POLITECNICO
MILANO 1863

RASD

Requirement Analysis and Specification Document

Authors:

David Leonardo Gomez, 10787907

Martin Anselmo, 10789796

Vinay Krishna Munnaluri, 10780972

Version: 1.0

Date: 29 November 2020

Professor: Elisabetta Di Nitto

Contents

1. Introduction

1.1. Purpose

This document will present a description of CLup, a system designed to solve the problem of crowds inside of supermarkets during the Covid-19 pandemic, explain the features and interfaces of the system, as well as the main functionalities and conditions under which the system will operate.
All of this is done through the analysis of the world and conditions where the system will live. This corresponds to use cases, scenarios, domain assumptions, requirements, interfaces and many more things.

1.2. Scope

CLup is a system created in the context of the Covid-19 pandemic, where large crowds of people are dangerous. It consists in a system for managing the number of people inside a supermarket, as well as the queues that can form outside. All of this so the customers of the supermarket can have a safer and more optimized experience when shopping, and at the same time, allow the managers of each store to monitor the number of people inside of it.

The three main functionalities offered by CLup are:

1. Lining up: A customer can request a number (ticket) for a virtual queue for entering the supermarket. They will be given a QR code that will be scanned when entering and exiting the store.
2. Book a visit: A customer will be able to request the booking of a visit to the supermarket, so they don’t have to wait in a queue if they know the time they are coming. In addition to this, the customer will be able to tell the system the approximate duration of their visit and what they will buy, so that the system can better manage the people inside of a store.
3. Alternatives for the customer: CLup will offer the customer suggestions in case the queue for a particular store is too long (going to a different store or in a different time).

The system will also include the interfaces for scanning the QR codes, so the store managers can correctly monitor the amount of people inside of the store. The scanning of said codes will allow the system to automatically calculate the queues waiting times, and do the predictive work required for suggesting alternatives.

The users will need to be authenticated for using all of the functionalities except the *lining up* feature, where it will be optional. This for the case of customers who don't have access to the technologies required, so they can go directly to the store, and ask the manager to generate a ticket for them.

1.2.1. World Phenomena

WP1	A store is full of customers
WP2	The queue for a store is long
WP3	A customer with Covid-19 enters a supermarket
WP4	A customer does not have a smartphone and wants to enter the store
WP5	One store of a supermarket chain is empty while another is full
WP6	One customer has a long shopping trip
WP7	One customer only needs an item from a specific section of the store

1.2.2. Shared Phenomena

SP1	The user requests a ticket for queueing for a store
SP2	The user enters or exits a store, scanning their ticket
SP3	The system notifies the user that their time to enter the store is approaching
SP4	The user requests the booking of a shopping time
SP5	The user tells the system the duration and which items will it buy in their visit
SP6	The system suggests the user with an alternative

SP7	The manager requests a physical ticket for a customer without a smartphone
SP8	The user enters his/her data into the system

1.2.3. Goals

G1	Allow managers to regulate the influx of people inside supermarkets
G2	Save people from having to line up outside of stores for long times
G3	Allow customers to have a safer shopping experience
G4	Allow customers to have a more efficient shopping experience
G5	Allow customers who don't have a smartphone to shop in this technological era

1.3. Definitions, Acronyms, Abbreviations

1.3.1. Definitions

Ticket	The QR code generated for entering into a store
Invalid Ticket/code	A QR code which is not valid to enter a store in that moment (Because it is not the time to enter, or because it already expired)
Manager	The person in charge of a store
Store	A branch or location of a supermarket.

1.3.3. Abbreviations

WPn	World phenomenon number n
SPn	Shared phenomenon number n
Gn	Goal number n
F1	Functionality 1: *Lining up*

F2	Functionality 2: *Booking a visit*
F3	Functionality 1: *Alternatives for the customer*
Rn	Requirement number n

1.4. Revision history

Date	**Modifications**
29/11	Initial version (0.1) ● Setup of the document ● Section 1 and general characteristics of the document
23/12	First version (1.0) ● First completed version of the whole document

1.5. References

- Assignment Document: “R&DD Assignment A.Y 2020-2021”

1.6. Document Structure

- **Chapter 1:** Contains an introduction to CLup, as well as its goals and objectives. It also includes a description of the document, its purpose and main goals as a whole.
- **Chapter 2:** Contains a more detailed description of the system to be developed, through UML descriptions and state charts. This is done so the reader can have a better understanding of how is the system going to work. Also includes main description of functionalities and target users of the system.
- **Chapter 3:** Contains the interfaces that CLup is going to use, functional and non functional requirements. Functional requirements are described using use cases, sequence diagrams and scenarios.
- **Chapter 4:** Contains the alloy specification of the system and the models generated from it, as well as explanations of the code and goal of it.

- **Chapter 5:** Contains the efforts spent by each member of the group.
- **Chapter 6:** Has the reference documents used.

2. Overall Description

2.1. Product perspective

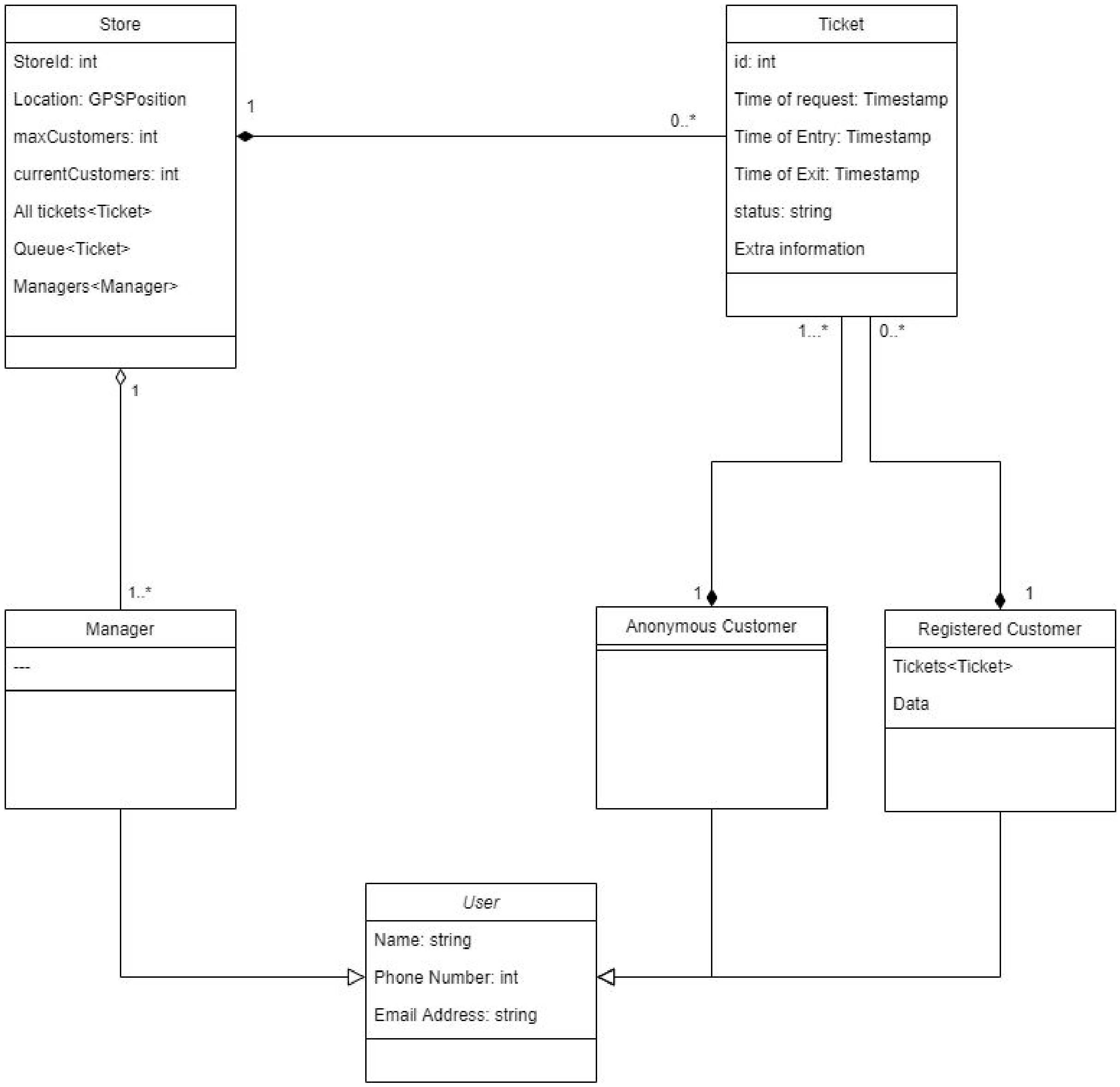
2.1.1. UML Description

The following UML diagram is a high level representation (excluding implementation details) of the system to be developed, considering all 3 main functionalities of the software. The reader must note the following characteristics

- CLup is a software which will be developed for one chain of supermarkets (with multiple stores that the users can shop in).

- In the diagram below, certain details are omitted (such as which data is saved in a registered user), which will be explained in more detail in the design document, as it is part of the implementation.

- Anonymous customers will be treated exactly as they registered counterparts, but no extra data will be saved on them, more than they mobile number as an unique identifier. In the case of an user without access to one (or who doesn't want to give his out), the manager will be able to hand out a ticket entering a generic one.
- Registered customers will only have to go through the process of registering once, and then will be able to login with their credentials.



Some details to be considered are:

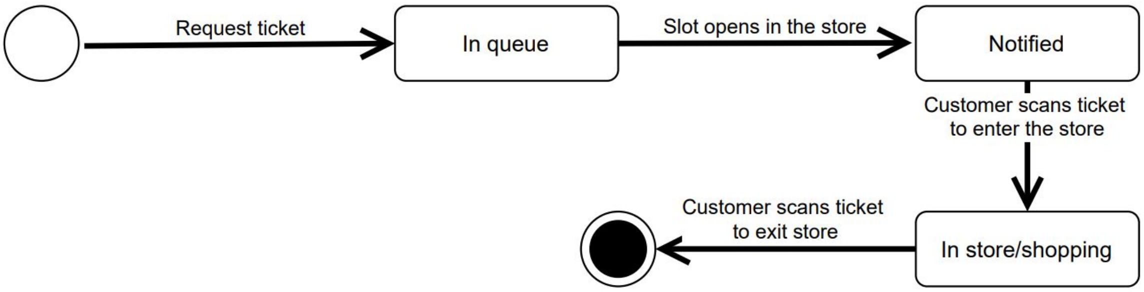
- In the Registered Customer class, *Data* will include:
    a. Favourite store
    b. Accumulated statistics on them
- In the Ticket class, *Extra information* considers the data entered when booking a visit:
    a. Supermarket categories to visit when shopping

2.1.2. State Charts

The following diagrams show the behaviour of certain parts of the system, considering how they experience state changes when conditions are given.

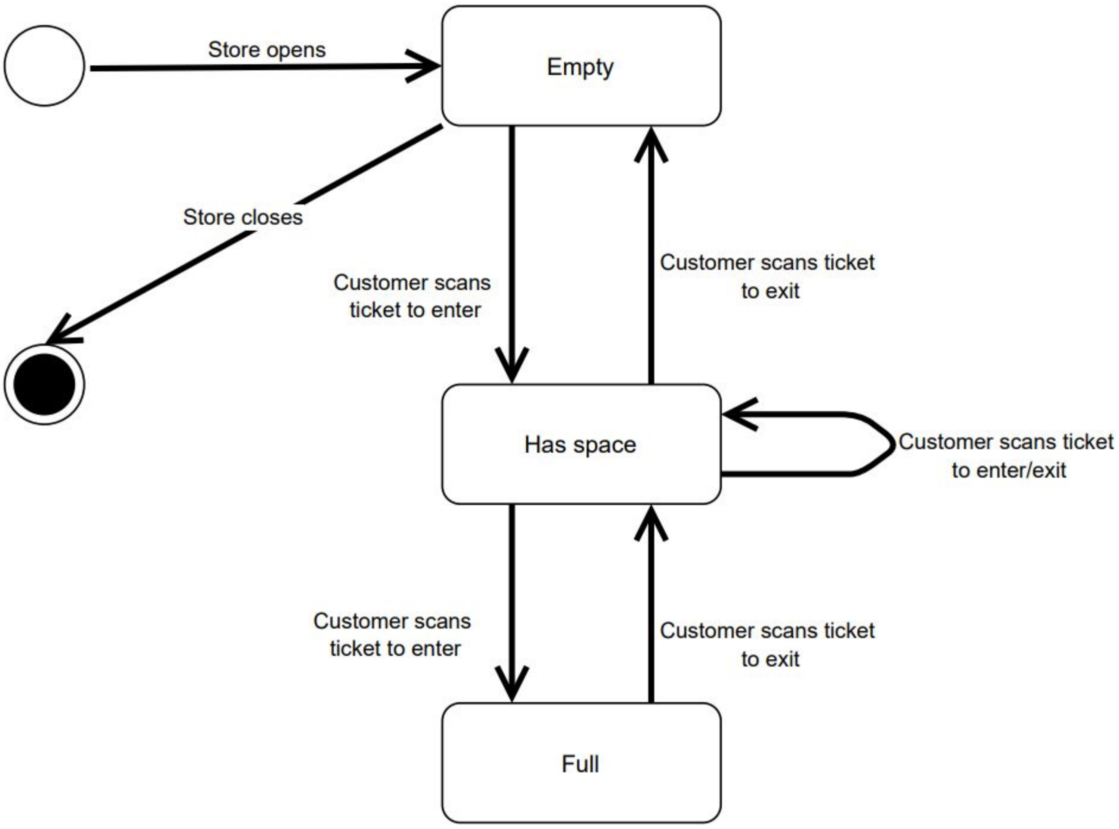
Customer state chart

The following diagram models the behaviour of the states that a customer can be in, from the moment he starts using the application to when he enters and then exits a store. Details on login are omitted.


Request ticket
In queue
Slot opens in the store
Notified
Customer scans ticket to enter the store
Customer scans ticket to exit store
In store/shopping

Store state chart

The diagram below shows how does the store will act during a day. The behaviour of the store is a key component for the management of the queue, which will only allow customers based on the space left one the store.


Store opens
Empty
Store closes
Customer scans ticket to enter
Customer scans ticket to exit
Has space
Customer scans ticket to enter/exit
Customer scans ticket to enter
Customer scans ticket to exit
Full

2.2. Product functions

In this section, the main functionalities of the system will be described in more detail, but the reader must keep in mind that many specific characteristics will still be in other parts of the document.

2.2.1. Lining up

This is the main functionality of CLup. It consists in allowing the user to get a ticket for a virtual queue for entering the supermarket, guaranteeing all of the Covid-19 safety measures regarding crowds. Asking for a ticket will give the customer a QR code, as well as an estimated time for when he has to arrive at the store. The user will have to specify the store location that he will go to, and his phone number, so an alert can be sent to him when the time is near.

In case a customer doesn't have access to a web browser, he will be able to go directly and ask for a ticket to a store manager, which will be able to print the QR code. For this, the store will have to provide any device with a supported browser as well as a printer.

For a smarter management of the queue, registered customers will only be able to line up once per day.

2.2.2. Book a visit

In case a customer wants to plan a visit to a supermarket, he will be able to book a visit. This consists in getting a ticket for entering the store in advance, so that he can specify when he is going to enter, rather than having to wait in the virtual queue. Doing this will also allow them to enter the approximate duration of their visit, as well as which sections of the store will they be in, so the system can better manage the queue for the rest of the customers.

2.2.3. Alternatives for the customer

In the case of a customer that requests a ticket, and the queue for that specific store is too long, the system will be able to suggest alternatives. This alternatives consists in:

1. Suggesting a different time for the visit. This can be a time where that particular store is emptier, given historical data, or to send a notification to the user when the queue is shorter.
2. Suggesting a different store. Based on the location of the customer, the system will be able to tell him if there are any other branches from the same chain that are emptier.

2.2.4. Manager functionalities

The system will include a section for the manager of the stores, which will contain the functionalities for scanning the QR codes from the customers, and a tool for monitoring the state of their store. The scanning of the QR codes will be done automatically by having a supported device running on the store, but during this document we will refer to the scanning as a manager functionality. In addition to the mentioned functionalities, there will be a section where managers can *edit* their stores, this is, change the number of people allowed and set up the sections where users can shop (for the *booking a visit* functionality)

2.3. User characteristics

There are two types of users that use CLup:

1. **Customer**: Someone who wants to shop in a store from the supermarket chain, and for doing so, must use CLup for requesting access to the stores. There are 2 types of customers regarding the level of authentication.

   1.1. Anonymous: Can only access functionality *lining up* of the system. From the main page of the web app, he will be able to request for a ticket just by entering his mobile phone number. A manager who gets a physical ticket for a customer will act as an anonymous one for the logic of the system, and can either enter the customer's number, or a generic one for managers. No session data will be stored here, so once the ticket is gone there is no way for checking back any info on it.

   1.2. Authenticated: Can access all of the functionalities of CLup, and his profile will contain relevant information regarding the use of the supermarkets, like average shopping time, and stores frequented, so the system can do better suggestions for him. Authenticated users will be able to check the store of their tickets for updates.

2. **Manager**: The persons in charge of a store. They have access to a different part of the system where they can scan the tickets from the customers, and monitor data on the store. The manager is also responsible for generating tickets for customers that request it at the store, for this, they will use the regular section of CLup, as explained before.

2.4. Assumptions, dependencies and constraints

2.4.1. Domain Assumptions

D1	The internet connection of the store works properly
D2	No customer enters the store without a ticket
D3	Every customer scans his/her ticket when entering/exiting the store
D4	Every customer who requests a ticket, uses it within an appropriate time of it getting ready
D5	Managers know the maximum capacity of their stores
D6	Customers don’t need to shop more than once per day at the same store

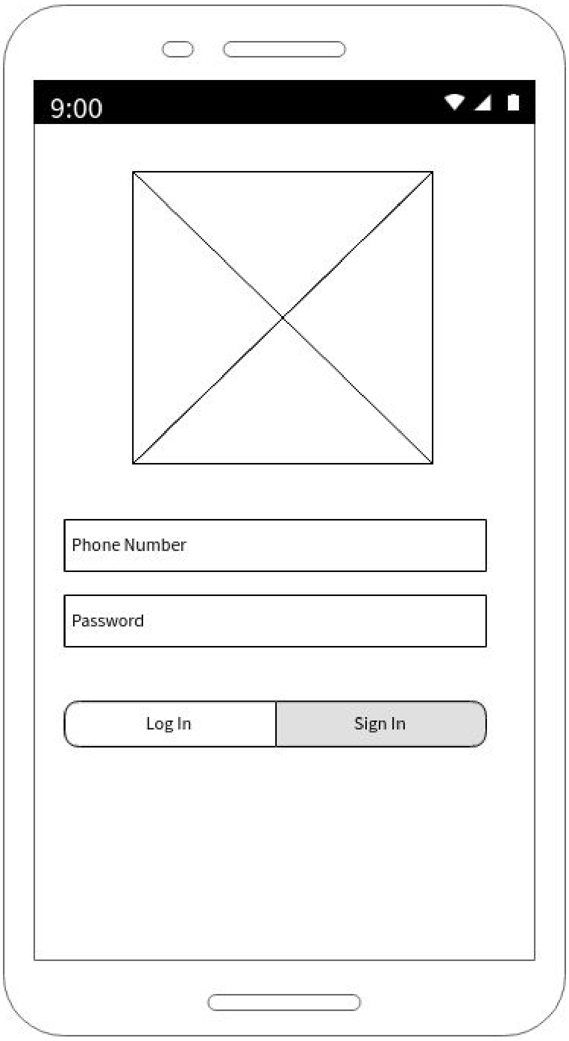
3. Specific Requirements

3.1. External Interface Requirements

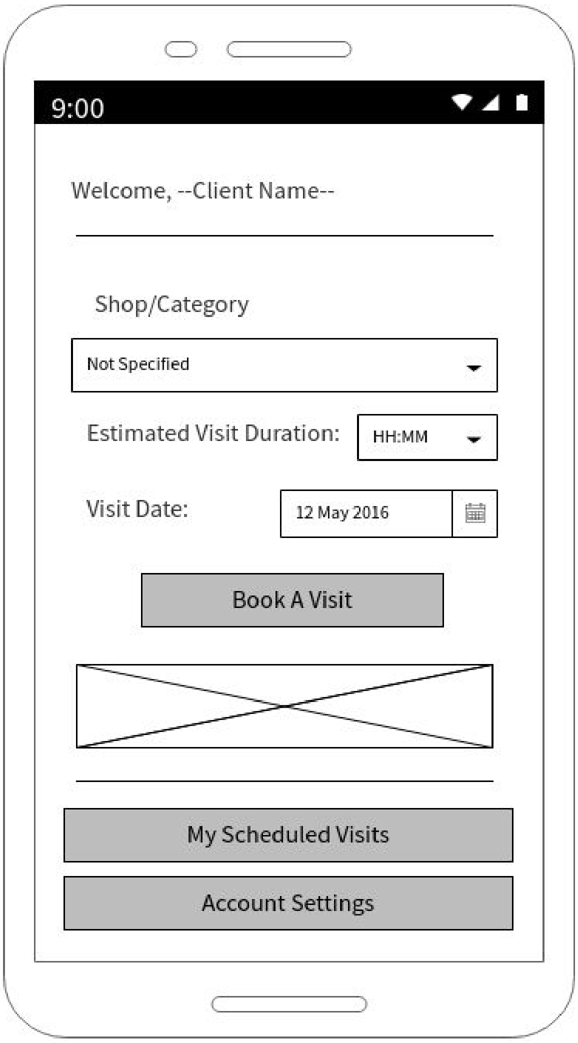
3.1.1. User Interfaces

The user interfaces are web pages, modelled in wireframes for mobile devices.

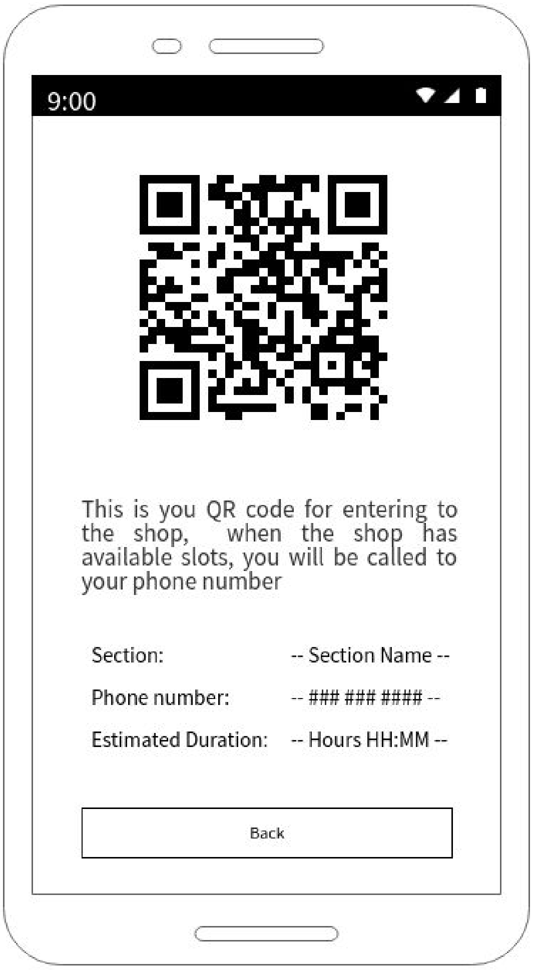
A. Customer Log In Screen



B. Booking Screen



C. QR Code Screen



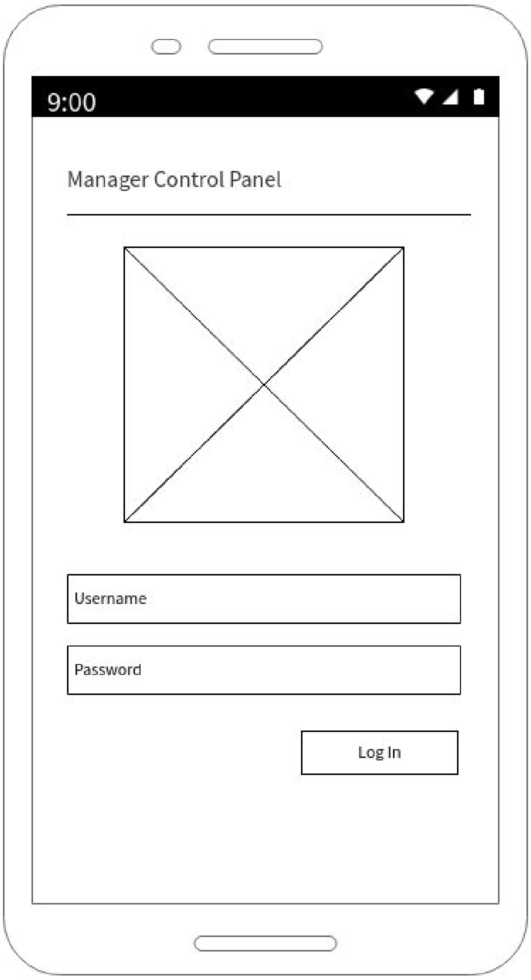
This is you QR code for entering to the shop, when the shop has available slots, you will be called to your phone number

Section: -- Section Name --

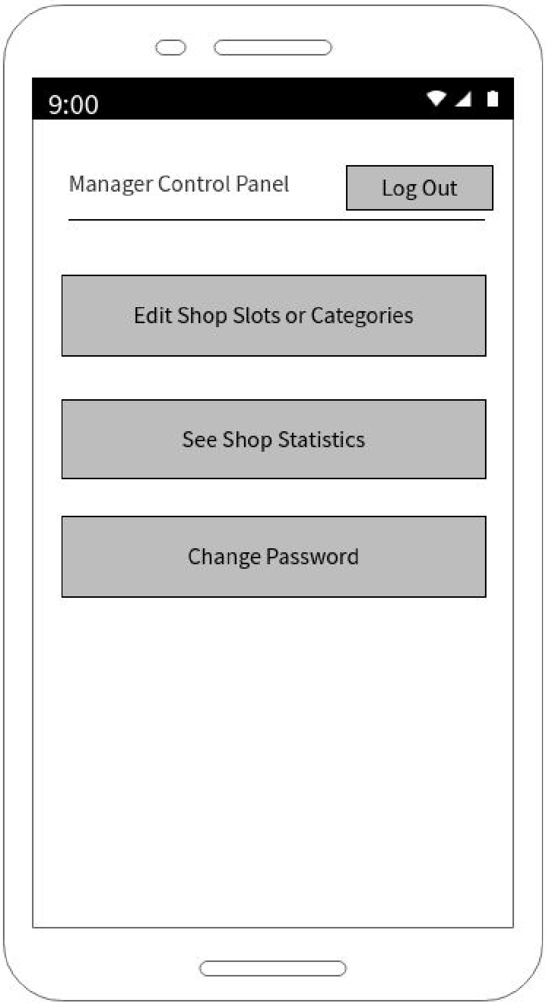
Phone number: -- ### ### #### --

Estimated Duration: -- Hours HH:MM --

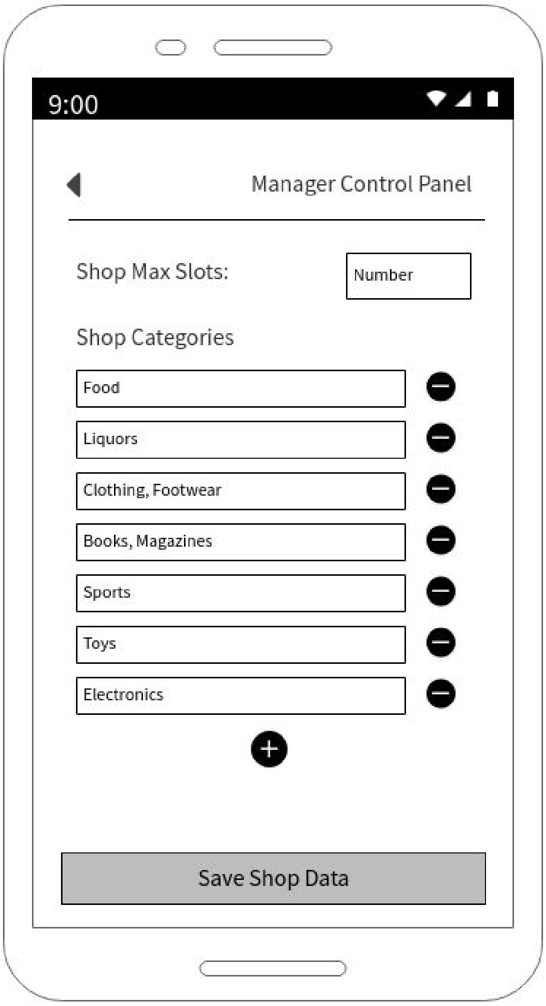
D. Manager Log In Screen



E. Admin Screen



F. Shop Edit Screen

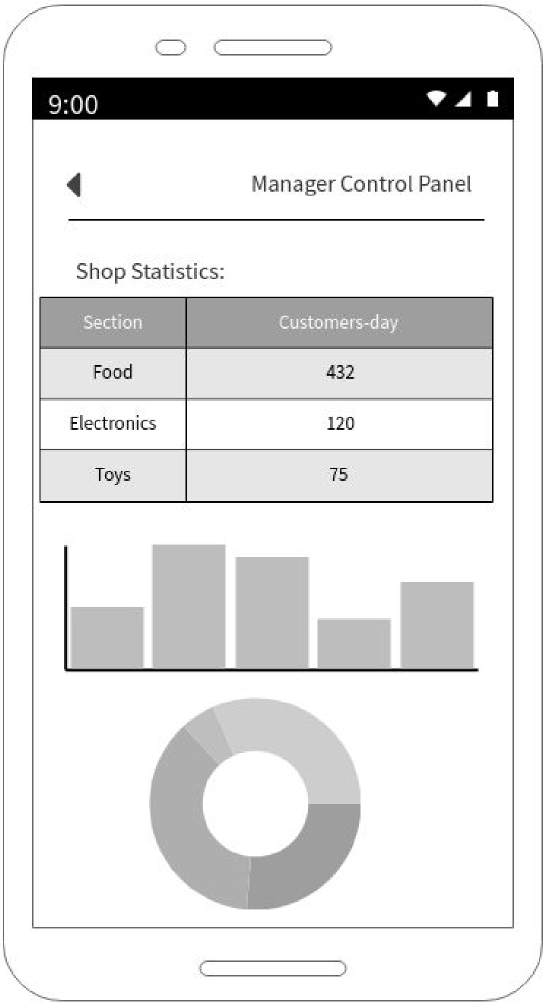


Shop Max Slots: Number

Shop Categories

- Food
- Liquors
- Clothing, Footwear
- Books, Magazines
- Sports
- Toys
- Electronics

G. Shop Statistics Screen



Shop Statistics:

Section	Customers-day
Food	432
Electronics	120
Toys	75

3.1.2. Hardware Interfaces

The system will be available as a web application, which will be able to run in any modern browser from a smartphone or a computer. For the functionalities available for the customers, namely *booking a visit, see scheduled visits* and *account settings,* displayed on the previous section, the user won't need anything extra for CLup to work, just a device with a screen in which he/she can display the QR code (or a printer for showing code at the physical store), and optionally a mobile phone where the user can be notified.

Moreover, a part of the system, which will be running in a store, will need a device with a camera, which the system will access through an *api*, for the customers' line QR code. Also, there must be a printer in the store, where the managers will be able to print the QR codes generated for the customers that do not have access to a modern device.

3.1.3. Software Interfaces

The system will use the following external interfaces:

- **SMS service**: Needed for sending the alerts to the registered users when his/her time is coming.
- **Map service:** This will be used for showing nearby stores, computing distances between users and stores and suggesting other locations.

3.1.4. Communication Interfaces

The client device will be connected to the system through an internet connection. For the customer, the internet connection will be optional after generating the QR code to be scanned. The device on the store will have to be connected to the internet at all times for it to work.

3.2. Functional Requirements

3.2.1. List of requirements

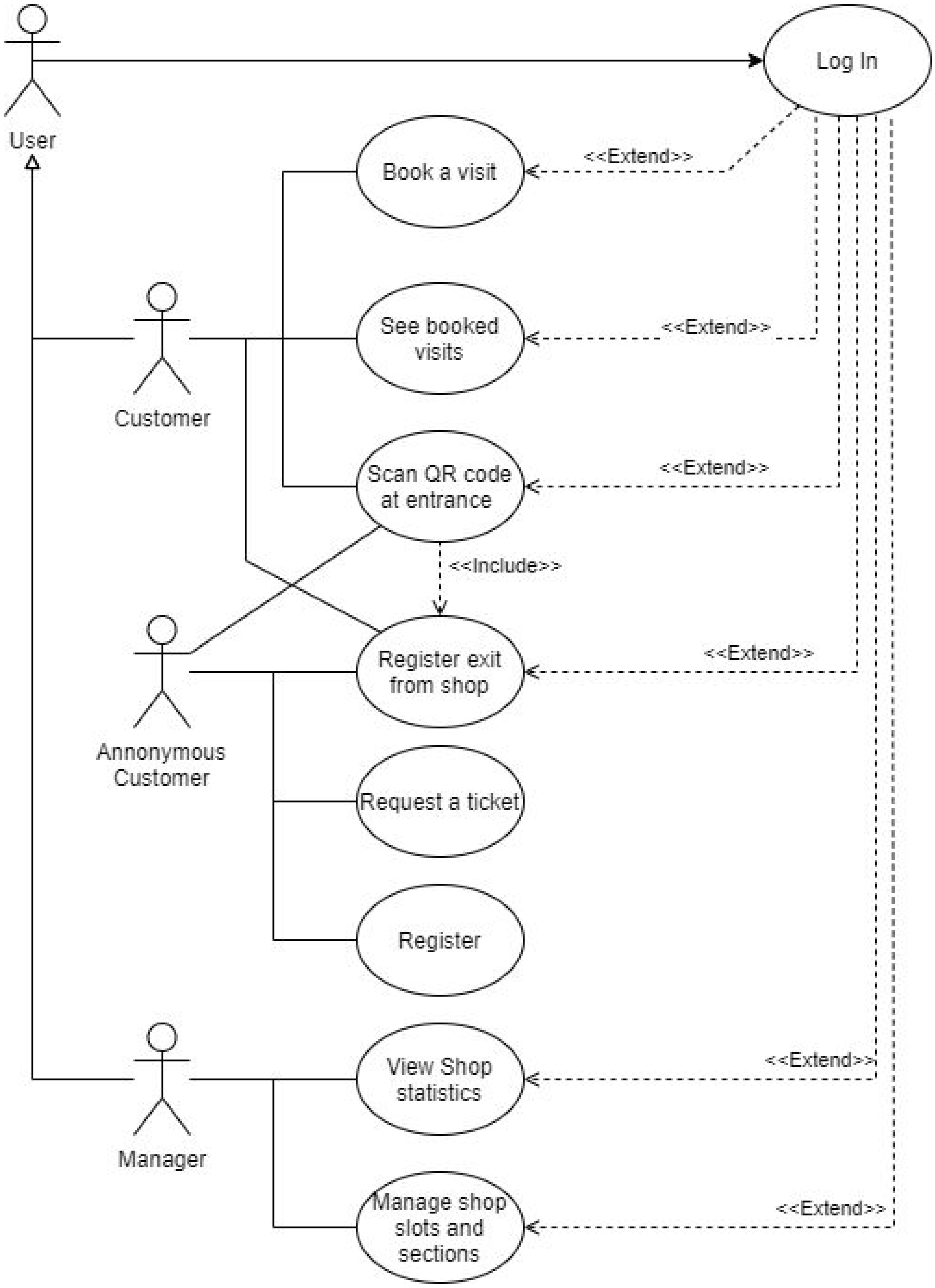
R1	An user can log in and out of the platform
R2	A registered customer can get a ticket through the platform
R2.1	A manager can produce a ticker for a customer
R2.2	An anonymous customer can get a ticket through the platform
R3	A registered customer can check the status of his code using the system
R4	The system can read QR codes from the customers
R4.1	The system can show an alert if a customer shows an invalid code
R5	A customer can get notified when their time for entering the store is near
R6	A customer can book a shopping visit
R6.1	A customer can specify how long will his/her shopping time will be
R6.2	A customer can specify to the system which items will him/her buy
R7	The system will be able to suggest an alternative store/time of shopping
R8	A manager can check how many people are currently in his/her store

3.2.2. Mapping

Goals	Domain Assumptions	Requirements
G1: Allow managers to regulate the influx of people inside supermarkets.	D1, D2, D3, D4, D5, D6	R2, R4, R6, R7, R8
G2: Save people from having to line up outside of stores for long times.	D2, D3, D4, D6	R1, R2, R4, R5, R6, R7
G3: Allow customers to have a safer shopping experience.	D2, D3, D5, D6	R2, R3, R5, R7, R8
G4: Allow customers to have a more efficient shopping experience.	D1, D2, D3, D4, D5, D6	R1, R2, R3, R4, R5, R6, R7, R8
G5: Allow customers who don't have a smartphone to shop in this technological era	D1	R2.2, R4

3.2.3. Use Cases

Use Case Diagram



Use Case Specification

A. Log In

a. Customer Login

ACTORS	Customer
GOALS	G2, G3
INPUT CONDITIONS	Customer already in the home web page
EVENTS FLOW	1. The user inserts his credentials on phone number and password fields. 2. The user clicks on the “Log in” button. 3. The system authenticates the credentials and redirects the user to the booking screen
OUTPUT CONDITIONS	Customer successfully logged into the systems and redirected to booking screen
EXCEPTIONS	- Not valid phone number - Incorrect password Exceptions handled on Step 3, and in case they occur, use case goes back to step 1

b. Manager Login

ACTORS	Manager
GOALS	G1
INPUT CONDITIONS	Manager already in the admin login screen
EVENTS FLOW	1. The user inserts his credentials on username and password fields. 2. The user clicks on the “Log in” button. 3. The system authenticates the credentials and redirects the manager to its admin screen
OUTPUT CONDITIONS	Customer successfully logged into the systems and redirected to booking page
EXCEPTIONS	- Incorrect username - Incorrect password Exceptions handled on Step 3, and in case they occur, use case goes back to step 1

B. Customer Register

ACTORS	Anonymous Customer
GOALS	G2, G3, G4
INPUT CONDITIONS	User already in the home web page
EVENTS FLOW	1. The user clicks on the “Sign In” button. 2. The user inserts his phone number and password in the fields. 3. The user clicks on the “Register” button. 4. The system stores the credentials and redirects the client to its booking screen.
OUTPUT CONDITIONS	New customer is successfully created and is redirected to booking screen
EXCEPTIONS	- Existent user with input phone number - Invalid phone number - Invalid password Exceptions are handled on Step 3, and in case they occur, the use case does not continue and screen displays error messages, use case resume when the user corrects the given data.

C. Book A Visit

ACTORS	Customer
GOALS	G4
INPUT CONDITIONS	Customer Logged in and correctly authenticated, screen displaying Booking Screen
EVENTS FLOW	1. The user select a shopping category or if he doesn’t want to specify leave it on “Not Specified” 2. The user selects an estimated duration for his visit using the selector button. 3. The user selects a date for its visit using the calendar button. 4. The user clicks on the “Book a Visit” button and the system generates a QR code for its visit on the selected date.
OUTPUT CONDITIONS	A booking has been successfully booked, a QR code is generated and user is redirected to QR code page

EXCEPTIONS	- Visit duration not selected - Visit date not selected The exceptions are handled by the UI webpage constraints by locking the “Book a Visit” button until fields are completed - Not available visits for the date The exception is handled when the “book a visit” button is pressed, there will be a message window displayed with options to the user, like scheduling at other available time/day, or make the booking with a high wait to enter.

D. See Booked Visits

ACTORS	Customer
GOALS	G3, G4
INPUT CONDITIONS	Customer Logged in and correctly authenticated, screen displaying Booking Screen
EVENTS FLOW	1. The user click on My Scheduled Visits 2. All the scheduled visits with date, shopping section and duration are displayed on screen. 3. The user clicks on a visit 4. The user is redirected to that visit’s QR code
OUTPUT CONDITIONS	User redirected to QR Code screen of a specific visit
EXCEPTIONS	- There are not scheduled visits Exception handled on step 2, the use case is finished and user is sent back to Booking Screen with a information screen displaying “You have no booked visits”

E. Scan QR Code at Entrance

ACTORS	Customer / Anonymous User
GOALS	G2, G3, G5
,INPUT CONDITIONS	User in front of the QR scanner device with a QR code
EVENTS FLOW	1. The user places its QR code on the scanner. 2. The system validates the shop occupancy and adds the customer to the occupancy count. 3. The user enters the shop
OUTPUT CONDITIONS	User validated inside the shop
EXCEPTIONS	- Wrong QR code read Exception handled by step 2, user is told to put the QR code on the scanner again. - Invalid QR code Exception handled by step 2, user is told to return on the date of QR code validity - Shop at full capacity. Exception handled by step 2, customer is told to wait, until the shop has an empty slot.

F. Register Shop Exit

ACTORS	Customer / Anonymous Customer
GOALS	G2, G3, G4
INPUT CONDITIONS	User in front of the QR Scanner device with a valid QR code
EVENTS FLOW	1. The user places its QR code on the scanner. 2. The system stores the shopping time and removes the customer to the occupancy count. 3. The user exits the shop
OUTPUT CONDITIONS	User out of the shop and its time spent is saved on the system

EXCEPTIONS	- Wrong QR code read Exception handled by step 2, user is told to put the QR code on the scanner again.

G. Request Ticket

ACTORS	Anonymous Customer
GOALS	G5
INPUT CONDITIONS	Customer in front of the shop
EVENTS FLOW	1. The user presses the device button. 2. The user gets a printed QR code with an estimated slot availability time.
OUTPUT CONDITIONS	Customer with a system's valid QR code
EXCEPTIONS	No possible software exceptions.

H. View Shop Statistics

ACTORS	Manager
GOALS	G2
INPUT CONDITIONS	Manager Logged in and correctly authenticated, screen displaying Admin Screen
EVENTS FLOW	1. Manager click on "View Statistics" button 2. Shop data statistics is displayed in sections, with tables and graphics 3. Manager clicks on the "back" button
OUTPUT CONDITIONS	Manager on the admin screen
EXCEPTIONS	- Wrong QR code read Exception handled by step 2, user is told to put the QR code on the scanner again.

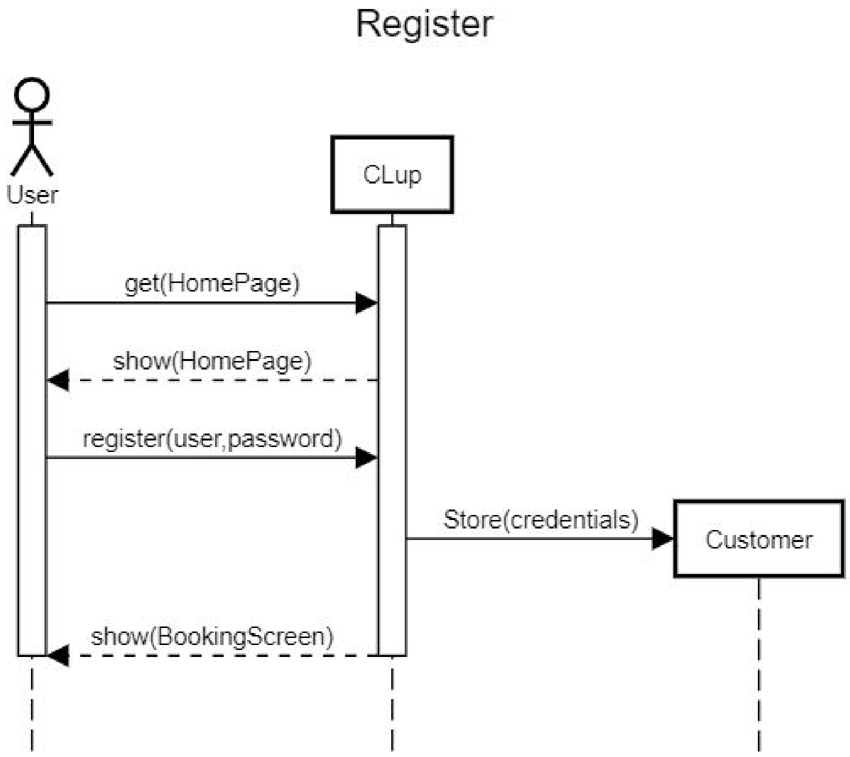
I. Manage Shop Slots And Sections

ACTORS	Manager
GOALS	G1

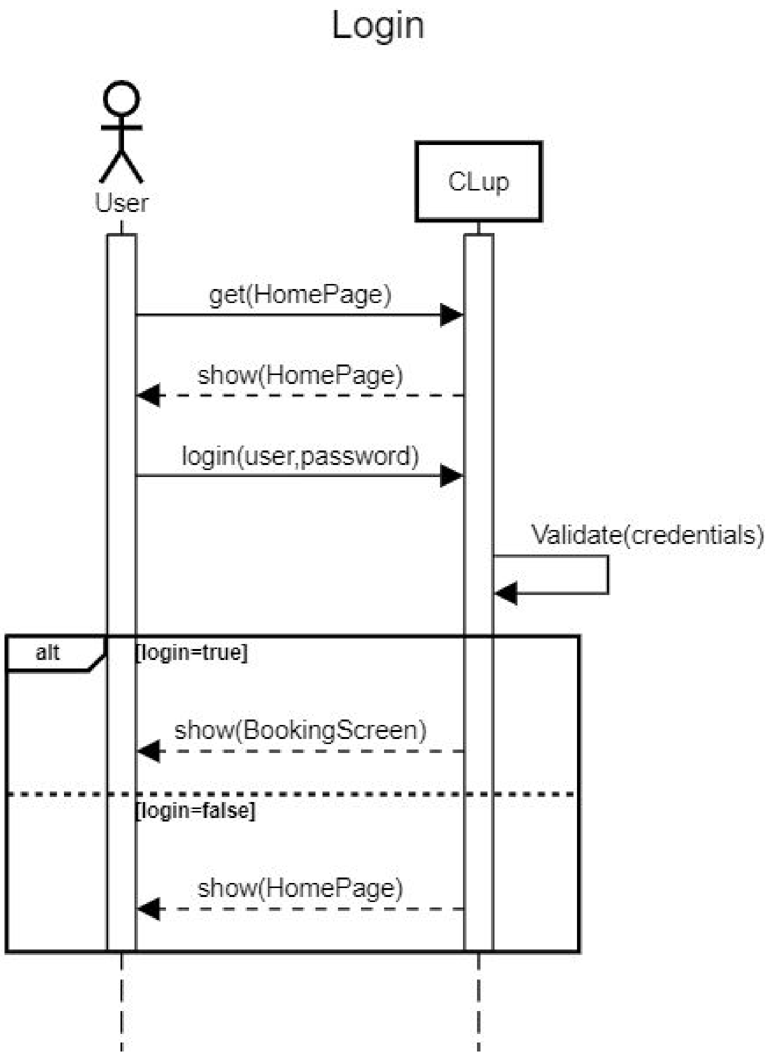
INPUT CONDITIONS	Manager Logged in and correctly authenticated, screen displaying Admin Screen
EVENTS FLOW	1. Manager click on “Edit Shop” button 2. Manager fills the form with a new slot number. 3. Manager adds or removes shop sections in the list. 4. Manager clicks on the “save changes” button
OUTPUT CONDITIONS	Shop data updated
EXCEPTIONS	- No sections added Handled in step 4, there are no sections on the shop - Invalid slots Handled in step 4, A not valid number was input on the field and must be corrected

3.2.4. Sequence Diagrams

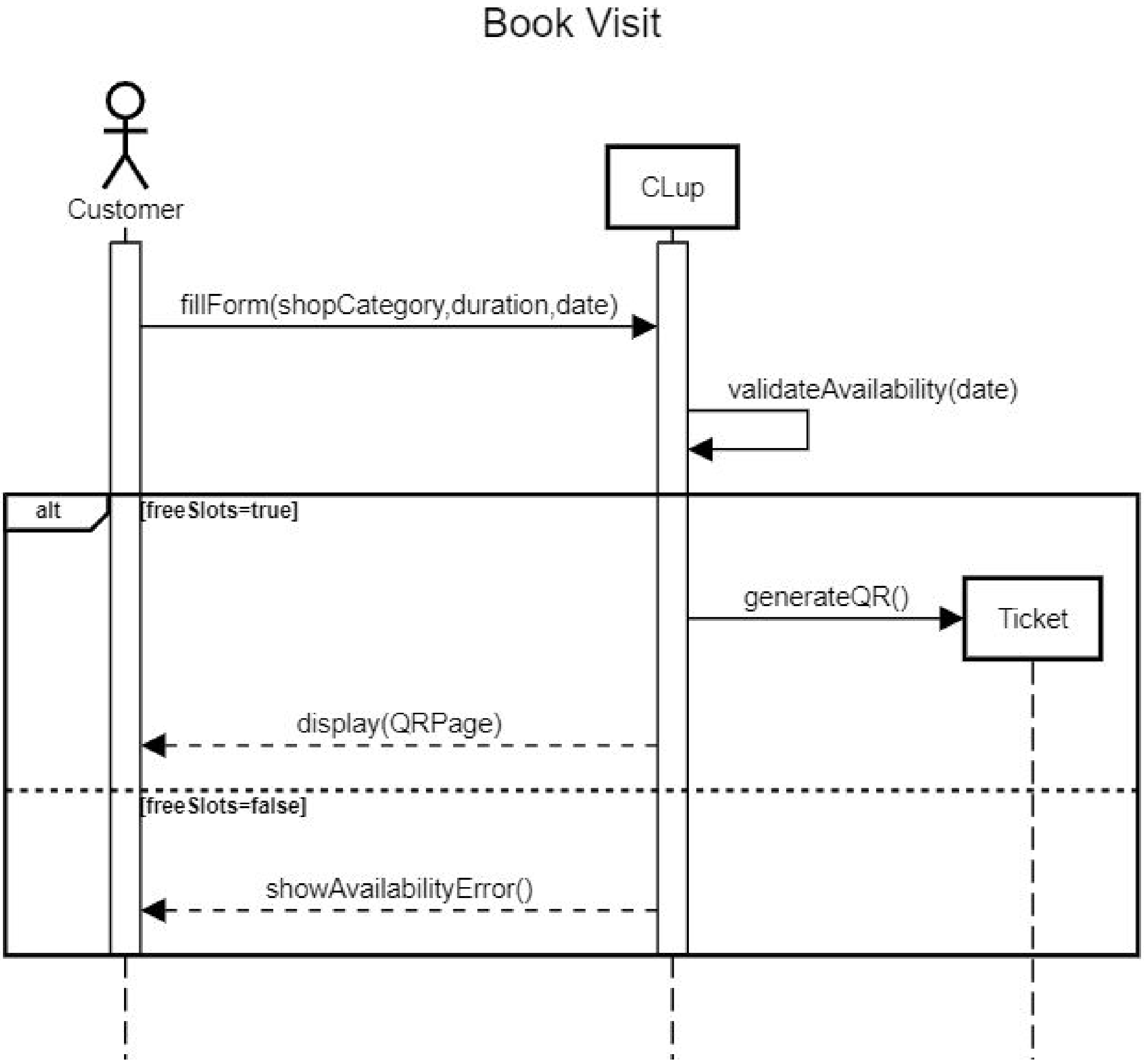
A. Register



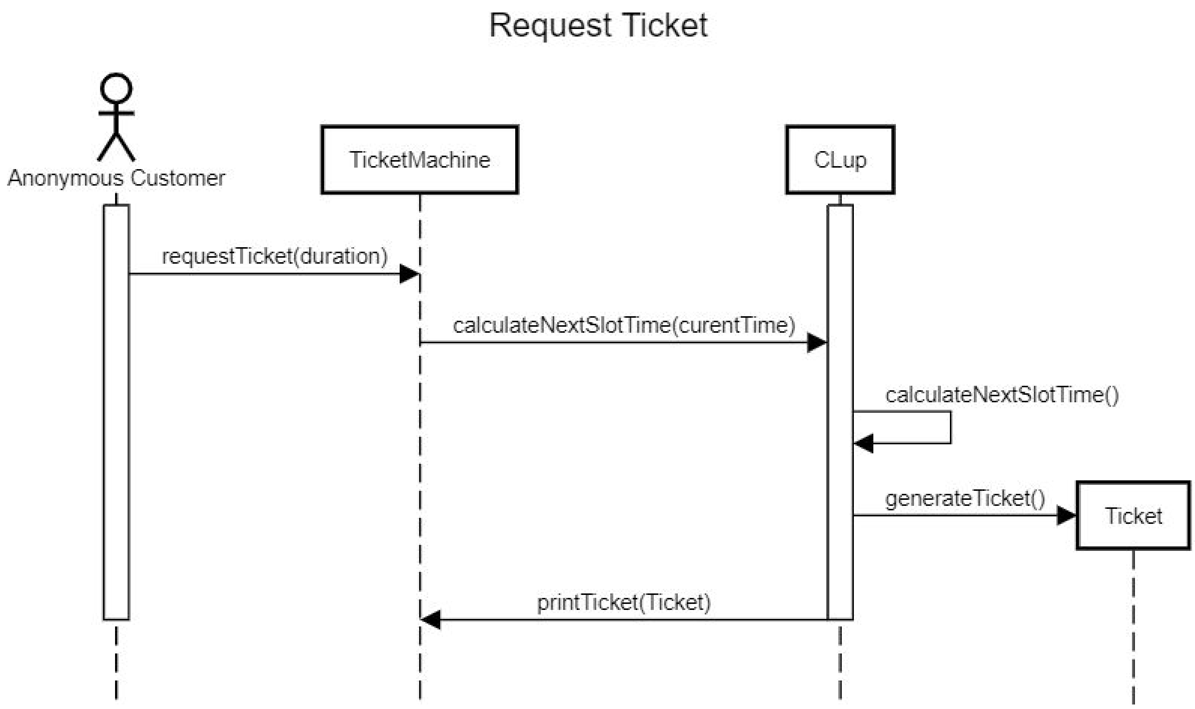
B. Login


Login
User
CLup
get(HomePage)
show(HomePage)
login(user,password)
Validate(credentials)
alt
[login=true]
show(BookingScreen)
[login=false]
show(HomePage)

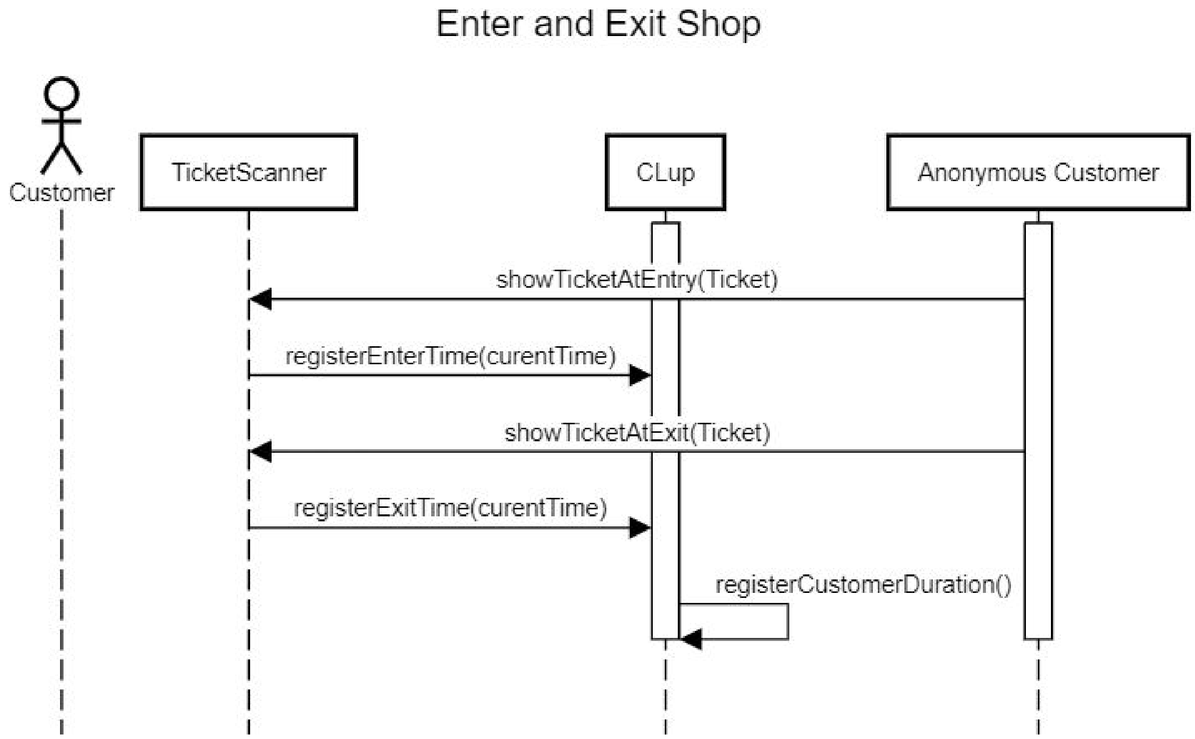
C. Book Visit


Book Visit
Customer
CLup
fillForm(shopCategory,duration,date)
validateAvailability(date)
alt
[freeSlots=true]
generateQR()
Ticket
display(QRPage)
[freeSlots=false]
showAvailabilityError()

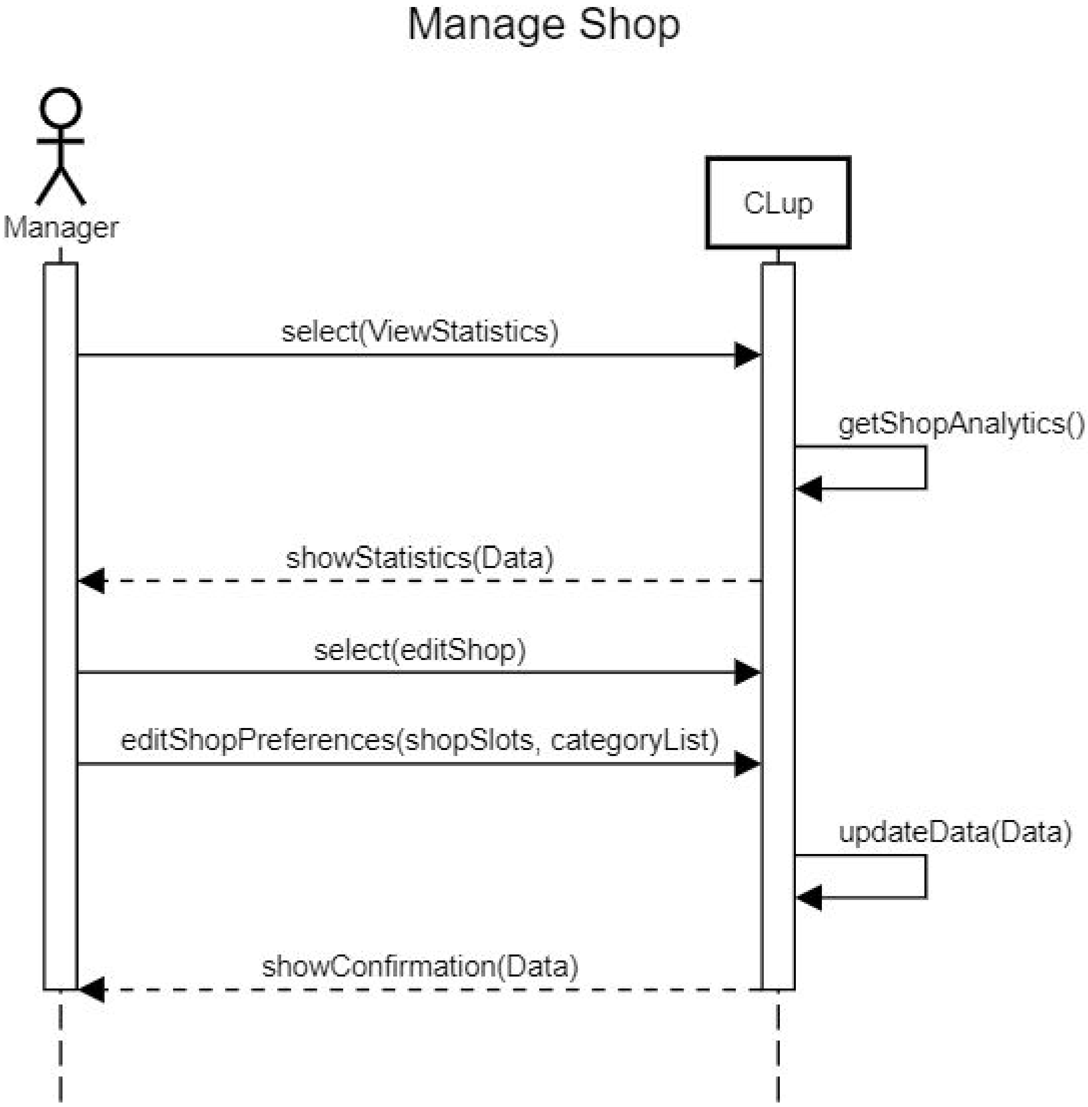
D. Request Ticket



E. Entry and Exit Shop


Enter and Exit Shop
Customer
TicketScanner
CLup
Anonymous Customer
showTicketAtEntry(Ticket)
registerEnterTime(curentTime)
showTicketAtExit(Ticket)
registerExitTime(curentTime)
registerCustomerDuration()

F. Manage Shop



3.2.5. Scenarios

Scenario 1:

A customer wants to buy some sports products on a nearby store, but with pandemic restrictions the shop has a virtual waiting line, so he uses its smartphone browser and goes to the shop webpage and clicks on the CLup virtual line where he already has a account, he selects the sports shop category for an estimated duration of 1 hour, for the same day, then he books the visit and receives the QR code, after an hour he receives a call and an automated service says that his slot in shop is open, he goes to shop, scans his code and enter. Before 45 minutes he buys some sport items and registers his code at the shop exit.

Scenario 2:

A customer wants to buy food for a meeting with friends, but he doesn't use a smartphone or computer so he goes to the grocery store and is told to get a line QR ticket, he gets to the ticket machine and presses the screen to get a ticket, the tickets displays that the estimated time for entry is 20 minutes, he waits around the shop for that time and then he enters the store to buy his items, at the exit he has lost the ticket, so he just marks his exit out of the shop on the scanner device.

Scenario 3:

The store manager wants to know how is the store going, and open a new section to expand the shop slots, he opens the admin webpage, log in, and first opens the shop statistics, he see that his shop has a big concentration on entertainment section, so he decides to split the section into sports, movies, toys and books, he goes back to the admin menu and clicks the shop edit button, removes the Entertainment section, and adds the other ones, decides to not add more slots, and saves all the configuration, and then logs out CLup.

Scenario 4:

A customer who lives in a very central part of the city (ie, close to many stores) wants to do some grocery shopping, so he enters CLup and decides to line up for a particular store. The system tells him that the queue wait time will be around 50 minutes but there is another branch nearby which only has a 10 minute wait time. The customer then lines up for this store and proceeds with his shopping as usual.

Scenario 5:

Some house owner is planning christmas dinner and realizes he will have to go shopping to have everything ready for then. He knows that this time of the year stores are very busy and he decides to try CLup to book a time at the store, which he heard about from some friends. He loads the page and registers, then he books a visit for some later in the week when he has time. As he knows exactly what he will be cooking, he tells the system which parts of the store he will be visiting. Later, the day comes and he gets notified in his mobile phone. He then proceeds to shop, scanning his ticket as usual with CLup.

3.3. Performance Requirements

The system should be able to handle the connection of at least 100.000 individuals at the same time, and respond to requests for creating and scanning QR codes in less than 5 seconds.

3.4. Design Constraints

3.4.1. Standards compliance

The system will make use of SMS and Automated Phone Calls using wireless communication networks accomplishing the Global System for Mobile Communications (GSM) series of standards.

Regarding the privacy of data, The General Data Protection Regulation (EU) (GDPR) is a regulation in EU law on data protection and privacy in the European Union (EU) and the European Economic Area (EEA), the system must provide principles and safeguards to protect all the user data.

3.4.2. Hardware limitations

- Web App
  - Modern browser
- Shop scanner
  - Camera
  - Screen
  - Wired internet connection

3.5. Software System Attributes

3.5.1. Reliability

The system must have a reliability of 80%, it can support non critical failures, and programmed maintenance on low traffic times (between 22:00 and 6:00), in case of critical failures, the system should be able to recover its state in less than 5 minutes

3.5.2. Availability

The system should be available to users and shop personnel 24/7, the system can support interruptions on the service on low traffic times (between 22:00 and 6:00).

3.5.3. Security

The system will store user data as name, number phone and shopping preferences. Even if the data flow between users and system is not private, security of the internal data is a concern.

3.5.4. Maintainability

In case of error or critical upgrade, when the system has a component that must be critically fixed or upgraded, it has a 95% chance that the component is fixed or upgraded within 24 hours.

3.5.5. Portability

As the system is working on a Web Application, any device that has an internet connection and is capable of running a modern browser, can access the system independently of the operative system or software specifications.

3.5.6. Scalability

As the system is intended for virtual lining, it should be capable of high scalability for being used by multiple shops, malls and other centres with lining entry, at the same time and also can grow with compatibility on mobile apps for a better user experience.

4. Formal Analysis using Alloy

4.1. Signatures

```
open util/integer

enum Name{ john, nick, paul, sam, mike, mark, nike}
enum State { Milan, Pavia, Centrale, Dergano }

abstract sig Position{}
abstract sig TicketStatus{}

//status of the tickets after scanning
one sig Scanned extends TicketStatus{}

//status of the ticket after booking
one sig Booked extends TicketStatus{}

//customer related data with unique id
sig Customer{
      id: one Int,
      name: disj Name,
      ticket: one Ticket
}
{
      id > 0
}

// float value for precision while  calculating latitude and longitude
sig Float{
      beforePoint: one Int,
      afterPoint: one Int
}
{
      afterPoint>0
}
```

```
//gps position to know location of store and calculate distance
sig GPSPosition extends Position{
      latitude: one Float,
      longitude: one Float
}
{
      latitude.beforePoint < 90 and latitude.beforePoint > -90
      longitude.beforePoint < 180 and longitude.beforePoint > -180
}

//store related general information
sig Store {
      location: one GPSPosition,
      city: disj State,
      state: disj State,
      currentCustomers: one Int,
      maxCust: one Int,
}

//time to store time of booking
sig Time{
      hours: one Int,
      minutes: one Int,
      seconds: one Int
}
{
      hours>=0
      minutes>0
      seconds>0
}

//ticket related information to store time of booking and status of ticket
sig Ticket {
      id: one Int,
      timeOfRequest: one Time,
      statusOfTicket: one TicketStatus
}
{
      id> 0
}

// database to store the customer and ticket data mapped
sig Database {
      row: Customer -> one Ticket
}
```

4.2. Facts

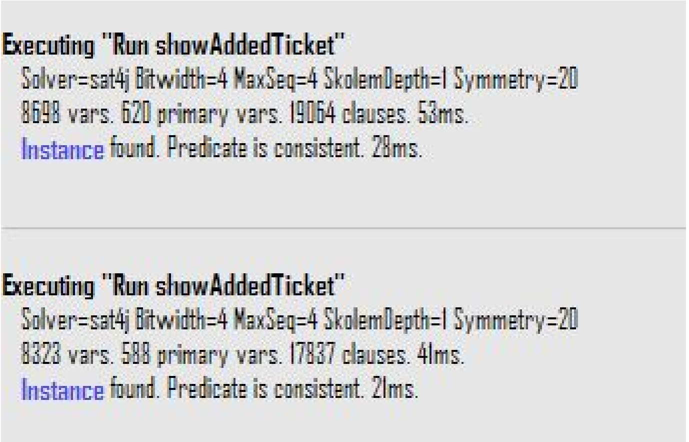
```
//every customer has ticket which is unique from all other tickets in
database
fact uniqueTicket{
      all disj I, I': Ticket | I.id != I'.id
}

//every customer has id which is unique from all other customers in
database
fact uniqueIdPerCustomer{
      all disj c1,c2: Customer | c1.id != c2.id
}

// number of customers is equal to number of tickets (as customers are
only able to line up once per day
// no two customer ticket ids should match
fact {
      #Customer = #Ticket
      all disj c,c': Customer | c.ticket.id != c'.ticket.id
      all disj c,c': Customer, d,d': Database | c.(d.row)  != c'.(d'.row)
      and c.(d.row) = c.ticket
}
```

4.3. Dynamic Model

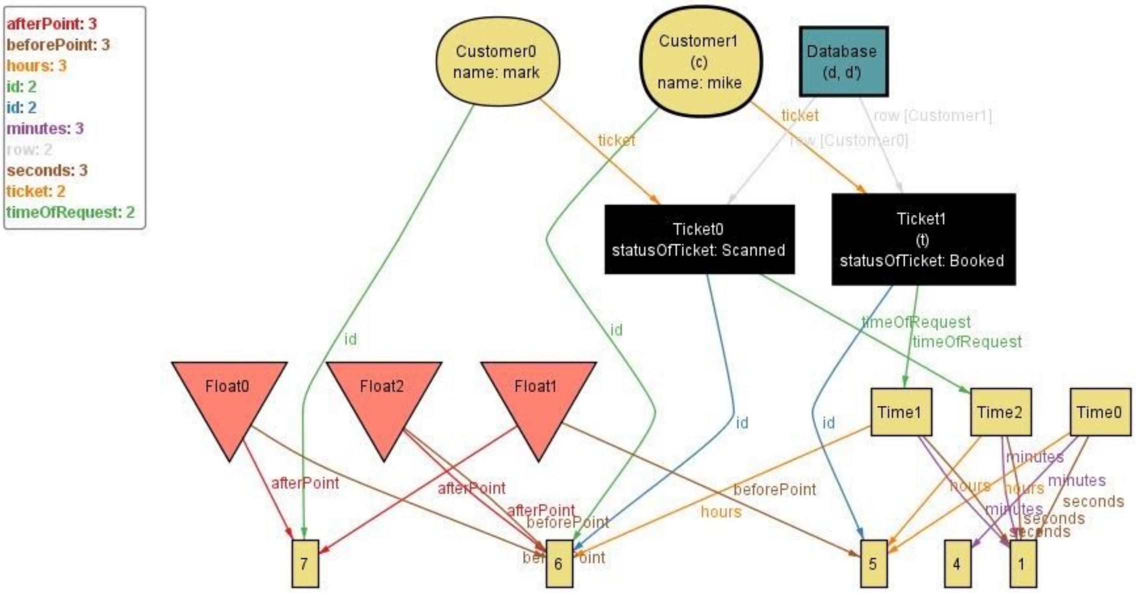
```
//bookticket that add new row to database and update status of new ticket
to booked
pred bookTicket [d, d': Database, c: Customer, t: Ticket] {
      d'.row = d.row + c-> t
      all disj d: Database | c.(d.row) = c.ticket implies
      c.ticket.statusOfTicket = Booked
}

// delTicket deletes row from the database which contains info related to
that customer
pred delTicket [d,d': Database, c: Customer] {
      d'.row = d.row - c->Ticket
}

// showAddedTicket books ticket and shows the booked ticket data
pred showAddedTicket [d,d': Database, c: Customer, t: Ticket] {
      t.statusOfTicket =  Booked
      bookTicket[d, d', c, t]
      #Customer.(d'.row) > 1
}

// after scanning ticket the status of ticket  in database should change
to scanned
pred ScanTicket[c:Customer]{

all disj d: Database | c.(d.row) = c.ticket implies
c.ticket.statusOfTicket = Scanned
}

//bookticket and delete ticket then database should be consistent from
before booking and after deleting ticket
assert delUndoesAddedTicket {
      all d,d',d'': Database, c: Customer, t: Ticket |
      no c.(d.row) and bookTicket[d,d',c,t] and delTicket[d',d'',c]
      implies
      d.row = d''.row}
//show whole world
Pred show {}
```

4.4. Results

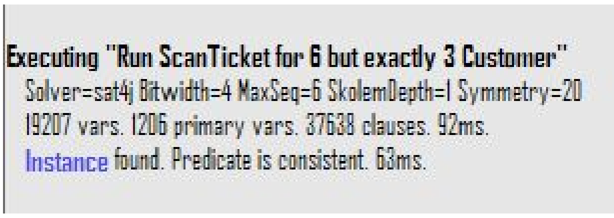
run showAddedTicket



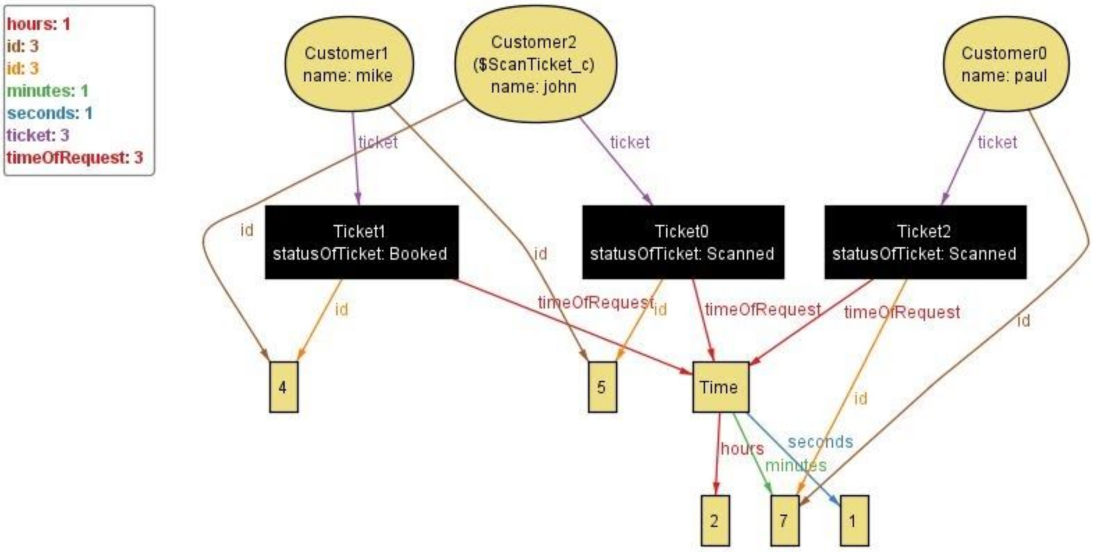
```
Executing "Run showAddedTicket"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   8698 vars. 620 primary vars. 19064 clauses. 53ms.
   Instance found. Predicate is consistent. 28ms.

Executing "Run showAddedTicket"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   8323 vars. 588 primary vars. 17837 clauses. 41ms.
   Instance found. Predicate is consistent. 21ms.
```



run ScanTicket for 6 but exactly 3 Customer

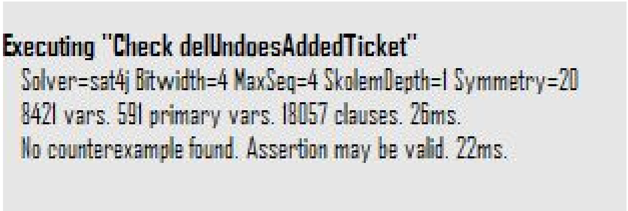


```
Executing "Run ScanTicket for 6 but exactly 3 Customer"
   Solver=sat4j Bitwidth=4 MaxSeq=6 SkolemDepth=1 Symmetry=20
   19207 vars. 1206 primary vars. 37638 clauses. 92ms.
   Instance found. Predicate is consistent. 63ms.
```



4.4.1. Proof of consistency

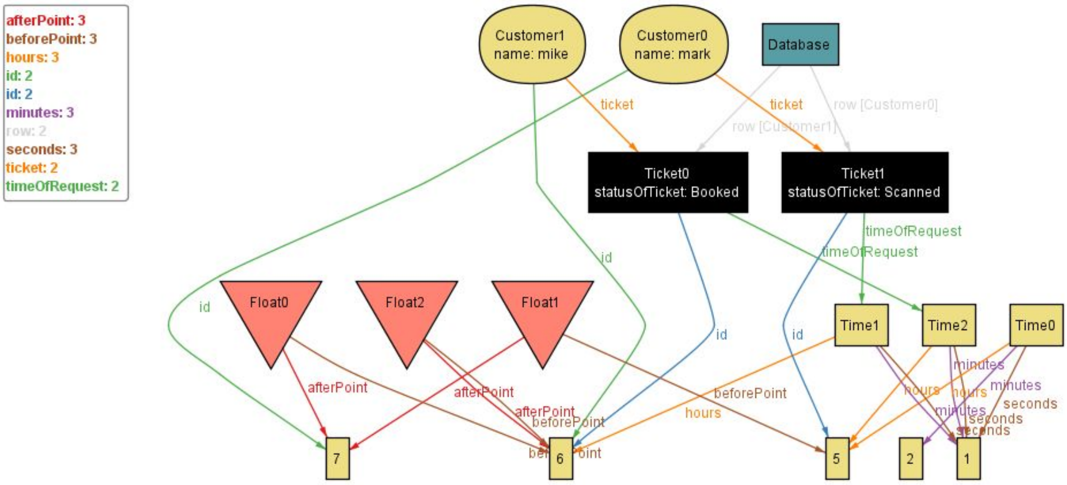
check delUndoesAddedTicket



```
Executing "Check delUndoesAddedTicket"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   8421 vars. 591 primary vars. 18057 clauses. 26ms.
   No counterexample found. Assertion may be valid. 22ms.
```

4.4.2. Generated World

run show



5. Effort Spent

5.1. Hours of Work

Topic	**Hours**
Initial setup of document	3hr
Wireframes design	2hr
Section 1 - Adding information	3hr
Section 2 - Description of the system	4hr
Section 2 - UML diagrams	2hr
Section 3 - Non functional requirements	1hr
Section 3 - Use Case Diagram and descriptions	3hr
Section 3 - Sequence Diagrams	2hr
Section 3 - Mappings and descriptions	4hr
Section 4 - Alloy code	6hr
Review of document	2hr

6. References

GSM Standards:

https://www.gsma.com/aboutus/open-standards-specifications

GDPR Law:

https://gdpr-info.eu/

Diagrams

https://draw.io

Alloy specification

https://alloytools.org/