POLITECNICO
MILANO 1863

M.Sc. Automation and Control Engineering

Software Engineering (for Automation)

Academic year 2021/2022

Requirements Analysis and Specification Document

CLup: Customers Line-up

Advisor: Prof. Matteo Giovanni Rossi

Students: Alessandro Nocentini alessandro.nocentini@mail.polimi.it

Alberto Valentini alberto.valentini@mail.polimi.it

Fausto Luca Pichierri faustoluca.pichierri@mail.polimi.it

Table of Contents

1. Introduction

1.1 Purpose

This document is intended to be a support for developers that want to implement a mobile phone application called Customer Line-up (CLup), thought to manage the influx of people inside a supermarket.
It provides a brief and at the same time complete description of the requirements and the features of the app.

1.2 Scope

The coronavirus emergency has put a strain on society on many levels, due to many countries imposing lockdowns that allow people to exit their homes only for essential needs, and enforcing strict rules even when people are justified in going out (such as limiting the number of accesses to buildings and keeping a distance of at least one meter between people).
In particular, grocery shopping a most essential need can become a challenge in the presence of such strict rules. Indeed, supermarkets need to restrict access to their stores to avoid having crowds inside, which typically results in long lines forming outside, which are themselves a source of hazards. In these trying times, people turn to technology, and in particular to software applications,to help navigate the challenges created by the imposed restrictions.
The goal of this project is to develop an easy-to-use application that, on the one side, allows store managers to regulate the influx of people in the building and, on the other side, saves people from having to line up and stand outside of stores for hours on end.
Note: CLup covers only the user experience side.

1.2.1 Goals

- **G1**: Allow Store Managers to regulate the influx of the people in their store
- **G2**: Avoid the creation of hazard gatherings outside the stores and crowds inside it.
- **G3**: Allow customers to save time booking their tickets from home

1.3 Glossary

1.3.1 Definitions

- Customer: person that has to buy something at the supermarket.
- User: customer with a smartphone that has downloaded CLup app and uses it.
- Non-User: customer that does not use the application.
- Store manager: person that administrates the store.
- Store capacity: maximum number of customers allowed in the store at the same time.
- QR Code: type of matrix barcode machine-readable.
- System: sum of hardware and software units dedicated to provide services and features guaranteed by the application.
- Ticket: Element generated by the system containing the QR Code.

- User city: the city in which the user looks for a supermarket.
- Time slot: A time window of half of an hour.
- Full time slot: time slot that has reached the maximum number of acceptable reservations set by the store manager.
- Past time slot: time slot that is no more bookable because its time window is over at the user time.
- Free time slot: time slot that is not past or full, so available for reservation.
- Reservation time: time window booked at the supermarket by the user.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- CLup: Customer Line-Up (name of the application).
- QR Code: Quick Response code.

1.3.3 Abbreviations

- Gn: Goal number n.
- Dn: Domain assumption number n.
- Rn: Requirement number n.

1.4 References

- Project proposal: link to the document
- Course slides on WeBeep.

2. Overall Description

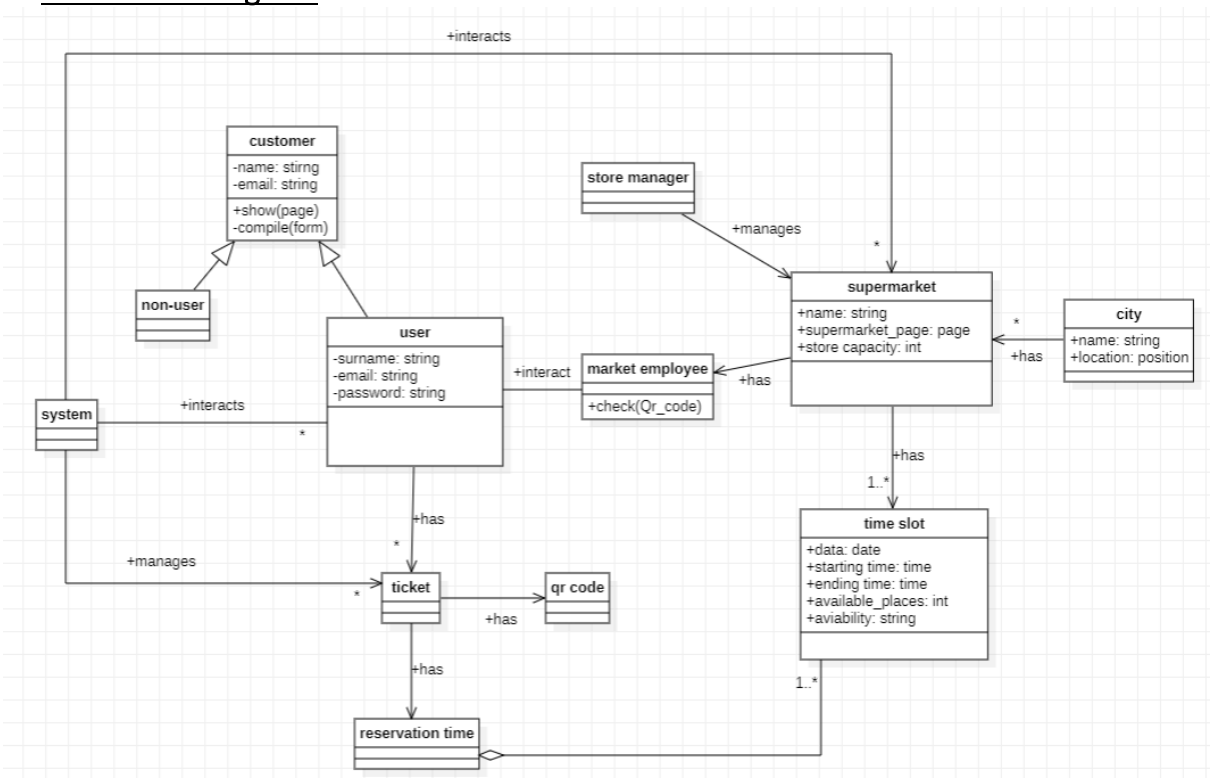
2.1 Product Functions

- **Sign up&Sign in**: the first time the user uses the application, they must sign up.
  Next times, they will access to CLup through the login procedure using the credentials registered in the sign-up phase.

- **Booking Management**: allows the user to look for a supermarket, among which of them that are supported by the application and are in the user city. Then, the user can choose free time slots from the supermarket timetable, provided that they are consecutive, in the same day, and they do not overlap with the already existing user tickets in the selected store.
  The reservation is stored and a virtual ticket composed of a QR code will be generated accordingly: it must contain info for a unique validity check. The QR code must be scanned at the entry of the supermarket. The ticket control is delegated to the supermarket, for instance, through an attendant at the entrance.
  Once the ticket is created, the user can always later check or delete it until the end time of reservation, when it is automatically deleted by the system.
  In order to allow the access to the store also to the non-users, the store capacity dedicated to the booking by app is equal to X% (X parameter decided by the store manager).

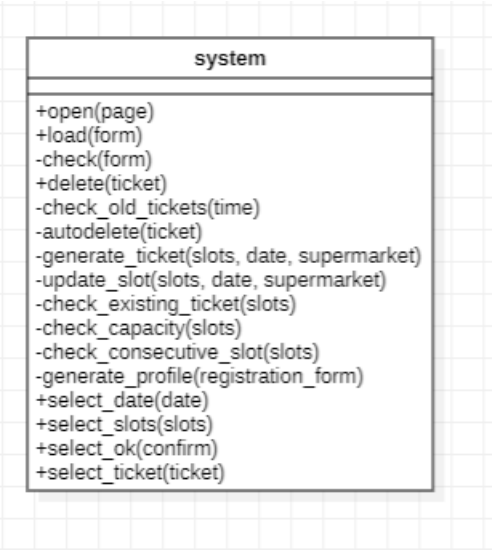
2.2 Product Perspective

2.2.1 Class diagram

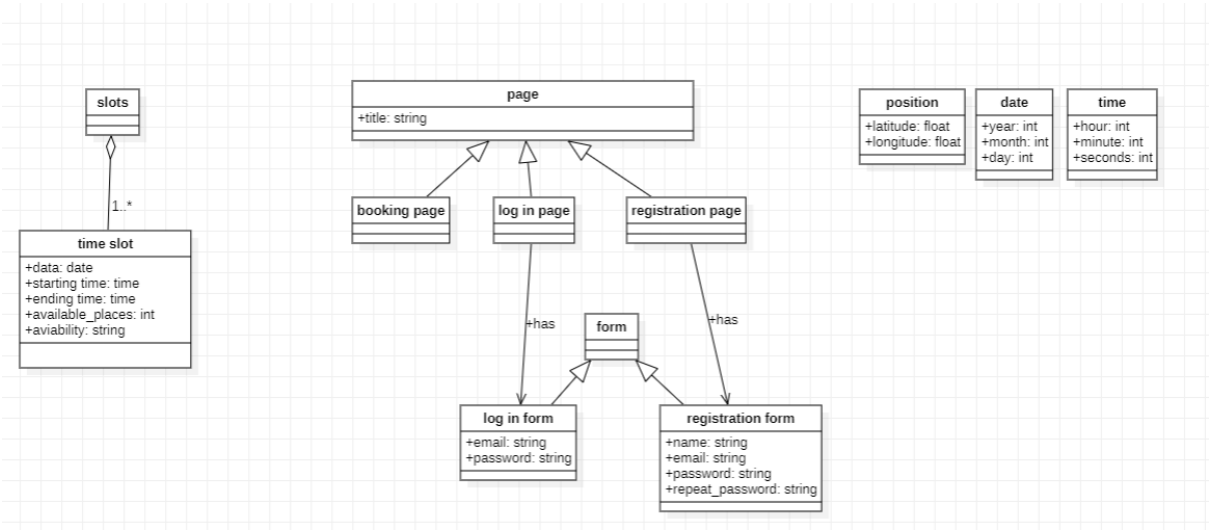


Class diagram for application domain

As displayed in the previous class diagram, the user interacts with the system in order to use the application and make a reservation. He also communicates with the market employee before entering in the store.
Below, many of the system's methods are shown and some classes capturing domain elements are defined.



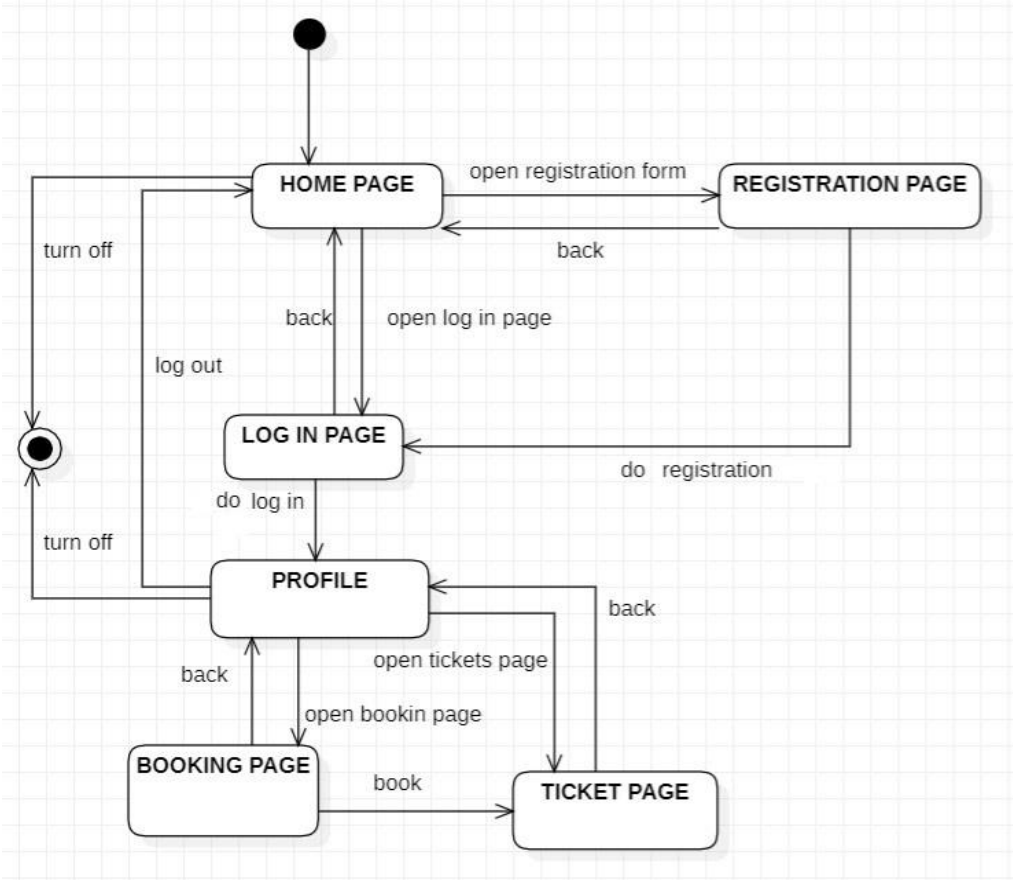
System's methods



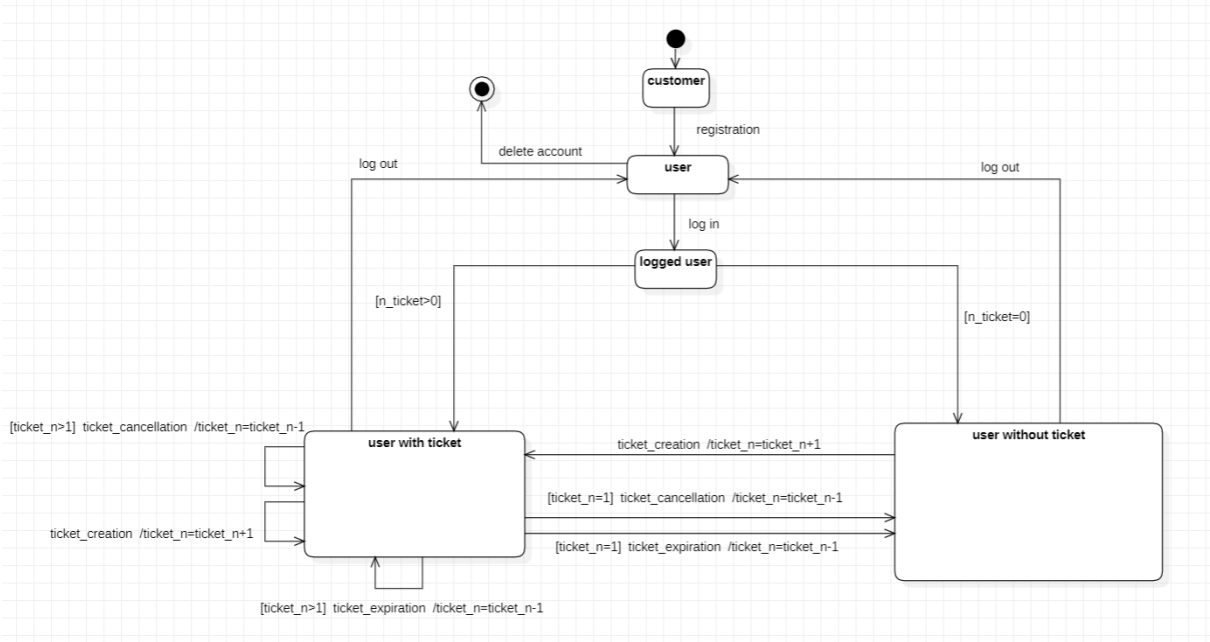
Classes capturing some domain elements

2.2.2 State chart diagrams

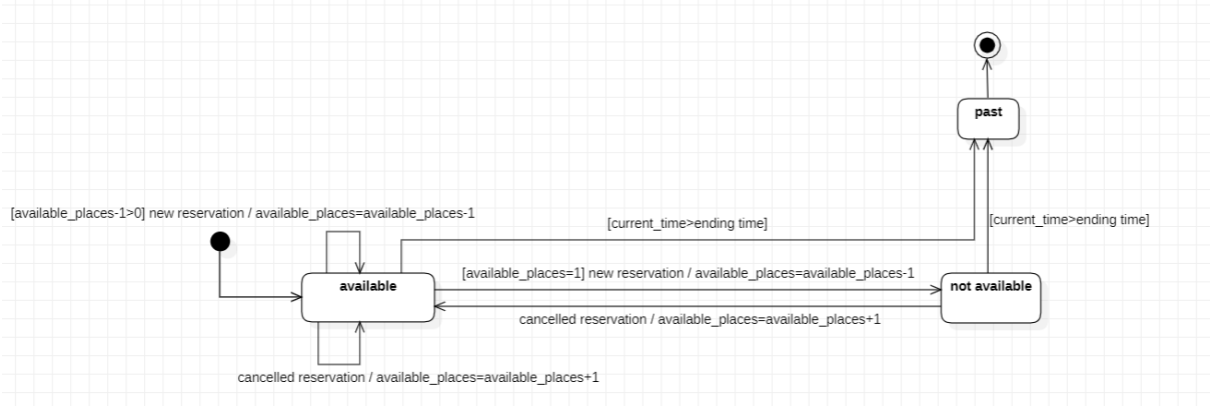
In this section, an overview of the display state's flow during the user's activity. Moreover, it is shown the different evolution of the user's state and of the slot aviablity. Finally, it is provided a complete shopping scenario.



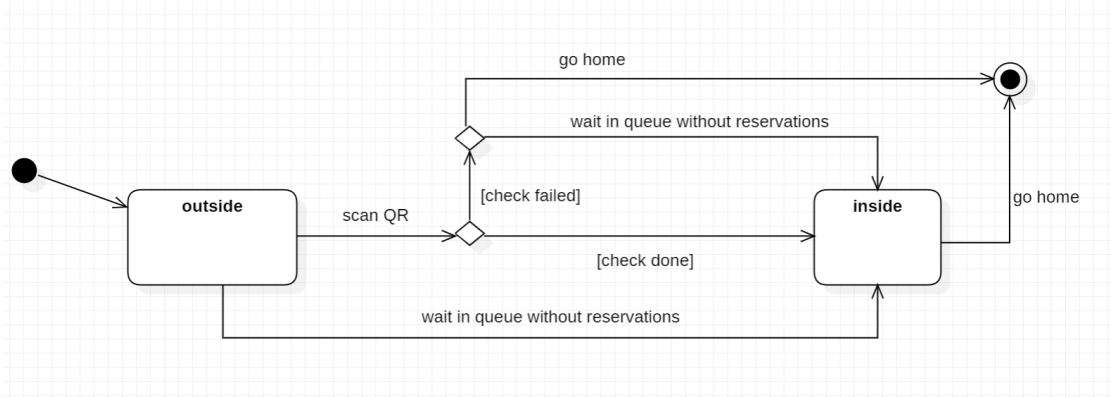
State chart diagram 1: Display



State chart diagram 2: User's state evolution

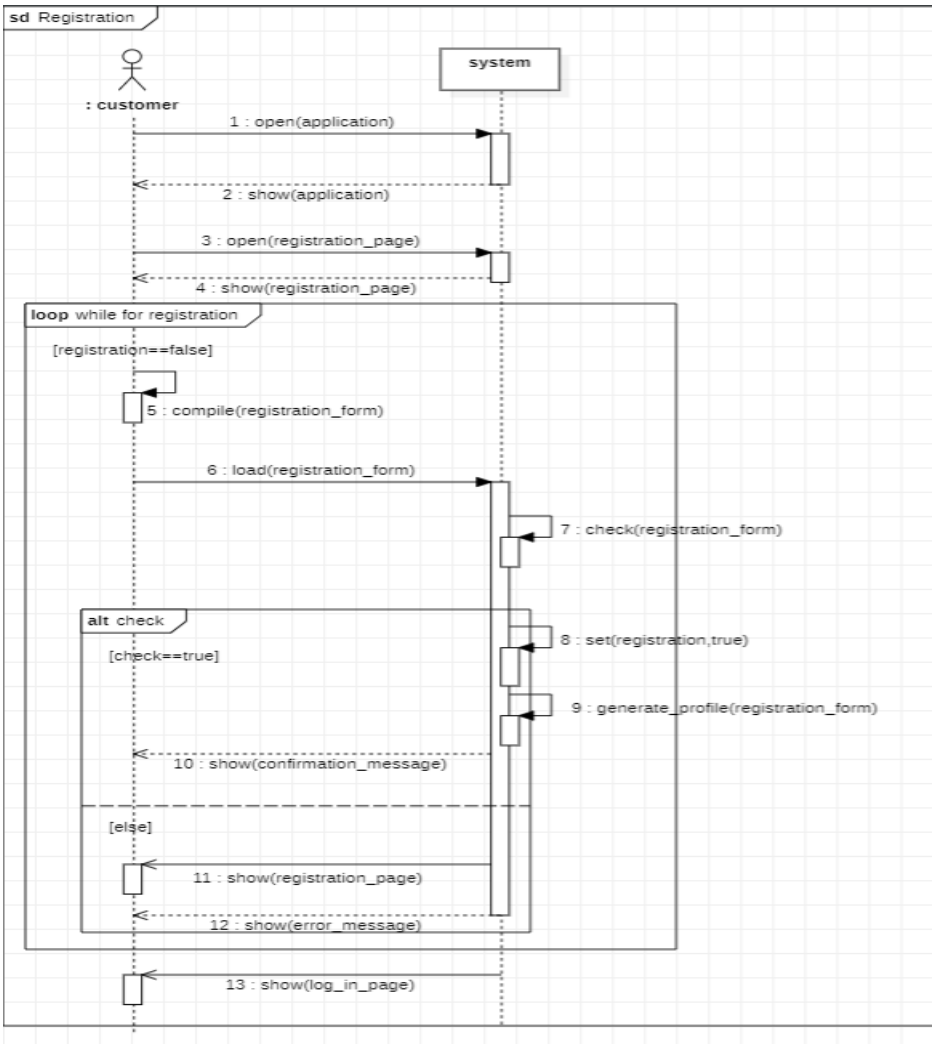


State chart diagram 3: Time slot availability

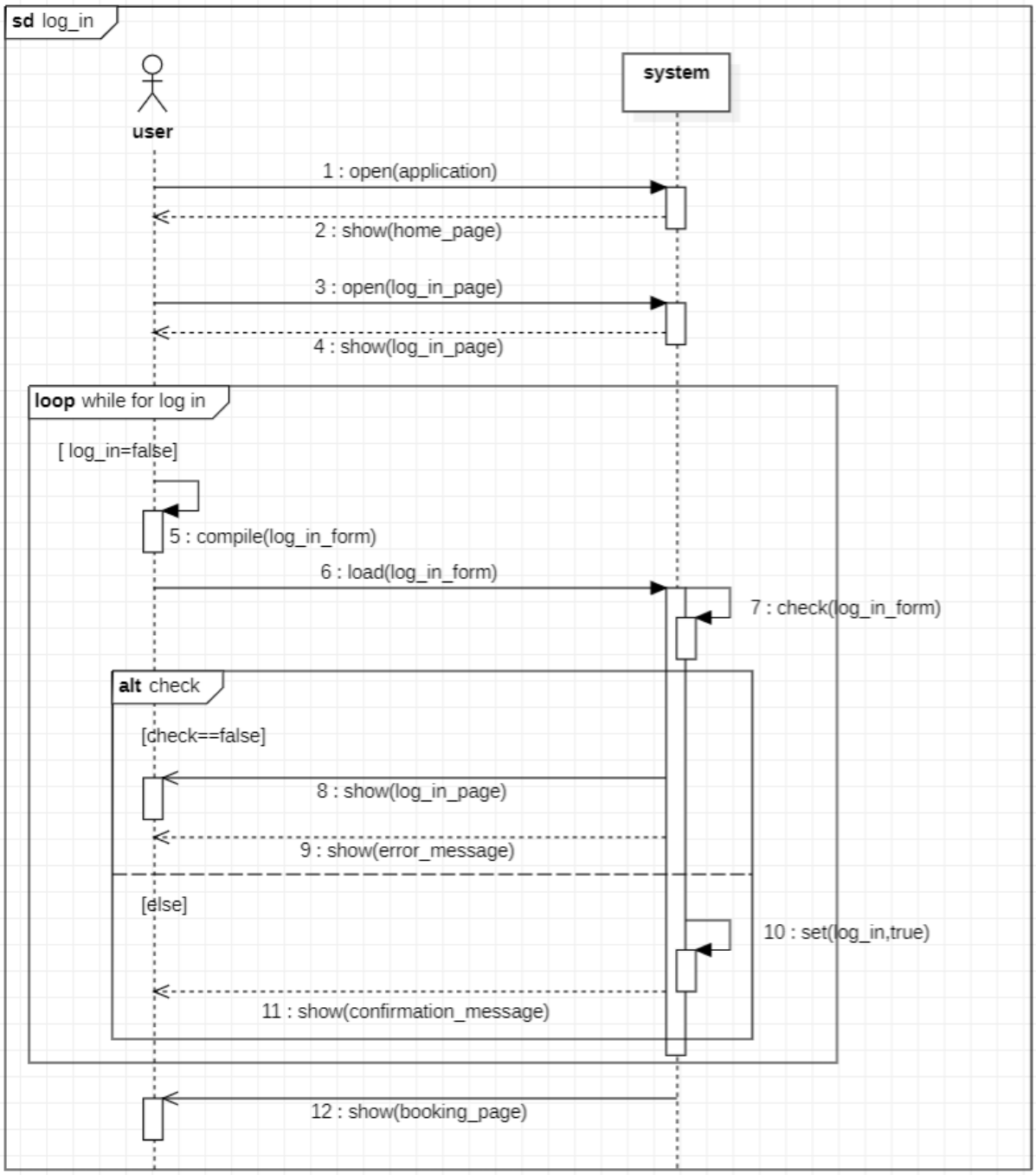


State chart diagram 4: Shopping scenario

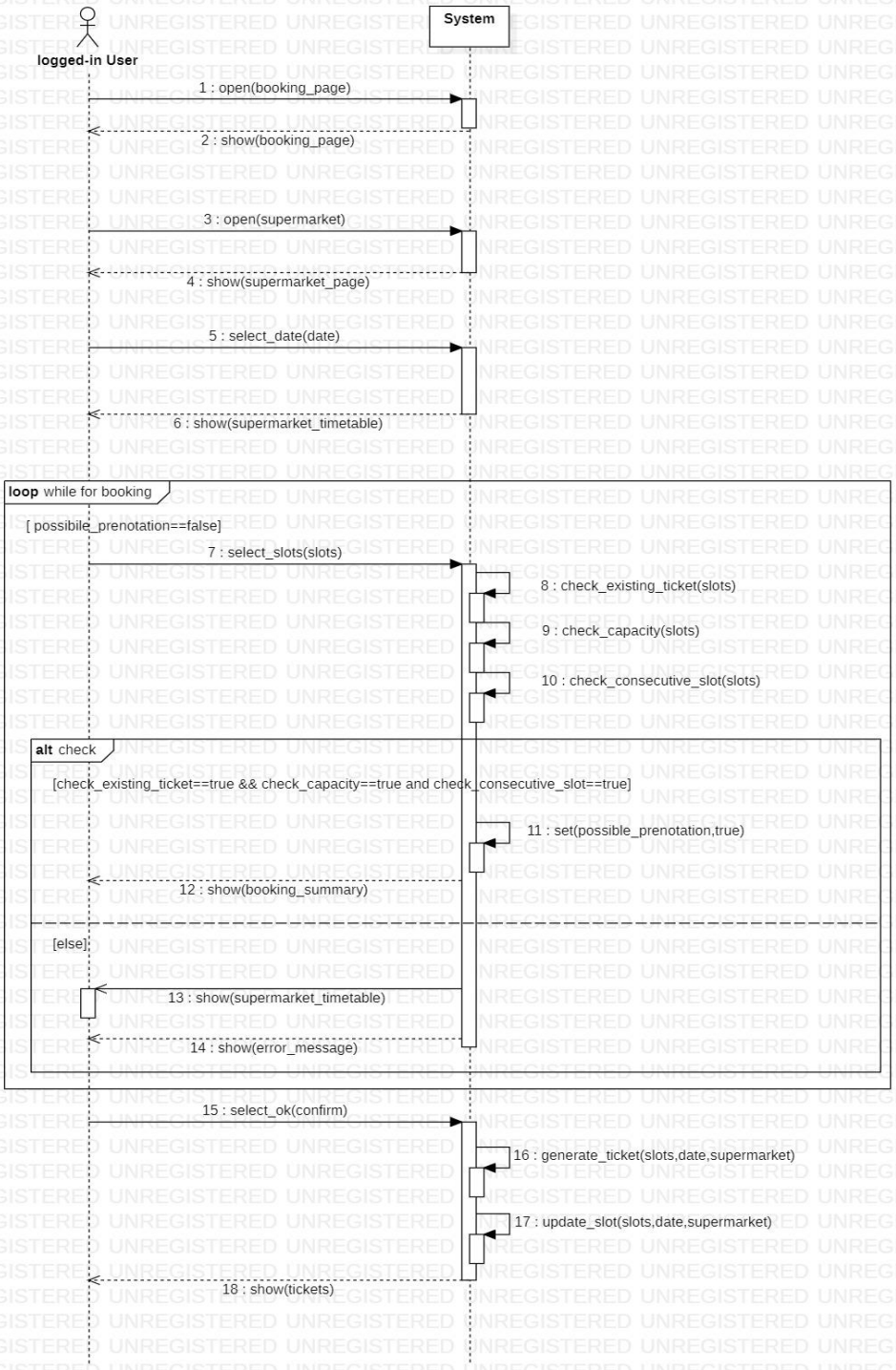
2.2.3 Sequence Diagrams



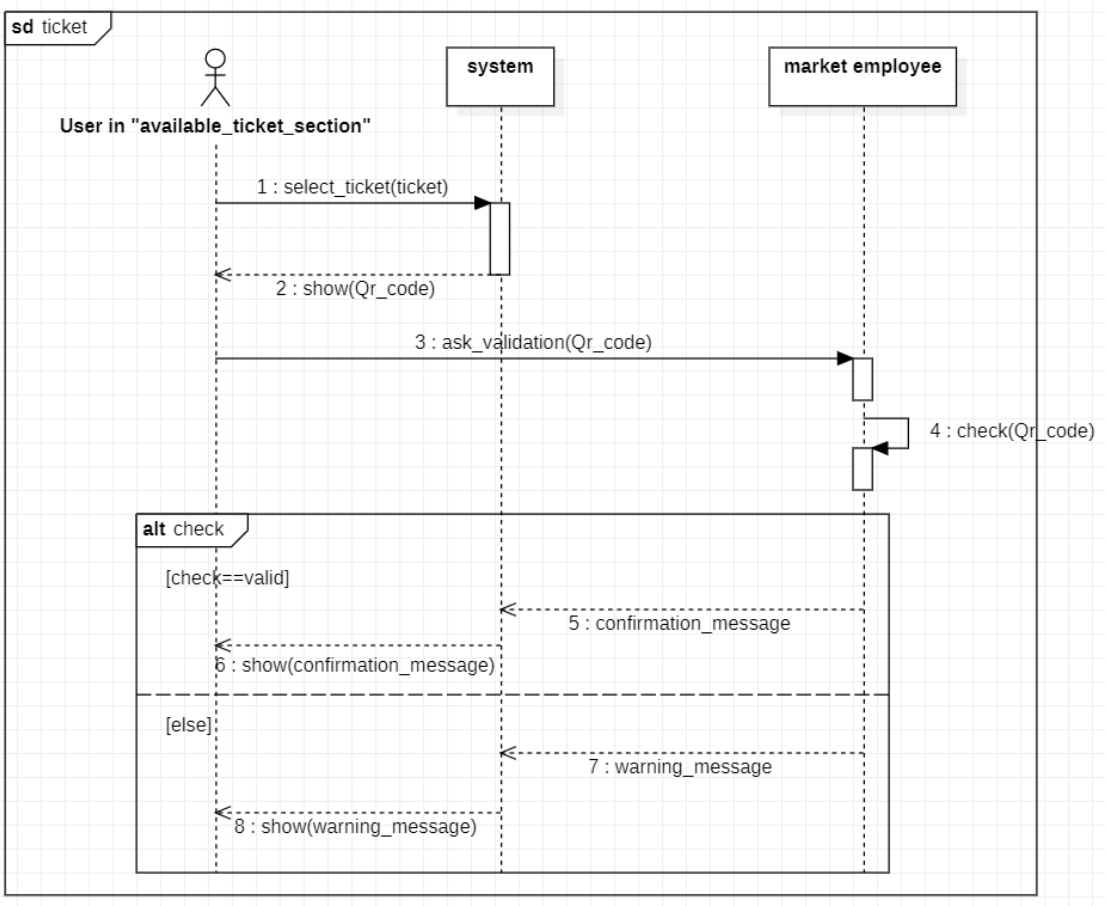
Sequence diagram 1: Registration



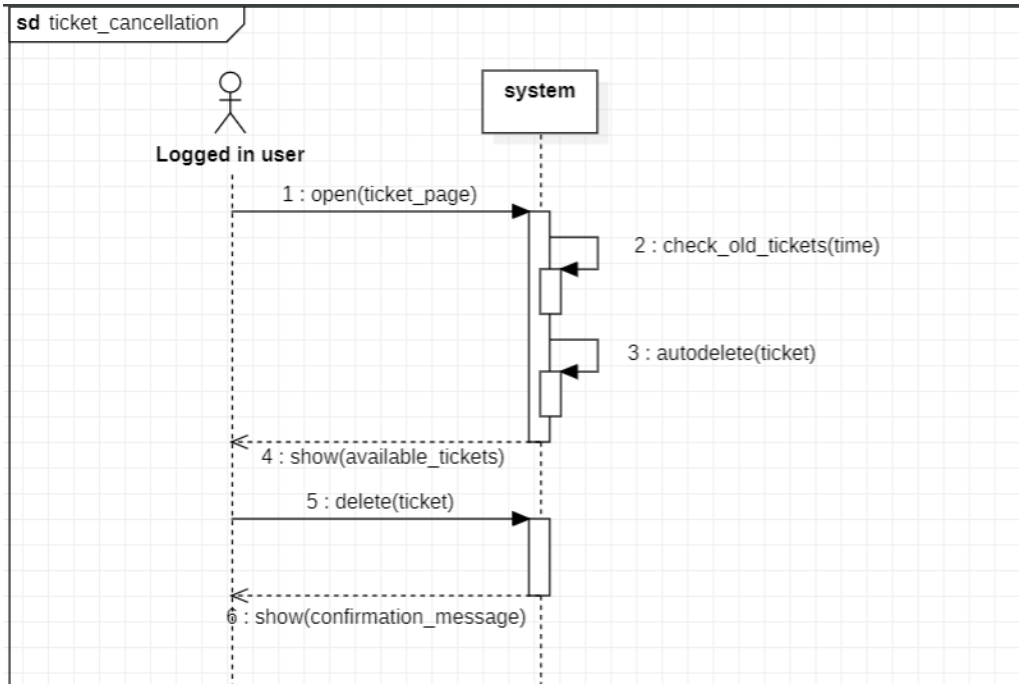
Sequence diagram 2: Log in



Sequence diagram 3: Booking



Sequence diagram 4: Ticket validation



Sequence diagram 5: Ticket cancellation

2.3 User Characteristics

The user is someone who wants to avoid spending time in long queues and, meanwhile, wants to stay safe avoiding gatherings. The application should designed to be able to be used by any range of ages and without distinction of any kind.

2.4 Domain Assumptions

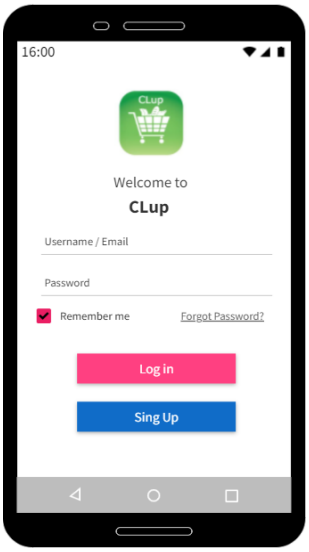
- **D1**: The user internet connection works properly.
- **D2**: Data given by users are assumed correct.
- **D3**: Data given by store managers are assumed correct.
- **D4**: Users expected to exceed the reservation time are few.
- **D5**: Each supermarket has a limited capacity grater than zero.
- **D6**: The supermarkets are open non-stop during their service.
- **D7**: Stores are uniquely identified.
- **D8**: The user's device, store's system and CLup's system have synchronized date and time.
- **D9**: Each supermarket accepts a limited numer of users, the remaining capacity is assigned to the non-users.
- **D10**: Each store has an attendant at the entrance who manages the influx of people and oversees the situation.

3. Specific Requirements

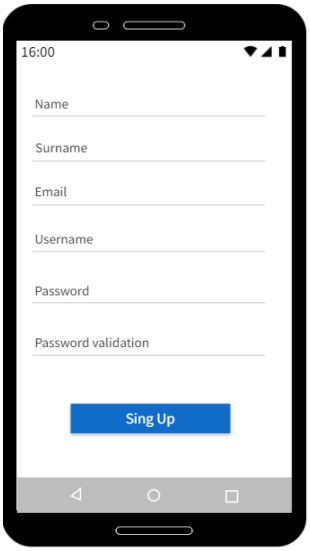
3.1 Preliminary user manual

The following pictures represent the main features of the app and how they might look to the user.

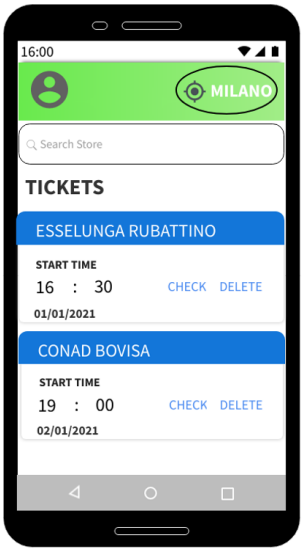
1. Log-in



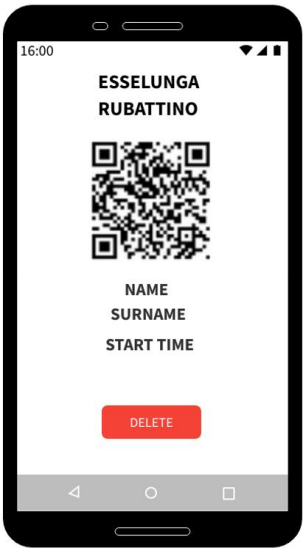
2. Sign up



3. Home

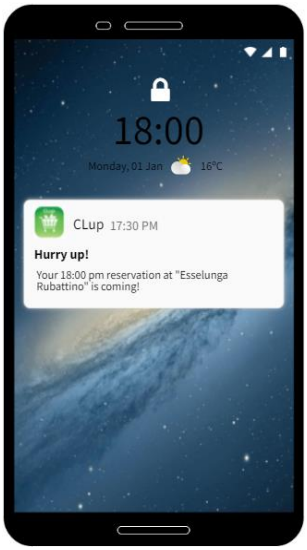


4. Check Ticket



5. Create ticket

6. Notification



When the user opens the application, he will find the log in form. Here, he can enter his credentials if he is already enrolled, otherwise, he can sign up from the registration page. To avoid this procedure at every use, the user can tick the “remember me” button. Once logged in, the user can find the already created ticket; he can cancel them, or decide to generate the corresponding Qr code. For a new reservation, the user can specify the city and search for the desired supermarket. Then, he has to choose the preferred date and time slots; finally he has to push on the “book” button. The user will receive an email reminding him not to lose his reserved spot.

3.2 Functional Requirements

Core:

- **R1**: The system must allow users to sign up if and only if they are not already registered.
- **R2**: The system must allow users to sign in if and only if already registered and the credentials inserted are valid.
- **R3**: The system must allow the user to log out.
- **R4**: The system must allow the user to search for and select the supermarket.
- **R5**: The system must allow the user to choose the city.
- **R6**: The system must restrict the supermarket research to the ones that are in the user city.
- **R7**: The system shows time slots of the selected market according to the timetable and crowding data.
- **R8**: The system must allow the user to select the day on which they want to book in the

selected supermarket, among the ones that the store manager makes available.

- **R9**: The system allows the user to make a reservation only in free time slots.
- **R10**: The system must generate one and only one ticket per reservation.
- **R11**: The system must update the time slot crowding of the supermarket for each ticket created/deleted in that store.
- **R12**: Each ticket generated by the system must have a unique QR code.
- **R13**: The system allows the user to make a reservation over more time slots only if they are all free, consecutive and in the same day.
- **R14**: The system allows the user to make a reservation only if it does not contain time slots already involved in existing reservations at the selected market on the same day.
- **R15**: the system must not allow the user to have concurrent reservations over a maximum number.
- **R16**: The system must allow the user to check their pending tickets.
- **R17**: The system must allow the user to delete their pending tickets.
- **R18**: The system does not allow the user to change single settings of already generated tickets.
- **R19**: The system must automatically delete the ticket at the end of its reservation time.

Medium priority:

- **R20**: The system should allow the user to update their data and delete their account.
- **R21**: The system should allow the user to auto-log in if they have previously ticked a "Remember me" box.
- **R22**: The system should allow the user to recover their credentials.
- **R23**: The system should alert the user when their account is created/deleted.
- **R24**: The system should filter supermarkets by name.
- **R25**: The system should remind the user about incoming reservations before their beginning.

Good to have:

- **R26**: The system should allow the user to select the app language.
- **R27**: The system should allow the user to contact the assistance.
- **R28**: The system should allow the user to download the QR ticket to use it offline.
- **R29**: The system should show the recent supermarket researches.
- **R30**: The system should memorize the favourite supermarkets.

3.3 Nonfunctional Requirements

Since the application is related to a daily activity like grocery shopping, the system should be able to manage a high amount of concurrent requests: depending on the cities size supported, one value could be a minimum of 1000 simultaneous users.
On the other hand, the app should be as lightweight as possible and guarantee fast time responses.

3.4 Software System Attributes

3.4.1 Reliability

The system is required to be robust and able to face potential faults that could happen, in order to guarantee the continuity of the service.

3.4.2 Availability

Because the application is designed for grocery stores, which might be open 7 days per week and sometimes 24h, it should guarantee a similar availability.

3.4.3 Security

The system does not use sensitive data, but would be good to have a technology that guarantees the protection of the user's data against intrusions from non-authorized people.

3.4.4 Maintainability

The system should be designed following a modular approach so that possible errors will be confined in stand-alone modules and facilitating correction and updates.
The maintenance should be made during the periods when the service is less used, like during nights or holidays, and an alert should be sent to the users the day before.

3.4.5 Portability

The application should be available for most Android and iOS devices in order to reach its goal of widespread usage.

3.4.6 Usability

The application should be used by any range of ages, so it should have a simple and intuitive user interface.