Software Engineering 2 - Prof. Di Nitto Elisabetta
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

CLup - Customer Line-up

RASD
Requirement Analysis and Specification Document

Andrea Franchini(10560276)
Ian Di Dio Lavore (10580652)
Luigi Fusco(10601210)

xx-xx-2020

Todo list

Contents

1 Introduction

1.1 Purpose

This document is the Requirement Analysis and Specification Document for the Customers Line-Up system. The purpose of this document is to describe the system focusing on scenarios, use cases, requirements and specifications, analyzing what the software will do, how it will be used and the constraints under which it will operate. This document is intended both for users and developers.

1.2 Scope

In recent times, it has become clear that avoiding gathering of groups of people in small spaces is critical for the safety of the population. During lockdowns some places become more crowded than others. This reason lead to the development of various solutions, all resulting roughly in restricting the capacity of a building, without having a real knowledge of the number of people inside (as ofter required by the law).

A definitive and well-articulated system is still missing, especially considering that most people aren't much tech-savvy, and especially elderly aren't familiar with the Internet. According to recent statistics[1], in Italy over 30% of the population has never used the Internet, in particular only the 10% of 75+ years old regularly use it; on the other hand many of those able to surf the web do it through a smartphone and lack digital skills.

The proposed system will focus on solving the problem posed by queues in front of groceries stores but it could be generalized for queues handling in multiple scenarios.

Customers Line-Up (CLup) is a system that allows supermarket managers to regulate the influx of people inside physical stores and giving customers a way to drastically reduce (or remove completely) the time spent waiting in a queue. The idea of CLup is being more akin to an open-source framework that can be adoped and improved, rather than it being a closed-source product.

This tool reaches the goal by offering a number of functionalities, including:

- access to the service via mobile app or website
- physical alternatives for people that do not have Internet access
- monitor and dynamically restrict the amount of people allowed in a store
- book a visit, notifying customers of any change in the schedule
- restrict the store selection by using filters
- suggest alternative stores and/or time frames
- track the time spent in the store by customers to provide better estimate of waiting times

1.2.1 World Phenomena

[WP1] new law on the limit on the maximum number of people in a store

[WP2] User reaches to the store

[WP3] User enters the store

[WP4] User exits the store

[WP5] User wants to shop

[WP6] User can't reach the store in time

[WP7] new store is built

[WP8] store is closed

[WP9] maximum number of people in a store is reached

[1] `https://www.istat.it/it/archivio/216672`

1.2.2 Shared Phenomena

[SP1] User makes a reservation

[SP2] User joins Queue

[SP3] User cancels reservation

[SP4] User leaves the Queue

[SP5] System cancels reservation

[SP6] System kicks User out of Queue

[SP7] User scans the QR code

[SP8] The system is notified that an User has finished shopping

[SP9] The system let an allowed User enter the store

[SP10] Manager adds a store

[SP11] Manager removes a store

[SP12] Manager changes the maximum number of people allowed in a store

[SP13] Manager changes time and size of reservation slots

1.2.3 Current System

While there are already existing similar services, some are usually independent from store chains and therefore have limited functionalities. Notable examples are:

- *Ufirst* allows customers to virtually get a number and get notified when it's their turn, offers an app for store managers and a tablet app for totems where a customer without a smartphone can get a ticket. Stores are required to register to make use of the service.
- *FilaIndiana* allows customer to know how many people are in queue in front of a store by using GPS data from its userbase. Doesn't allow to book because it's indipendent from store chains. It's similar to what Google Maps offers with location-based crowd prediction.
- *QuandoSpesa* works similarly to *FilaIndiana*, but allows users to select a certain time slot that is guaranteed to be the less crowded for all the app users.

CLup is a service that supermarket chains can implement alongside their existing services. The system is as independent as possible from existing infrastructures, and it can be used with minimal setup.

1.2.4 Goals

[G1] allow users to avoid crowds inside and outside the store when shopping

  [G1.1] allow users to join a virtual queue

  [G1.2] allow users to book a visit at a specific time

[G2] allow users with special needs to have priority over regular users when shopping

[G3] allow users to save time when shopping by avoiding long waits in line

[G4] allow managers to have control over the the maximum number of people in the store by managing access methods

[G5] allow managers to monitor the current and past status of the store (number of people in the store, fullness of the queues and slots)

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- *User* (also *Customer* or *Visitor*): A person that uses the system to shop at a store.
- *Registered User*: A User that has registered an Account within the System.
- *System Manager*: A stakeholder (owner, employee, manager etc.) of the Store chain that can tweak the parameters of the System and access informations and statistics.
- *Account*: A reference to a specific User in the System, that allows to track the User across multiple visits.
- *Reservation* (or *Booking*): Arrangement made between a User and the System in which the System shall grant the User access to Store at the arranged time.
- *Visit*: The time frame in which the User enters the store, shops and exits.
- *Time slot*: The time at which a Customer with a Reservation is expected arrive at the store.
- *Store*: Any physical location (e.g.: building) where it is possible to utilize the System.
- *Totem*: A physical device with a touchscreen display and an attached printer that allows Customers to join the Virtual Queue.
- *Virtual Queue*: the virtual equivalent of a physical queue in front of the store, regulating the access of people by ordering them.
- *Web App*: A web application, consisting of a back-end and a front-end accessible from a web browser.
- *Line*: Synonim for *queue*.

1.3.2 Acronyms

- RASD: Requirement Analysis and Specification Document
- API: Application Programming Interface
- CLup: Customer Line-up
- REST: REpresentational State Transfers

1.3.3 Abbreviations

- [Gn]: n-goal.
- [Dn]: n-domain assumption.
- [Rn]: n-functional requirement.
- [WPn]: n-world phenomenon.
- [SPn]: n-shared phenomenon.

1.4 Revision History

1.5 Reference History

- Problem Specification Document: “Assignment AY 2020-21.pdf”
- `https://standards.ieee.org/standard/29148-2018.html`

1.6 Document Structure

The document is structured in three sections:

- In the *Introduction* we provide a brief explanation of the problem, what the proposed solution consist of and how it differs from existing systems, reference for used resources, language and revisions.
- In the second section, we provide an overall description of the product and its functions, with the help of scenarios to illustrate various situations. We also analyze the possible userbase of the service and then base assumptions and constraint upon them.
- In the third section we explain the more technical details by analyzing the software, hardware and user interfaces with the help of UML diagrams (sequence, activity) and mockups. We explore also aspects related to reliability, availability, security, maintainability and portability.
- In the fourth section we analyze formally the proposed models with the help of the Alloy Tool.

2 Overall Description

2.1 Product Perspective

Customers Line-Up is developed for both shop managers and customers. The intent is to provide functionalities adding value to the interactions between the two. Managers have access to a website that will help them to avoid large crowds inside and outside their stores, providing them with useful analytics. Customers may avoid queues by booking visits to stores via website or mobile application, and will be guided in selectiong the best place and time. Customers must register an account in order to utilize the website or the mobile app. Customers who do not possess an Internet-connected device may still utilize the service via physical totems outside the stores.

The system will be developed from scratch, giving great flexibility and scalability. The privacy of the customers will be guaranteed according to the latest privacy related norms.

2.1.1 Entities diagram

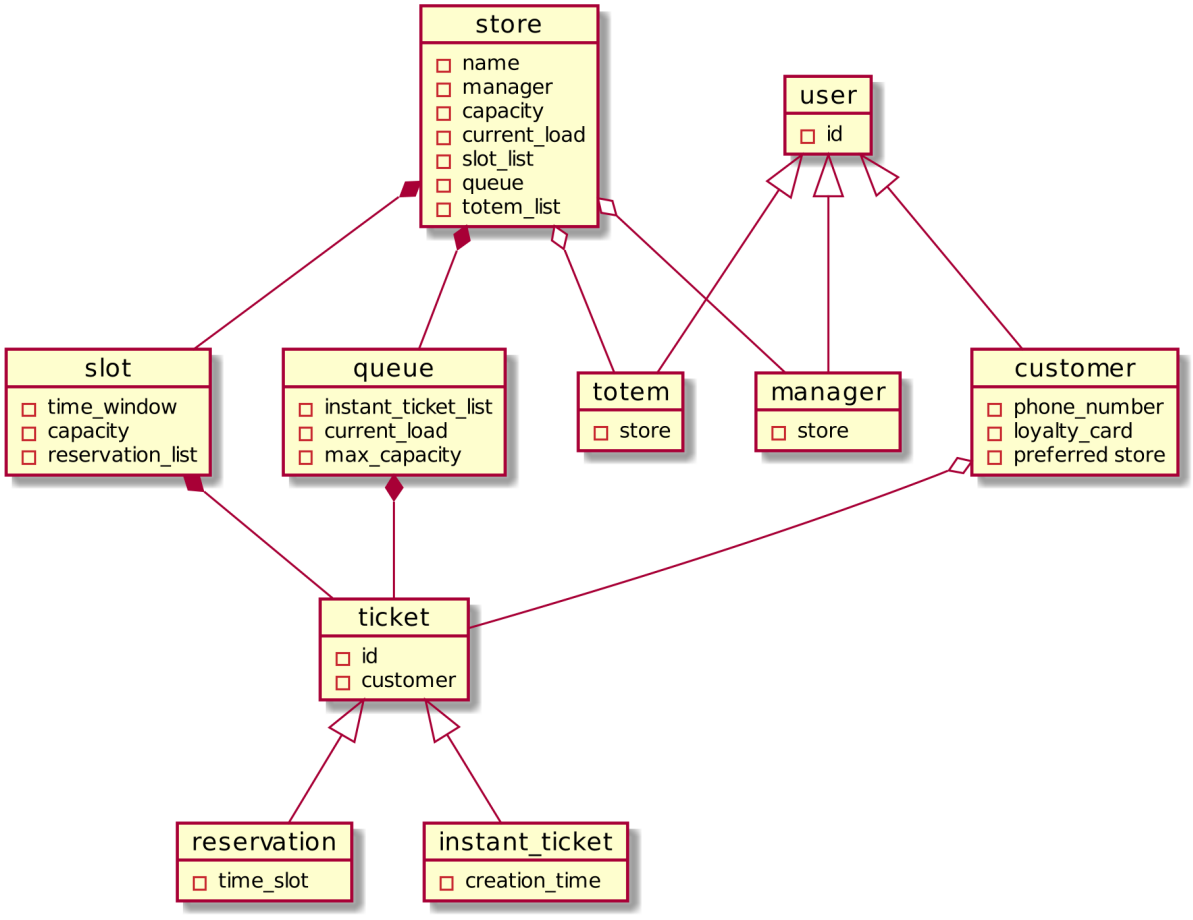


Figure 1: High level view of the main entities of the system

2.1.2 State Diagrams

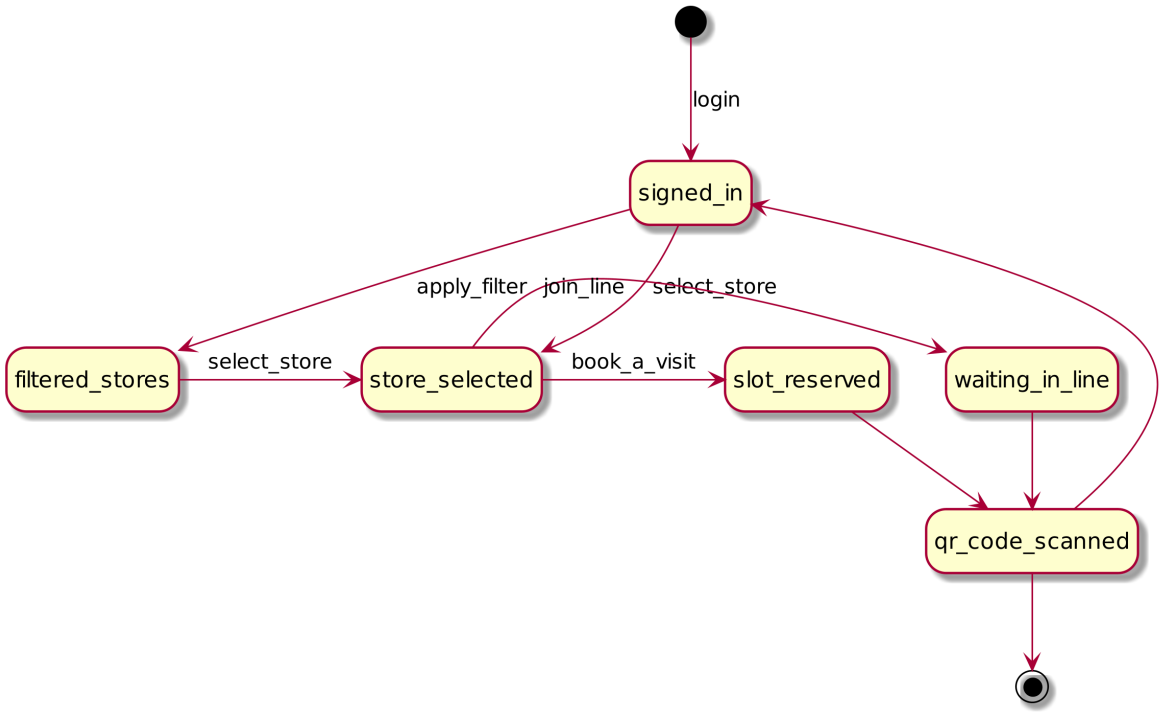


Figure 2: Client state diagram

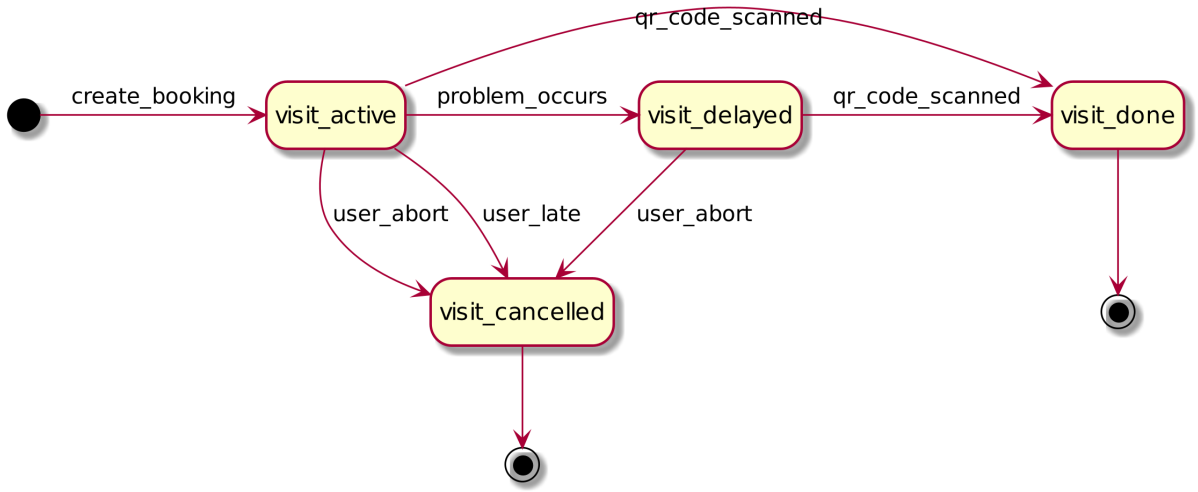


Figure 3: Book a visit state diagram

2.1.3 Scenarios

A. **Customer with the mobile app arrives in time**
Ian wants to buy groceries to make a cake. Ian uses CLup to get a ticket for the supermarket with the shortest queue in his area. The app provides Ian with a QR code and an estimate of the time when he'll be granted access to the store. Ian arrives at the supermarket in the correct time slot, scans the code generated by the app and he is granted access to the store. Once he pays for his groceries the system register the exit of a user from the store.

B. **Customer with no knowledge of the booking system**
Pino is an elderly man. Pino knows nothing about Smartphones or Computers. Pino needs to buy a cake for his nephew's birthday party, so he decides to go to the local supermarket. When he arrives, he notices that the doors of the supermarket aren't opening. He reads the sign pointing him to a totem. As soon as he approaches the machine, the machine activates itself an starts speaking with a reassuring voice. The machine allows Pino to get a ticket for the queue to enter the store and instructs Pino on how to do so. As soon as the time is up, Pino places his ticket onto the reader beside the door of the store, and he is granted access.

C. **Customer cancels the reservation**
Luigi, after booking a visit to the store, remembers that he had a visit to the dentist at the same time. Since Luigi cares about others, he cancels his reservation, freeing up a time slot to be used by other customers.

D. **Customer is unable to provide their code**
Andrea books visit and reaches the store in time, but has forgot to charge his phone, which turns off as he pulls it out of his pocket in order to scan his code. Andrea goes at the totem, gets a new queue ticket with a QR code and an estimate of the remaining time in queue, then he waits at an adequate distance for his turn.

E. **Manager adjusts the number of users allowed without reservation**
Ada is a store manager that notices that most of the customers at her store are elderly and do not make use of the app. She also notices that very few customers actually use the app, and the system is reserving too many spots for customers using the app. She navigates to the web panel of the service, logs in with her credentials, and navigates to the correct section, then she increase the number of customers allowed without reservation. The system automatically decreases the number of customers allowed by booking with the app.

F. **Store employee creates fast-line ticket for pregnant lady**
The store manager has enabled a dedicated queue for pregnant ladies and people with disabilities Mark is the store employee assigned to control the flow at the entrance of the store. Laura is a pregnant lady that hasn't got a reservation, but she deserves a ticket for the dedicated queue (or even instant access if possible). Laura goes to Mark and asks him to provide her with a fast-line ticket. Mark uses his credential on the totem to create a special ticket and gives it to Laura. Laura enters the store with the provided ticket and is able to get the things that she needed.

2.2 Product Functions

The functions of Customer Line-Up can be clearly divided in two categories, based on the type of the stakeholder that is being addressed.

2.2.1 Manager functions

The manager is the owner of the store or store chain that is using the system. The functions targeted at the manager regard the management of the queue and the knowledge of statistics about the behavior of the clients. The system will let managers select the type of commercial exercise (whether single store or chain), manage independently every physical store, select the number of slots dedicated to reservation and the ones dedicated to a regular queue, as well as to create a high-priority queue for special categories of customers. At the same time the system will provide info about the number of people who are currently in a store, how the number of people changes over time and the average visit length.

2.2.2 Customer functions

The customer is the person who visits a store. The functions targeted at the customer regard the possibility of skipping queues. The system will let people book visits at specific time slots or queue up at the moment. If the user is in the queue, they will be updated live with their estimated time of entrance in the store. If the user has booked a visit, they will be notified immediately if the system realizes that their visit has become unfeasible and automatically assign a new time or, in the worst case, cancel completely the visit. The system will offer the possibility of creating an account and of logging in. The system will allow the user to setup ad-hoc notification when specified conditions are met (for example a combination of store, days of the week

and a timeframe). The system will enable the user to book a visit applying filters to stores based on the following criteria: distance from the current position of the user, available days of the week and time frame of the available slots.

2.3 User Characteristics

Customers Line-Up is mainly aimed at essential and widely used services. Because of this its audience will be wide and diversified, and the system will be easy to use and accessible in several of ways, accounting in particular for people with disabilities or people who are not familiar with technology. On one side of the system there is the system manager (single or multiple), who will monitor how the system is used and obtain useful information. They are usually already familiar with othe customer relationship managers and already know what to expext from a control panel. On the other side there is the customer, who uses the system in order to avoid boring lines and to prevent contact with others. The main categories of customer are:

- **Tech-friendly**
  People who are familiar with modern technologies. They find it easy to navigate the menus of a complex application. They are able to use the system in an autonomous way and are the ones who will benefit the most from the more complex and advanced features.
- **Tech-unfriendly**
  People who are not familiar with modern technologies. They have problems navigating complex application, and are more accustomed to talking to humans. They might need aid using the system or misuse the system. They benefit from a system designed around clarity and simplicity, or from different, easier ways of using the system. This category includes people with disabilities.

The objective of Customers Line-Up is to be as inclusive as possible, providing utilities targeted at all possible users.

2.4 Assumptions, Dependencies, and Constraints

2.4.1 Domain Assumptions

[D1] The number of people in a store cannot go over a certain fixed amount.

[D2] If an user enters the store they will exit the store before it closes.

[D3] The system reliably counts (e.g. with a turnstyle, through an employee) the number of people entering and exiting the store.

[D4] There's no mismatch between the real state of the store and its representation by the system.

[D5] A customer cannot enter the store without using the system.

[D6] External software dependencies the system relies on always provide true data and never fail.

2.4.2 Dependencies

- The system requires access to a third party maps API.
- The system needs a third party API in order to send SMS to customers' phones.
- The mobile app shall be deployed on major mobile phones operating systems (Android, iOS).
- The website requires a modern browser (no legacy browser support).
- The application deployed on the totem has to be compatible with its operative system.

2.4.3 Constraints

- The system shall be compliant to local laws and regulations, in particular users data should be treated according to the GDPR. This means that users should be always able to request their data.
- The system should in first place limit itself to collect only the necessary ones to function, such as the user telephone number.
- To better protect the users' sensitive information, such as their telephone number, their data should be encrypted.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The application is designed to allow the customer to make a reservation for a certain hour or to register in the current queue of a supermarket. The first time the user opens the app, they are prompted to register for an account: they have to provide their mobile phone number (Fig. 4a), then they'll receive an SMS with a confirmation code (Fig. 4b) to insert. If the code inserted is the one sent to the phone number inserted in the first screen, the account creation is successful, otherwise the user can ask for a new SMS or go back and enter another phone number.

After a successful registration, the user sees a map (Fig 4c) with stores nearby their current location (they might have to allow the app using their location through the OS API), otherwise they can manually search stores by typing an address. The user can narrow their search through filters (Fig. 4d). Users can also see shortcuts to stores they have previously marked as "favorites".

Upon selecting a store (Fig. 4e), the current number of people in queue is shown to the user, as well as the estimated waiting time in queue. They can choose to enter the queue as soon as possible, or to pick an available time slot. The system also suggests possible alternatives.

If they choose to pick a time slot, they are shown a list of time slots sorted by day (Fig. 4f). Upon confirmation, the user receive a receipt (Fig. 4g) with the date and time of their reservation, and a QR code that they'll need to show when they present themselves at the store.

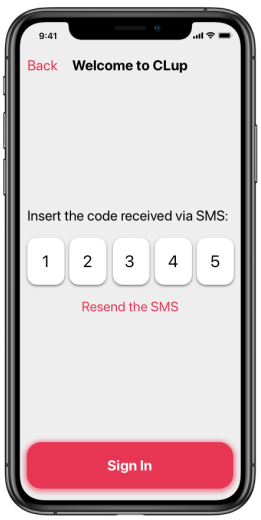
Should the customer queue for the first available time slot, then they will receive a similar receipt (Fig. 4h), that has a countdown of the time remaining in queue before their visit.

In either case, users can cancel their reservations/queue from the receipt screen.

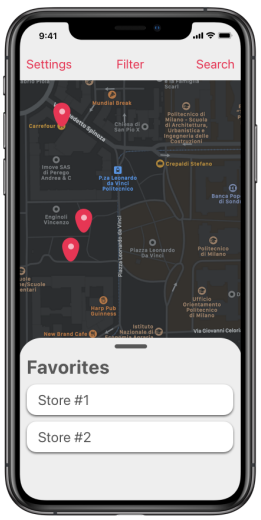
Figure 4: Mock-up of the Application User Interface

(a) Welcome Screen

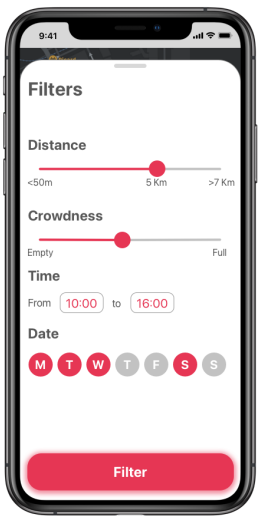
(b) Confirm Login



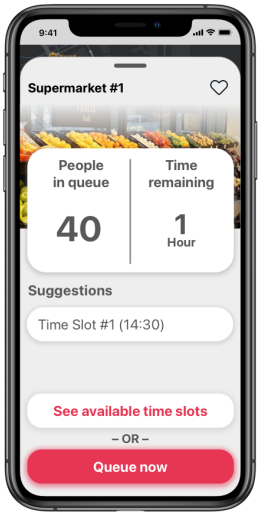
(c) Select Store



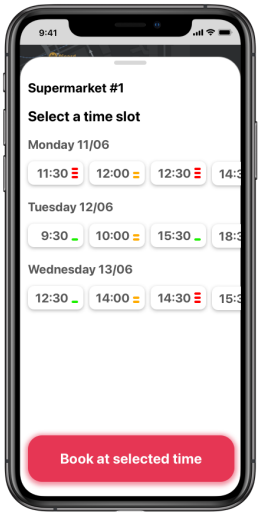
(d) Filters



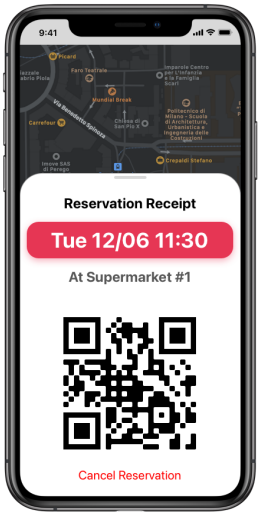
(e) Store Details



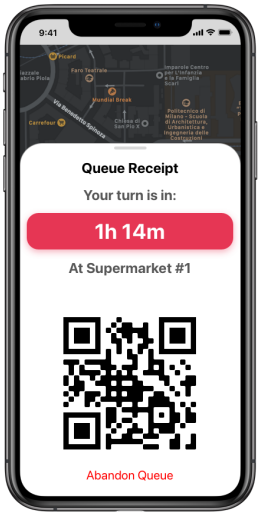
(f) Select Timeslot



(g) Reservation Receipt



(h) Queue Ticket



3.1.2 Hardware Interfaces

An interactive totem (Fig. 5) with a touchscreen display should be made available at each store entrance, to account for Customers that do not have an Internet connected device at their disposal. The totem is connected to Internet or the store local network. It mirrors the key features of the web application, but it doesn't require the customer to authenticate. Customers is able to enter the queue, but they won't be able to be notified of possible changes/delays. The reservation receipt is printed and the customer must keep it until they're admitted to the store, otherwise they won't be able to enter. The store will also have a monitor showing the current status of the queue and the users in queue who have to enter the store.

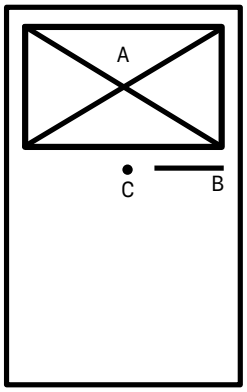


Figure 5: The totem interface: (A) is the touchscreen display, (B) is where the printed ticket is emitted, (C) is a proximity sensor that wakes the totem when a client is nearby.

3.1.3 Software Interfaces

In order to provide its full functionality to all users, the system will make use of the following software interfaces (APIs):

- **Maps**: to implement the functionality allowing the Users to look for a store the system will interface with a digital map service. Store Managers will be able to register on the map the position of their store.
- **Geolocalization**: geolocalization through GPS (when available) or other means (cellular, WI-FI or IP location) will be used in the web app and in the mobile app in order to show the User their current position and help them find a nearby store.
- **Mobile OS Notification Service**: notifications will be sent to warn and update Users in real time about their Reservations (if any) and the status of the Virtual Line (if it is joined).
- **SMS service**: an SMS with a secret code will be sent to the user upon login in order to verify their identity through a third party service.

3.1.4 Communication Interfaces

The backend of the system will expose a unified REST compliant API to communicate with all clients using HTTPS and TCP/IP. The User will need a working Internet connection as well as cellular network upon login. The Totem will be provided with WiFi, wired or cellular connectivity, depending on the physical constraints of each store.

3.2 Functional Requirements

Definition of use case diagrams, use cases and associated sequence/activity diagrams, and mapping on requirements

[R1] Allow a User to sign up for an Account after providing a mobile phone number.

[R2] Allow a Registered User to find Stores nearby a specified location.

[R3] Allow a Registered User to filter out stores based on available timeframes, days and distance.

[R4] Allow a Registered User to get in the virtual line at a specified store.

[R5] Allow a Totem User to get in the virtual line of the store where the totem is installed.

[R6] Allow a Registered User to preview an estimate of the queue time.

[R7] Allow a Registered User to book one visit to a specific store.

[R8] Allow a Registered User to cancel their reservation.

[R9] Allow a Registered User to leave the virtual queue.

[R10] Allow a Registered User and a Totem User to retrieve a scannable QR Code/Barcode that they must present in order to be granted access to a store.

[R11] The System notifies the Users affected by delay.

[R12] The System cancels User reservations in case of a major delay.

[R13] The System enforces the limits on the allowed number of concurrent Customers inside a store by restricting the access at the entry points (for example, automatic doors or turnstile).

[R14] The System grants a User with a reservation access only within a short time (set by the manager) after the User's time of reservation.

[R15] Allow System Managers to set the division of the maximum number of people allowed between the normal queue, the priority queue for people with special needs and the book a visit slot capacity.

[R16] The System calculates the average shopping time by recording every time a user enters and exits the store.

[R17] Allow System Managers to set a limit to the people allowed into the store at a time.

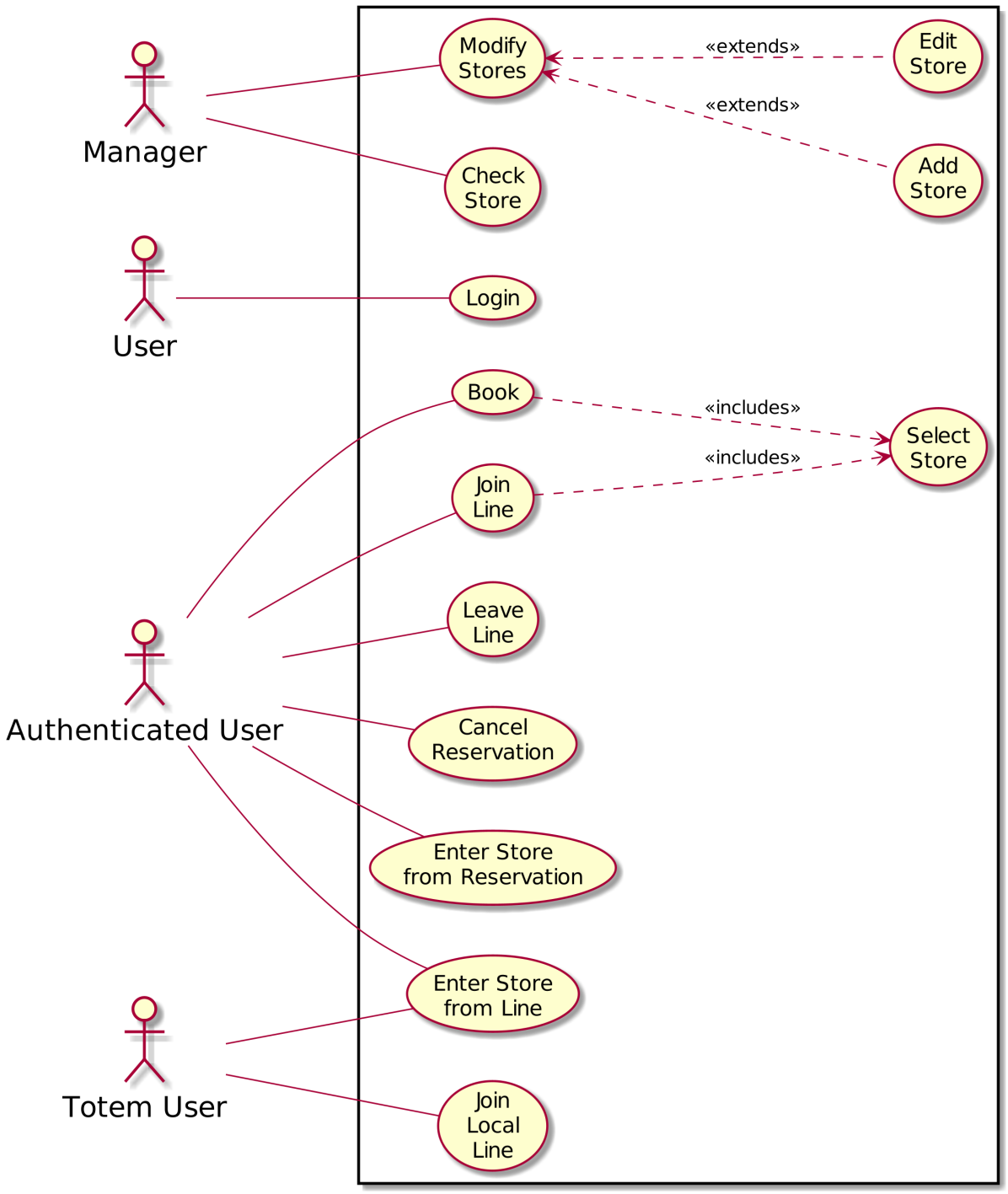
[R18] Allow System Managers to choose the frequency and size of the time slots.

[R19] Allow System Managers to know the average time spent in the store.

[R20] Allow System Managers to know the current and past number of people in the store.

[R21] Allow System Managers to check the current status of the queue and of the time slots.

Goal	**Requirements**	**Assumptions**
G1	R1, R2, R3, R4, R5, R6, R7, R10, R13, R14, R17	D1, D2, D3, D5
G2	R15	D3, D4, D5
G3	R6, R7, R11, R12	D2, D5, D6
G4	R10, R13, R14, R15, R17, R18	D1, D3, D4
G5	R16, R19, R20, R21	D1, D3, D4, D5, D6


Manager
User
Authenticated User
Totem User
Modify Stores
Edit Store
«extends»
«extends»
Add Store
Check Store
Login
Book
«includes»
Select Store
«includes»
Join Line
Leave Line
Cancel Reservation
Enter Store from Reservation
Enter Store from Line
Join Local Line

3.2.1 Use Case: Login

Name	Login
Actors	Customer
Entry Condition	The Customer has installed the application on their device.
Event Flow	1. The Customer opens the application. 2. The Customer may read Privacy Policy and the Terms of Service by clicking on the respective links. If they create an account, it's implied that they accept them. 3. The Customer enters their mobile phone number and taps "Continue". 4. The Customer enters the received token via SMS and taps "Sign In".
Exit Condition	The Customer is now logged-in and they may utilize the application.
Exceptions	• The Customer inputs invalid data. • The Customer doesn't fill the required data.
Notes	When an exception occurs, the user is informed by a human-readable message. In case the user thinks the SMS hasn't been sent/received, they can ask the system for a new token.

3.2.2 Use Case: Select Store

Name	Select Store
Actors	Authenticated User
Entry Condition	An authenticated User, with no pending reservation, opens the application.
Event Flow	1. (Optional) The Authenticated User applies some filters to restrict the available stores. 2. The System provides the user with a list of the supermarkets near him. 3. The Authenticated User is able to see a preview of the number of people in queue and the estimated waiting time. 4. The user selects the desired store. 5. The user is now able to see full screen the waiting time and the number of people in queue (and possibly other stats on the store), and is given the possibility to join the line or make a reservation if time slots are available.
Exit Condition	The Authenticated User has selected a store that suits their need, and is given the possibility to join the line or make a reservation.
Exceptions	• There are no open stores in the area. • There are no stores in the area.
Notes	When an exception occurs the users is promted to check the application again later.

3.2.3 Use Case: Join Queue

Name	Join Queue
Actors	Authenticated User
Entry Condition	Authenticated User selects a Supermarket in the application
Event Flow	1. The Authenticated User pushes the "Queue Now" button. 2. The system provides the Queue Receipt with the generated QR code and the estimated time remaining.
Exit Condition	The customer is now in the queue waiting for their turn, and will receive notifications about the state of the queue.
Exceptions	
Notes	

3.2.4 Use Case: Book

Name	Book
Actors	Authenticated User
Entry Condition	Authenticated User selects a Supermarket in the application
Event Flow	1. One of the following path: (a) The Authenticated User pushes the "Make a Reservation" button. • The user chooses a suitable time among the available ones. (b) The Authenticated User selects the less crowded slot proposed by the application. 2. The system provides the Reservation Receipt with the generated QR code.
Exit Condition	The customer has now a pending reservation.
Exceptions	
Notes	

3.2.5 Use Case: Leave Queue

Name	Leave Queue
Actors	Authenticated User
Entry Condition	Authenticated user enters the home page of the application and is currently in Line.
Event Flow	1. The Authenticated User clicks the "Leave Current Queue" button. 2. The Authenticated User confirms the selection.
Exit Condition	The customer has now left the queue.
Exceptions	
Notes	

3.2.6 Use Case: Cancel Reservation

Name	Cancel Reservation
Actors	Authenticated User
Entry Condition	Authenticated user enters the home page of the application and has a pending Reservation.
Event Flow	1. The Authenticated User clicks the "Cancel Current Reservation" button. 2. The Authenticated User confirms the selection.
Exit Condition	The customer has now canceled the reservation.
Exceptions	
Notes	

3.2.7 Use Case: Join Local Queue

Name	Join Local Queue
Actors	Totem User
Entry Condition	A user approaches the totem in front of a store while the store is open.
Event Flow	1. The user clicks the button appearing on screen. 2. The totem prints a ticket containing a QR code and the estimated waiting time.
Exit Condition	The user is now in queue.
Exceptions	
Notes	

3.2.8 Use Case: Enter Store from Queue

Name	Enter Store from Queue
Actors	Registered User, Totem User
Entry Condition	A user is in Queue.
Event Flow	1. The user waits until the ID on their ticket comes up on the monitor at the store or until a notification on the app appears. 2. The user enters the store by letting the totem read their QR code, shops, and exits the store.
Exit Condition	The user has shopped.
Exceptions	• There is a major delay in the store. • The time remaining to wait in the queue exeeds the opening hours of the store.
Notes	When an exception occurs, the user is asked to leave, through a notification on the app or at the store at the monitor.

3.2.9 Use Case: Enter Store from Reservation

Name	Enter Store from Reservation
Actors	Authenticated User
Entry Condition	User has a reservation
Event Flow	1. The user reaches the store at the specified time. 2. The user enters the store by letting the totem read their QR code, shops, and exits the store.
Exit Condition	The user has shopped.
Exceptions	• There is a major delay. • The user does not reach the store in time.
Notes	When there is a major delay the reservation is either postponed with the confirmation of the user or automatically cancelled, depending on the estimated waiting time. If the user does not reach the store in time the reservation is automatically canceled.

3.2.10 Use Case: Edit Store

Name	Edit Store
Actors	Manger
Entry Condition	The manager knows their own credentials for the web panel.
Event Flow	1. The manager opens the apposite web portal. 2. The manager inserts their credentials and confirm the login. 3. The manager chooses from a menu the operation they want to perform. 4. The manager edits the parameters of the system.
Exit Condition	The new parameters are applied.
Exceptions	• The Manager inputs the wrong credentials.
Notes	In case of a failed login, the Manager is informed by the System on how to retrieve their credentials, had they been forgotten.

3.2.11 Use Case: Add Store

Name	Add Store
Actors	Manger
Entry Condition	The Manager is logged into the web panel.
Event Flow	1. The manager chooses from a menu the option to add a new store. 2. The manager enters the details of the new store in a form. 3. The manager submits the form.
Exit Condition	A new store is added to system.
Exceptions	• The store can’t be added.
Notes	In case of exceptions, the manager is informed by a human-readable message.

3.2.12 Sequence Diagrams

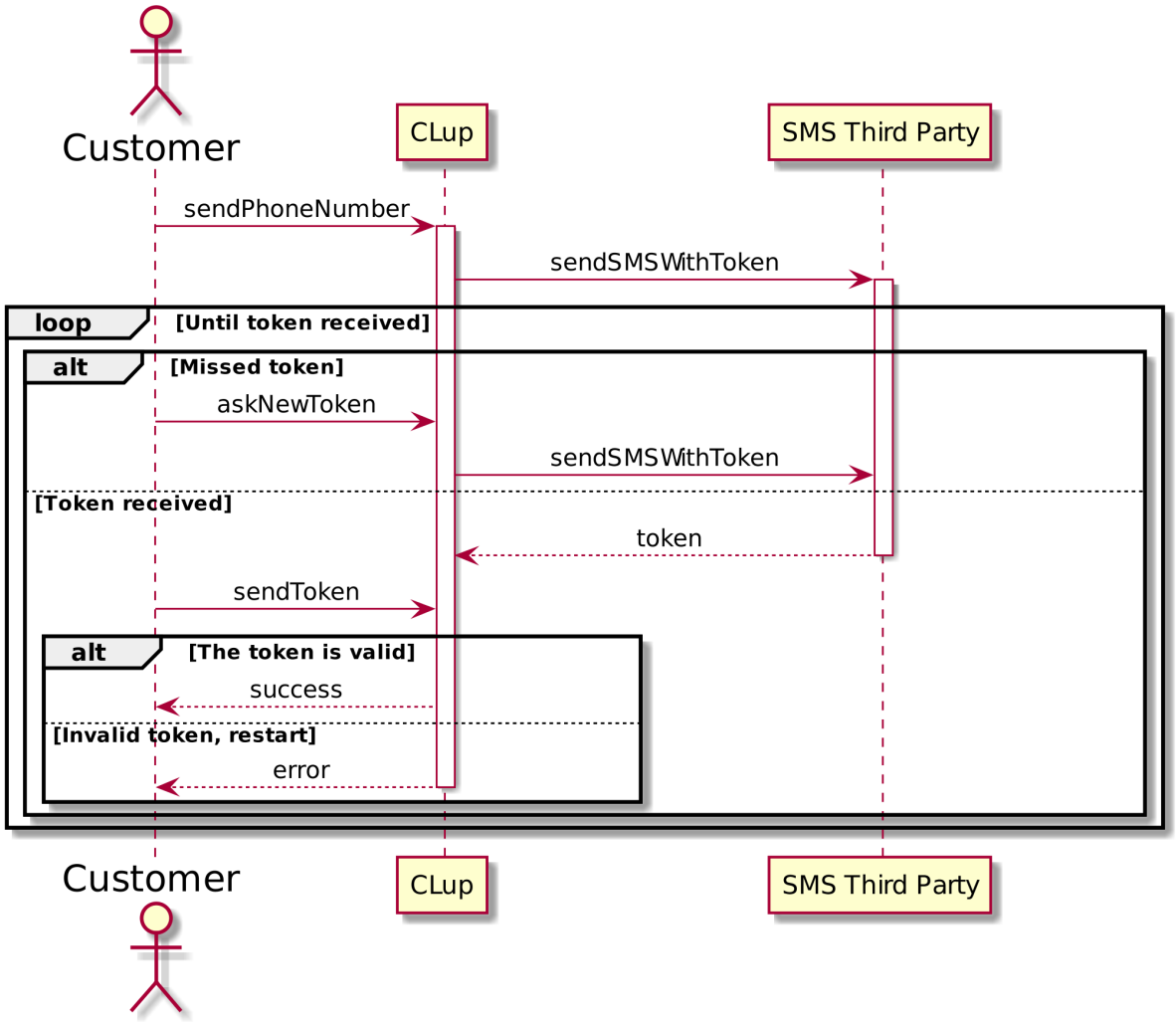


Figure 6: Login Sequence Diagrams

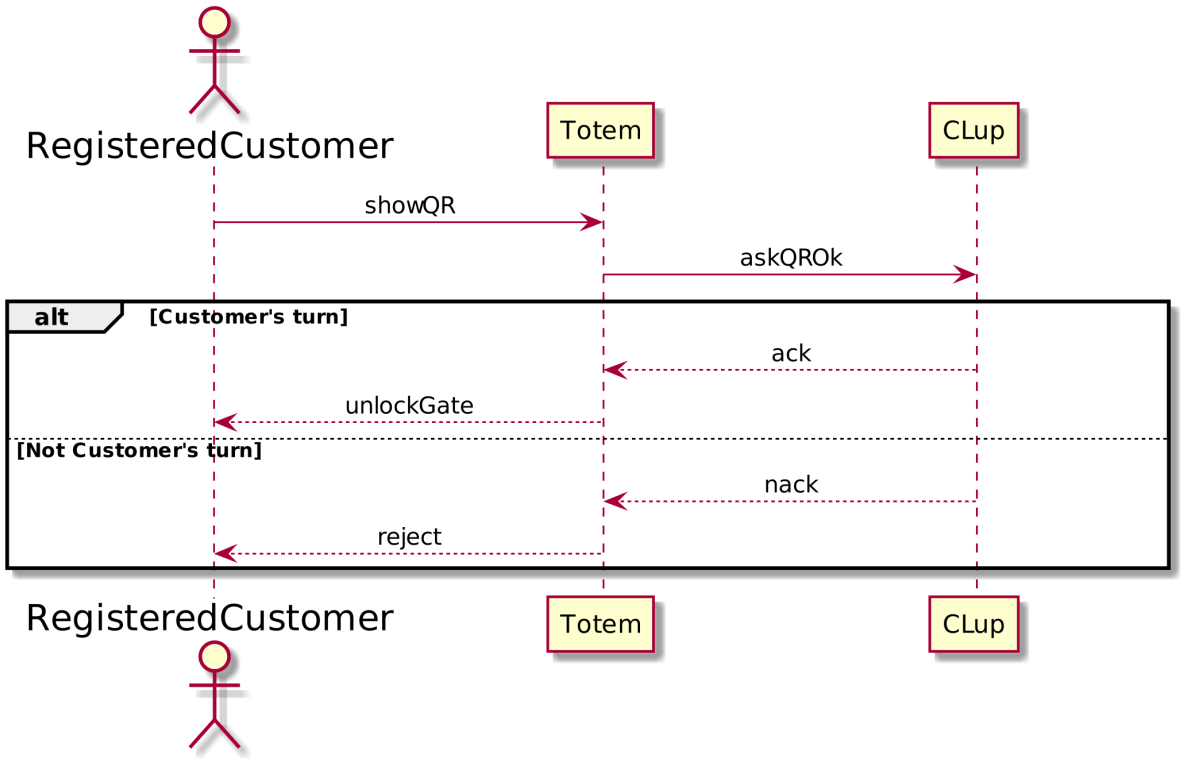


Figure 7: Enter Store from Queue Sequence Diagram

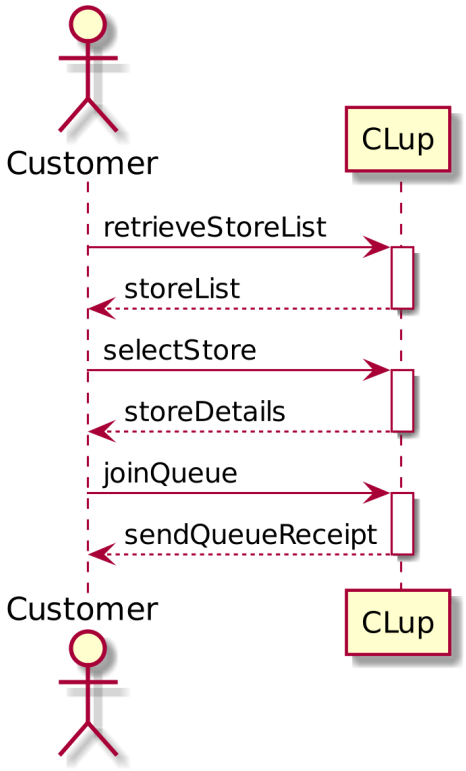


Figure 8: Join Queue

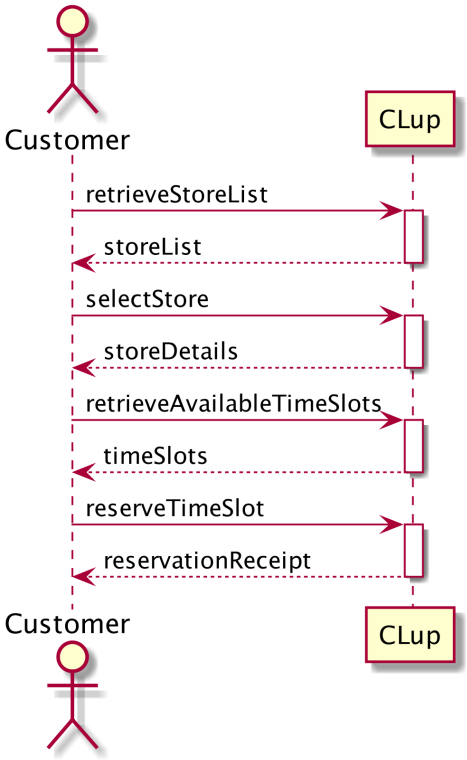


Figure 9: Reserve Timeslot

3.3 Performance Requirements

The system backend needs to handle a fair amount of customers, but little amount of data. It needs to be scalable, capable of load-balancing and reactive in order to notify customers on time. Protection against denial of service attacks would be necessary. Since each store chain has to deploy its own version of this system, they should estimate server resources based on the average amount of customers visiting their stores. Client-side and web applications must be responsive, fluid and handle correctly asynchronous interactions with the server (especially when the connection quality is bad), in order to provide the best experience possible to the end user.

3.4 Design Constraints

3.4.1 Standards Compliance

The system shall keep track of interactions with the system by the customers (i.e. joining the queue, entering the store, exiting the store). The data should be named accordingly to this document (RASD) and the Design Document (DD). All users sensible data shall be handled according to existing regulations, such as the GDPR.

3.4.2 Hardware Limitations

- Mobile Application
  - iOS or Android smartphone
  - Cellular data connection
  - WiFi
  - GPS
- Web Application
  - Internet connection
  - Computer/Tablet with a display
  - Major recent browser release (Firefox or WebKit based browser)
- Backend
  - Scalable webserver
  - Database Management System

3.5 Software System Attributes

3.5.1 Reliability

The system infrastructure should have the lowest MTTR reasonably achievable; should the system need maintenance, it should take place outside store opening hours or during the weekend. It's important to prevent data loss through redundant storage methods.

3.5.2 Availability

The whole system doesn't need to be functional 24/7, but certain parts (i.e. backend for the web and mobile app) must be functional at all times to allow user to book a visit at any hour of the day; the totems can be powered off after a certain hour to save energy. The managers web portal should not be heavy on computing resources therefore it's feasible to keep it online all the time.

3.5.3 Security

To protect customers sensible data (i.e. phone number, visits to stores, location) the system should collect in first place the required minimum data possible, and store them in an encrypted database. The connections to the system should follow modern standard protocols (HTTPS over TLS) and be encrypted as well. Since managers must log in with username and password, those data should be encrypted and the password salted and hashed.

3.5.4 Maintainability

All the code produced should be well commented and tested. The system structure should provide the ability to deploy updates in the backend without the user noticing it. System components should be created to be as modular as possible. Mobile applications can be updated through their OS app store to release new features and fixes.

3.5.5 Portability

The application must be available on major mobile OSs (Android and iOS) targeting at least the current release, even though the previous would be ideal. The web application must run at least on the most recent browsers, but browser support can be limited. It would be a smart choice to reuse most of the codebase across mobile and web by using cross platform tools.

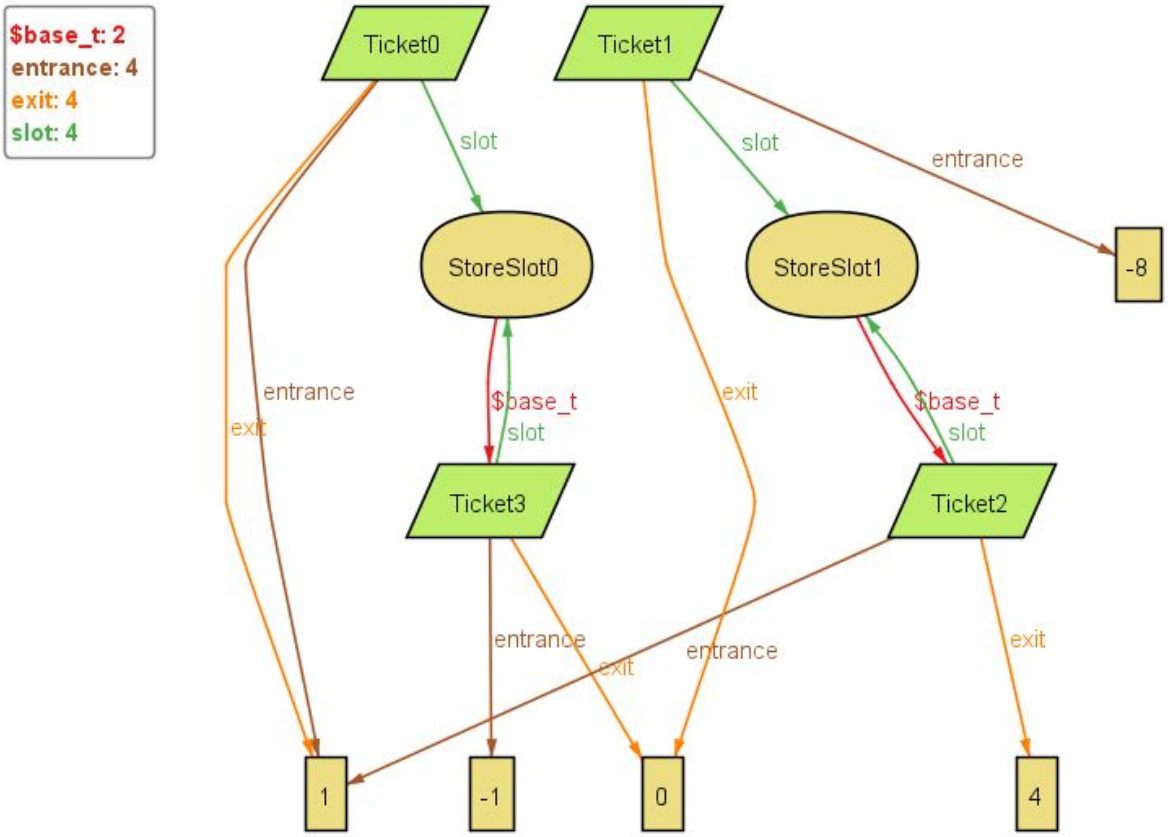
4 Formal Analysis using Alloy

Alloy is a declarative language commonly used to describe, model, and verify complex systems. Its objective is creating small models and checking their correctness, using the power of first order logic. We used Alloy to model and verify the behavior of specific parts of the application.

4.1 Queue Modeling

Queue management is one of the most critical aspects of the application. We implemented the concept of queue as a set of ordered Tickets. We made the assumption that each customer will stay at least one unit of time inside of the store to be able to model it through Alloy Int variables. To create a fully populated world we used one fact to have all the slot assigned to at least one ticket. We checked that no slot inside of the store is assigned to multiple people at the same time, by respecting the registered entrance and exit time.

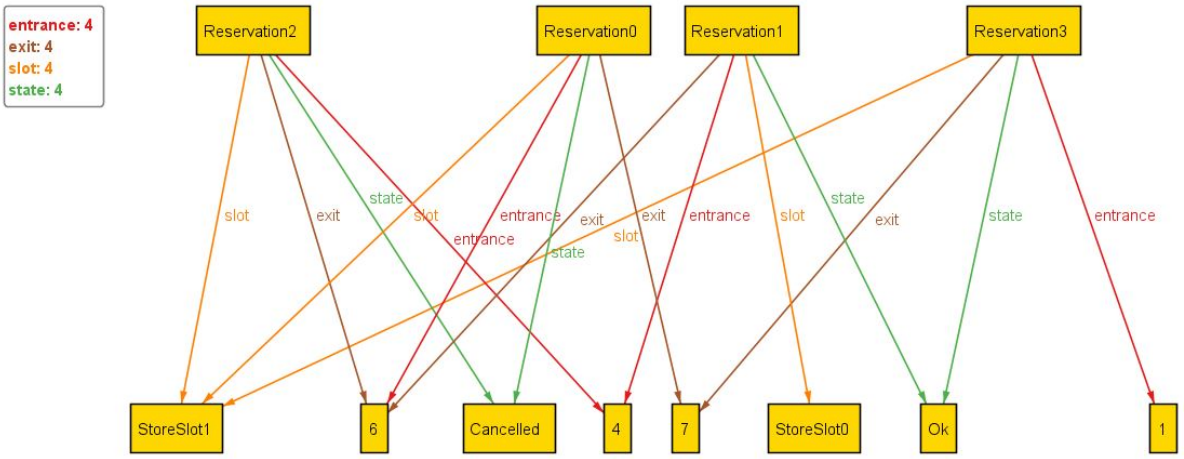
```
sig StoreSlot {
}

sig Ticket {
  entrance: one Int,
  exit : one Int,
  slot: one StoreSlot
}{
  exit>= sum[entrance+1] //ticket dura un ora
}


fact no_StoreSlot_without_ticket{
  no s: StoreSlot | no t: Ticket | s = t.slot
}

fact one_person_at_max_for_each_StoreSlot {
  all disjoint t1, t2: Ticket | (t1.slot = t2.slot) =>
     (t1.exit<t2.entrance or t2.exit <t1.entrance )
}

assert no_StoreSlot_with_two_people{
 no disjoint t1,t2 : Ticket | (t1.slot = t2.slot) and
  ( (t1.entrance<t2.entrance and t1.exit>=t2.entrance) or
  (t2.entrance<t1.entrance and t2.exit>=t1.entrance))
}

pred base {
  #Ticket >= 4
  #StoreSlot >= 2
}

run base for  40

check no_StoreSlot_with_two_people
```


$base_t: 2
entrance: 4
exit: 4
slot: 4
Ticket0
Ticket1
slot
slot
entrance
StoreSlot0
StoreSlot1
-8
entrance
exit
$base_t
slot
exit
$base_t
slot
Ticket3
Ticket2
entrance
exit
entrance
exit
1
-1
0
4

4.2 Reservation Modeling

Reservations are another critical aspect of the applications. Users who make a reservation expect to shop at the specific time of the reservation. In extreme cases reservations can be automatically cancelled by the system. This should happen only in case of another user taking too long to shop. The objective of this Alloy model is to verify that the presence of two concurent reservations at the same slot implies that one of the two has been cancelled by the system.

```
sig StoreSlot {
}

enum State { Ok, Cancelled }

sig Reservation {
        slot: StoreSlot,
        entrance: one Int,
        exit: one Int,
        state: one State
}{
        entrance >= 0
        exit > entrance
}

// a minimum time of 2 hours is kept between reservations to the same slot
fact baseRespectedTime {
        all disjoint r1, r2: Reservation | (r1.slot = r2.slot) =>
                (r1.entrance >= sum[r2.entrance+2] or r2.entrance >= sum[r1.entrance+2])
}

// there are not valid concurrent reservation for the same slot
fact noConcurrentReservation {
        all disjoint r1, r2: Reservation |
                (r1.state = Ok and r2.state = Ok and r1.slot = r2.slot)
                        => (r1.entrance >= r2.exit or r2.entrance >= r2.exit)
}

// if a reservation is cancelled it means that someone who came before was shopping slowly
fact noUselessCancellations {
        all r1: Reservation | r1.state = Cancelled => one r2: Reservation |
                (r2.state = Ok and r1.slot = r2.slot and
                r2.entrance < r1.entrance and r1.entrance < r2.exit)
}

// if there are two concurrent reservations for the same slot one of them is cancelled
assert reservationCancelledIfConcurrent {
        all disjoint r1, r2: Reservation |
                (r1.slot = r2.slot and
                        ((r1.entrance >= r2.entrance and r1.exit < r2.exit) or
                        (r2.entrance >= r1.entrance and r2.exit < r1.exit))
                ) =>
                        ((r1.state = Cancelled) or r2.state = Cancelled)

}

pred base {
        #StoreSlot = 2
        #Reservation >= 4
        #Cancelled = 1
}

run base for 10

check reservationCancelledIfConcurrent
```


entrance: 4
exit: 4
slot: 4
state: 4
Reservation2
Reservation0
Reservation1
Reservation3
slot
exit
state
slot
entrance
entrance
exit
state
slot
exit
entrance
slot
state
exit
state
entrance
StoreSlot1
6
Cancelled
4
7
StoreSlot0
Ok
1