one q: Queue |  t in q.tickets iff t.passStatus = Valid
}

fact noSameTicketQueueNumberInSameQueue {
  all q: Queue | no disj t1, t2: Ticket | t1 in q.tickets and t2 in q.tickets and t1.queueNumber = t2.queueNumber
}

-- Booking
```

```
fact oneBookingBelongToSameStoreSlots {
  all b: Booking | no disj s1, s2: Store | b in s1.timeSlots.bookings and b in s2.timeSlots.bookings
}

fact noBookingWithoutTimeslot {
  all b: Booking | some ts: TimeSlot |  b in ts.bookings
}

fact noTwoBookingOfSamePersonOnSameDay {
  no disj b1, b2: Booking | b1.date = b2.date and b1.email = b2.email
}

-- Booking Mail
fact noBookingEmailWithoutStore {
  all e: Email | one b: Booking | b.email  = e
}

-- Date
fact noDateWithoutStorePassOrTimeSlot {
  (all d: Date | one sp: StorePass | sp.date = d) or
  (all d: Date | one ts: TimeSlot | ts.date = d) or
  (all d: Date | one dt: DateTime | dt.date = d)
}

-- Time
fact noTimeWithoutStorePass {
  (all t: Time | one sp: StorePass | sp.arrivalTime = t) or
  (all t: Time | one b: Booking | b.departureTime = t) or
  (all t: Time | one o: OpeningHours | o.from = t or o.to = t) or
  (all t: Time | one ts: TimeSlot | ts.timeStart = t or ts.timeEnd = t) or
  (all t: Time | one dt: DateTime | dt.time = t)
}

-- Date Time
fact noDateTimeWithoutStorePass {
  all dt: DateTime | one sp: StorePass | sp.issuedAt = dt
}

---------------------------------
-- Predicates
---------------------------------
pred world1 {
  #Store = 1
  #ItemCategory = 1
  #OpeningHours = 1
  #Admin = 0
  #TimeSlot = 0
  #Ticket = 2

  one t: Ticket |  (t.passStatus = Active or t.passStatus = Expired
    )
}
run world1 for 2

pred world2 [q: Queue] {
  #Store = 1
  #q.tickets = 0
  #Booking = 2
  #TimeSlot = 2

  one t: TimeSlot | #t.bookings = 0
}
run world2 for 2

pred world3 {
  #Store = 2
  #Queue.tickets = 0
  #TimeSlot = 0
  #ItemCategory = 2
}
run world3 for 5
```

4.1.1 First World

In the first world (Figure 4.1) the focus is on the **store** and the **ticket feature**. The store entity contains all its information, manager and employee user. The store has also a ticket queue which contains only tickets which status is *VALID*. In case of a ticket *ACTIVE* or *EXPIRED*, the ticket is removed from the queue.

4.1.2 Second World

In the second world (Figure 4.2) the focus is on the **booking feature**. Store has timeslots which were hidden in the previous world. They represent a slice of time that can be booked. Every booking must be linked to at least one timeslot. This is because a customer may require to stay in the store longer than just a single timeslot.

4.1.3 Third World

In the third world (Figure 4.3) the focus is on a **multiple store** implementation. Each store has its information, manager and employee users. It is notable that the CLup admin is *super partes* and has no relation with the number of store.

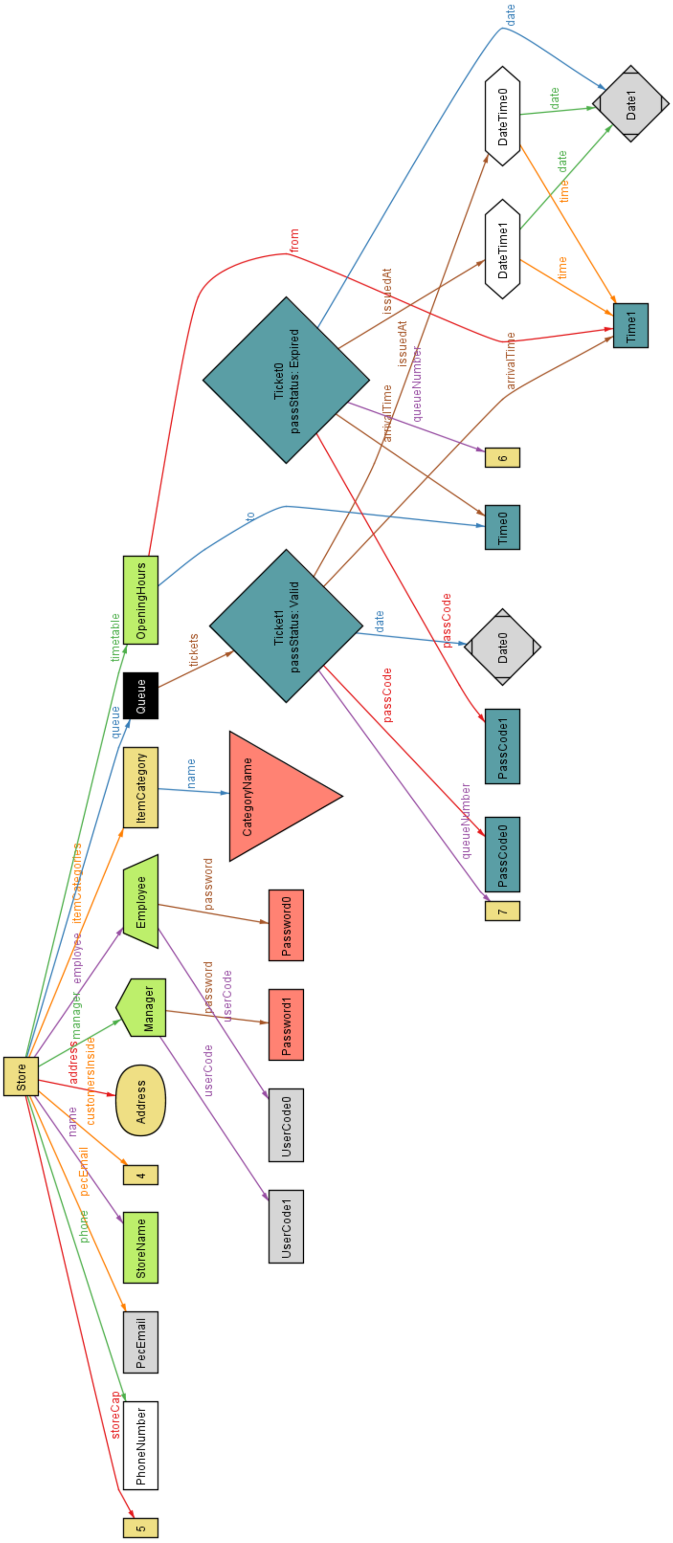


Figure 4.1: First World.

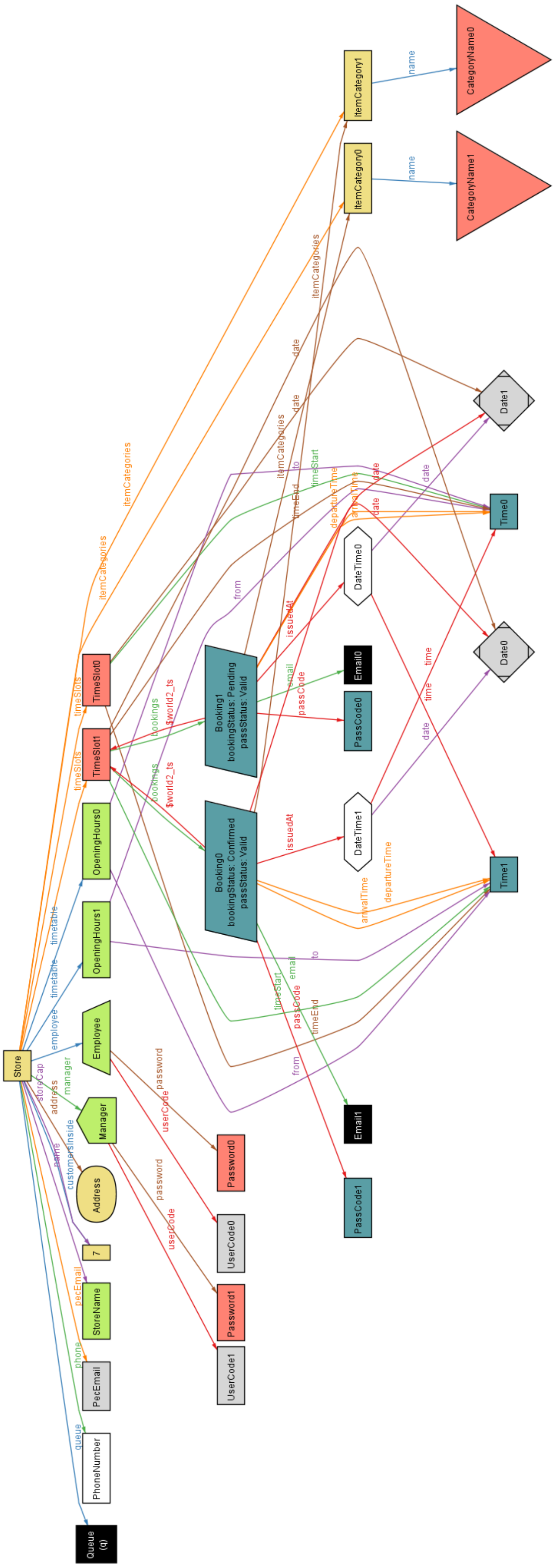


Figure 4.2: Second World.

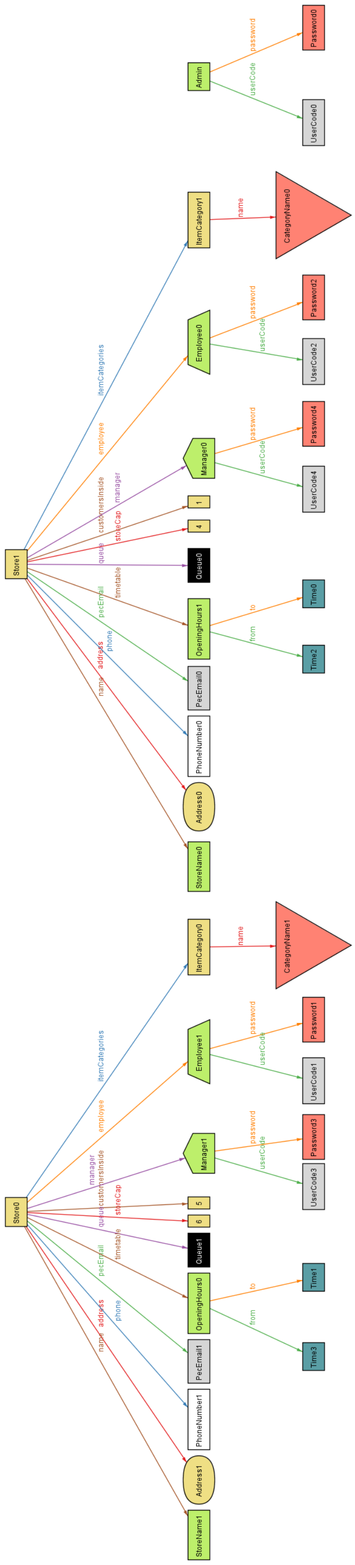


Figure 4.3: Third World.

Chapter 5

Effort Spent

5.1 Teamwork

Task	Hours
Initial briefing	1
Introduction	4
Domain Assumptions	3
Functional Requirements	5
Document Revision	9

5.2 Samuele Negrini

Task	Hours
Product Functions	5
Constraints	1.5
Functional Requirements	8
Use Cases	6
Traceability Matrix	4
System Software Attributes	4
Other Requirements	3

5.3 Giorgio Piazza

Task	Hours
Product Perspective	5
User Characteristics	2
External Interface Requirements	8
Scenarios	1
Use Cases	1
Formal Analysis Using Alloy	12
Document Revision	1