



POLITECNICO DI MILANO 1863

SOFTWARE ENGINEERING 2 PROJECT

REQUIREMENT ANALYSIS AND SPECIFICATION DOCUMENT (RASD)

---

CLup

Version 1.1

---

*Authors*

SAMAN FEKRI  
PARNIYA SAEEDZADEH

*Supervisors*

Prof. ELISABETTA DI NITTO  
Prof. MATTEO ROSSI

Click [here](#) for Github repository

December 23, 2020

**Copyright:** Copyright © 2020, Saman Fekri - Parniya Saeedzadeh – All rights reserved

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose	3
1.2	Scope	3
1.2.1	World phenomena	3
1.2.2	Shared phenomena	4
1.2.3	Goals	4
1.3	Definitions, Acronyms, Abbreviations	4
1.3.1	Definitions	4
1.3.2	Acronyms	5
1.3.3	Abbreviations	5
1.4	Revision history	5
1.5	Reference Documents	5
1.6	Document Structure	5
<b>2</b>	<b>Overall Description</b>	<b>7</b>
2.1	Product perspective	7
2.1.1	UML Description	7
2.1.2	State Diagrams	8
2.2	Product functions	9
2.3	User characteristic	10
2.4	Assumptions, dependencies and constraints	10
<b>3</b>	<b>Specific Requirements</b>	<b>12</b>
3.1	External Interfaces	12
3.1.1	User Interface	12
3.1.2	Hardware Interfaces	15
3.1.3	Software Interfaces	15
3.2	Functional Requirements	16
3.2.1	List of requirements	16
3.2.2	Mapping	17
3.2.3	Use cases	20
3.2.4	Sequence Diagrams	29
3.2.5	Scenarios	36
3.3	Performance Requirements	37
3.4	Design Constraints - Non-Functional Requirements	37
3.4.1	Standards compliance	37
3.4.2	Hardware limitations	37
3.4.3	Any other constraint	37
3.5	Software system attributes	37
3.5.1	Simple User Interface	37
3.5.2	Reliability	37
3.5.3	Availability	38
3.5.4	Security	38
3.5.5	Maintainability	38
3.5.6	Portability	38
3.5.7	Scalability	38
<b>4</b>	<b>Formal Analysis Using Alloy</b>	<b>39</b>
4.1	Alloy Code	39
4.2	Meta Model	45
4.3	Result for predictions	46
<b>5</b>	<b>Effort spent</b>	<b>54</b>
<b>6</b>	<b>References</b>	<b>55</b>

## List of Figures

1	High level Class Diagram	7
2	User book a visit or insert to lineup queue	8
3	User get ticket from ticket machine	8
4	Manager create a shop	8
5	User shopping	9
6	Ticket states	9
7	Icon Mockup	12
8	Mockup	13
9	Mockup	14
10	Mockup	15
11	Use case diagram	28
12	Sequence diagram for register	29
13	Sequence diagram for login	29
14	Sequence diagram for create a shop	30
15	Sequence diagram for update shop	31
16	Sequence diagram for add to line-up	31
17	Sequence diagram for book a visit	32
18	Sequence diagram for see shop lists	32
19	Sequence diagram for generate ticket	33
20	Sequence diagram for notify approach	33
21	Sequence diagram for notify call user turn	33
22	Sequence diagram for scan for entrance	34
23	Sequence diagram for user checkout	34
24	Sequence diagram for clerk add offline user in queue	35
25	Meta model	45

# 1 Introduction

## 1.1 Purpose

This document focuses on Requirements Analysis and Specification Document (RASD) and contains the description of the main goals, the domain and its representation through some models, the analysis of the scenario with the uses cases that describe them, the list of the most important requirements and specifications that characterize the development of the software described below.

It also includes the research about the interfaces, functional and non-functional requirements and the attributes that distinguish the quality of the system.

This document has the purpose to guide the developer in the realization of the software called CLup, a Customers Line-up application.

Finally, to understand better the development of the document, it contains the history that describes how it is made, with the references used and the description of its structure.

## 1.2 Scope

The software wants give them the possibility to line up for a super market and notify them when it's their turn.

- **basic function:** Allow application to retrieve a number for users to be in the line waiting for their numbers to be close. Then they will go and stand in the line but in a way that they wont waste their time standing in a queue for long time
- **advanced function 1:** The second functionality point out about booking a visit from application for supermarket which is similar to booking a visit for museums but has some differences. This feature let application for customer to indicate the approximate expected duration of the visit. Alternatively, for long- term customers, this time could be inferred by the system based on an analysis of the previous visits. The application might also allow users to indicate, if not the exact list of items that they intend to purchase, the categories of items that they intend to buy. This would allow the application to plan visits in a finer way, for example allowing more people in the store, if it knows that they are going to buy
- **advanced function 2:** Other feature is that the application might have include a suggestion of alternative slots (in the same day, or in different days) for visiting the store, to balance out the number of people in the store, the suggestion of different stores of the same chain (or even of different chains, if the application is chain- independent) if the preferred one is not available, or the periodic notification of available slots in a day/time range.

### 1.2.1 World phenomena

WP1	User wants to go shopping
WP2	User get a ticket from ticket machine
WP3	User approaches the shop
WP4	User show the QR code to scanner
WP5	User checkouts the shop

### 1.2.2 Shared phenomena

<b>SP1</b>	Receive a notification that it is user turn for shopping
<b>SP2</b>	Receive a notification that it is better user to approach to shop based on user's location
<b>SP3</b>	Generate QR code
<b>SP4</b>	User choose which category wants to buy
<b>SP5</b>	Manager define his store
<b>SP6</b>	Receive information for booking a visit
<b>SP7</b>	Suggest free slots to user
<b>SP8</b>	Read QR code for entering the store

### 1.2.3 Goals

<b>G1</b>	Allow user to line up for a specific store
<b>G2</b>	Allow user to book a visit for a specific store
<b>G3</b>	Allow store to generate a ticket for whom they don't use electronic devices
<b>G4</b>	Estimate the waiting time for each person
<b>G5</b>	Notify user to approach store
<b>G6</b>	Notify user when it is their turn
<b>G7</b>	Suggest people free slots to book a visit
<b>G8</b>	Allow manager to define their store in the system
<b>G9</b>	Allow users to enter the store
<b>G10</b>	Allow users checkout from the store

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

<b>Manager</b>	A person who manages the shop
<b>User</b>	A regular citizen who wants to shop
<b>Clerk</b>	A person who works for a specific shop
<b>Lineup</b>	An imaginary queue of current person who wants go to shop
<b>Ticket Machine</b>	A stand that clerk can get and print ticket
<b>Scanner</b>	A device that scans QR code
<b>QR Code</b>	is a type of matrix barcode (or two-dimensional barcode)

### 1.3.2 Acronyms

<b>RASD</b>	Requirement Analysis and Specification Document
<b>GPS</b>	Global Positioning System
<b>app</b>	Application
<b>API</b>	Application Programming Interface
<b>QR Code</b>	Quick Responsible code
<b>DDoS</b>	Distributed Denial of Service

### 1.3.3 Abbreviations

<b>WPn</b>	World Phenomenon number n
<b>SPn</b>	Shared Phenomenon number n
<b>Gn</b>	Goal number n
<b>BS</b>	Basic Service of CLup
<b>AF1</b>	Advance Function 1 of CLup
<b>AF2</b>	Advance Function 2 of CLup
<b>Rn</b>	Requirement number n
<b>Dn</b>	Domain assumption number n
<b>Cn</b>	Constraint number n

## 1.4 Revision history

Date	Modifications
<b>14/12/2020</b>	First version
<b>23/12/2020</b>	<ul style="list-style-type: none"> <li>Adding alloy models and alloy code.</li> <li>Update Class Diagram.</li> <li>Adding Mockup images.</li> </ul>

## 1.5 Reference Documents

- Specification Document: "R&DD Assignment A.Y. 2020-2021.pdf"
- Slides of the lectures.

## 1.6 Document Structure

This document is divided in six sections.

- Chapter 1** describes the purpose of this document and contains the description of the given problem we want to solve with our application. We state the goals of CLup and we describe the phenomena related to the "world" where it will be used and the ones related to our system.
- Chapter 2** is about presenting the product perspective, including details on how we abstracted the problem using a class diagram. We describe the main functions of the application using also some state diagrams. We resent the needs of the potential users of the application. Finally we state the domain assumptions and the dependencies.

3. **Chapter 3** contains the external interface requirements, including: user interfaces, hardware interfaces, software interfaces and communication interfaces. We define the functional requirements and the use cases. We use class diagrams and sequence diagrams to describe better the use cases and the interaction between different parts of the system. Lastly we include the performance requirements and the software system attributes.
4. **Chapter 4** contains a model written using the Alloy language in order to describe formally the application
5. **Chapter 5** contains the tables where we reported for each group member the hour spent working on the project
6. **Chapter 6** include the reference documents.

## 2 Overall Description

### 2.1 Product perspective

#### 2.1.1 UML Description

The UML below describes at high-level the model of the system to be developed. It considers the basic service together with the advanced function 1 and advanced function 2 previously specified. The UML does not include every class that will be necessary to define the complete architecture of the system.

CLup has more functions than basic service. The manager registers to the application with all necessary information and the manager could activate the advance function 1 or advance function 2 at any time. The user who use mobile could simply download the application on his/her device and use it and user who doesn't have mobile could easily go near the shop and get a ticket from ticket machine.

Here we can identify the main aspect of CLup:

- The user could request to be in line for a shop and application shows estimated waiting time to him/her.
- The user could book a visit for a shop. This booking contains the date and time user wants to go shopping. Besides, user can add the categories of item he/she has in shopping list. The application could suggest the user free slots and user book them.
- The application based the current location of users must notify them and ask them to approach the shop in a suitable time.
- The application must notify people when it's their turn to go shopping.
- At the entrance time, the QR code generated in the app must be scanned to ensure they come in the right time.
- At the checkout, the clerk must scan the QR code of user and the system must add shopping information (like duration of shopping, category of item which user buy) to user history.

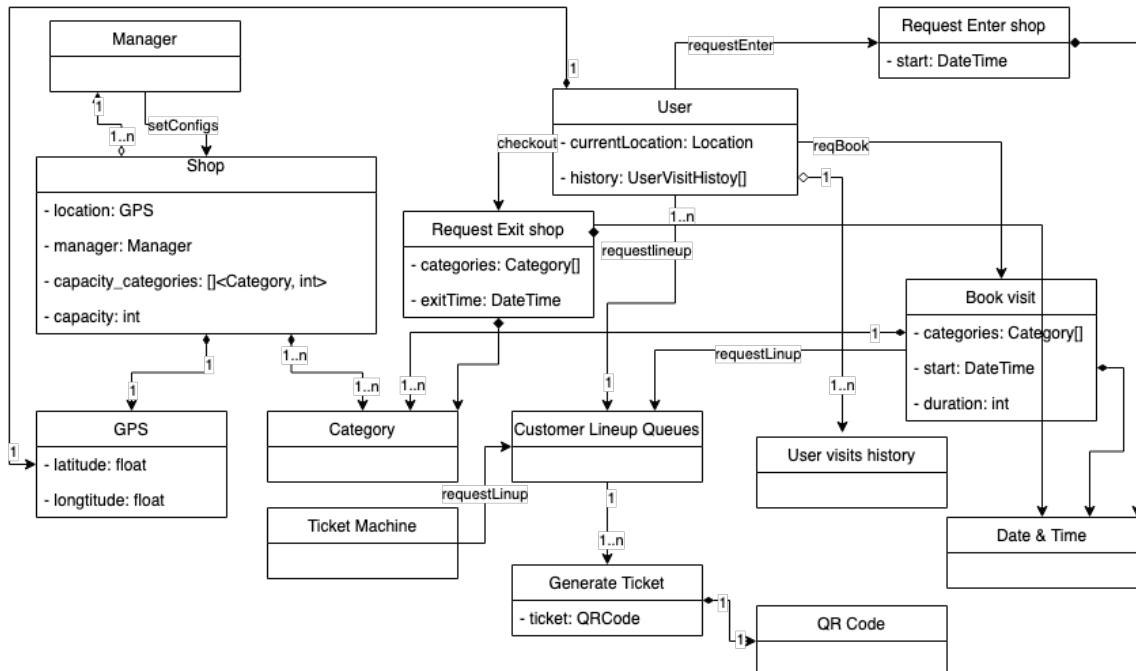


Figure 1. High level Class Diagram

### 2.1.2 State Diagrams

Now we analyse the some important functions of the application, modelling their behaviours and analyze their behaviour to have the expected functionality. we report these diagrams below.

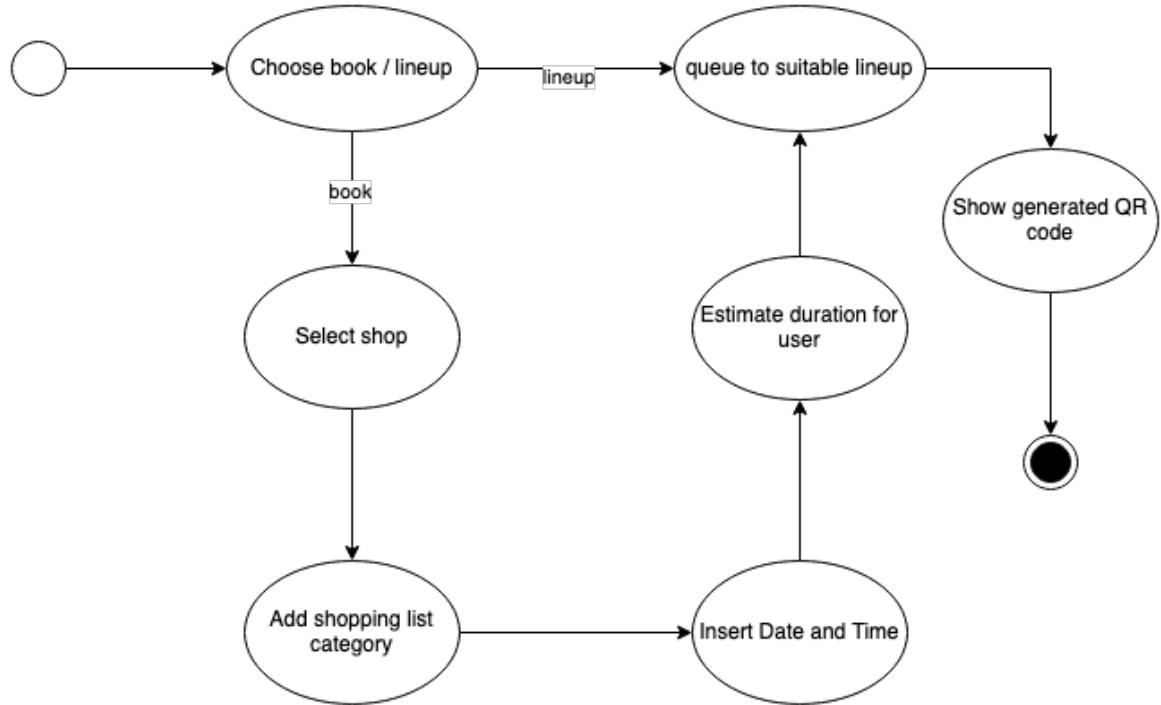


Figure 2. User book a visit or insert to lineup queue

In this (Figure 2), we model a user whom has a cell phone and wants to go shopping. As you can see, the user can choose to go to queue or book a visit for a time in future.

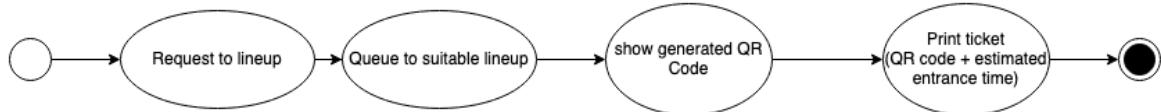


Figure 3. User get ticket from ticket machine

In this (Figure 3), we model a user who want to use ticket machine and do not use the app. In this case, user only can add himself/herself to the current line up of the shop.

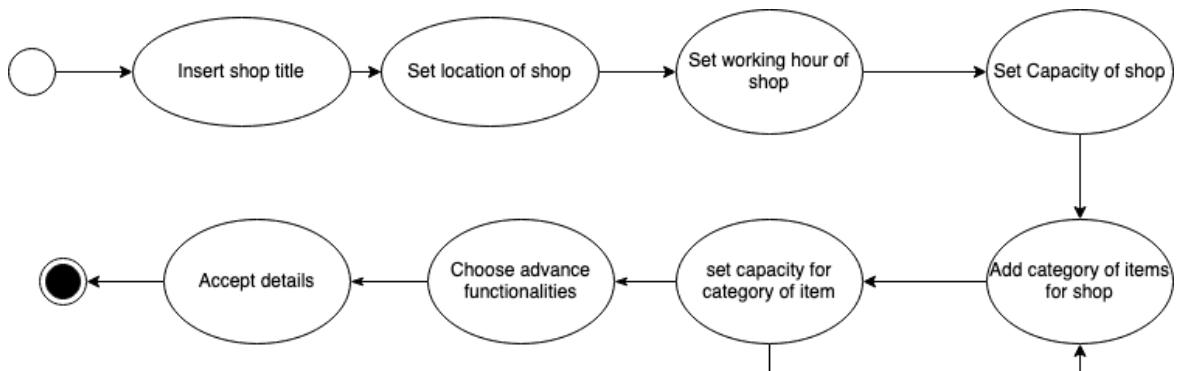


Figure 4. Manager create a shop

In this (Figure 4), shows how a manager could create a shop and add necessary information for creating a shop in our system.

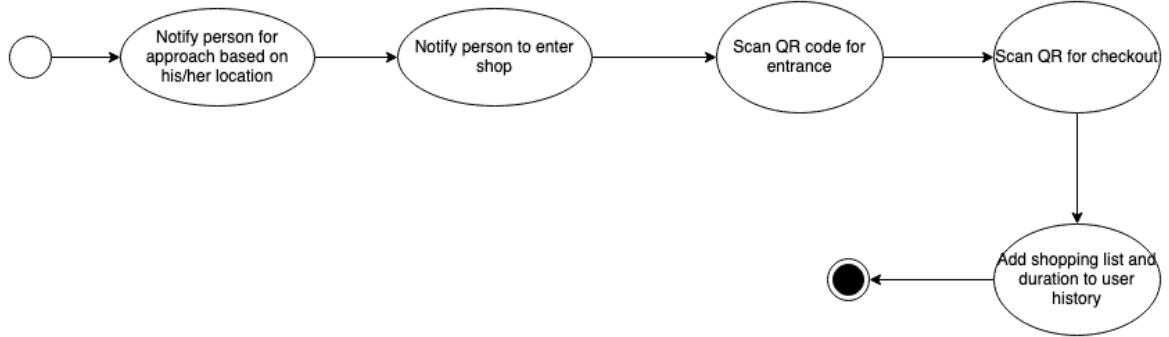


Figure 5. User shopping

In this (Figure 5), we model the behaviour of user from entrance to shop until checkout. In the checkout time, we re-scan the QR code of user to insert data user shopping list and duration to user history. we could use, this information to estimate the behaviour of each user and improve waiting time estimation in our system.

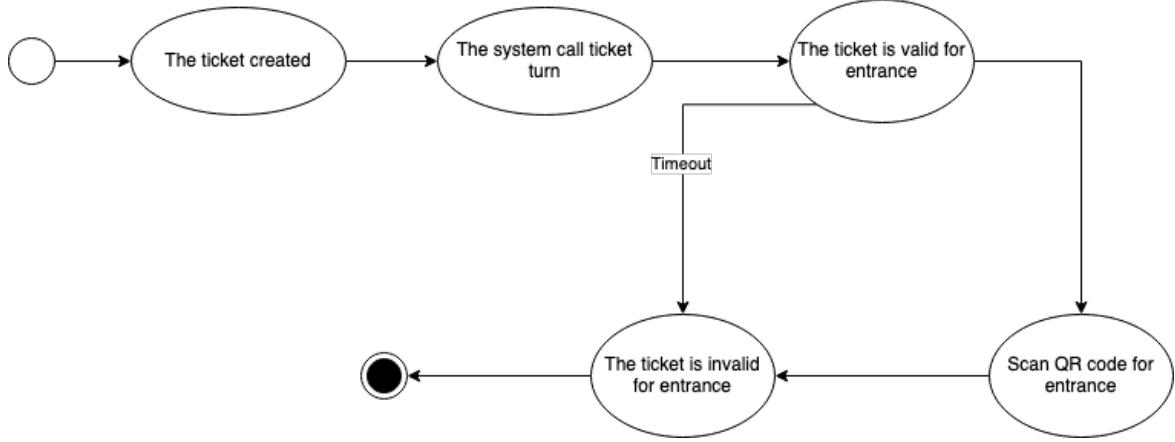


Figure 6. Ticket states

In this (Figure 6), the life cycle of a ticket shows. This helps us to understand when a ticket is valid for enter the shop.

## 2.2 Product functions

1. retrieving a number to line up: the main functionality of the application is the possibility of retrieving a number which gives them their position in the queue. after then it will be easier for customers to access the supermarket without standing in a line for so long. this way forces people to first approach the building and then wait in close proximity (though not in a line) until their number is called.
2. generate QR codes: this feature would be scanned upon entering the store, thus allowing store managers to monitor entrances.
3. estimation process: There is a real risk that the approach does not work in the case the customer arrives to the grocery store after his/her number is called, or too early, as in this case we would get back into a physical line situation. This implies that the system should provide customers with a reasonably precise estimation of the waiting time and should alert them taking into account the time they need to get to the shop from the place they currently are.
4. booking a visit: this function indicate that u can reserve a slot to go to supermarket, its almost like reserving a slot to go to museum or exhibition but they have slight differences like the

duration that take to be in a supermarket which we are not able to estimate their time of shopping. so we have to mention a feature which request customer to indicate the approximate expected duration of the visit.

5. system for the customers analysis: there's another option that works for long-term customers in which the time spent by customers is analyzed and the system is gonna predict the average time for that specefic customer based on their pervious visits.
6. indicating the category of items: The application might also allow users to indicate, if not the exact list of items that they intend to purchase, the categories of items that they intend to buy. This would allow the application to plan visits in a finer way, for example allowing more people in the store, if it knows that they are going to buy different things, hence they will occupy different spaces in the store when they visit (thus respecting the requirement that people keep enough distance between them).

### 2.3 User characteristic

The actors of the application are the following:

1. **user:** customers who download this application to go shopping for the groceries but due to Covid-19 stores and supermarkets shouldn't be full of people and they should follow the social distant rule so without application they used to stand in long lines with social distance but with this application which generate QR codes and with other functionalities, we minimize the time spent by each customer in a line by retrieving a number and send them to be in line if their number is getting closer. also they can book a visit from some days before to go for shopping at supermarkets.
2. **Clerk:** they would scan the QR codes which customers already have by app, and they allow customers to enter. Moreover, stores should also have the possibility to hand out "tickets" on the spot, thus acting as proxies for the customers because maybe some customers do not have the access to the required technology
3. **Manager:** the manager is defining the store on our system and he/she assigns the capacity of either the whole store or each section of the that.

### 2.4 Assumptions, dependencies and constraints

Domain assumptions:

<b>D1</b>	The internet works properly
<b>D2</b>	Its better for users to have smartphones
<b>D3</b>	users should be at the spot right on the specific time which is mentioned in the app
<b>D4</b>	the time estimation by the app is correct
<b>D5</b>	Clerks are connected to device Which is registered for the specific shop.
<b>D6</b>	the app is always available and able to generate QR codes
<b>D7</b>	QR code on the ticket of each user must be scan first then let the user to enter the shop.
<b>D8</b>	the QR code shown by users to scanner is valid
<b>D9</b>	At the checkout, QR code on the ticket of each user must be scanned.
<b>D10</b>	the slots shown by app is correct and updated every second for booking part
<b>D11</b>	the QR code generator which called ticket machine for those who doesn't have smartphones works properly all the time
<b>D12</b>	QR code scanner just scan the valid QR codes which means it shouldn't scan after the estimated time.
<b>D13</b>	users do not have much delay at the supermarket and they exit at almost certain time.

- |            |  |
|------------|--|
| <b>D14</b> | Ticket machine should have enough paper to print QR codes for those who doesn't have smartphone.             |
| <b>D15</b> | if users don't have electronic device, they can use QR codes at the supermarket in person by ticket machine. |

Constraints:

- |           |   |
|-----------|---|
| <b>C1</b> | each ticket is valid for entrance from time the system call it turns until 30 minutes later |
|-----------|---|

### 3 Specific Requirements

#### 3.1 External Interfaces

##### 3.1.1 User Interface

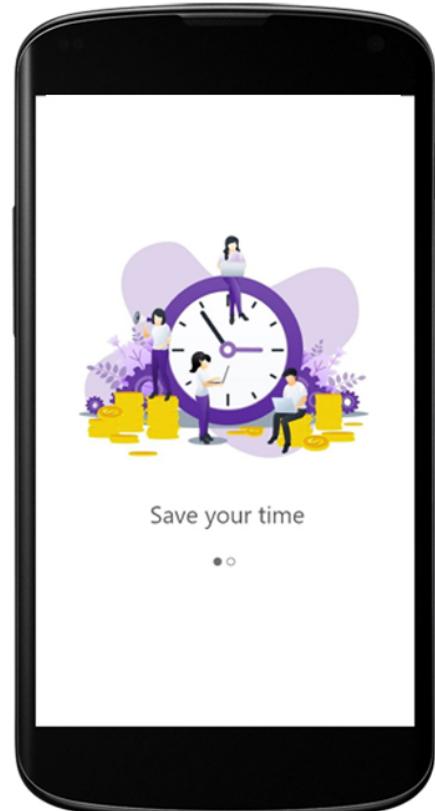
The application that we are using should be able to generate QR codes and will appear in the screen for customers to use. before that, we assume that users already inserted the specific shop they want to go and also choose between two mode of lining up and book a visit for that supermarket later. the app show estimated time and the number of people before that user so they can save their time and mange their life better. after all they will scan their QR code and enter the supermarket. A smartphone and within that, the application, is the suitable thing that answer the needs. the following mockups give the idea of: 1. The logo of the application looks like; 2. The splash screens; 3. And the 4 first pages of the application;



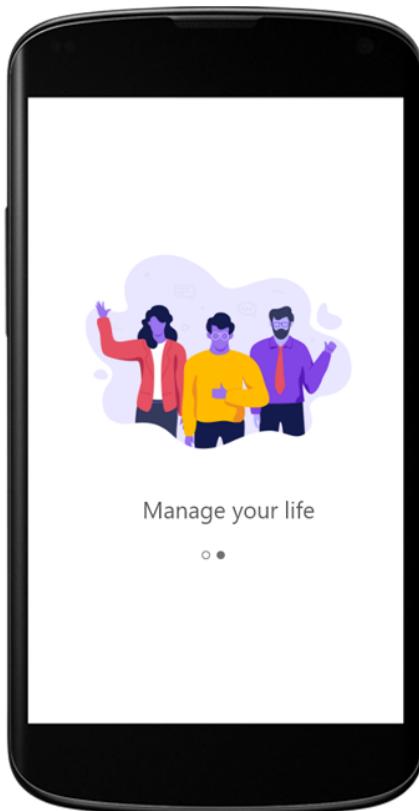
*Figure 7. Icon Mockup*



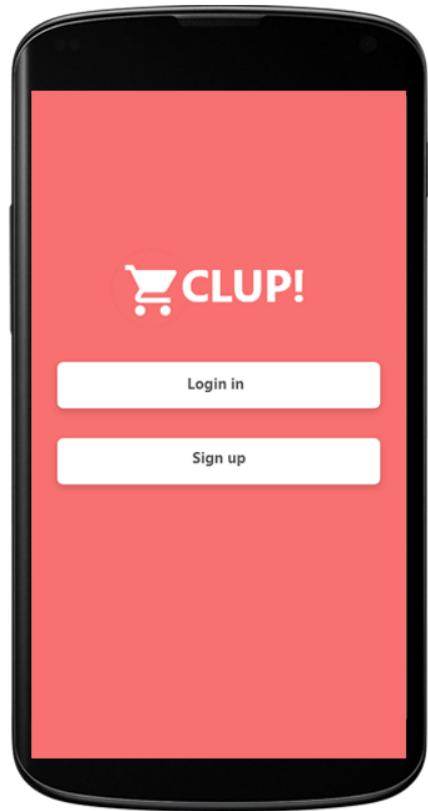
(a) Opening page Mockup



(b) Splash Screen Mockup

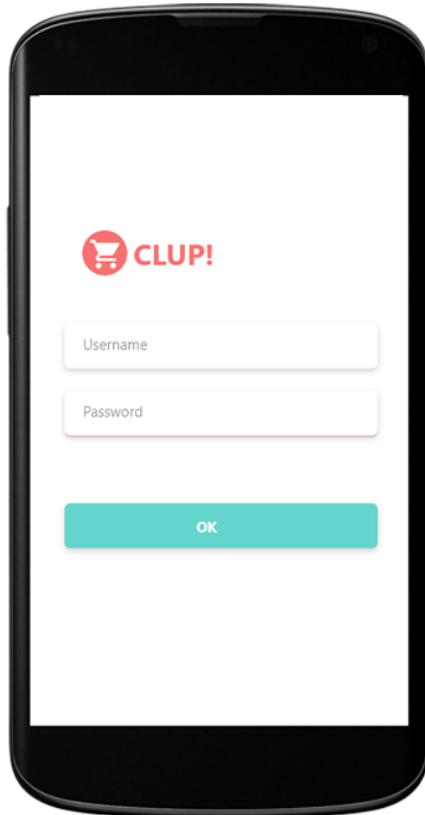


(c) Splash Screen page Mockup

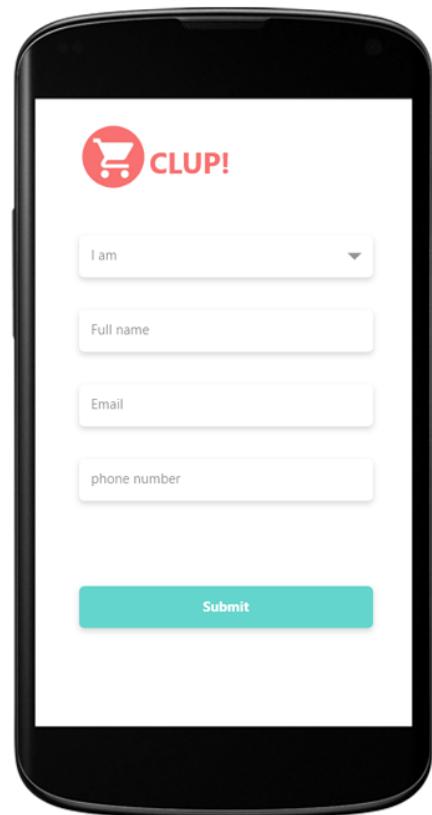


(d) Sign up/log in page Mockup

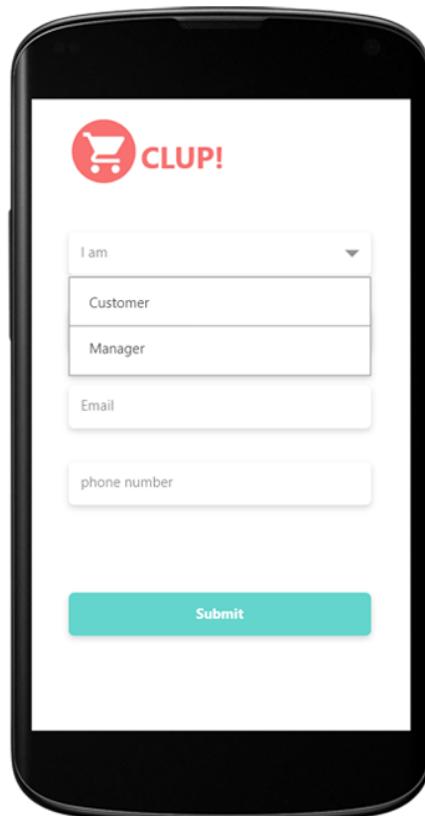
Figure 8. Mockup



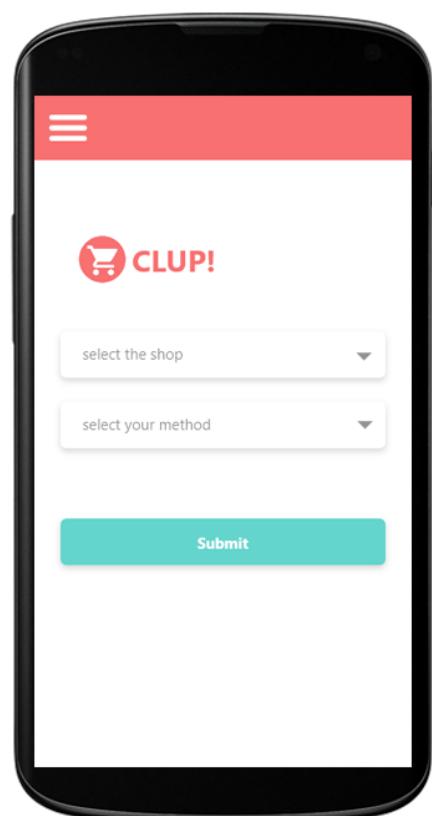
(a) Log in page Mockup



(b) Sign up page Mockup

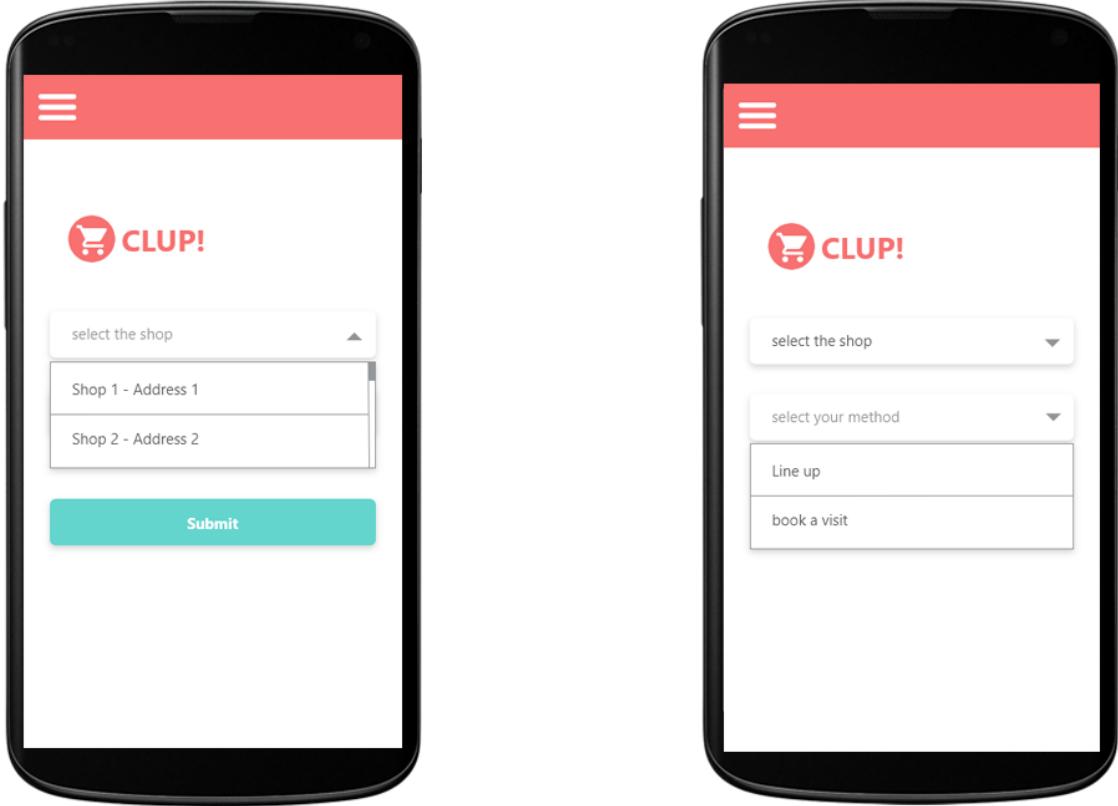


(c) Sign up page II Mockup



(d) Home page Mockup

Figure 9. Mockup



(a) Home page II Mockup

(b) Home page III Mockup

*Figure 10. Mockup*

Other mockups will be presented in the Design Document part for User Interfaces.

### 3.1.2 Hardware Interfaces

The software application contains three main functions and two types of actors, which require different kinds of hardware interfaces.

- Regarding BS: The users must own a smartphone in order to generate QR codes to access supermarkets. They can also use a ticket machine for this purpose so that they can have the code in person. The managers, instead, can access the system only through the App using a smartphone. They first have to register themselves to the system and then add their shops to the system. There is also a scanner which scan the validity of the tickets.
- Regarding AF1 : for booking a visit, users should have smartphone to get their QR codes otherwise they cannot use this feature.

### 3.1.3 software Interfaces

The system uses the following external interfaces:

City map We assume that the system uses a public API to provide: 1. to the customer the possibility to see the shops location by selecting its branch name. 2. to the manager to add their shops location to the system

notification server: This app has a system which it sends notification to customers when their turn is close

estimation time: The application shows an estimated time in which users can manage their time to be on the spot in time.

Communication Interfaces: The device connects to CLUp via internet connection.

## 3.2 Functional Requirements

### 3.2.1 List of requirements

<b>R1</b>	Users certified with an authentication
<b>R2</b>	Managers certified with an authentication
<b>R3</b>	Managers should register to the application with extra mandatory fields
<b>R4</b>	Only managers can create shop
<b>R5</b>	Only managers can update their shop
<b>R6</b>	Managers can activate advance function 1 (allow users booking) on their shop
<b>R7</b>	Managers can activate advance function 2 (send suggestions) on their shop
<b>R8</b>	Users must accept the GPS location for the application
<b>R9</b>	Users could see the list of shops are around them
<b>R10</b>	Users could see the list of shops are around them
<b>R11</b>	Users could see their ticket (QR code) and estimated waiting time
<b>R12</b>	Users could "book a time" to visit specific shop
<b>R13</b>	Users could choose for which categories they go shopping when they book a visit
<b>R14</b>	The system must generate suitable QR code
<b>R15</b>	The system must estimate the waiting time for each user
<b>R16</b>	The system must make current line-up queue based on users booking
<b>R17</b>	The system sends a notification to user to approach shop based on current location of user and location of shop
<b>R18</b>	The system sends a notification to user to go to the shop when it's him/her turn
<b>R19</b>	After user ticket scanned for entrance, system change status of ticket to invalidate for entrance
<b>R20</b>	The clerk must scan the user ticket
<b>R21</b>	The clerk could add a person to current line-up queue
<b>R22</b>	The clerk could print ticket
<b>R23</b>	After the user checked out, the system analyze shopping list and duration time and add it to the user's visit history
<b>R24</b>	The system could estimate the category of shopping and duration of shopping for specific user
<b>R25</b>	The system estimate the time based on user characteristic
<b>R26</b>	Users can ask a clerk to give them a ticket
<b>R27</b>	If specific time past from user turn to enter his/her ticket will be invalidate for entrance
<b>R28</b>	If managers update a shop the estimated time and book visits will be recalculate
<b>R29</b>	Manager choose the capacity of shop
<b>R30</b>	Manager choose the category of each section in the shop and their capacity
<b>R31</b>	Manager choose working hour of the shop

### 3.2.2 Mapping

Goal	Domain assumption	Requirements
G1	D1,D2,D4,D6,D10,D11	R1,R9,R10,R11,R14,R16
G2	D1,D2,D6,D7,D8,D9,D10,D12	R1,R9,R11,R12,R13,R14,R16,R21,R22
G3	D5,D7,D8,D9,D11,D12,D14,D15	R11,R14,R26
G4	D3,D4,D9,D10,D13	R1,R8,R10,R11,R12,R13,R14,R15,R16,R23, R24,R25,R28,R29,R30,R31
G5	D1,D2,D4	R1,R8,R15,R17
G6	D1,D2,D4	R1,R8,R17
G8	D1,D2,D5	R2,R3,R4,R5,R6,R29,R30,R31
G9	D3,D4,D5,D7,D8,D12	R19,R20,R27
G10	D5,D9	R20,R23,R24

<b>G1</b>	Allow user to line up for a specific store
<b>D1</b>	The internet works properly
<b>D2</b>	Its better for users to have smartphones
<b>D4</b>	Estimate waiting time
<b>D6</b>	the app is always available and able to generate QR codes
<b>D10</b>	the slots shown by app is correct and updated every second for booking part
<b>D11</b>	the QR code generator which called ticket machine for those who doesn't have smartphones works properly all the time
<b>R1</b>	Users certified with an authentication
<b>R9</b>	Users could see the list of shops are around them
<b>R10</b>	Users could see the list of shops are around them
<b>R11</b>	Users could see their ticket (QR code) and estimated waiting time
<b>R14</b>	The system must generate suitable QR code
<b>R16</b>	The system must make current line-up queue based on users booking
<b>G2</b>	Allow user to book a visit for a specific store
<b>D1</b>	The internet works properly
<b>D2</b>	Its better for users to have smartphones
<b>D6</b>	the app is always available and able to generate QR codes
<b>D7</b>	QR code on the ticket of each user must be scan first then let the user to enter the shop.
<b>D8</b>	the QR code shown by users to scanner is valid
<b>D9</b>	At the checkout, QR code on the ticket of each user must be scanned.
<b>D10</b>	the slots shown by app is correct and updated every second for booking part
<b>D12</b>	QR code scanner just scan the valid QR codes which means it shouldn't scan after the estimated time

<b>R1</b>	Users certified with an authentication
<b>R9</b>	Users could see the list of shops are around them
<b>R11</b>	Users could see their ticket (QR code) and estimated waiting time
<b>R12</b>	Users could "book a time" to visit specific shop
<b>R13</b>	Users could choose for which categories they go shopping when they book a visit
<b>R14</b>	The system must generate suitable QR code
<b>R16</b>	The system must make current line-up queue based on users booking
<b>R21</b>	The clerk could add a person to current line-up queue
<b>R22</b>	The clerk could print ticket
<b>G3</b>	Allow store to generate a ticket for whom they don't use electronic devices
<b>D6</b>	the app is always available and able to generate QR codes
<b>D7</b>	QR code on the ticket of each user must be scan first then let the user to enter the shop.
<b>D8</b>	the QR code shown by users to scanner is valid
<b>D9</b>	At the checkout, QR code on the ticket of each user must be scanned.
<b>D11</b>	the QR code generator which called ticket machine for those who doesn't have smartphones works properly all the time
<b>D12</b>	QR code scanner just scan the valid QR codes which means it shouldn't scan after the estimated time
<b>D14</b>	Ticket machine should have enough paper to print QR codes for those who doesn't have smartphone
<b>D15</b>	if users don't have electronic device, they can use QR codes at the supermarket in person by ticket machine
<b>R11</b>	Users could see their ticket (QR code) and estimated waiting time
<b>R14</b>	The system must generate suitable QR code
<b>R26</b>	Users can ask a clerk to give them a ticket
<b>G4</b>	Estimate the waiting time for each person
<b>D3</b>	users should be at the spot right on the specific time which is mentioned in the app
<b>D4</b>	Estimate waiting time
<b>D9</b>	At the checkout, QR code on the ticket of each user must be scanned.
<b>D10</b>	the slots shown by app is correct and updated every second for booking part
<b>D13</b>	users do not have much delay at the supermarket and they exit at almost certain time
<b>R1</b>	Users certified with an authentication
<b>R8</b>	Users must accept the GPS location for the application
<b>R10</b>	Users could see the list of shops are around them
<b>R11</b>	Users could see their ticket (QR code) and estimated waiting time
<b>R12</b>	Users could "book a time" to visit specific shop
<b>R13</b>	Users could choose for which categories they go shopping when they book a visit

<b>R14</b>	The system must generate suitable QR code
<b>R15</b>	The system must estimate the waiting time for each user
<b>R16</b>	The system must make current line-up queue based on users booking
<b>R23</b>	After the user checked out, the system analyze shopping list and duration time and add it to the user's visit history
<b>R24</b>	The system could estimate the category of shopping and duration of shopping for specific user
<b>R25</b>	The system estimate the time based on user characteristic
<b>R28</b>	If managers update a shop the estimated time and book visits will be recalculate
<b>R29</b>	Manager choose the capacity of shop
<b>R30</b>	Manager choose the category of each section in the shop and their capacity
<b>R31</b>	Manager choose working hour of the shop
<b>G5</b>	Notify user to approach store
<b>D1</b>	The internet works properly
<b>D2</b>	Its better for users to have smartphones
<b>D4</b>	Estimate waiting time
<b>R1</b>	Users certified with an authentication
<b>R8</b>	Users must accept the GPS location for the application
<b>R15</b>	The system must estimate the waiting time for each user
<b>R17</b>	The system sends a notification to user to approach shop based on current location of user and location of shop
<b>G6</b>	Notify user when it is their turn
<b>D1</b>	Estimate waiting time
<b>D2</b>	Estimate waiting time
<b>D4</b>	Estimate waiting time
<b>R1</b>	Users certified with an authentication
<b>R8</b>	Users must accept the GPS location for the application
<b>R17</b>	The system sends a notification to user to approach shop based on current location of user and location of shop
<b>G8</b>	Allow manager to define their store in the system
<b>D1</b>	The internet works properly
<b>D2</b>	Its better for users to have smartphones
<b>D5</b>	Clerks are connected to device Which is registered for the specific shop
<b>R2</b>	Managers certified with an authentication
<b>R3</b>	Managers should register to the application with extra mandatory fields
<b>R4</b>	Only managers can create shop
<b>R5</b>	Only managers can update their shop

<b>R6</b>	Managers can activate advance function 1 (allow users booking) on their shop
<b>R29</b>	Manager choose the capacity of shop
<b>R30</b>	Manager choose the category of each section in the shop and their capacity
<b>R31</b>	Manager choose working hour of the shop
<b>G9</b>	Allow users to enter the store
<b>D3</b>	users should be at the spot right on the specific time which is mentioned in the app
<b>D4</b>	the time estimation by the app is correct
<b>D5</b>	Clerks are connected to device Which is registered for the specific shop
<b>D7</b>	QR code on the ticket of each user must be scan first then let the user to enter the shop.
<b>D8</b>	the QR code shown by users to scanner is valid
<b>D12</b>	QR code scanner just scan the valid QR codes which means it shouldn't scan after the estimated time
<b>R19</b>	After user ticket scanned for entrance, system change status of ticket to invalidate for entrance
<b>R20</b>	The clerk must scan the user ticket
<b>R27</b>	If specific time past from user turn to enter his/her ticket will be invalidate for entrance
<b>G10</b>	Allow users checkout from the store
<b>D6</b>	the app is always available and able to generate QR codes
<b>D9</b>	At the checkout, QR code on the ticket of each user must be scanned.
<b>R20</b>	The clerk must scan the user ticket
<b>R23</b>	After the user checked out, the system analyze shopping list and duration time and add it to the user's visit history
<b>R24</b>	The system could estimate the category of shopping and duration of shopping for specific user

### 3.2.3 Use cases

#### 3.2.3.1 Use case Description

- Register

<b>Name</b>	Register
<b>ID</b>	UC1
<b>Actors</b>	Super User
<b>Entry Condition</b>	Super User has internet connection on his/her device

<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. Super user see the landing page</li> <li>2. Super user click on the "sign up" button</li> <li>3. Super user fill all the mandatory fields</li> <li>4. Super user choose his type between manager or user</li> <li>5. Super user fill extra fields based on his type</li> <li>6. Super user click on "Register" button</li> <li>7. The system validate the filled data</li> <li>8. The system confirm the registration</li> <li>9. The system save the super user on database</li> </ol>
<b>Exit Condition</b>	Super user is successfully registered in the system
<b>Exceptions</b>	<p>If the below conditions happened the application return a suitable error in message and return to register page.</p> <ul style="list-style-type: none"> <li>– Super user is already existed in the system</li> <li>– Super user doesn't fill all mandatory fields</li> <li>– Super user sends invalid data</li> </ul>

- **Login**

<b>Name</b>	Login
<b>ID</b>	UC2
<b>Actors</b>	Super User
<b>Entry Condition</b>	Super User has internet connection and it already registered to the application
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. Super user see the landing page</li> <li>2. Super user click on the "sign in" button</li> <li>3. Super user fill username and password fields</li> <li>4. Super user click on "Login" button</li> <li>5. The system check the credential</li> <li>6. The system confirm the registration</li> <li>7. The system return the home page of user</li> </ol>
<b>Exit Condition</b>	Super user has access to the service of CLUp
<b>Exceptions</b>	<p>If the below conditions happened the application return a suitable error in message and return to login page.</p> <ul style="list-style-type: none"> <li>– Super user is not exist in the system</li> <li>– Super user doesn't fill all mandatory fields</li> <li>– Super user password is invalid</li> </ul>

- **Create shop**

<b>Name</b>	Create Shop
<b>ID</b>	UC3
<b>Actors</b>	Manager
<b>Entry Condition</b>	Manager has internet connection
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. Manager see the home page</li> <li>2. Manager click on the "add a shop" button</li> <li>3. Manager set shop title</li> <li>4. Manager set location of the shop</li> <li>5. Manager set start and end hour of the shop</li> <li>6. Manager set capacity of the shop</li> <li>7. manager click on "new category"</li> <li>8. manager set capacity and title of category</li> <li>9. click on "end" button or return to 7.</li> <li>10. manager choose AF1 or AF2 enable by checkbox.</li> <li>11. manager click on "Submit" button.</li> <li>12. The system shows all the detail of shop.</li> <li>13. manager click on "Accept" button.</li> <li>14. The system add shop to data base.</li> </ol>
<b>Exit Condition</b>	Shop successfully created
<b>Exceptions</b>	If the data of shop is not valid the system will return error

- Update shop

<b>Name</b>	Create Shop
<b>ID</b>	UC4
<b>Actors</b>	Manager
<b>Entry Condition</b>	Manager has internet connection and the shop is exists
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. Manager see the home page</li> <li>2. Manager click on the "edit" button in specific shop</li> <li>3. Manager change fields it like</li> <li>4. manager click on "Submit" button.</li> <li>5. The system shows all the detail of shop.</li> <li>6. manager click on "Accept" button.</li> <li>7. The system update the shop in data base.</li> <li>8. The system update recalculate the estimated time by new values.</li> </ol>
<b>Exit Condition</b>	Shop successfully updated

<b>Exceptions</b>	If the below conditions happened the application return a suitable error. <ul style="list-style-type: none"> <li>– Shop is not existed in the system.</li> <li>– Manager doesn't have access to change that shop.</li> <li>– Fields values are invalid.</li> </ul>
-------------------	--

- **See list of shops**

<b>Name</b>	<b>See list of shops</b>
<b>ID</b>	UC5
<b>Actors</b>	User
<b>Entry Condition</b>	User has internet connection and must accept GPS permission
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. User get latitude and longitude from GPS</li> <li>2. User send a request to get list of shops</li> <li>3. The system extract latitude and longitude from user request</li> <li>4. The system get shop list from database</li> <li>5. The system order shops based on latitude and longitude and characteristic of user.</li> <li>6. The system shows the list of shops to user.</li> </ol>
<b>Exit Condition</b>	User see list of shops
<b>Exceptions</b>	None

- **Generate ticket**

<b>Name</b>	<b>Generate ticket</b>
<b>ID</b>	UC6
<b>Actors</b>	User, Clerk
<b>Entry Condition</b>	user or clerk wants to add to a line-up queue
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The system extract the user turn.</li> <li>2. The system extract the user id.</li> <li>3. The system extract the shop id.</li> <li>4. The system extract the line-up queue id.</li> <li>5. The system generate a QR code based on those information.</li> <li>6. The system saves generated QR code to database.</li> <li>7. The system shows the generated QR code to user.</li> </ol>
<b>Exit Condition</b>	QR code generated successfully
<b>Exceptions</b>	None

- **Estimate waiting time**

<b>Name</b>	<b>Estimate waiting time</b>
<b>ID</b>	UC7
<b>Actors</b>	User, Clerk
<b>Entry Condition</b>	user or clerk wants to add to a line-up queue
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The system extract the users from line-up queue.</li> <li>2. The system extract the shop.</li> <li>3. The system extract the new user wants to add in line up.</li> <li>4. The system get history of users shopping from database.</li> <li>5. The system extract new user wants shop from which categories.</li> <li>6. based on previous behavior of user and categories system estimate shopping duration of user.</li> <li>7. based on previous users in queue, system estimate waiting time for user.</li> </ol>
<b>Exit Condition</b>	The system estimate the user waiting time
<b>Exceptions</b>	Check the shops, users and line-up queue are really exists. if they are not exist send a proper message.

- **Add to Line-up Queue**

<b>Name</b>	<b>Add to Line-up Queue</b>
<b>ID</b>	UC8
<b>Actors</b>	User, Clerk
<b>Entry Condition</b>	user choose specific shop to go to one of it lines up queue or user book a visit, clerk wants to add person do not use application to a line-up queue
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. User choose a shop to add line up or book a visit for that shop or clerk click on add to line-up.</li> <li>2. The system find that line-up queue</li> <li>3. The system get the generate number of user in queue.</li> <li>4. The system pass user, shop and queue to [UC6].</li> <li>5. The system get the QR code of ticket.</li> <li>6. The system pass user, shop and queue to [UC7].</li> <li>7. The system get the estimation waiting time for user.</li> <li>8. The system get current location of user.</li> <li>9. The system get calculate traveling time estimation for user.</li> <li>10. The system set interval to send notification for approach a user.</li> <li>11. The system add user to correct line-up queue.</li> <li>12. The system send ticket and estimated time to user.</li> </ol>
<b>Exit Condition</b>	add user to line-up queue
<b>Exceptions</b>	None

- Book a visit

<b>Name</b>	<b>Book a visit</b>
<b>ID</b>	UC9
<b>Actors</b>	User
<b>Entry Condition</b>	The shop AF1 is activate and user choose book
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. User choose a shop.</li> <li>2. User click on "book a visit" button.</li> <li>3. The user choose date and time.</li> <li>4. The system returns items' category of that shop.</li> <li>5. User add category of items wants to buy.</li> <li>6. Based on date user choose system finds a suitable queue.</li> <li>7. Pass the parameters to [UC8] and get the result.</li> <li>8. Get the ticket and show it.</li> </ol>
<b>Exit Condition</b>	user successfully book a visit
<b>Exceptions</b>	If user the queue capacity is full

- See ticket

<b>Name</b>	<b>See ticket</b>
<b>ID</b>	UC10
<b>Actors</b>	User
<b>Entry Condition</b>	The user has a ticket for that shop
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. User choose a shop.</li> <li>2. User click on "ticket" button.</li> <li>3. The system find the ticket.</li> <li>4. The system calculate remaining time.</li> <li>5. The system return the QR code and estimation time.</li> </ol>
<b>Exit Condition</b>	user get the ticket
<b>Exceptions</b>	If user is doesn't have the ticket for that shop system return a message

- Notify user for approach

<b>Name</b>	<b>Notify user for approach</b>
<b>ID</b>	UC11
<b>Actors</b>	Time
<b>Entry Condition</b>	Timer that [UC8] set, finished

<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The system sends a request to notification system.</li> <li>2. The notification system send a notification to user.</li> </ol>
<b>Exit Condition</b>	user get notification
<b>Exceptions</b>	none

- Print ticket

<b>Name</b>	<b>Print ticket</b>
<b>ID</b>	UC13
<b>Actors</b>	Clerk
<b>Entry Condition</b>	A person doesn't use mobile ask clerk for a ticket
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The clerk click "ticket" on ticket machine.</li> <li>2. The system add an anonymous user to queue by [UC9].</li> <li>3. The system show ticket.</li> <li>4. The clerk click on "print" button.</li> <li>5. The printer print the ticket.</li> </ol>
<b>Exit Condition</b>	anonymous user got to queue and ticket printed
<b>Exceptions</b>	none

- Scan QR code for entrance

<b>Name</b>	<b>Scan QR code for entrance</b>
<b>ID</b>	UC14
<b>Actors</b>	Clerk
<b>Entry Condition</b>	A person want to enter the shop
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The person show his/her ticket on mobile or printed ticket.</li> <li>2. The clerk scan QR code.</li> <li>3. The system check can the user enter.</li> <li>4. The system remove user from line-up queue.</li> <li>5. The system change ticket status invalid for entrance.</li> <li>6. The system set start time for user.</li> </ol>
<b>Exit Condition</b>	user can enter the shop
<b>Exceptions</b>	If it isn't user turn red flash light blink on scanner

- checkout

<b>Name</b>	<b>checkout</b>
<b>ID</b>	UC15

<b>Actors</b>	Clerk
<b>Entry Condition</b>	A person want to checkout the shop
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The person show his/her ticket on mobile or printed ticket.</li> <li>2. The clerk scan QR code.</li> <li>3. The system find the user.</li> <li>4. The clerk scan bar code of items with his/her scanner.</li> <li>5. The system find category of items and add to user history.</li> <li>6. The system set end time for user and calculate the duration.</li> <li>7. The system add user history to database.</li> <li>8. The system call the next person turn.</li> <li>9. The system set a 1 minute timer to notify next person.</li> </ol>
<b>Exit Condition</b>	user checkout and exit the shop
<b>Exceptions</b>	If user lost his ticket act him as an anonymous user.

- **checkout**

<b>Name</b>	checkout
<b>ID</b>	UC16
<b>Actors</b>	Time
<b>Entry Condition</b>	Timer that [UC15] set finished.
<b>Event flow</b>	<ol style="list-style-type: none"> <li>1. The system change ticket of user to valid.</li> <li>2. The system sends a request to notification system.</li> <li>3. The notification system send a notification to user.</li> </ol>
<b>Exit Condition</b>	user get notification
<b>Exceptions</b>	None

### 3.2.3.2 Use case Diagram

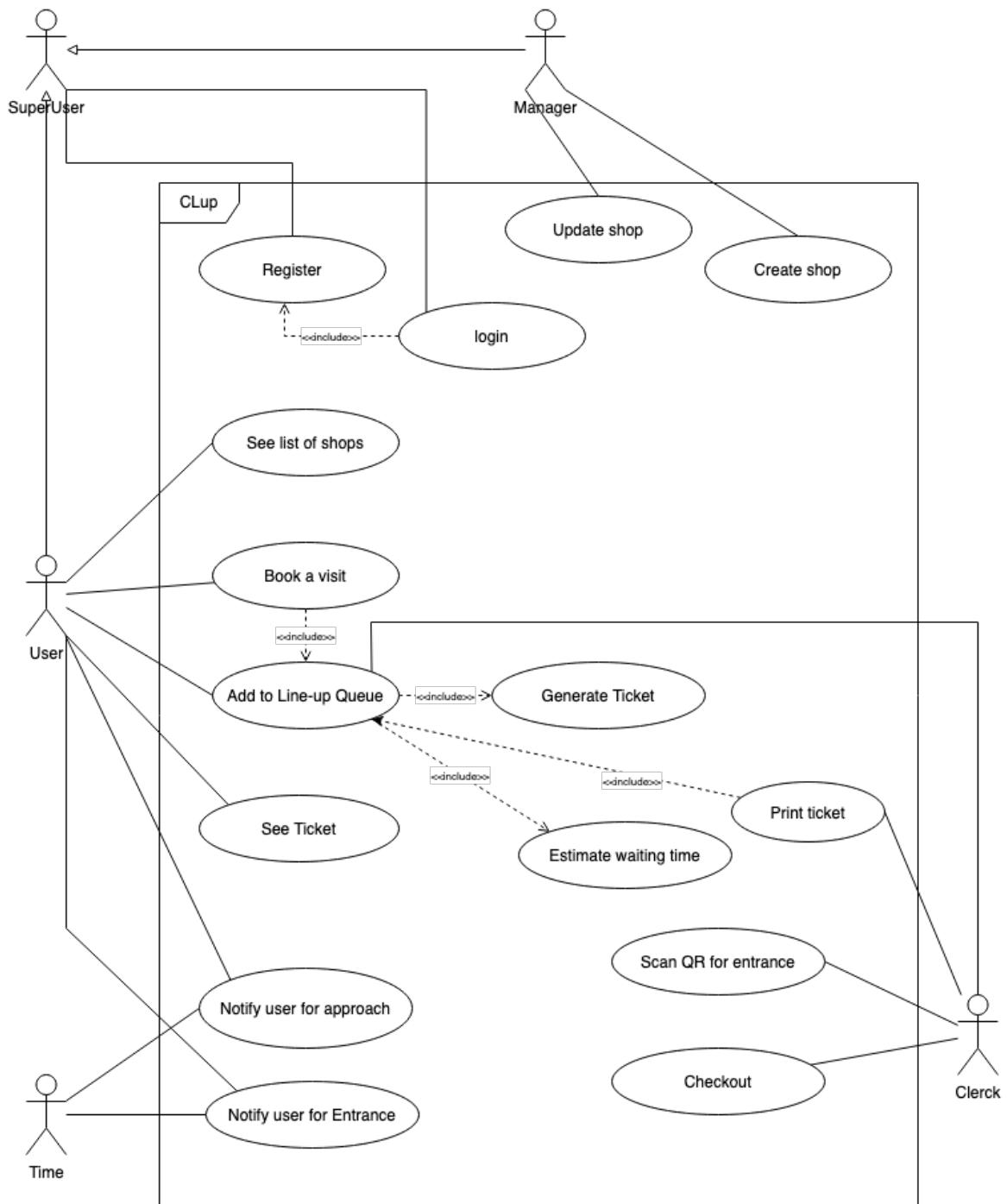


Figure 11. Use case diagram

### 3.2.4 Sequence Diagrams

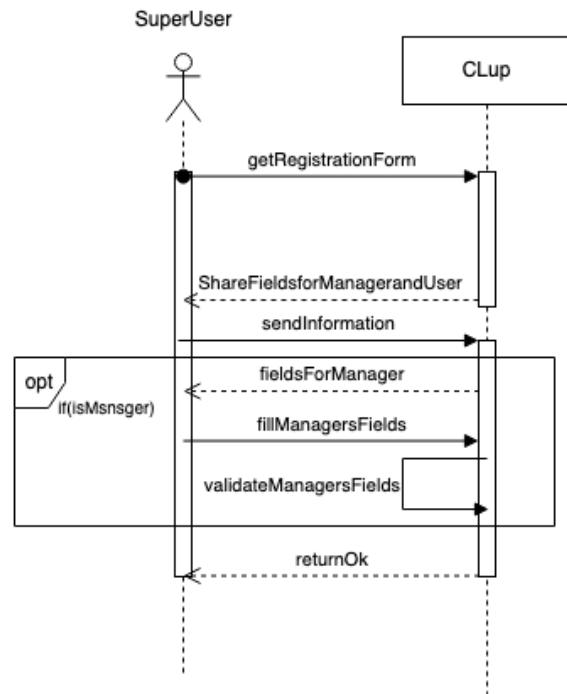


Figure 12. Sequence diagram for register

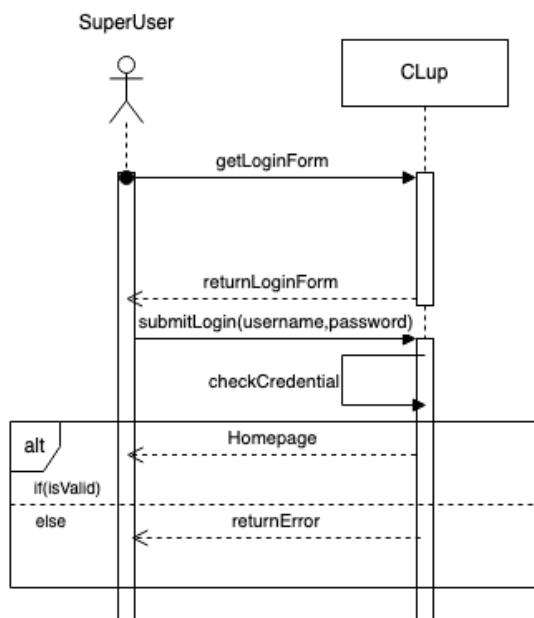


Figure 13. Sequence diagram for login

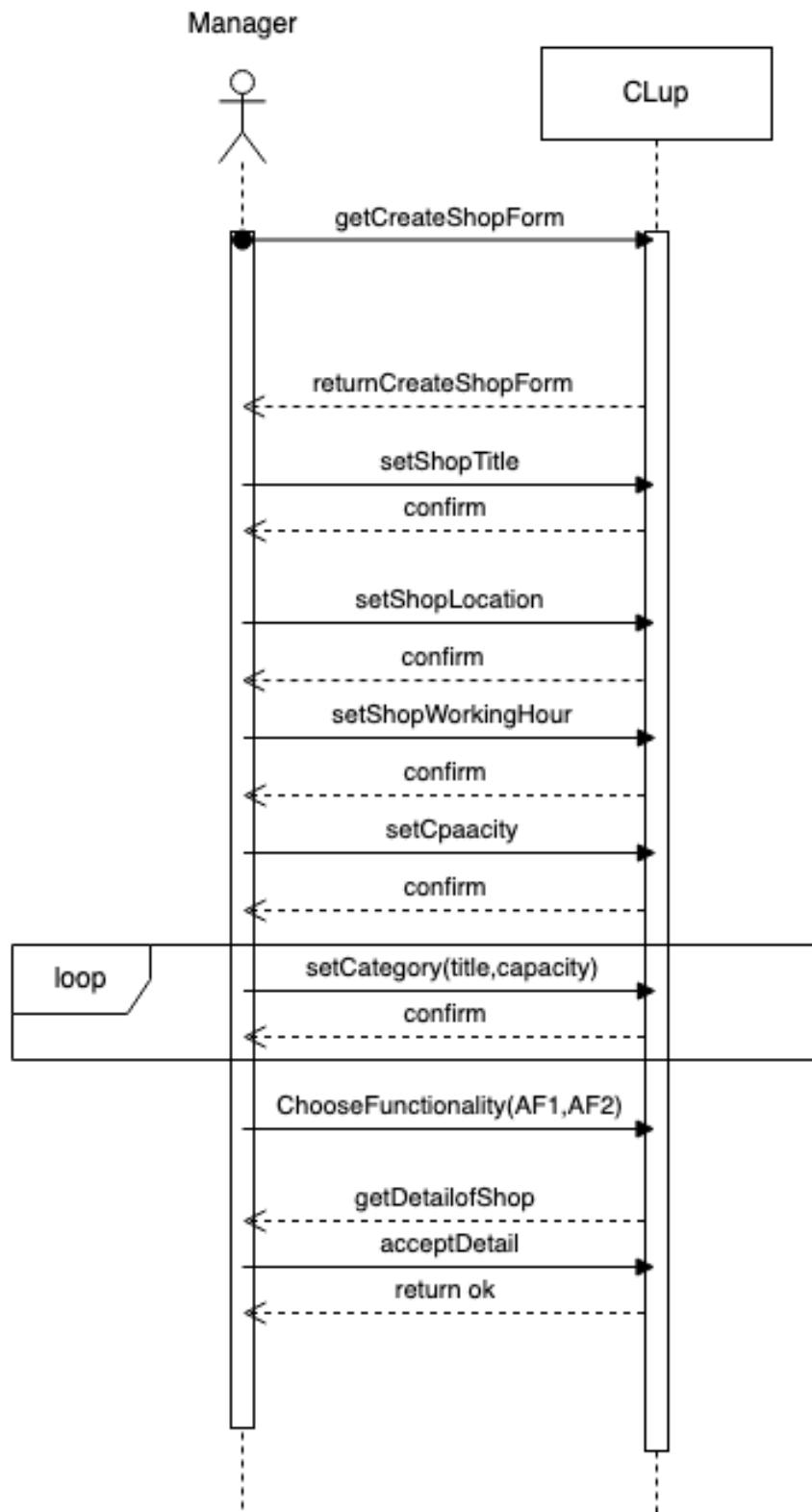


Figure 14. Sequence diagram for create a shop

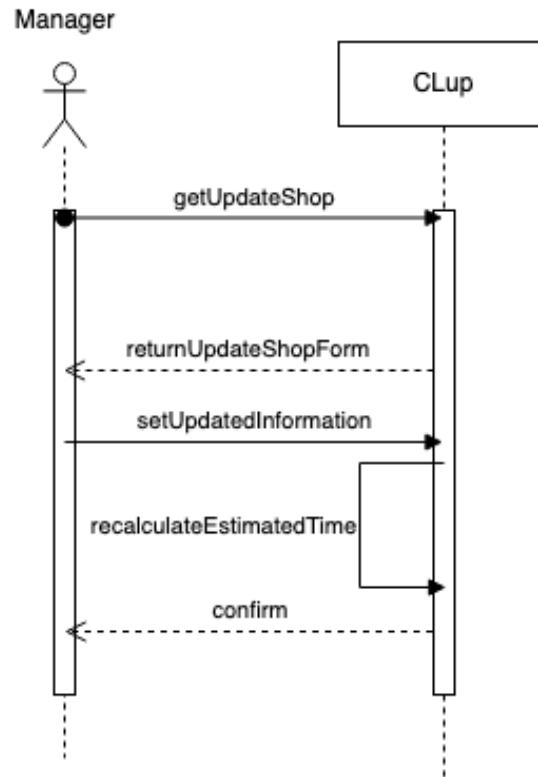


Figure 15. Sequence diagram for update shop

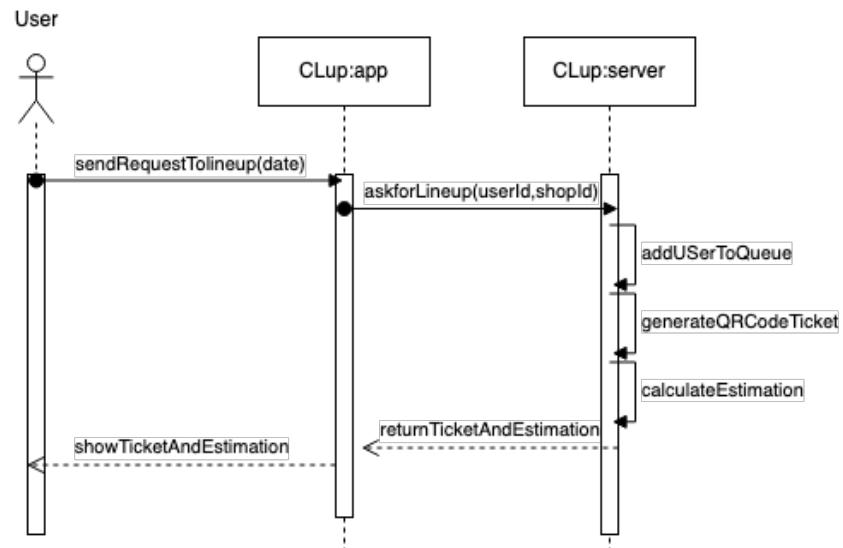


Figure 16. Sequence diagram for add to line-up

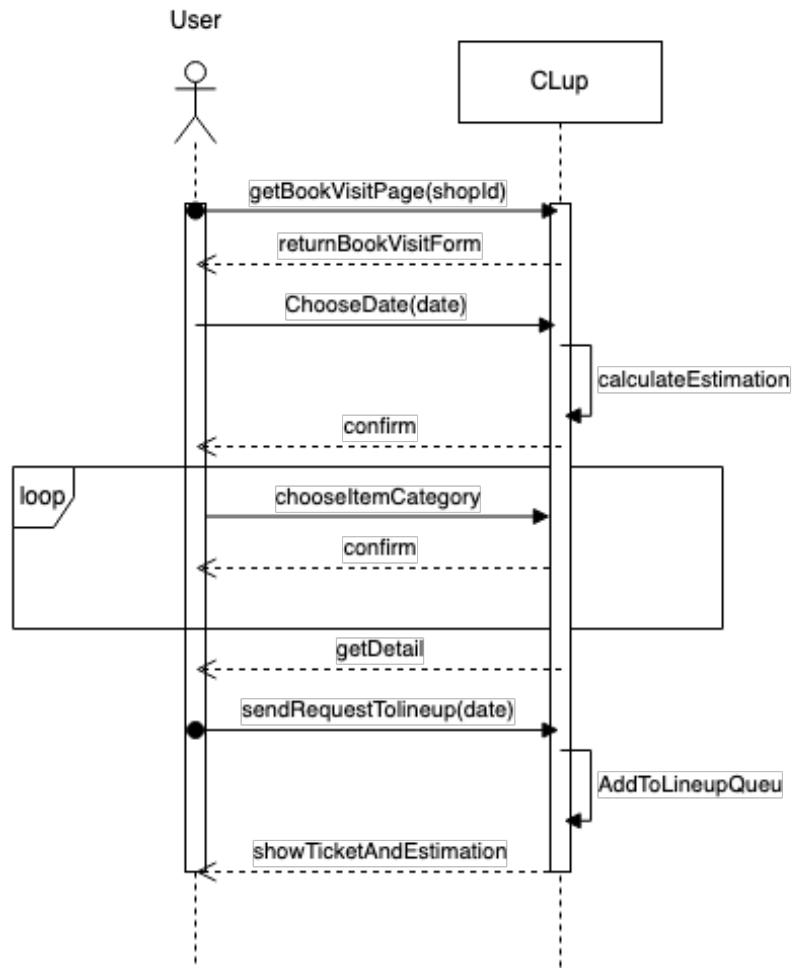


Figure 17. Sequence diagram for book a visit

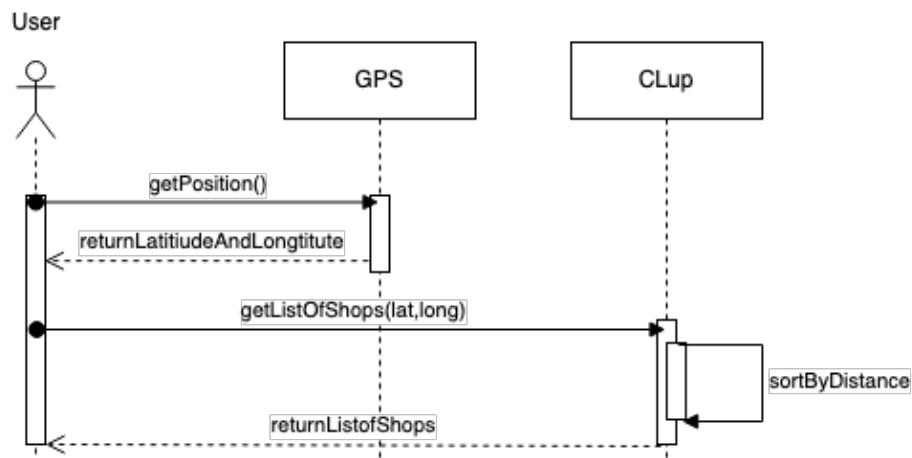


Figure 18. Sequence diagram for see shop lists

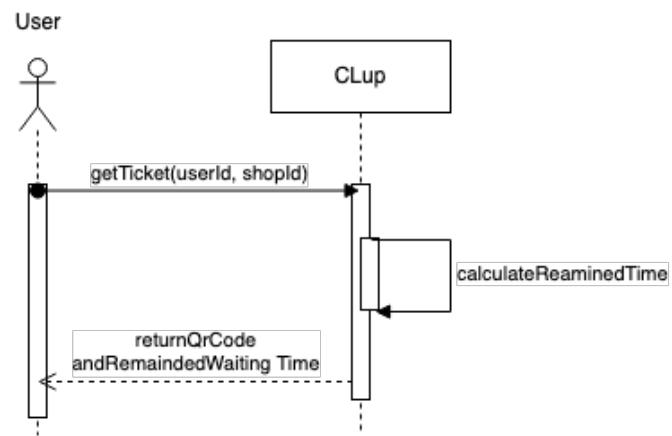


Figure 19. Sequence diagram for generate ticket

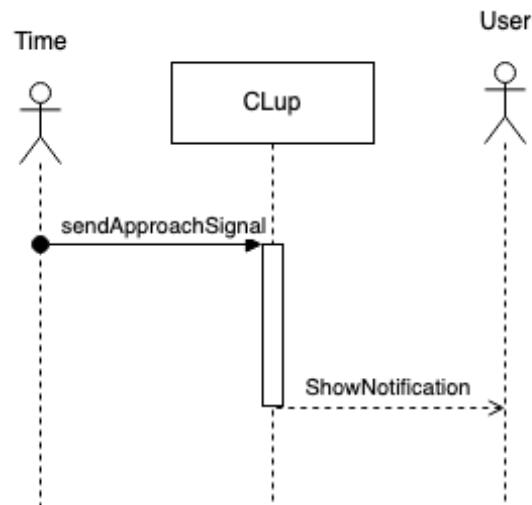


Figure 20. Sequence diagram for notify approach

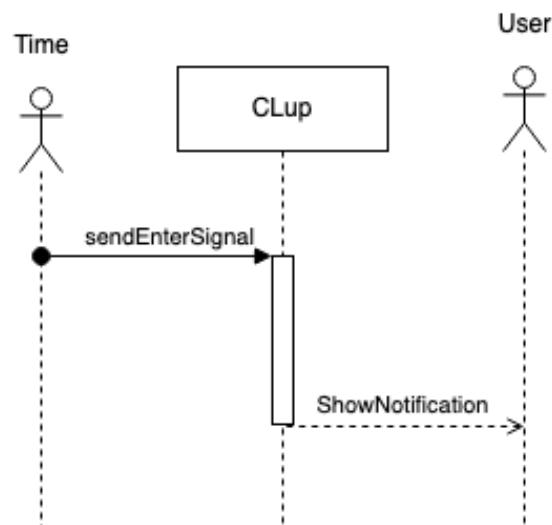


Figure 21. Sequence diagram for notify call user turn

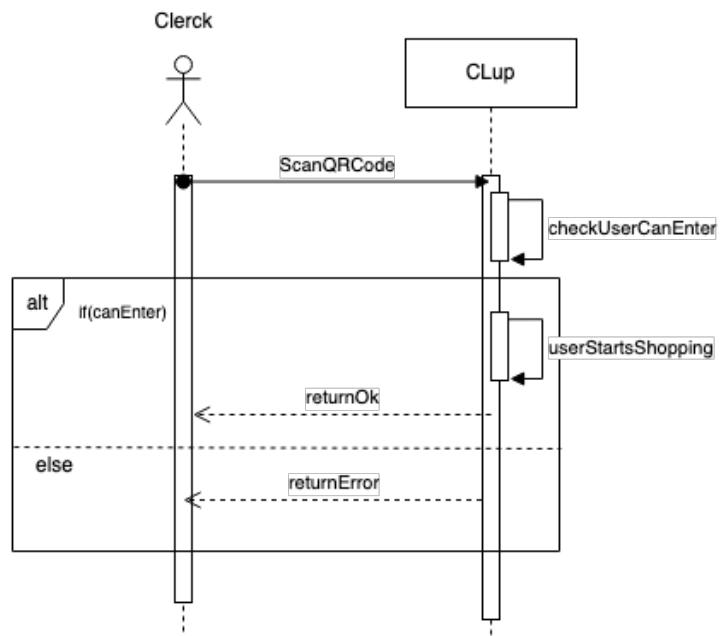


Figure 22. Sequence diagram for scan for entrance

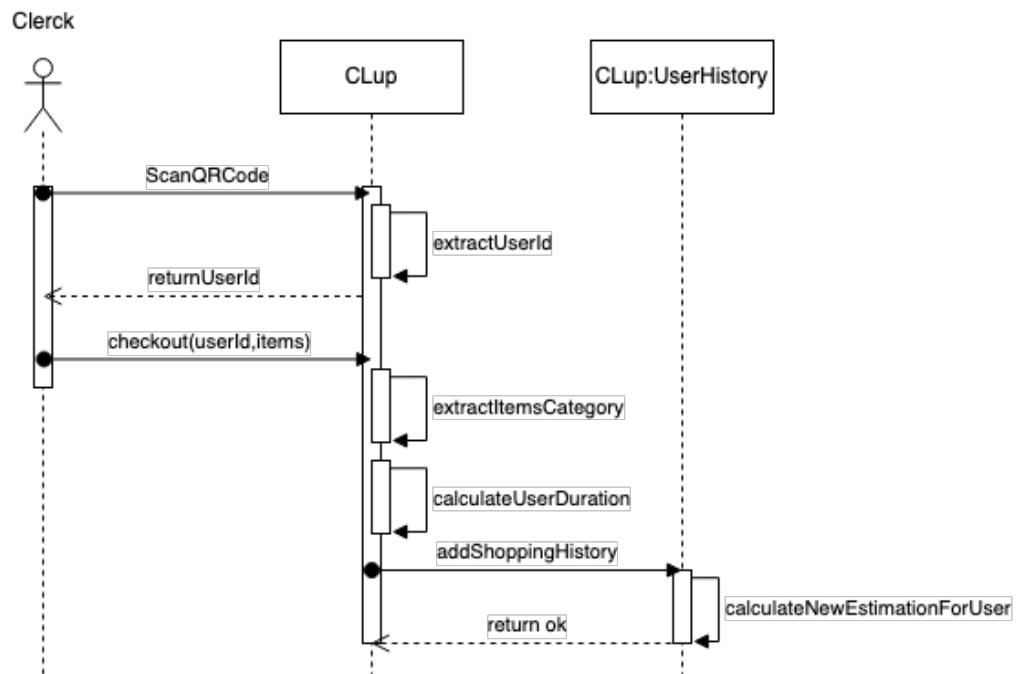


Figure 23. Sequence diagram for user checkout

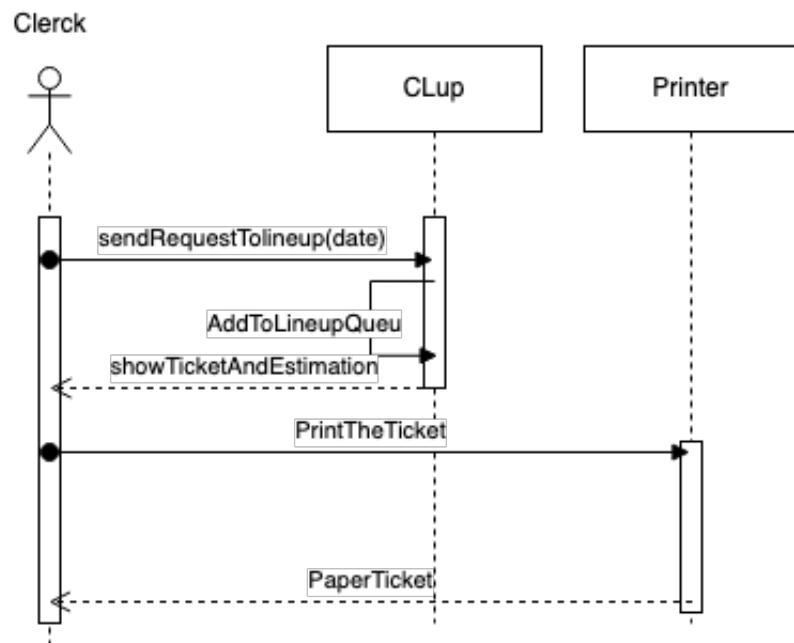


Figure 24. Sequence diagram for clerk add offline user in queue

### 3.2.5 Scenarios

#### Scenario 1: General

Sarah is an employed person who is busy all the time so managing her time is really a big deal for her this she always look for things which saves her time. on the other hand, since COVID-19 has spread all over the world, governments try to keep people's social distancing everywhere. so for supermarkets it becomes a trouble since for each supermarket, maximum 10 people can be in there and the others should wait outside in a line and it causes meters of line. in this case Sarah tends to use this application as she really cares about her time. using this app is mandatory though so it is really useful for her.

now she is running out of groceries and she want to buy some stuff today so she downloaded the application from google play and starts using it. firstly she has to sign up in the application. after inserting some information about herself, she access the first page which she should insert the supermarket name which she wants to go and then she selects the "Line up" methods as she wants to go to supermarket as soon as possible.

no she enters the page which she can see the information she inserted and and also the exact location by google map that she can access. so finally she touches the "Generate QR code" button and a bar code appears on her screen.

she can see the estimated time and number of people before her so she can manage her time to be there 5 minutes before and enter the supermarket and shop as always.

#### Scenario 2: Book a Visit

Peter wants to visit his family after so long of being far from them because he's a student and due to COVID-19 he couldn't come back to his country. so meantime he finds a flight to his country next week. so he wants to buy some stuff for his family and bring for them. so he opens the application and select the nearest supermarket and choose "Book a visit" method as he wants to go to supermarket in some days. so as soon as he select "book a visit" another part appears t which he has to select a slot which is free. so he clicks on that and selected 22th of December which was available and then he picks the perfect available time for him which is 5 P.M and finally click on the "generate QR code" button and download the pdf in case he wont be able to have connection to scan the code.

#### Scenario 3: Canceling request

Joe wants to cook pizza for his guest but he figured out that he ran out of cheese so he calls his friend to borrow some but he did not reply. so immediately he opens the CLUp application to go to supermarket to buy it. so he opens the app and choose "line-up" method and get his QR code. he looked at the estimated time and got panic cause it is about an hour and his guest is on way to his home. so he decided to cancel the request and go borrow from his neighbor. so at the bottom of the page he touched the "delete this request" part.

#### Scenario 4: invalidation of QR code

Anna is a girl who is not really on time. she is invited to her friends party tonight so she wants to buy a champagne for her friend from a supermarket. so she opens up the application and does the process for line up method till she got her QR code and her estimated time is in 30 minutes. afterward she started to get ready for that birthday party, and she forgets to check the time so as soon as she saw the clock she figured out that she just have 2 minutes to get there so she runs to get there but she is a bit late but she decides to stay in the line to see what will be happen. after 5 minutes its her turn to scan her code but her code is not valid anymore and both application and scan system shows that the QR code cannot be scanned anymore so she gets out of the line and generate another code.

#### Scenario 5: Getting tickets in person

Alina is a cute grandma to her 5 grandchild. most of the time she hold a party for them at weekends and invite them for warm traditional food. this weekend she is gonna make carbonara so she needs some ingredients. she is actually not really familiar with technology and smartphones since she is 82 years old. as always she gets ready to go to supermarket. she knows that since the COVID-19 has spread, she has to get a QR code to enter. so she goes to the ticket machine and press the "print" and then she has the code in her hand. now she can see that the estimated time for her is written 15

minutes so she decides to go and have a walk for 10 minutes in the park and come back. after about 10 minutes she comes back and stand in the line and after 5 minutes she scans her code and enter the supermarket.

#### **Scenario 6: Manager registration**

Paolo is a manager of the branch Via Benedetto Carrefour and he wants to register his supermarket in the application so that it gonna be available for people to use codes to access that. so he downloads the application from google play and enter the application as a manager after signing up. so he access the form which is about the supermarket information. first off all he enters the shop title as "Via Benedetto" carrefour and then click on the location part and insert the exact location on map. for working hours he sets from 8 A.M to 10 P.M and for the capacity of the shop he sets 12 people at the same time. moreover he select 3 categories as "Protein", "kitchen equipment" and "bread" and 4 people for each. lastly, he selects both methods for customers which are lining up and booking a visit. after submitting he sees that his request has been saved successfully and he can see the request history from the hamburger menu at top left.

### **3.3 Performance Requirements**

The system should be able to guarantee simultaneous connection. In the worst case, 500,000 user works with this application in a day. besides, many of stores are working between 8am - 8pm so we must expect some burst in our requests. Based on the previous assumptions, this system must be able to handle approximately 30 request per seconds.

### **3.4 Design Constraints - Non-Functional Requirements**

#### **3.4.1 Standards compliance**

The code should follow the requirements contained in this document. Furthermore, its comments should be clear and focused.

#### **3.4.2 Hardware limitations**

The software application requires a mobile device able to generate QR codes. In alternative, clerks can use scanners to receive the codes. also the QR codes can be generated by the ticket machine which is the second device we need. Both devices can be able to send data to the software via internet connection.

#### **3.4.3 Any other constraint**

A user has a limitation of at most 10 QR codes to generate in a single day.

### **3.5 Software system attributes**

#### **3.5.1 Simple User Interface**

The user interface has to be as simple and intuitive as possible, the application should allow an average user to set up an account and start using the application understanding its functionality in no more than a dozen minutes. This application must be designed for all range of users and it must be suitable for people have disabilities. In addition there should be a complete tutorial to make it easy using the application.

#### **3.5.2 Reliability**

The application provides a reliable service in which individual users can easily log in and go to a line up or book a visit in the most optimal way. Furthermore, it warranties that the chain of custody of the information coming from the users is never broken, and the information is never altered. This would provide a secure and reliable system.

### **3.5.3 Availability**

The application must offer availability in the order of 99% granting with in 8am-8pm(most of shops works in this hours). The lack of service must be minimal. It's better to better to maintain system at nights because people mostly do not use this application at night and many shops are closed. Even at night, we could put some resources at sleep mode to reduce the cost of application.

### **3.5.4 Security**

The users' location is a very important data and must be encrypted so this part must be more secure than other data stores in the application. Moreover, in case of password recovery this should never be sent in clear. The system must be behind a proxy servers (like cloudflare) to prevent the DDoS attacks.

### **3.5.5 Maintainability**

The application must keep a service log in order to fix bugs more easily. The app should be developed using a micro-service approach, so adding new functions shouldn't require to change the previous code, and we can have different instance of services in different time of day. This application needs infrastructure like kubernetes to orchestrate the containers.

### **3.5.6 Portability**

The user side must be available in different platform like Android, IOS and Web. The backend side must be deployed with docker to easily deploy on any server.

### **3.5.7 Scalability**

Because this application is very useful to citizens and now it's a pandemic and people tend to be in home and not in the long lines, so many people will use this app and many managers add their shops in this app thus, so the system must have a good infrastructure which could balance the loads and easily we must add new resources to it.

## 4 Formal Analysis Using Alloy

### 4.1 Alloy Code

```
sig Float {}  
// Geo location  
sig Position {  
    latitude: one Float,  
    longitude: one Float,  
}  
  
// time stamp which start from 1 jan 1970  
sig DateTime {  
    timestamp: one Int  
} {  
    timestamp >= 0  
}  
  
// define Super user  
sig Username, Password, Certificate {}  
abstract sig SuperUser {  
    id: one Int,  
    username: one Username,  
    password: one Password  
} {  
    id >= 0  
}  
  
// define Manager  
sig Manager extends SuperUser {  
    certificate: one Certificate,  
} {  
    id > 0  
}  
  
// define User  
sig User extends SuperUser {  
    currentLocation: one Position,  
    historyVisit: set UserVisitHistory,  
    tickets: some Ticket  
} {  
    all t: tickets | t.qr.userId = id  
}  
  
// This is an anonymous user which get the ticket offline we  
choose id = 0 for them  
sig OfflineUser extends User {} {  
    id = 0  
}
```

```

// define category
sig Category {
    capacity: one Int
} {
    capacity >= 0
}

// define the shop
sig Shop {
    id: one Int,
    manager: one Manager,
    capacity: one Int,
    categories: set Category,
    location: one Position,
    queue: one LineupQueue,
    start: one DateTime,
    end: one DateTime
} {
    id >0
    capacity >= 0
    start.timestamp < end.timestamp // shop can close
        shop after time it start it
}

// define User history
sig UserVisitHistory {
    start: one DateTime,
    end: one DateTime,
    categories: some Category
} {
    start.timestamp < end.timestamp // checkout time
        must be after user enter the shop
}

// define Ticket
// QR must encode the shopId and userId to a qrcode
sig QRCode {
    shopId: one Int,
    userId: one Int
}

// State of ticket for entrance
abstract sig State {}
one sig Valid extends State{}
one sig Invalid extends State{}

sig Ticket {
    qr: one QRCode,
    estimatedEntranceTime: one DateTime,
    state: one State
}

```

```

}

// define line up queue
sig LineupQueue {
    tickets: set Ticket,
    capacity: one Int
} {
    capacity >= #tickets // number of people are in the
    queue must be less than its capacity
}

// shops cannot have same queues
fact {
    no disj s1, s2: Shop | s1.queue = s2.queue
}

// define book a visit
sig BookAVisit {
    duration: lone Int,
    categories: some Category,
    selectedTime: one DateTime,
    ticket: one Ticket
}

////////// FACTS

// no same id and username for super users
fact {
    no disj p1, p2: SuperUser | p1.id = p2.id or p1.
        username = p2.username
}

// no same id for shop
fact {
    no disj s1, s2: Shop | s1.id = s2.id
}

// no same QR for each ticket
fact {
    no disj t1, t2: Ticket | t1.qr = t2.qr
}

// selected time must be available on that shop lineup
fact {
    all b: BookAVisit | one s: Shop |
        b.ticket.qr.shopId = s.id and
        b.selectedTime.timestamp >= s.start.
        timestamp and
}

```

```

        b.selectedTime.timestamp < s.end.timestamp
    }

// no same ticket for books
fact {
    no disj b1, b2: BookAVisit | b1.ticket = b2.ticket
}

// each ticket shop and user must exists
fact {
    all t: Ticket | one s: Shop | t.qr.shopId = s.id
    all t: Ticket | one u: User | t.qr.userId = u.id
}
///////////////////////////////
// PREDICTIONS

// create shop
pred createShop(m: Manager, s: Shop, l: Position, st:
    DateTime, et: DateTime) {
    s.manager = m
    s.location = l
    s.start = st
    s.end = et
}

run createShop

// add to line up
pred addToLineup(u: User, s: Shop, t: Ticket) {
    t.state = Invalid
    t.qr.shopId = s.id
    t.qr.userId = u.id
    s.queue.tickets = s.queue.tickets + t
}

run addToLineup

// book a Visit
pred bookVisit(u: User, s: Shop, t: Ticket, st: DateTime, b:
    BookAVisit) {
    t.state = Invalid
    t.qr.shopId = s.id
    t.qr.userId = u.id
    s.queue.tickets = s.queue.tickets + t
    b.ticket = t
    b.selectedTime = st
}

run bookVisit

```

```

// offline add to lineup


pred offlineLineup(u: OfflineUser, s: Shop, t: Ticket) {


    t.state = Invalid
    t.qr.shopId = s.id
    t.qr.userId = u.id
    s.queue.tickets = s.queue.tickets + t
}



run offlineLineup



// notify for entrance


pred notifyEntrance(u: User, t: Ticket) {


    t.qr.userId = u.id
    t.state = Invalid
}



run notifyEntrance



// enter the shop


pred allowEnter (u: User, s: Shop, t: Ticket, h: UserVisitHistory, n: DateTime){


    t.qr.userId = u.id
    t.qr.shopId = s.id
    t.state = Valid
    t in s.queue.tickets
    h.start = n
    u.historyVisit = u.historyVisit + h
}



run allowEnter



pred rejectEnter (u: User, s: Shop, t: Ticket){


    t.qr.userId = u.id
    t.qr.shopId = s.id
    t.state = Invalid or t not in s.queue.tickets
}



run rejectEnter



// checkout


pred checkout (u: User, s: Shop, t: Ticket, h: UserVisitHistory, n: DateTime, c: some Category){


    t.qr.userId = u.id
    t.qr.shopId = s.id
    t.state = Valid
    s.queue.tickets = s.queue.tickets - t
    h.end = n
    h in u.historyVisit
    h.categories = c

```

}

**run** checkout

## 4.2 Meta Model

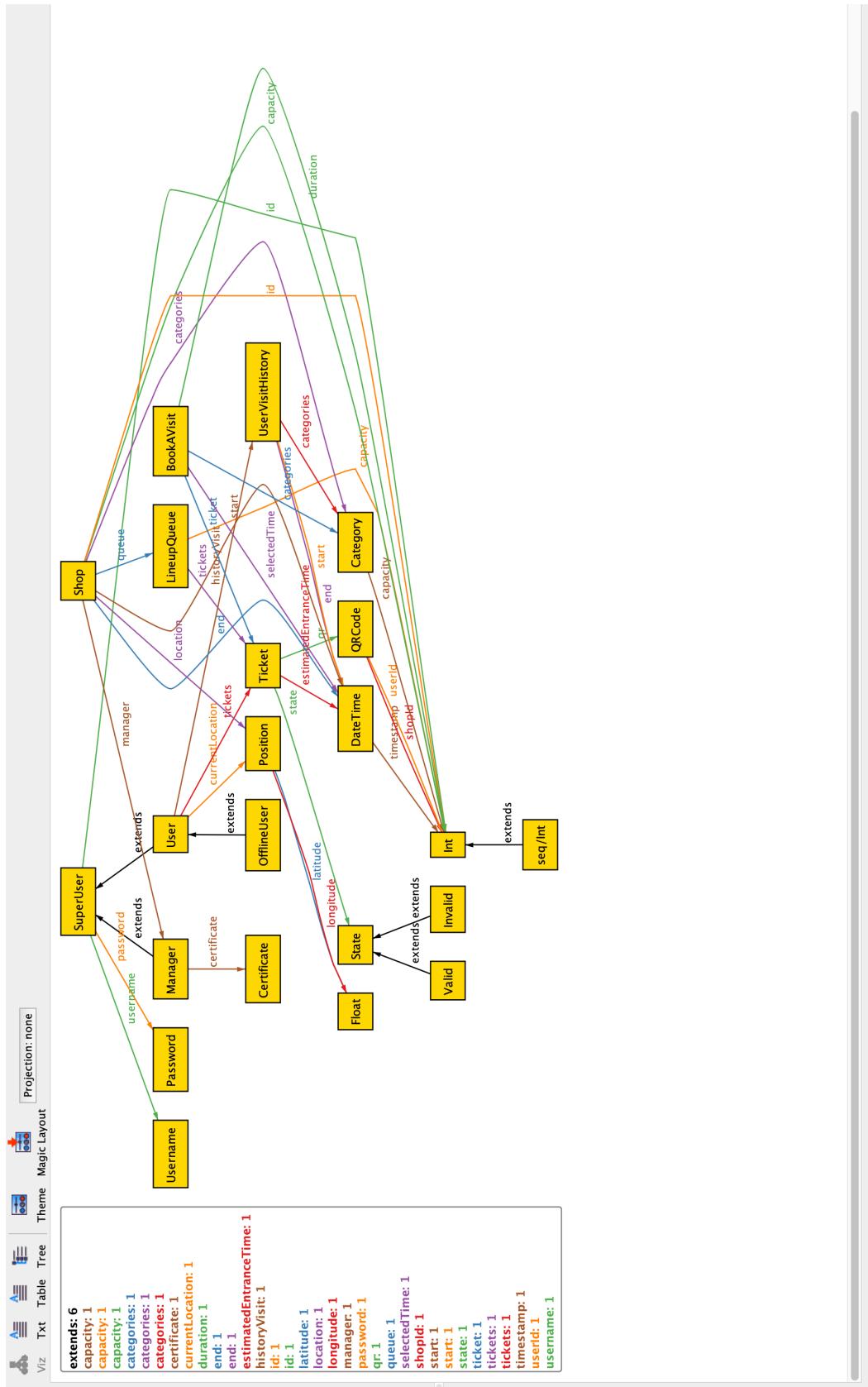
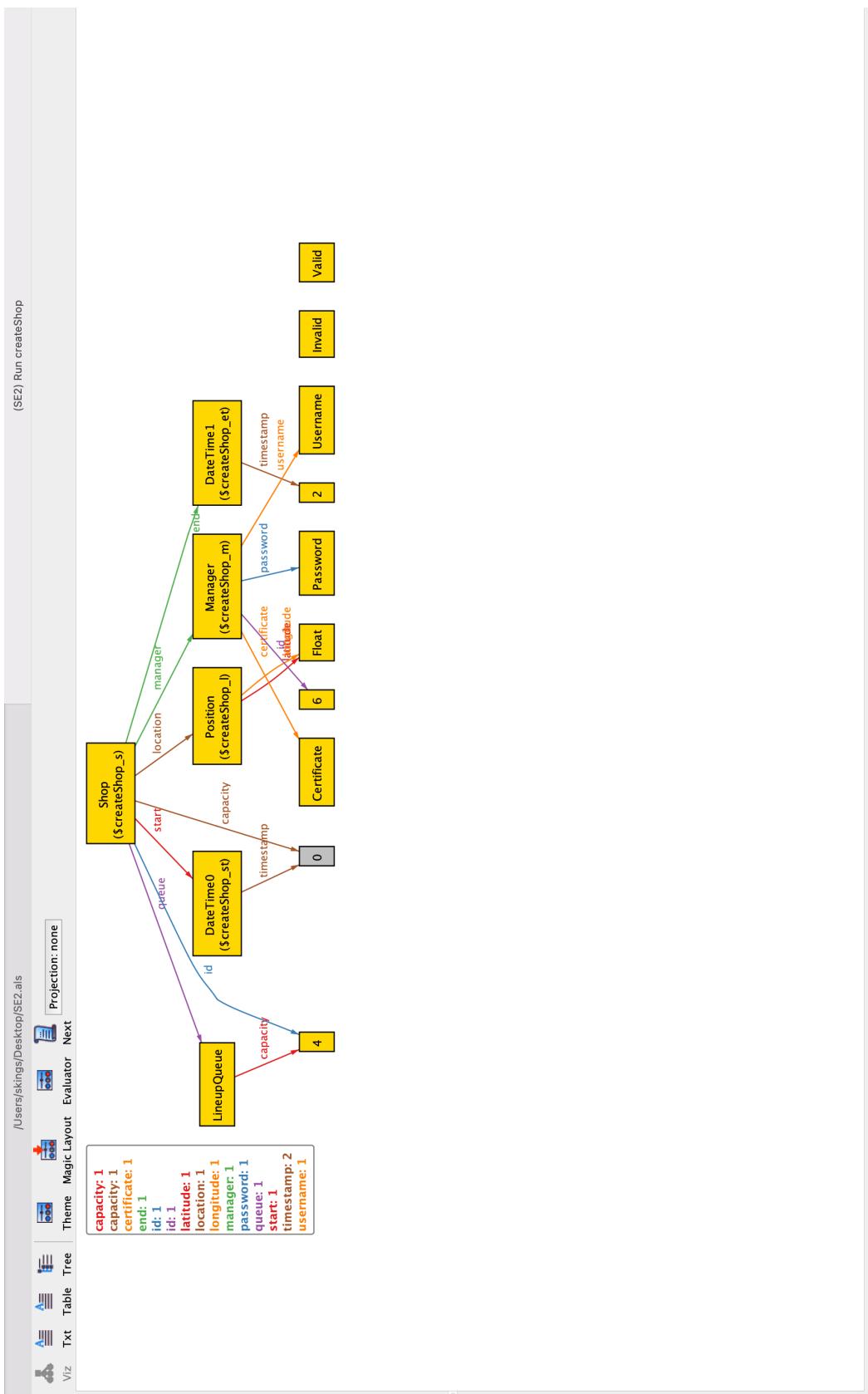


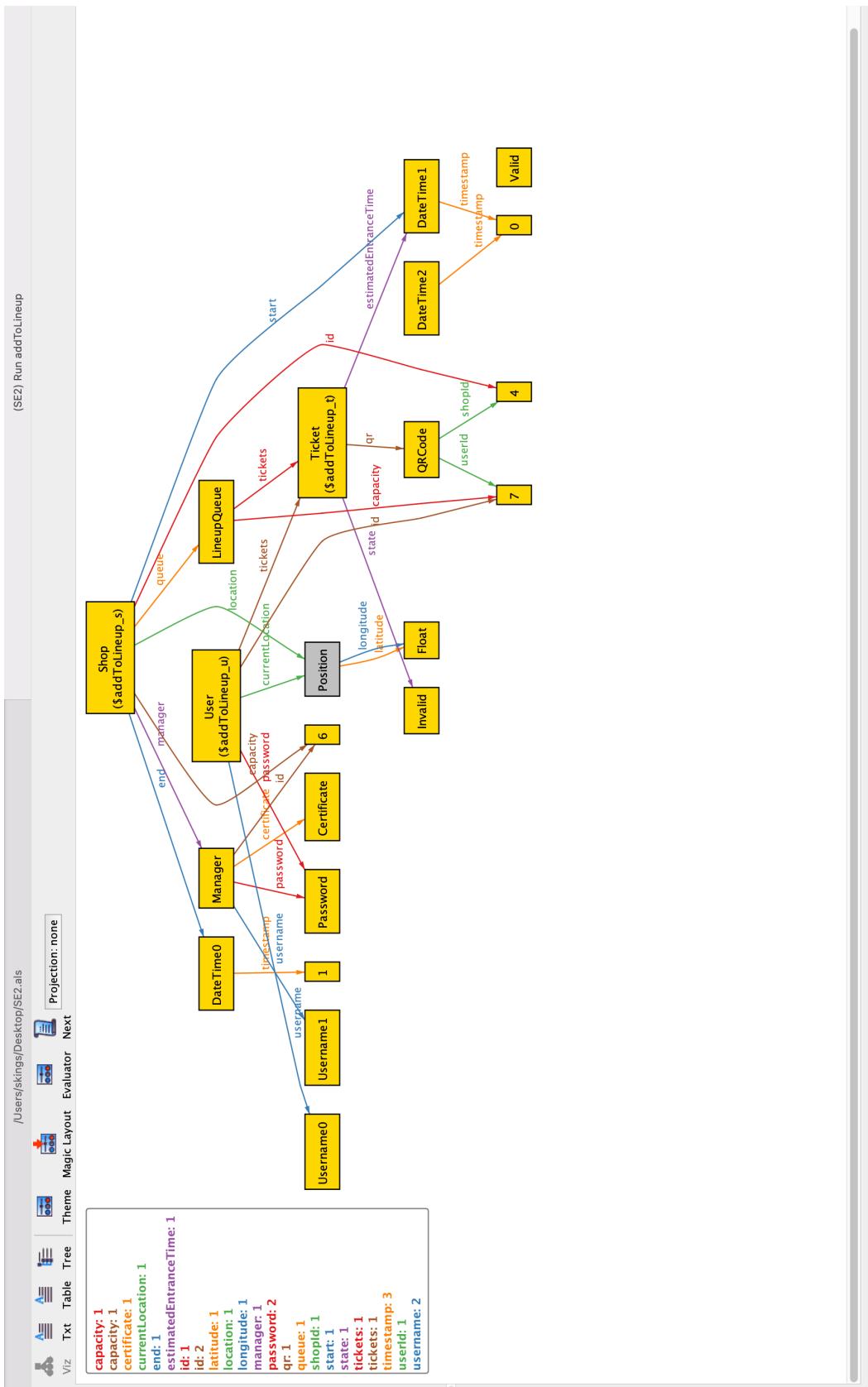
Figure 25. Meta model

### 4.3 Result for predictions

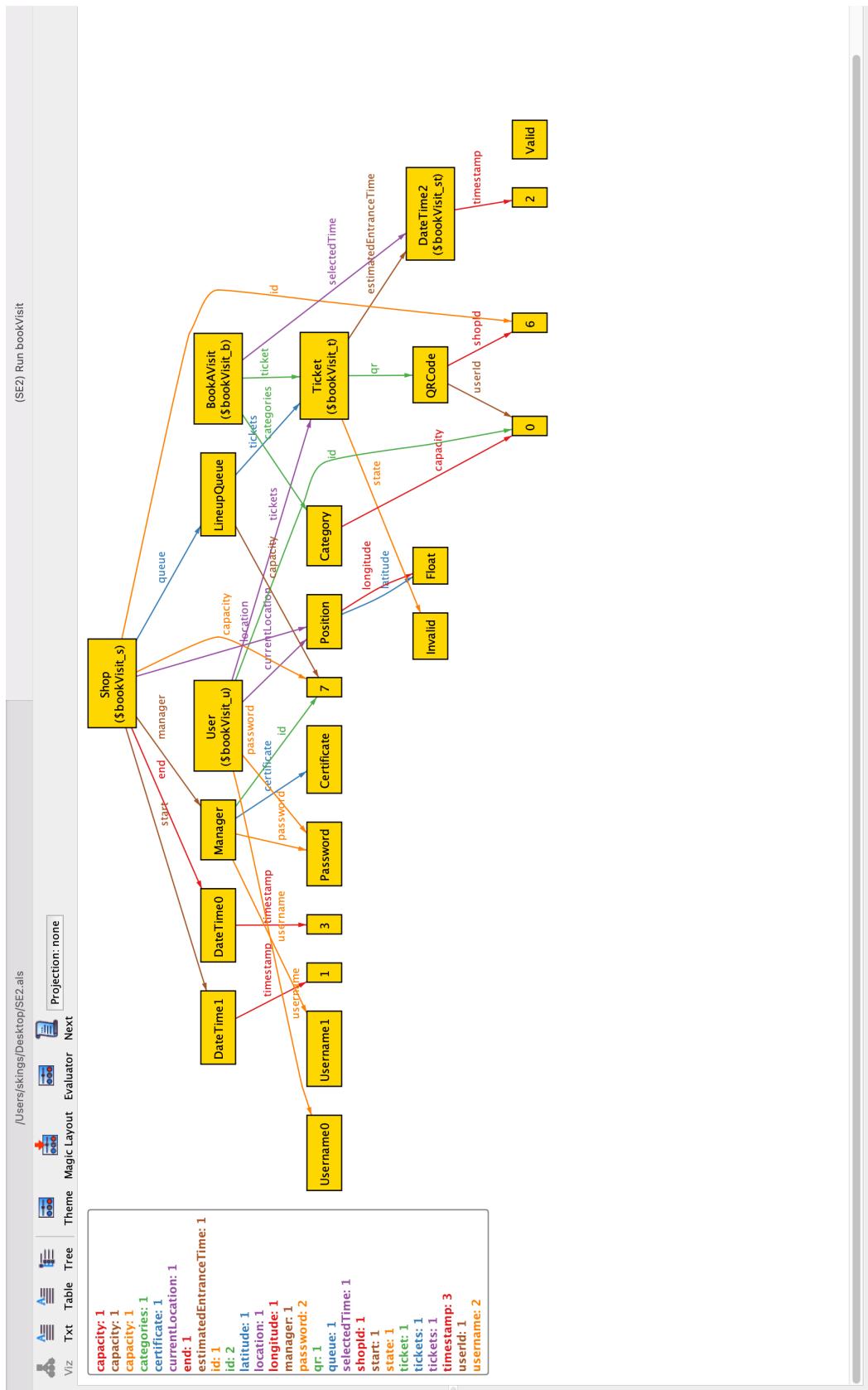
1. Create shop



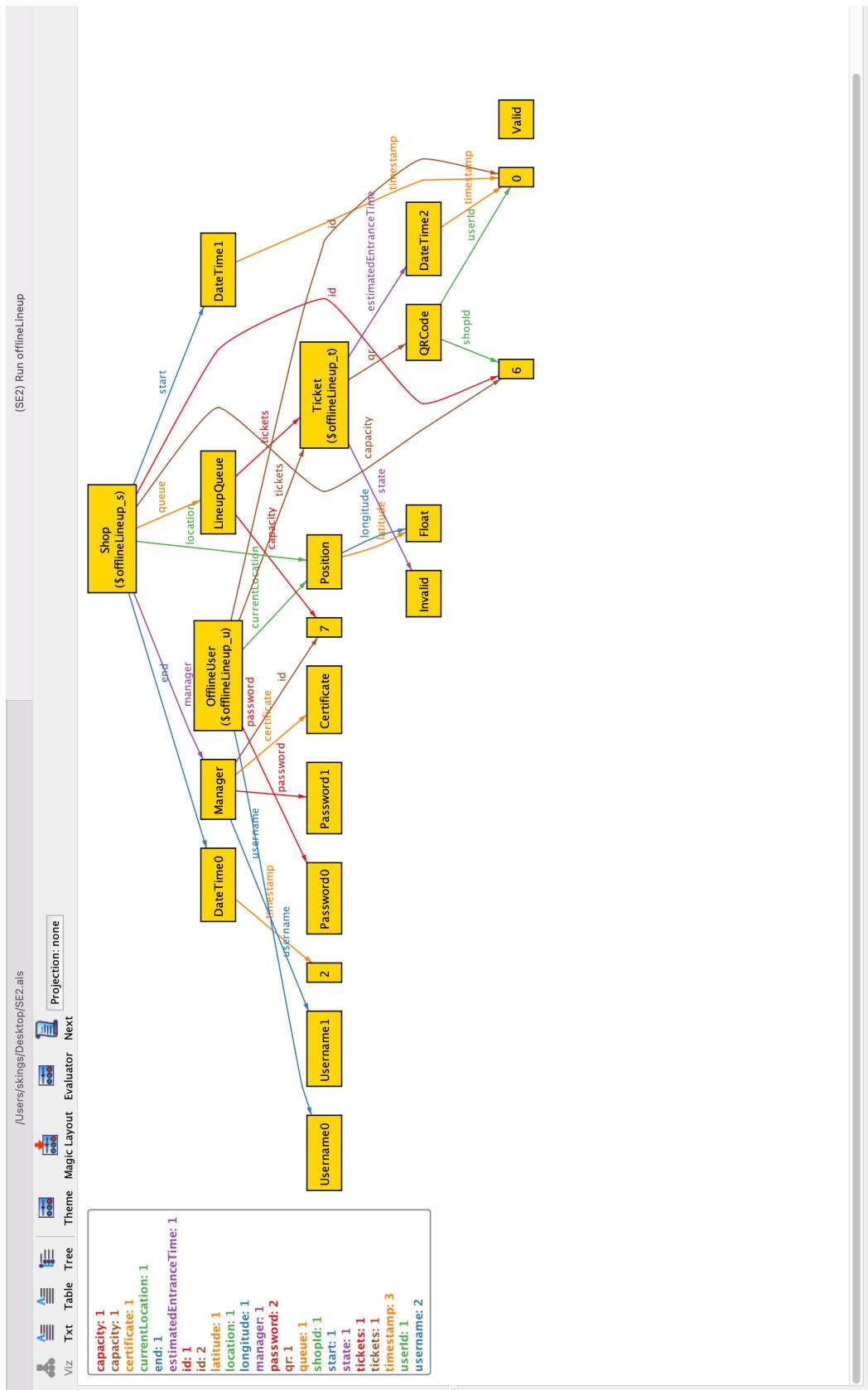
## 2. Add to line-up



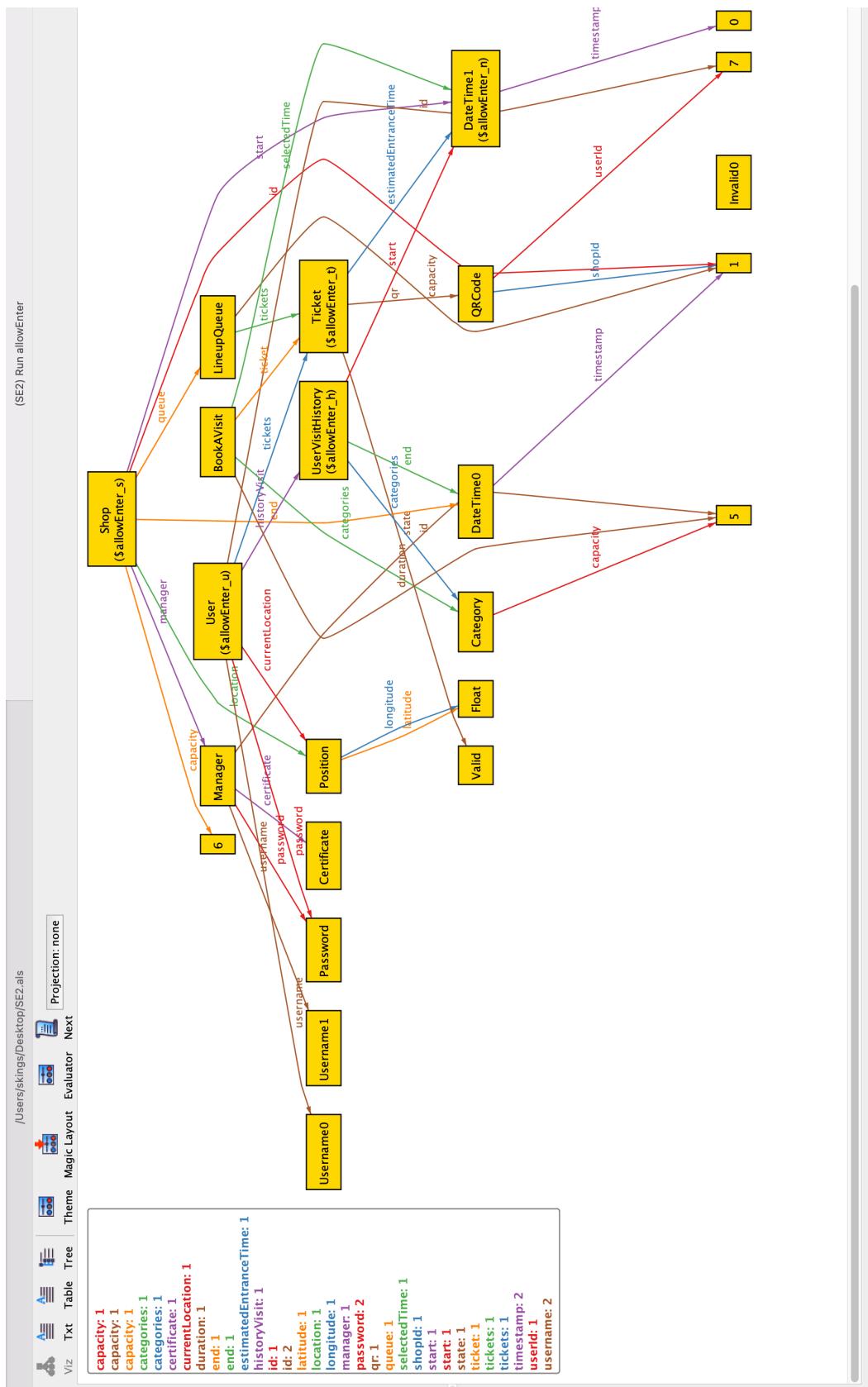
### 3. Book a visit



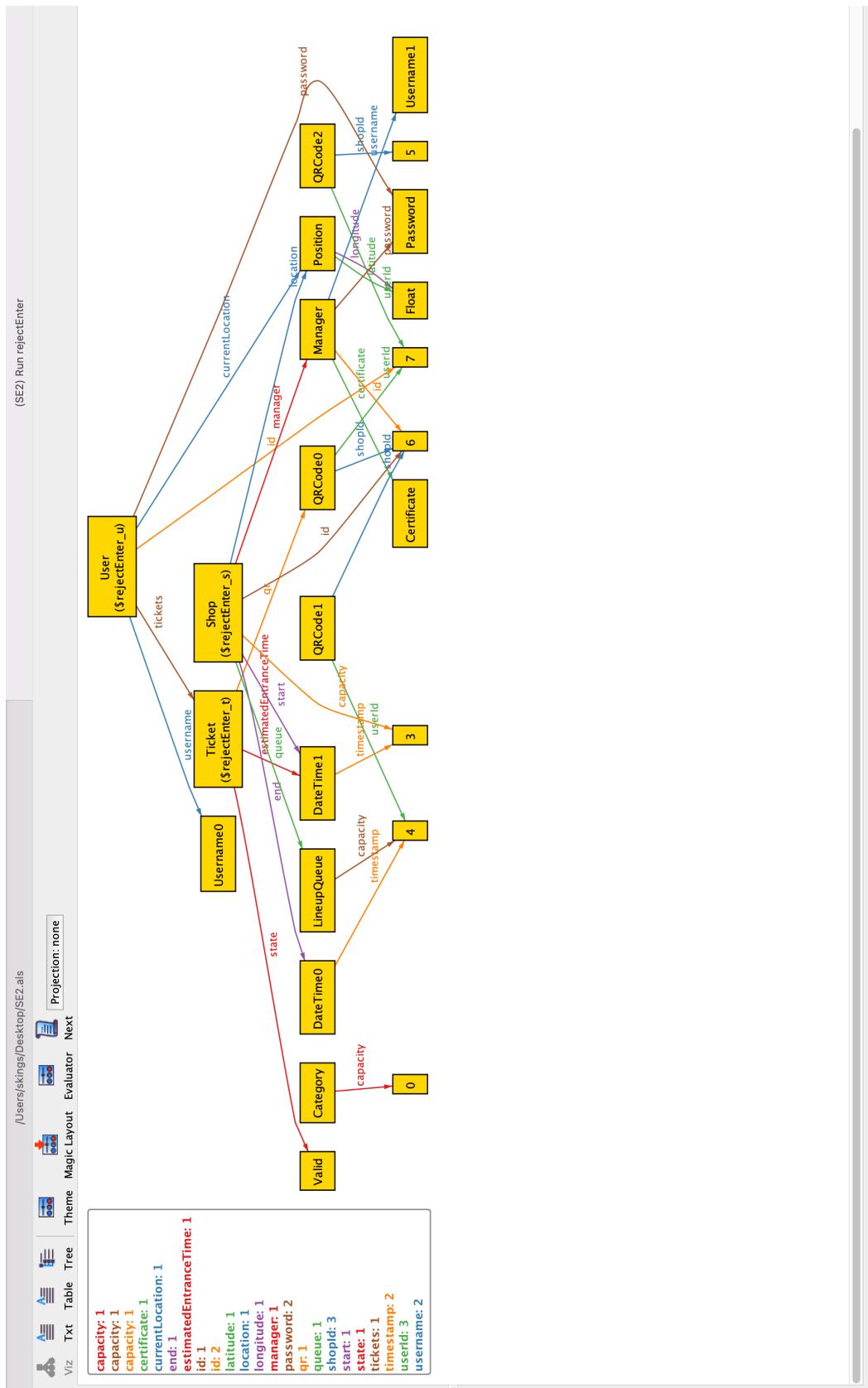
#### 4. Get an offline ticket



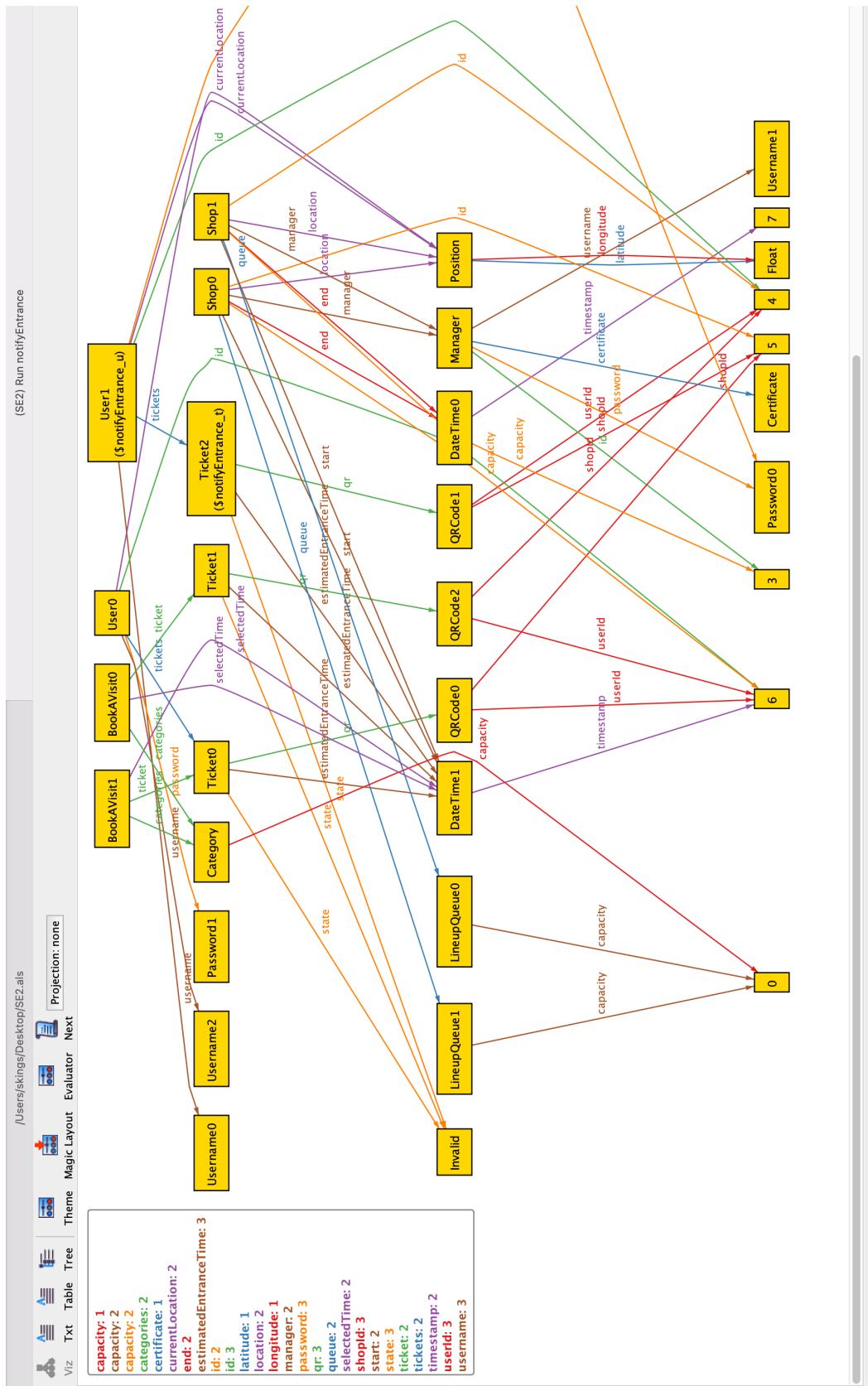
## 5. Allow Entrance



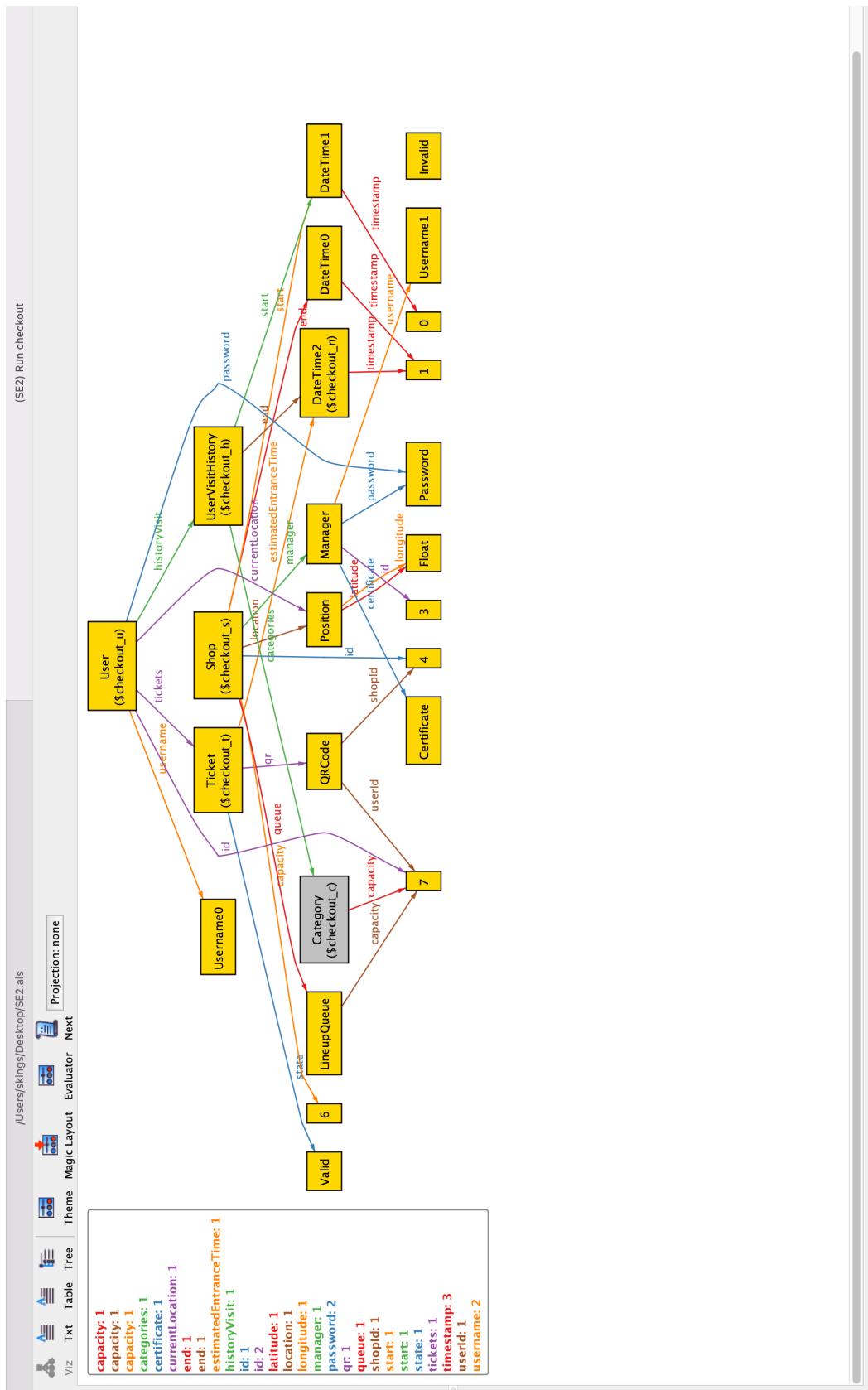
## 6. Reject Entrance



## 7. Notify Entrance



## 8. Checkout



## 5 Effort spent

Saman Fekri:

Topic	Hours (hh:mm)
First meeting	1:30
Discussion on first part	3:00
World & Shared phenomena	1:00
UML Description & State charts	4:00
User characteristics	2:00
List of Requirements	3:00
Mapping	2:00
Use cases	4:00
Sequence Diagram	6:00
Scenario	1:30
Software System Attributes	3:00
Alloy	12:00
Document Revision	2:00

Parniya Saeedzadeh:

Topic	Hours (hh:mm)
First meeting	1:30
Discussion on first part	3:00
Goals	1:30
UML Description & state charts	3:00
Product functions	2:00
Domain assumptions	2:00
External Interface Requirements(+Mockups)	10:00
mapping	4:00
Use cases	2:00
Sequence Diagrams	2:00
Scenarios	2:00
Design Constraints - Non-Functional Requirements	2:00
Alloy	5:00
Document Revision	2:00

## 6 References

- All diagrams are draw by the [app.diagrams.net](http://app.diagrams.net) (draw.io)
- Alloy code development had been support by [MIT](http://MIT)