POLITECNICO
MILANO 1863

Software Engineering 2
CLup project by
Robert Medvedec
Toma Sikora

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Robet Medvedec, Toma Sikora
Version:	1.0
Date:	15-November-2020
Download page:	https://github.com/robertodavinci/Software_Engineering_2_Project_Medvedec_Sikora
Copyright:	

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The purpose of this document is to serve as a Requirement Analysis and Specification Document (RASD) for the development of the CLup - Customer Line-up application.

It will clearly introduce the problem at hand, propose an adequate solution and explain it in detail. It will do so through the use of appropriate language, with software engineers, system and requirement analysts, and product testers as the target audience.

The document fully details the scope and the basic functions of the system. Furthermore, it expands the core functionality with conceivable upgrades and improvements. It characterizes requirements, assumptions, and constraints of the system and outlines its goals.

CLup is a mobile application that helps grocery store chains manage the influx of customers and reduce crowding both inside and outside of the store. Moreover, it enables people to reserve a timeslot in a specific store through the "book a visit" feature.

1.2 Scope

1.2.1 Description of the problem

Faced with a worldwide pandemic of the COVID-19 virus countries across the world imposed strict health measures in line with the recommendations of the World Health Organization.

Most governments introduced decrees that limited the movement of the population to a certain degree. They did so in hope of reducing the spread of the virus. Only essential movement, such as: going to work, grocery shopping or outdoor exercise, was deemed acceptable.

Although successful in the mitigation of the disease, the act put a serious strain on society on many levels. For the time being, people are slowly becoming annoyed with the measures and uncertainty, occasionally leading to protests.

To help reduce the stress and anxiety, many aspects of everyday life involving close contact can be considered and improved upon.

1.2.2 Proposed solution

This project aims to help with, and resolve the issues surrounding grocery shopping. As we all know, grocery shopping is an essential activity which involves close contact inside the store. Since the COVID-19 virus spreads mainly through airborne particles, this activity plays a key role in its mitigation.

To reduce crowding inside the stores, supermarkets need to restrict access to their store and keep the number of people inside below the optimal maximum capacity. A solution to this seemingly easy task can be found by looking at the way banks and other community services handle the issue. In the most common situation, after retrieving a personal number at a printer, people wait in a line according to their ticket number.

This system can serve as a good base for the solution but does not come without flaws. Firstly, the interaction with the ticket printer normally includes the use of a touchscreen. To avoid constant wiping and sanitizing of the screen, the system should not use a printer's touchscreen as an interface.

Furthermore, while waiting outside for a number to be called dramatically reduces indoor crowding, if the current number is shown on a screen in the store, outdoor crowding is inevitable. To avoid larger gatherings, a person should be able to have an idea of how long until their number is called without nearing the store. In such a way, a person could enter a store without physically waiting in a line and avoid close contact.

To resolve the issues that arise with available solutions because of the pandemic, a software application can be used. The application could provide a virtual counterpart to a physical line up in front of a store. The main idea is to enable store customers to enter a queue from home (or wherever they find themselves) through simple interaction with the application.

Upon the request, the person receives a number and a QR code. The person waits until that number is called to approach the store, and before entering, camera or store personnel, if available, scans the code to check its validity. Such a system allows store managers to monitor entrance in the store and control the influx of the customers.

Introducing such a system has its drawbacks and consequences, some more significant than others. The most obvious consequence is that upon implementing such a system, to attain the number of people inside the store under a certain threshold, all customers will be obligated to use it to enter the store. Creating a system that is usable, intuitive, and clear for all demographics of a society is hard, but necessary. The application should, therefore, be very simple to use. Moreover, to make sure every customer can access the grocery store, a solution should also be available for people who do not have access to the technology required by the system. The easiest solution is to have a traditional ticket printer in front of the store.

A simple intervention of removing the touchscreen and printing a new ticket as soon as the last one is taken addresses the problems regarding the printer. Another important area to address is the effectiveness of the mechanism. Just like in real life physical lines, if a person arrives before/after their number is called the system should send the person back in line. To minimize the frequency of such events, the developed solution should be capable of calculating a reasonably precise estimation of the wait time for the customer. A basic approach is to simply provide the number of people in line that are ahead of you, but more precise estimations can and should be implemented. Furthermore, to avoid the loss of one's place in line, the system should send occasional notifications to the user, to remind them of and update the estimated waiting time.

After having resolved all conceived problems of the system, one more suggestion can be proposed to enhance its convenience and user experience. Besides managing crowding inside the store and real-time queueing, the application will give customers the option to "book a visit" to the grocery store. This feature will allow them to view available time slots for their grocery shop and book the most convenient one. Also, during the booking process, a person will have an option to indicate an approximated duration of their visit to further improve the accuracy of the wait time estimation of the system.

1.2.3 Domain

The target audience for this application includes every person that shops for groceries in a store. Almost all demographics fall into this category, specifically people of age that use a smartphone (although a solution is given even for those who do not), have access to a store and live in fairly densely populated areas.

To use the application the person would have to have a smartphone and know how to use it, along with internet connection so the application can communicate with the database. This excludes some groups of the society, especially elderly ones, but still vast majority of young and working-class people should have easy access to it.

1.2.4 Goals [Gn]

G1 Allow the user to "line up" /retrieve a number.

- **G1.1** Allow the user to retrieve a number through the application.
- **G1.2** Allow the user to retrieve a number physically from the printer.

G2 Allow the store manager to control the entrance of the user via QR code scanning.

G3 Allow the user to get precise calculations of the wait time.

G4 Allow the user to get updates/notifications on the estimated wait time.

G5 Allow the user to "book a visit" to the store.

- **G5.1** Allow the user to "book a visit" to the store without indicating the expected duration of the visit.
- **G5.2** Allow the user to "book a visit" to the store with indicating the expected duration of the visit.

G6 Allow the user to automatically indicate the expected visit duration while booking a visit.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Application**: a computer (mobile) program that is designed for a particular purpose.
- **QR code**: a machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera or a scanner.
- **Smartphone**: a mobile phone that performs many of the functions of a computer, typically having a touchscreen interface, internet access, and an operating system capable of running downloaded apps.

1.3.2 Acronyms

- **RASD**: Requirement Analysis and Specification Document.
- **COVID-19**: Virus responsible for the spread of the coronavirus disease 2019.
- **CLup**: Customer Line-up.
- **API**: Application programming interface, computing interface which defines interactions between multiple software intermediaries

1.3.3 Abbreviations

- **Gn**: nth goal.
- **Dn**: nth domain assumption.
- **Rn**: nth functional requirement.
- **App**: Application.

1.4 Revision History

- **Version 0.1**: First edition; 7.11.2020.
- **Version 0.2**: First check and added some stuff; 9.11.2020.
- **Version 1.0**: First .tex document created and added all together; 15.11.2020.

1.5 Reference Documents

- Specification document "R&DD Assignment A.Y. 2020-2021.pdf"
- Alloy Dynamic Model example:http://homepage.cs.uiowa.edu/~tinelli/classes/181/Spring10/Notes/09-dynamic-models.pdf
- Presentations Software Engineering 2, Politecnico di Milano

1.6 Document Structure

The structure of this document is divided into six chapters. The first chapter gives an elaborate introduction to the problem at hand and the proposed solution. In the first part of the chapter the purpose of the document and the goals of the project are presented. In the second part the project's scope is given, outlining the description of the problem, proposing a solution, and defining the domain of the problem. The third part includes the definitions, acronyms, and abbreviations necessary to understand the project. Fourth and fifth part of the chapter provide an oversight of the revision history of this document and a list of reference documents, and the last part presents the structure of the document.

The second chapter gives an overall description of the product. In the beginning introducing the product perspective, containing scenarios and further details on the shared phenomena and a domain model. It also introduces the product functions with the most important requirements and user characteristics. In the end, necessary assumptions are displayed together with the system's dependencies and constraints.

The third part includes all the specific requirements of the system, explained in more detail where necessary, to help the development team. It includes external interface requirements, functional and performance requirements, design constraints and software system attributes. The fourth part provides a formal analysis using Alloy, to prove the feasibility and soundness of the system. A formal model is presented and described and so are some worlds obtained by running it.

The fifth part provides information about the number of hours each group member has spent working on each part of this document. The sixth part contains a list of references such as the tools used to create the content of the document.

2 Overall Description

2.1 Product perspective

2.1.1 Internal structure

The proposed system uses an application on the server side that is connected with the APIs to the Android or the iOS app on the phone.

The way of communication between the parts of the system and logic behind the system is presented in the following figure:

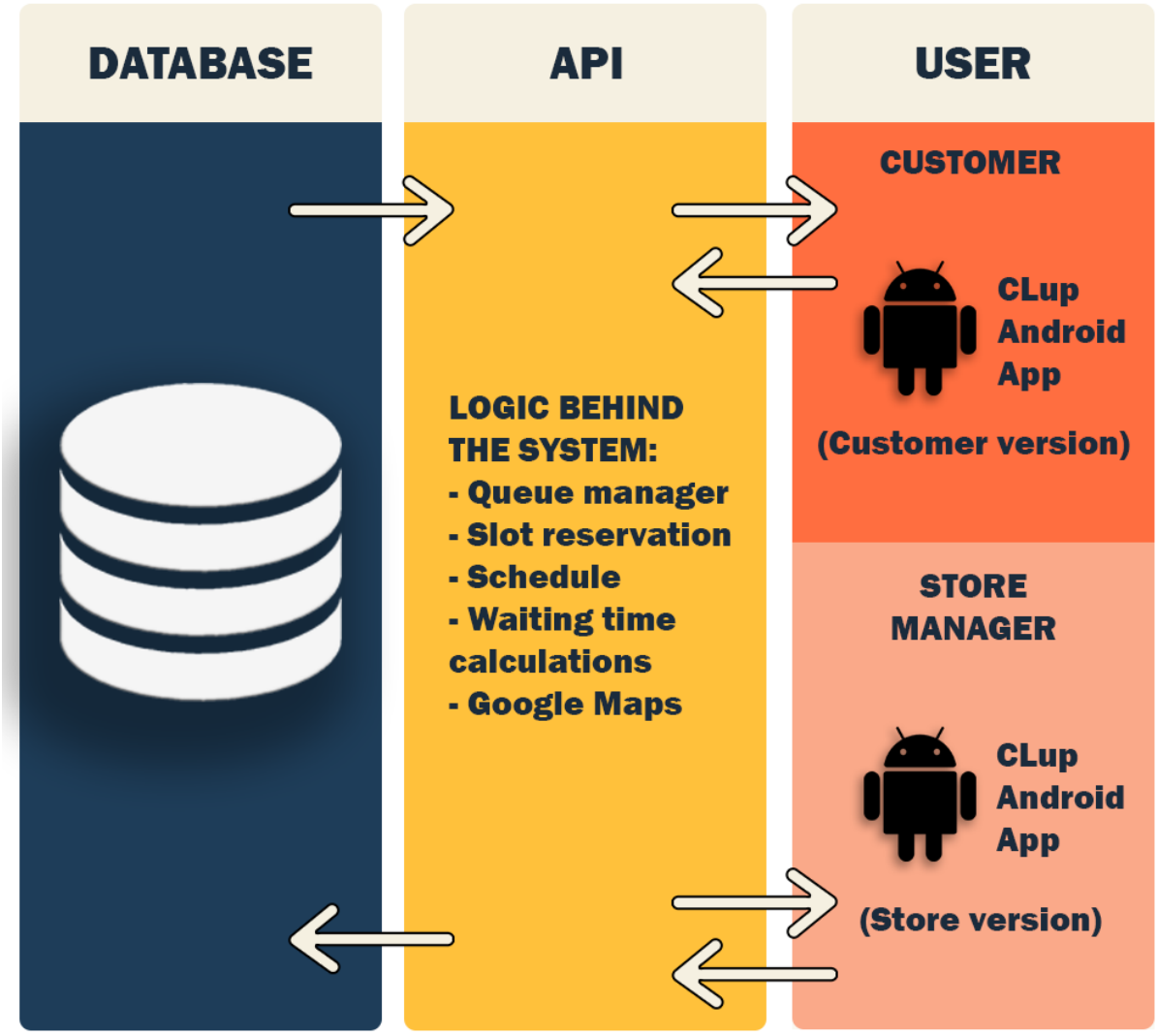

Figure 1: **Internal structure of the system**

The main part of the logic behind the system is on the server side. All the data taken from the users is stored in the database and all the data that users request, like stores' working hours, location, and available time slots, are also taken straight from it.

Since there is no need for users to have accounts, their personal data is not permanently stored in the database, therefore allowing users to use the app quickly and safely.

API is used to do all the work in between. Calculating waiting time, slot reservation, scheduling, queue management, and calculating distance is done based on the request by the user and with using the data provided by the database.

Another part of the app is the store manager version, which requires login to connect a store manager to the specific store, so he/she can manage the influx of people to and from the store.

State chart diagrams

Core functioning of the system is shown in this chapter using the state chart diagrams. These are simplified representations of the process that is happening during the most basic functionalities of the application and the system. Following state charts will be further explained and deepened through other diagrams in the following chapters.

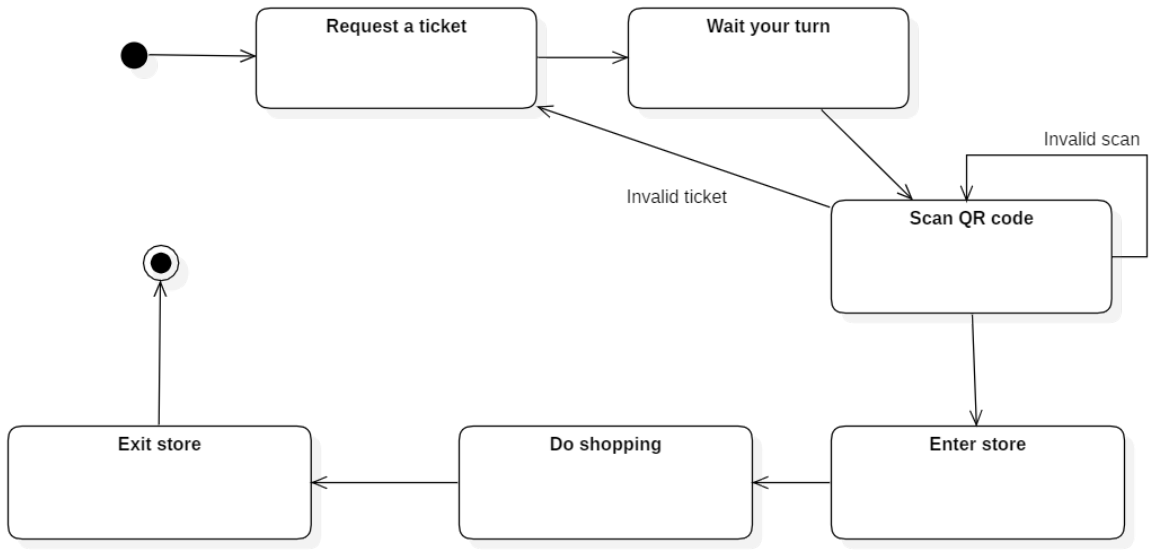

Figure 2: **State chart diagram 1 - Request and scan**

State chart 1 represents the most basic function of the CLup app - requesting and using a ticket. This diagram can be used both for the ticket request through the app or for taking the ticket from the printing machine. The mechanism works the same way. Once the ticket has been acquired by the customer, and the ticket's number gets called (either by the store manager or by the screen that shows the queue), the user must enter the store and present the ticket on his way in. The ticket will then be scanned and either accepted or denied by the store manager. After successfully entering the store, the customer can do the shopping and exit the store normally.

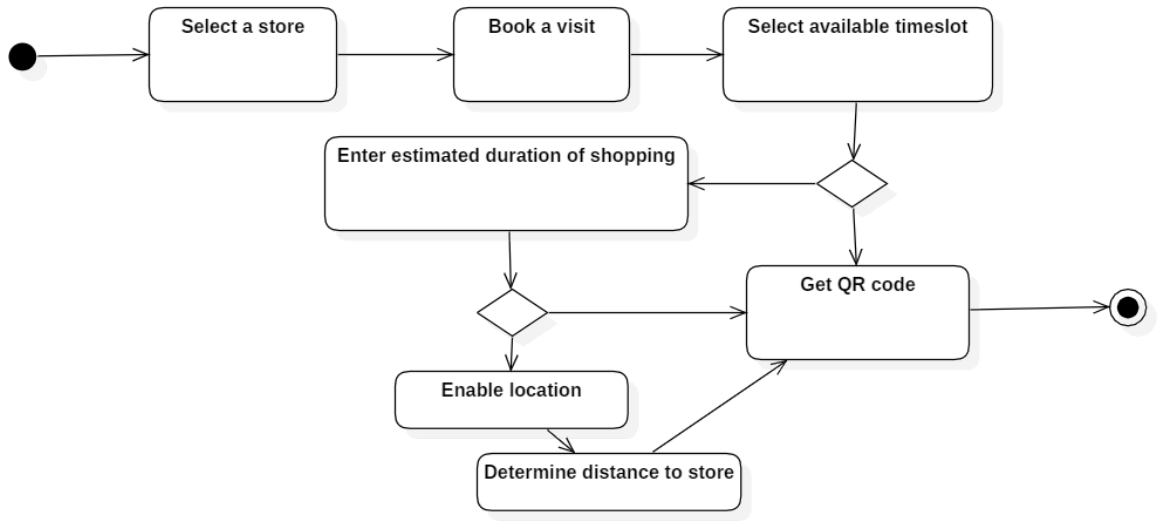

Figure 3: **State chart diagram 2 - Book a visit**

State chart 2 shows the usage of the app's second feature - "Book a visit". This can be done exclusively through the app and it's not possible to do it in person. Booking a visit is aimed for people who have less time to wait in the line and is projected to be the most effective way of stopping any additional virus spreading, as there will be less people in front of the stores at the same time. The customer must select a store and an available timeslot through the app. The customer will then be presented with a ticket that can be used only during that timeslot and in that specific store which customer has specified. Customers also have an option of entering estimated duration of shopping as well as allowing the app to access location services of the device, which would then be used to calculate the full duration of the customer's visit to the store. This would not only help the customers, but also help regulate the traffic in stores. However, these two options are optional as they slow down the reservation process and acquire confidential information from the customers, which we want to avoid doing if not needed.

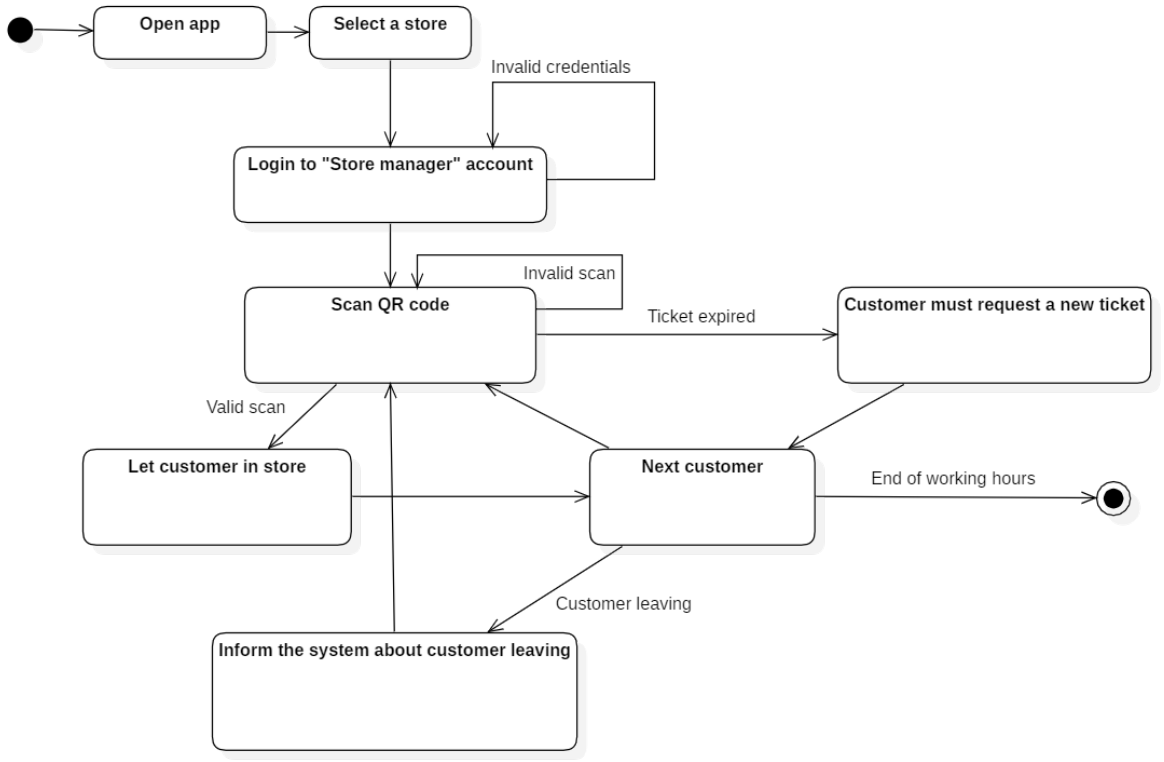

Figure 4: **State chart diagram 3 - Store manager**

State chart 3 describes the flow of the whole application usage for the store manager. Since every store manager is connected to a specific store that is in the system, it must provide the login details, which will be given during the store registration process. This diagram also shows how the store manager can control the influx of people to the store, simply by scanning their tickets' QR code when a customer enters and pressing the button when a customer exists. That way the store manager can keep the store under capacity at all times and make sure that no customer with an invalid ticket can enter the store.

2.1.2 Scenarios

Scenario 1

During the COVID-19 pandemic the government has forbidden unnecessary trips outside of the living space, therefore only allowing people to go to the stores to buy groceries. Among the measures, the government has also allowed only up to 60 people in closed spaces. This has created a problem since the only place people can go is the store and there are often a lot of people waiting outside in line and by doing so, spreading the infection. The store owner decided to use CLup system to reduce the number of people outside of the store. He installed a small screen just inside the store so that the people can see the number displayed through the glass. He also put up a small printing device for the tickets as well as a store manager at the entrance so that the influx of people can be controlled. He also gives 5% discount to everybody who uses CLup app by scanning their QR code at the cash register. This way he encourages people to use the app and not come to the store prematurely, so that the infection would not spread. This can also increase the profit a store makes since a lot of people would decide to go to the other nearby store if the line is too big. The lines have now been reduced and people can also make reservations so that they know exactly when they are going to be able to enter the store without waiting.

Scenario 2

Esselunga, one of the biggest supermarket chains in Italy, has a problem in their stores. Every working day of the week they have so called "rush hours" between 11 and 13, which is lunch time, and between 16 and 18, right after people finish their work. Because of the new restrictions, they are unable to allow everybody to go in which results in people going to the other stores and not waiting for their turn, since they do not have that much time. Every Esselunga store in Milano has therefore started using the CLup system. All Esselunga stores are put by the administrators in the system, along with their addresses and working hours. This way it will be easier for the users to find the exact Esselunga store they are looking for. Now customers can either reserve their spot in advance and plan their quick shopping accordingly, or just request a ticket when they are leaving their working space, so their waiting time is minimal. Both the stores and users have better shopping experience.

Scenario 3

A small store right next to the university and student housing area is having issues with too many students going to the store at once. Even though the store is only 50m away from the student housing, the students that are stuck in line decide to wait outside of the store and hang out with other students, therefore spreading the disease. The store owner has introduced the CLup system and completely disallowed people from entering the store without using the app. This way, students can easily make reservations and come to the store at a certain time, as well as just take a virtual ticket and come just a minute or two before it is their turn to enter. Since nearly every customer is a student and all of the students have access to the smartphones and Internet, the store owner hasn't lost any profit, and the students can now go to the store without waiting in line and risking getting the disease from one of their colleagues.

Scenario 4

Ludovico is an elderly citizen fighting diabetes. Although not as physically capable as he once was, he still goes for a long walk twice a week to and from a grocery store in the neighborhood. Such a physical activity helps him keep his blood sugar levels low, keeping a healthy body as well as the mind. Because of the new government recommendations, he would often have to wait in line in front of the grocery store. In fear of catching the virus, he slowly stopped his weekly routine, ordering his supplies online. As time passed, he started noticing that he tires more easily and that his blood sugar levels were rising. Luckily, his grandson introduced him to the CLup application which allows him to rebuild his old habit, while minimizing the health risks. As he knows exactly how much it takes him to get to the store, he uses the "book a visit" feature and indicates the time he wants to be able to enter the store.

Scenario 5

Umberto is a hard-working middle-aged man, with two children and a loving wife. Although content with his current life situation, he often finds himself balancing between order and chaos. Work from 9 to 5, late lunch, or early dinner, then watching over the children while his wife is away. As she does not drive a car, he also has to fit in grocery shopping in his schedule. While all was in order before the pandemic, the new restrictions brought long lines in front of stores and often pushed his schedule into chaos. His problem was solved by switching to a store using the CLup application. He can now "book a visit" days before his planned trip to the grocery store and be certain that he will not have to wait long lines to get in.

2.2 Product functions

Functions of the final product should provide easy and accessible use for everyone, regardless of their experience and knowledge regarding technology and smartphones.

These functions are mentioned on several places in the document, but their most thorough explanation can be found here.

2.2.1 Generating a scannable QR code

The most important part of the system. It must be able to generate a QR code quickly and responsively on the screen that replaces a real-life ticket. By generating this code, it should also "insert" the virtual ticket into the current ticket queue that is interpolated together with the real-life tickets, which are handed out at the store entrance. This code should remain visible in the application until it expires, so even if the user exits the application it does not get lost. No additional data nor any permissions are needed from the user to get the code. Generation request of the QR code is done by pressing a single button that is presented on the main screen of the application, so that even the less experienced users have no trouble generating the QR code.

2.2.2 Managing available shopping time slots

To support "book a visit" function, the system must maintain a schedule that is specific for each store. This means reserving a specific timeslot up front so that the user can just come to the store without generating a ticket in advance. However, the ticket still needs to be generated and inserted into queue in real-time. This part is to be done by the system itself and without any help of the user. The QR code must remain active until the specific time slot has passed. The system will insert the virtual ticket in the queue at the right moment either by using the average buying time or by temporarily reducing the maximum amount of people that are allowed to be at the store, so that a specific slot is reserved for the user.

2.2.3 Calculating average waiting time

The system needs to be able to calculate average waiting time if the user is to take the ticket at that specific moment. This can be done by calculating the average shopping time in a certain day or time of day and multiplying it with the number of customers that are currently waiting in the queue. While this will not provide a perfect estimation, since it is possible that every customer shops for a lot longer or a lot shorter time, the deviation should not be off by a lot. By doing this we would ensure that there are no crowds in front of the stores as the customers would be prompted to arrive only when there are a couple of minutes until their turn. This part of the system should also provide notifications for the virtual ticket holders and inform them of how many customers are currently in front of them, and therefore how many minutes until they will be able to enter the store.

2.2.4 Calculating time needed to get to the store

By providing their location, which has to be allowed by the user, the system can calculate the exact amount of time that user needs to get to the store. This is done

by using one of the Google Maps APIs which allow the distance calculation of two specific locations as well as the time needed to get from one point to another. Together with the estimated shopping time provided by the user, this would allow the user to get the exact amount of time needed to make a shopping trip. While this function has no advantages to the effectiveness of the system itself, we believe that it is an important function for the user, and it would make the usage of the application much more enjoyable experience.

2.3 User characteristics

There are four different categories of users in this application:

2.3.1 User

A person using the CLup application, not necessarily registered. The person can request to be queued in line, get an approximation of the wait time, and get updates on the estimated time left until their number is called. Furthermore, the person can request to inspect and reserve available time slots for the "book a visit" feature during which they can also indicate the estimated visit duration.

2.3.2 Store manager

An employee of the grocery store that uses the CLup application in charge of monitoring and controlling the influx of customers of the store. The person scans the QR code of incoming customers to prevent irregularities and registers their exit time (specific information about which customer exactly is exiting is not necessary as this feature is used only for calculating wait time statistics). The person uses a different UI from an ordinary user and their credentials are added to the system directly through installation.

2.3.3 System manager

An employee of CLup in charge of maintaining and updating the application. The person has administrative authority in the application setup during the system's installation.

2.3.4 Physical customer

A person that for some reason is unable to line-up through the CLup application, and therefore must take a physical ticket from the printer. The person can take the ticket with minimal physical contact and wait until their number is called by the store manager.

2.4 Assumptions, dependencies and constants

2.4.1 Assumptions

Domain assumptions

D1 The store manager's username must be unique.

D2 The registered store manager's password secure.

D3 The user's device provides accurate GPS information.

D4 The system can use data about the registered user to calculate estimated wait time.

D5 The system can correctly save data about enter and exit times of anonymous customers, to calculate estimated wait time.

D6 The system can correctly save data to and pull data from available time slot schema in the database.

D7 The user always has internet connection for the device.

2.4.2 Dependencies and constraints

Hardware limitations and dependencies

The platform the CLup application is developed for is a device with uninterrupted internet connection(2G/3G/4G/5G) and GPS signal. For the store manager's installation, a functional camera is needed.

Software limitations and dependencies

The CLup application will be developed for the Android operating system with possible versions for other operating systems, such as iOS, in the future.

User permissions and agreements

The user will have to agree to the general terms and conditions agreement as part of the installation process to use the application. Also, other permissions such as the access to the devices location for more precise wait time estimation can be allowed, if the user wants to. Naturally, no user information will be used for commercial purposes.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The CLup application interface will have to serve two types of users: customers and store managers. Opening screen of the application allows the user to pick a store or to login as a store manager. Depending on the choice, another screen is presented. For a customer, a screen with the options to retrieve a ticket, book a visit or change the store. For a store manager, a screen containing their camera view, and buttons to confirm the scanned ticket, notify the system about a customer exit and to log out of the application.

The following images show some basic concepts on how the app should look on both Android and iOS devices.

Figure 5: **Logo concept**

Figure 6: **App icon concept**

Figure 7: **Android app 1**

Figure 8: **iOS app 1**

Figure 9: **Android app 2**

Figure 10: **iOS app 2**

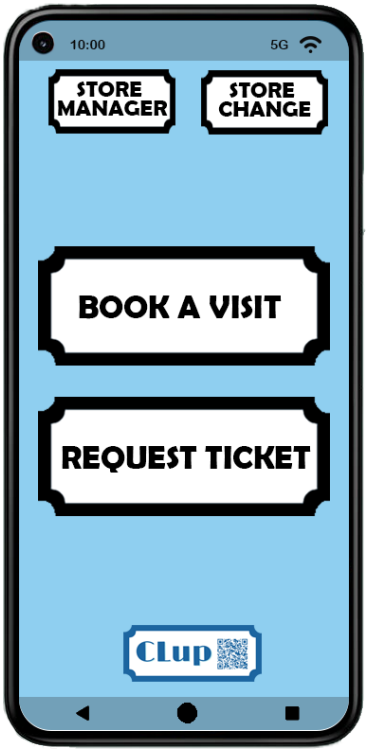

Figure 11: **Android app 3**

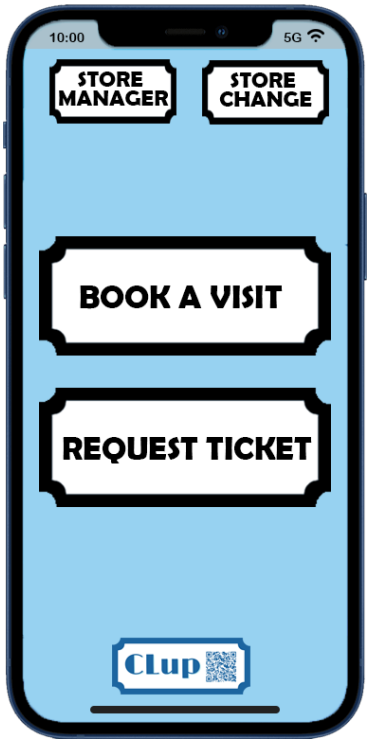

Figure 12: **iOS app 3**

Figure 13: **Android app 3**

Figure 14: **iOS app 3**

3.1.2 Hardware Interfaces

Depending on the current user the CLup application will require access to some hardware interfaces. If the current user is a customer, the application will require the device's camera and if the current user is the store manager the GPS location data will be needed. The application will require no further hardware interfaces.

3.1.3 Software Interfaces

The CLup application will not require any specific software interfaces.

3.1.4 Communication Interfaces

The most important communication will occur between the device and the database. The decision on the specific communication interface which will be used depends on the database, and is, therefore, left to the developers.

3.2 Functional Requirements

G1 Allow the user to "line up"/retrieve a number.

G1.1 Allow the user to retrieve a number through the application.

R1 The user must be able to select a specific store in which they want to do the shopping.

R2 The user must be able to request a number and a ticket.

R3 The user must be able to receive a number and a ticket.

G1.2 Allow the user to retrieve a number physically from the printer.

R4 The user must be able to physically retrieve a ticket from the printer containing a number and a QR code.

D8 The user has internet connection for the device at all times.

G2 Allow the store manager to control the entrance of the user via QR code scanning.

R5 The store manager must be able to scan a QR code.

R6 The store manager must be informed by the application if a user tries to enter the store out of order.

R7 The store manager must be informed when the capacity of the store is full.

R8 The store manager must be able to alert the system whenever a customer exits the store.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G3 Allow the user to get precise calculations of the wait time.

R9 Allow the user to receive a precise estimation of wait time when retrieving a number.

R10 The system must provide the user with an estimation of wait time based on data.

D4 The system can use data about the registered user to calculate estimated wait time.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G4 Allow the user to get updates/notifications on the estimated wait time.

R11 The system must be able to update its estimated wait time in real time.

R12 The system must be able to send an update to the user in specific intervals regarding estimated wait time until it's their turn.

D4 The system can use data about the registered user to calculate estimated wait time.

D6 The system can correctly save data about enter and exit times of anonymous customers, in order to calculate estimated wait time.

D8 The user has internet connection for the device at all times.

G5 Allow the user to "book a visit" to the store.

G5.1 Allow the user to "book a visit" to the store without indicating the expected duration of the visit.

R13 The user must be able to request to see all the available timeslots in that specific store.

R14 The system must be able to provide the user with the list of all available timeslots upon the request.

R15 The user must be able to select a specific timeslot.

R16 The user must be able to receive a confirmation of his timeslot reservation, along with a number and a ticket.

R17 Allow the user to be at most five minutes late for his reservation before cancelling his ticket.

G5.2 Allow the user to "book a visit" to the store with indicating the expected duration of the visit.

R18 The user must be able to specify expected duration of his visit to the store.

D3 The user's device provides accurate GPS information.

D7 The system can correctly save data to and pull data from available time slot schema in the database.

D8 The user has internet connection for the device at all times.

G6 Allow the store manager to login to his store manager account with credentials.

R19 The store manager must be provided with the login credentials upon request to the system administrator.

D1 The store manager's username must be unique.

D2 The store manager's password must be secure.

Requirement, [Rn]	Goals, [Gn]	Domains, [Dn]
R1	G1.1	D8
R2	G1.1	D8
R3	G1.1	D8
R4	G1.2	D8
R5	G2	D6, D8
R6	G2	D6, D8
R7	G2	D6, D8
R8	G2	D6, D8
R9	G3	D4, D6, D8
R10	G3	D4, D6, D8
R11	G4	D4, D6, D8
R12	G4	D4, D6, D8
R13	G5.1	D3, D7, D8
R14	G5.1	D3, D7, D8
R15	G5.1	D3, D7, D8
R16	G5.1	D3, D7, D8
R17	G5.1	D3, D7, D8
R18	G5.2	D3, D7, D8
R19	G6	D1, D2

Table 1: **Mapping table**

3.2.1 Use cases

Name	**Book a visit**
Actors	**User**
Goals	**G5**
Entry condition	User has the application installed Desired store is in the CLup database
Event flow	User opens the CLup application User selects a store User selects "Book a visit" System sends user a list of available timeslots User selects a specific timeslot User inputs the estimated shopping time (optional) User enables location services (optional) If user has enabled location services System calculates distance to the store User receives estimated traveling time to the store User receives a number and a QR code
Exit condition	Ticket request is successful
Exception	There are no available timeslots, user can search for a another store
Special requirements	/
UMLs	**Use case**, **Sequence**, **State chart**

Table 2: **Use case - Book a visit**

Name	**Retrieve a number (in person)**
Actors	**User, Store manager**
Goals	**G1**
Entry condition	User present at the store Desired store is in the CLup database
Event flow	User goes to the ticket machine and takes a ticket User waits for his number Store manager scans user's QR code User enters the store
Exit condition	User exits the store
Exception	The ticket printer is not working, user waits for it to be fixed The QR code cannnot be scanned due to damaged ticket, user gets another ticket
Special requirements	A new ticket gets printed in less than two seconds after the old ticket got taken
UMLs	**State chart**

Table 3: **Use case - Retrieve a number (in person)**

Name	**Retrieve a number (through the application)**
Actors	**User, store manager**
Goals	**G1**
Entry condition	User has the application installed Desired store is in the CLup database
Event flow	User opens the application User selects a store User selects "Get ticket" User receives a number, a QR code, and an estimated wait time User waits for his number User gets to the store Store manager scans user's QR code User enters the store
Exit condition	User exits the store
Exception	The QR code cannnot be scanned due to broken screen, user gets physical ticket ticket
Special requirements	The ticket number has to be inserted in the queue immediately after its generation
UMLs	**Sequence**, **Use case**

Table 4: **Use case - Retrieve a number (through the application)**

Name	**Control the number of customers entering the store**
Actors	**Store manager, user**
Goals	**G2**
Entry condition	Store manager has downloaded the application Store manager is logged in to his CLup account Desired store is in the CLup database
Event flow	Scan the user's QR code If the QR code is valid Let the user inside of the store If the QR code is not valid Send the user back to the virtual line
Exit condition	End of working hours
Exception	The QR code cannnot be scanned due to damaged ticket, user gets another ticket
Special requirements	/
UMLs	**State chart**

Table 5: **Use case - Control the number of customers entering the store**

Name	**Control the number of customers exiting the store**
Actors	**Store manager, user**
Goals	**G2**
Entry condition	Store manager has downloaded the application Store manager is logged in to his CLup account Desired store is in the CLup database
Event flow	User exits the store Store manager alerts the system that an user has left the store
Exit condition	End of working hours
Exception	/
Special requirements	/
UMLs	**State chart**

Table 6: **Use case - Control the number of customers exiting the store**

Name	**Store manager login**
Actors	**Store manger**
Goals	**G6, G2**
Entry condition	Store manager has downloaded the application
Event flow	Open the CLup application Login to the store manager account
Exit condition	Store manager has successfully logged in to his account
Exception	Store manager inserts invalid credentials, store manager checks his credentials and tries again
Special requirements	/
UMLs	**Sequence**

Table 7: **Use case - Store manager login**

Name	**Calculate estimated wait time**
Actors	**User**
Goals	**G3, G4, G5**
Entry condition	User requested a ticket or booked a visit
Event flow	System calculates estimated wait time User receives estimated wait time
Exit condition	Estimated wait time displayed on user's device
Exception	/
Special requirements	Estimated wait time has to be calculated in less than five sconds
UMLs	/

Table 8: **Use case - Calculate estimated wait time**

Name	**Calculate distance to the store**
Actors	**User**
Goals	**G5**
Entry condition	User has enabled location services User booked a visit
Event flow	System calculates distance to the store User receives estimated traveling time to the store
Exit condition	Estimated travel time displayed on user's device
Exception	/
Special requirements	Estimated travel time has to be calculated in less than five seconds
UMLs	/

Table 9: **Use case - Calculate distance to the store**

Name	**Store registration**
Actors	**System manager, store manager**
Goals	**G6**
Entry condition	/
Event flow	Store manager requests putting the store in the system's database System manager reqests additional data about the store Store manager sends the data to the system manager System manager inserts the store into the system's database System manager sends the confirmation and the login credentials to the store manager
Exit condition	Store manager has received his account credentials
Exception	Store manager provides invalid information about the store, store manager corrects invalid data Store manager provides information about a store inside the database, system rejects the input and notifies the store manager
Special requirements	/
UMLs	**Sequence**

Table 10: **Use case - Store registration**

3.2.2 Use case diagrams

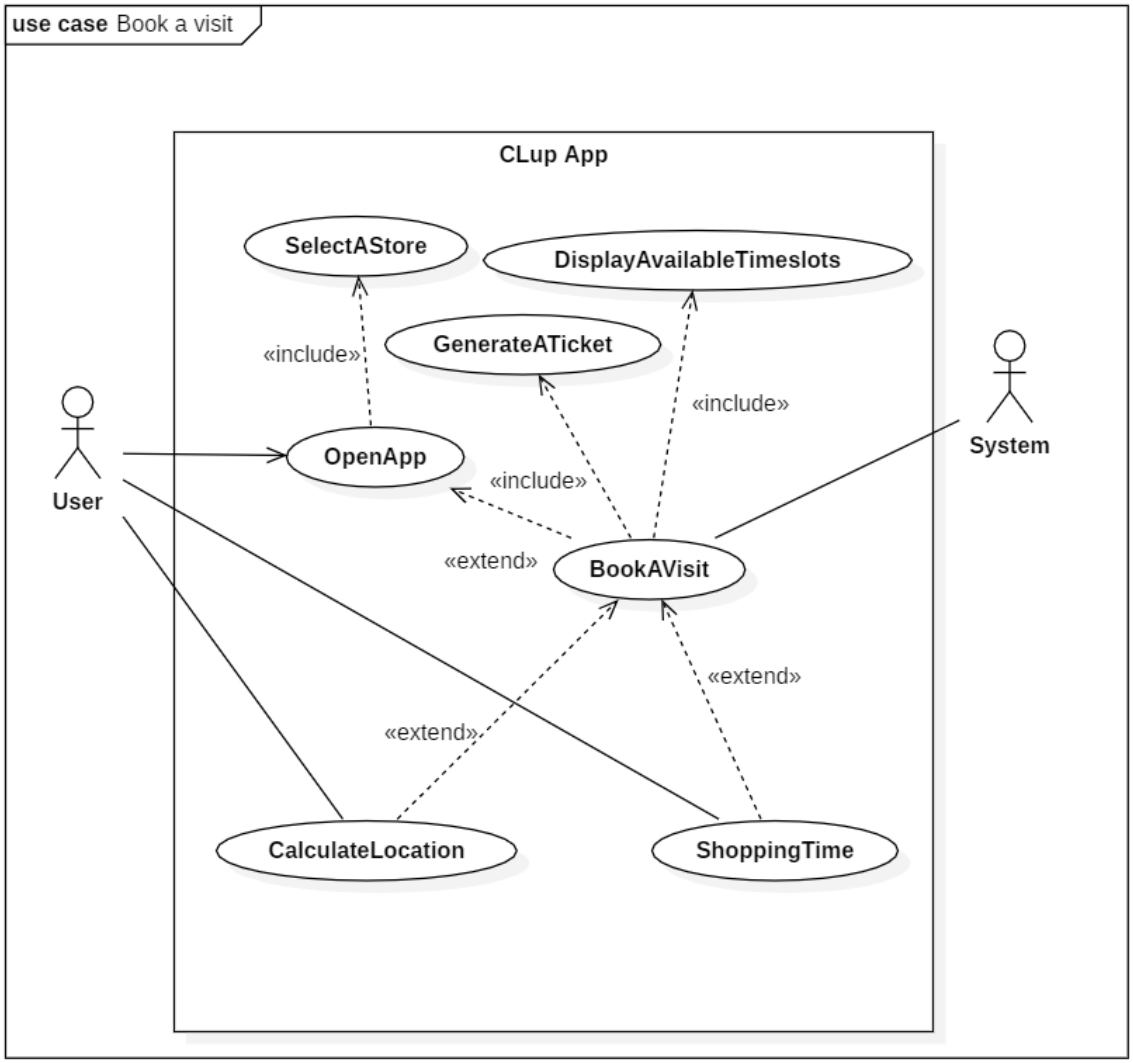

Figure 15: **Use case diagram 1 - Book a visit**

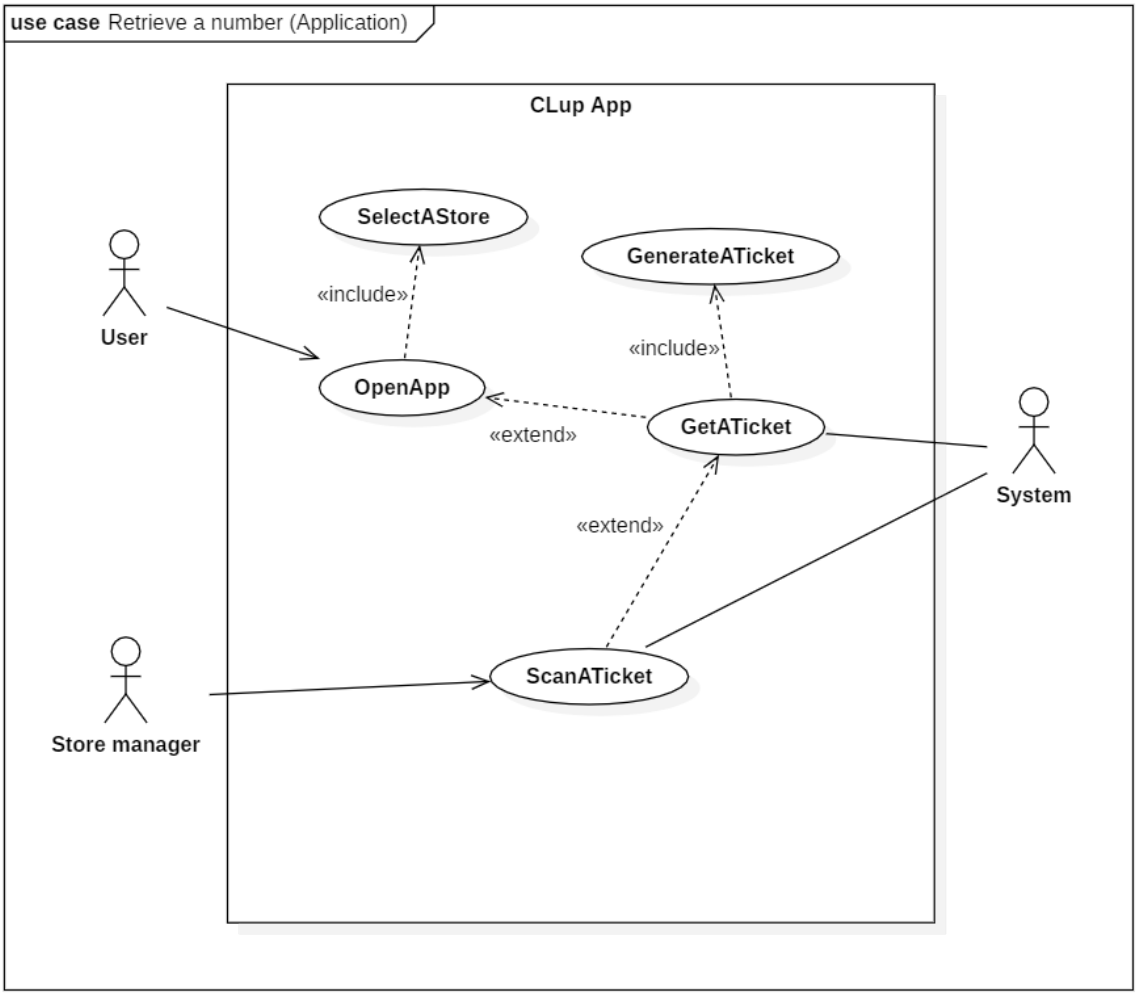

Figure 16: **Use case diagram 2 - Retrieve a number**

3.2.3 Sequence diagrams

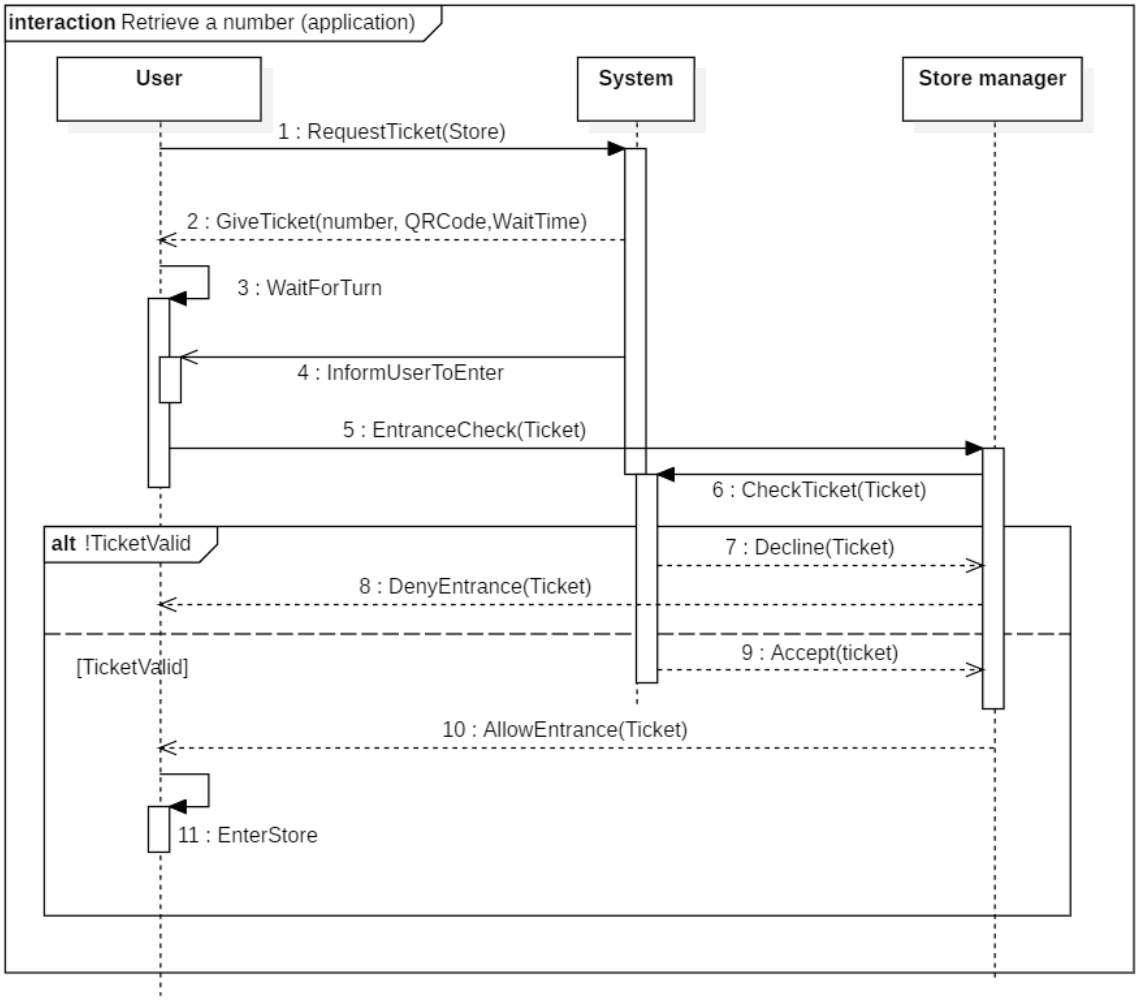

Figure 17: **Sequence diagram 1 - Retrieve a number**

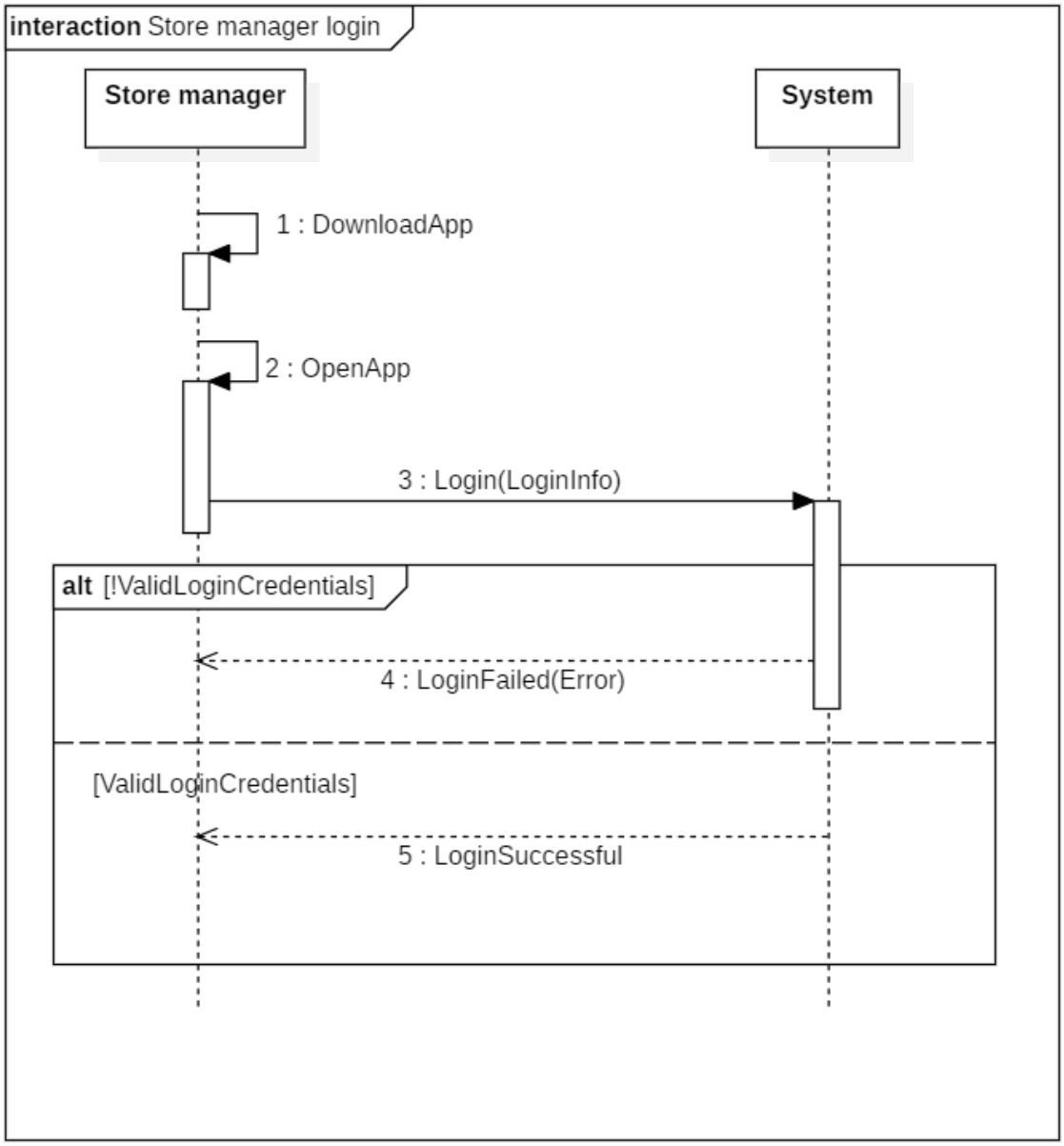

Figure 18: **Sequence diagram 2 - Store manager login**

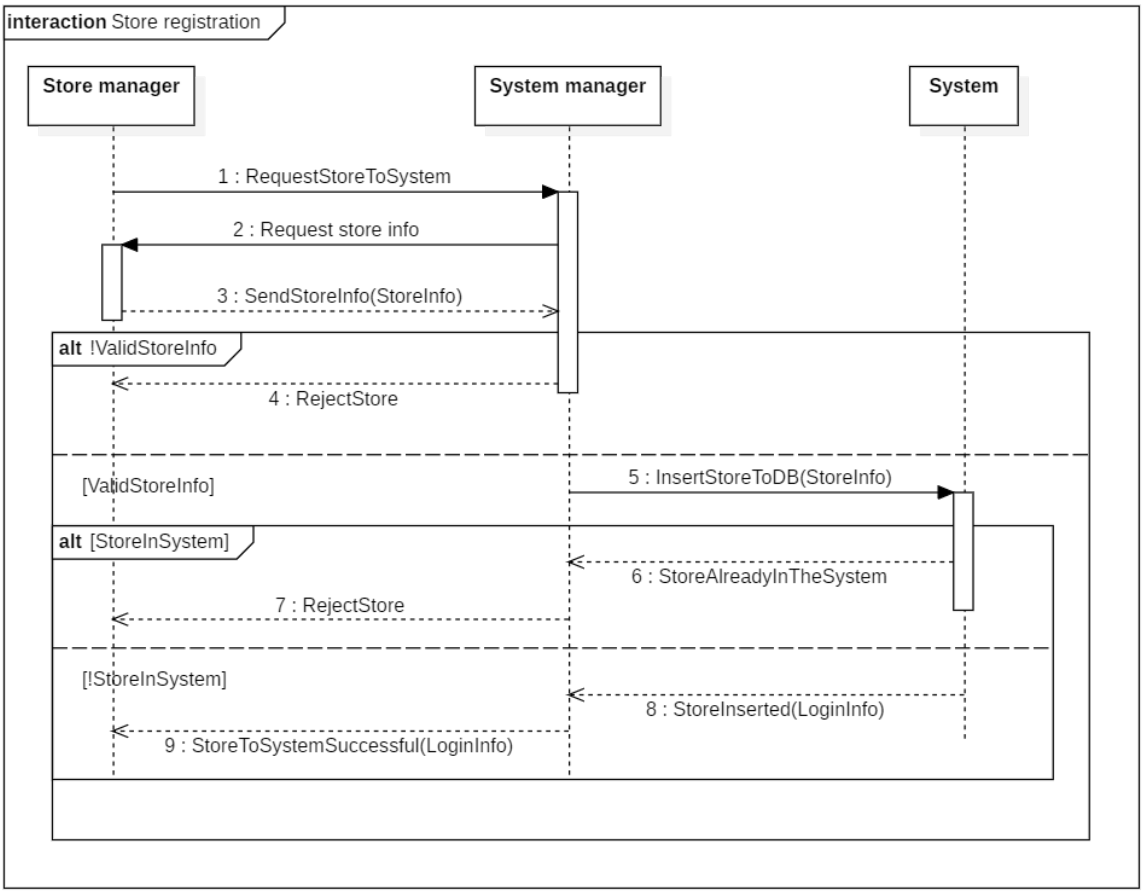

Figure 19: **Sequence diagram 3 - Store registration**

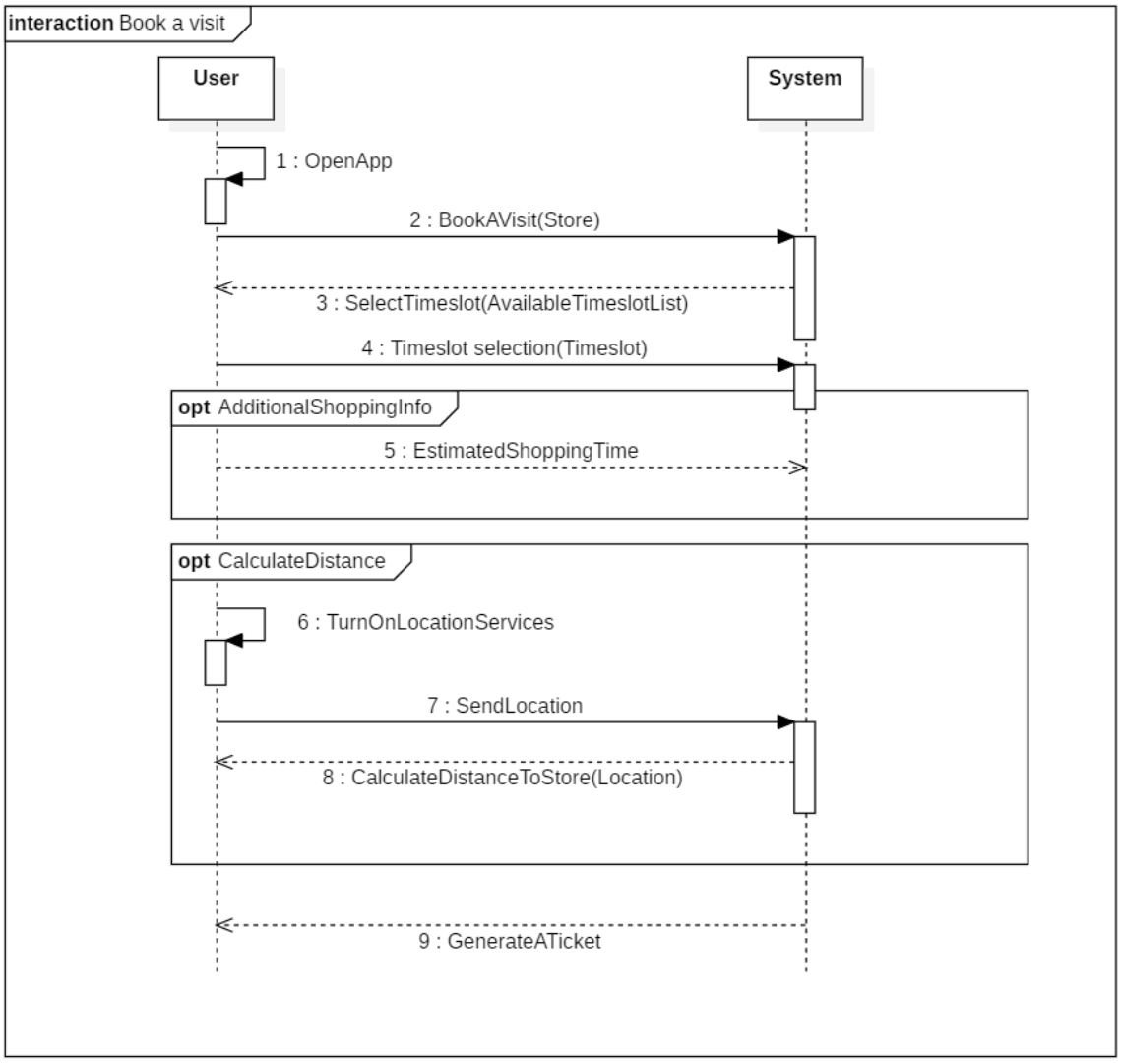

Figure 20: **Sequence diagram 4 - Book a visit**

3.3 Performance requirements

The system is relatively simple and doesn't work with a lot of data. What the system should do properly is control the queue for every specific store. It should also sync all of the tickets that are printed out in real life, and ones that are "printed" virtually. The same queue should feature both types of tickets and work adequately. That means that when a ticket is printed, it should be inserted into queue within ten seconds, to prevent errors in the queue. It is not of a major importance if a couple of people switch places due to slow reaction time from the system, but it should be avoided if possible.

When user want to print a ticket virtually, the system has to provide it with an average waiting time within a couple of seconds. This is a piece of data that is going to be stored in the database of every store, and it should be updated at least every 30 minutes based on the current traffic and other available data, like day of the week and time of day.

When calculating the distance from the user's current location to the desired store, the information should be available to the user in ten seconds at most. If the system is for any reason unable to calculate the distance it should display a message that it it unable to do it and continue to the next step.

3.4 Design constraints

3.4.1 Standards compliance

The entire application must be the subject to the General Data Protection Regulation (GDPR), a regulation in the EU law to protect the personal data of all individuals within the European Union (EU) and the European Economic Area (EEA).

3.4.2 Hardware limitations

The application is only planned to be made for Android OS and iOS. Any device that doesn't use Android or iOS, but uses Windows Mobile OS is excluded from the use. Any non-smartphone mobile device will not support the application.

The application will only work on Android devices that support Android 5.0 (Lollipop) or above due to security reasons.

The application will only work on iOS devices that support iOS 7.0 or above due to security reasons.

Device in use must have access to the internet connection during the ticket request. After it acquires the ticket, the internet connection is no longer required.

3.4.3 Privacy limitations

The application should not collect nor ask any personal information from the user. It should only request for access to publish push notifications, as well as location, but only if the user decides to use "Book a visit feature". No other data is to be collected at all.

3.5 Software system attributes

3.5.1 Reliability and availability

The system must be available at all times during the day. The only times that the system is allowed to be down, due to maintenance or any other reason, is during the night hours, while the stores are not working. In that time the use of application is nearly non-existent as only the "Book a visit" function is working, and the number of users using the function should be close to zero.

In case the system has stopped working during the day, a notification has to be sent to every user that uses the application, and the queueing in stores should be taken over by the store manager until the system is back up.

3.5.2 Security

Since there is no users' personal data that is stored within the system, the security by itself is at the very high level to begin with. Logins from the store managers should be encrypted and protected. Transfer of this data to the store managers should be done safely, either by using two-way encryption or in-person.

3.5.3 Maintainability

The system can be written in any language that will guarantee high level of maintainability and security at all times. It should also allow users to use the older version of the app, which means that every change made to the app must be made with legacy in mind. The code must have a clean and detailed documentation so that people who work on it have no problems upgrading the app and pushing new updates if necessary.

3.5.4 Portability

As already mentioned in the section Hardware limitations, the application should only work on Android and iOS mobile devices, with specific OS version limitations.

Application should not natively be supported with desktop devices since users must provide their QR code for scanning when entering the store, which would not be possible on a desktop computer, and not very usable on a laptop.

4 Formal Analysis Using Alloy

```
open util/integer

abstract sig TixState{}

one sig WAITING extends TixState{}
one sig INSTORE extends TixState{}
one sig EXPIRED extends TixState{}

abstract sig Location{}
sig GPSCoordinates extends Location {
        store: some Store

        // GPSCoordinates have been simplified for
          ↪ clearer view

        //latitude: one Int,
        //longitude: one Int
        }
        /*{
                latitude < 90 and latitude > -90
                longitude < 180 and longitude >
                  ↪ -180
        }*/

// Simplified version of address only includes
  ↪ GPSCoordinates
/*
sig Address extends Location {
        country: one Country,
        city: one City,
        street: one Street,
        number: one Int
        }

*/
```



```
  // Working time has been simplified for clearer
    ↪ view
sig WorkingTime{
        store: some Store
        }


        // One buyer has one ticket and optional
          ↪ GPS coordinates
sig Buyer{
        ticket: one Ticket,
        locGPS: lone GPSCoordinates,
}

sig QRCode{}

sig Store{
        allTickets: set Ticket,
        maxCustomers: one Int,
        storeId: one Int,
        timeOfWork: one WorkingTime,
        locGPS: one GPSCoordinates,
}{
        maxCustomers > 0 and storeId > 0
}


        // A tickets has a number, a parent store,
          ↪ a ticket state, and a QR code
        // Ticket state can be WAITING if the
          ↪ ticket is still in queue,
        // INSTORE if the buyer entered the store,
          ↪ and EXPIRED, if the
        // ticket has expired
sig Ticket{
        tixNum: one Int,
        parentStore: one Store,
        tixState: one TixState,
```



```
        qr: one QRCode
}{
        tixNum > 0 and tixNum ≤ parentStore.
          ↪ maxCustomers
}

    // There must be less tickets than the number
      ↪ of maximum customers
fact {
        (all s:Store | #s.allTickets < s.
          ↪ maxCustomers)
}

    // No two tickets with the same number can
      ↪ exist in the same store
fact {
        all disj t1,t2:Ticket | (t1.tixNum ≠ t2.
          ↪ tixNum and t1.parentStore = t2.
          ↪ parentStore )
}

        // Every ticket must have only one owner (
          ↪ buyer)
fact{
        all disj b1,b2:Buyer | (b1.ticket ≠ b2.
          ↪ ticket)
}

fact {
        all disj s1,s2:Store | s1.storeId ≠ s2.
          ↪ storeId
}

        // Every ticket has to belong to only one
          ↪ store
fact {
        (all s:Store | all t:s.allTickets | t.
          ↪ parentStore.storeId = s.storeId)
```



```
}

fact {
        (all s:Store | all w:WorkingTime | #s ≥ #w)
}


pred show{
        (some s:Store | #s.allTickets > 1 )
}

run show
```


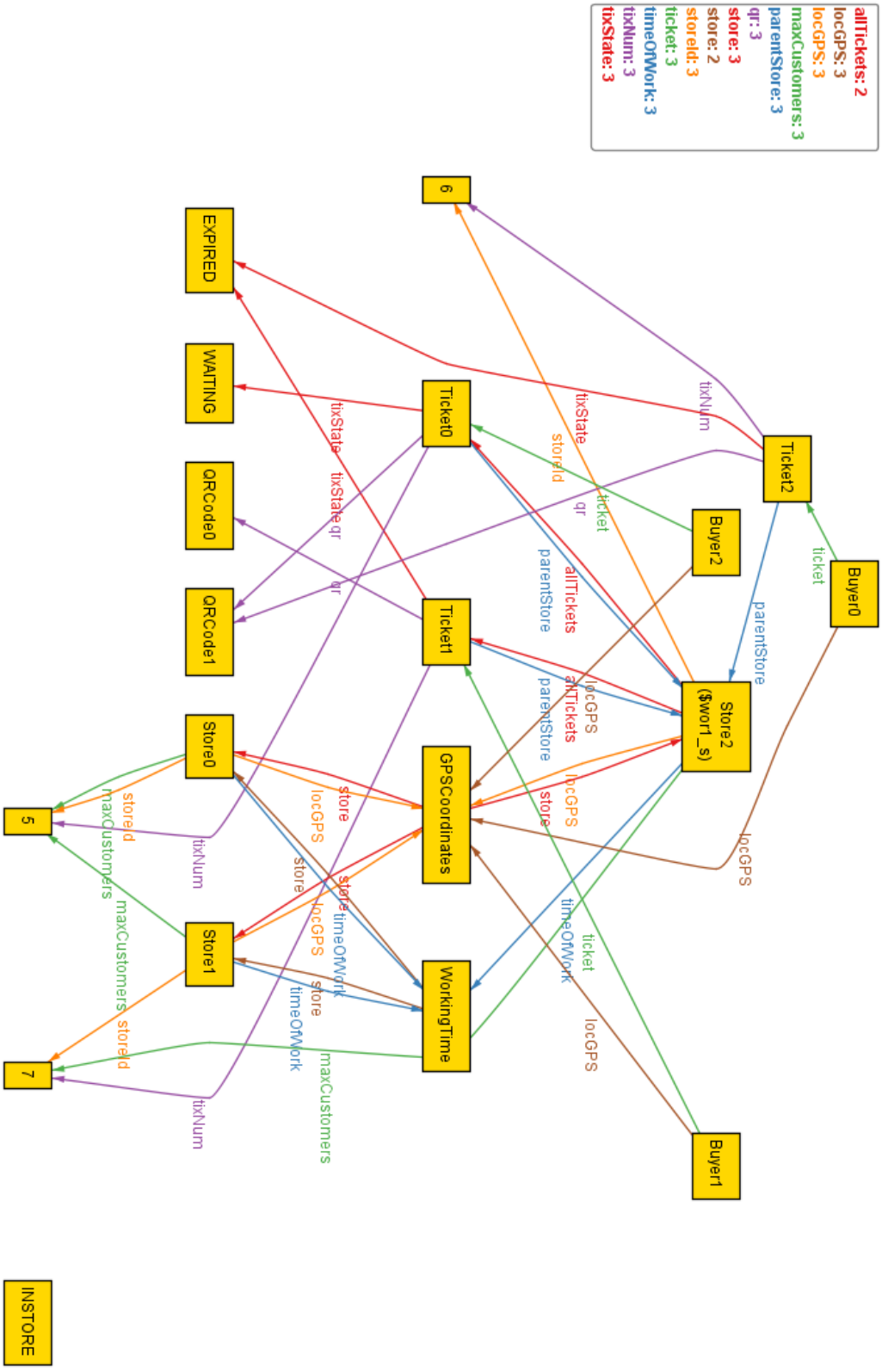

Figure 21: **Alloy graph 1**

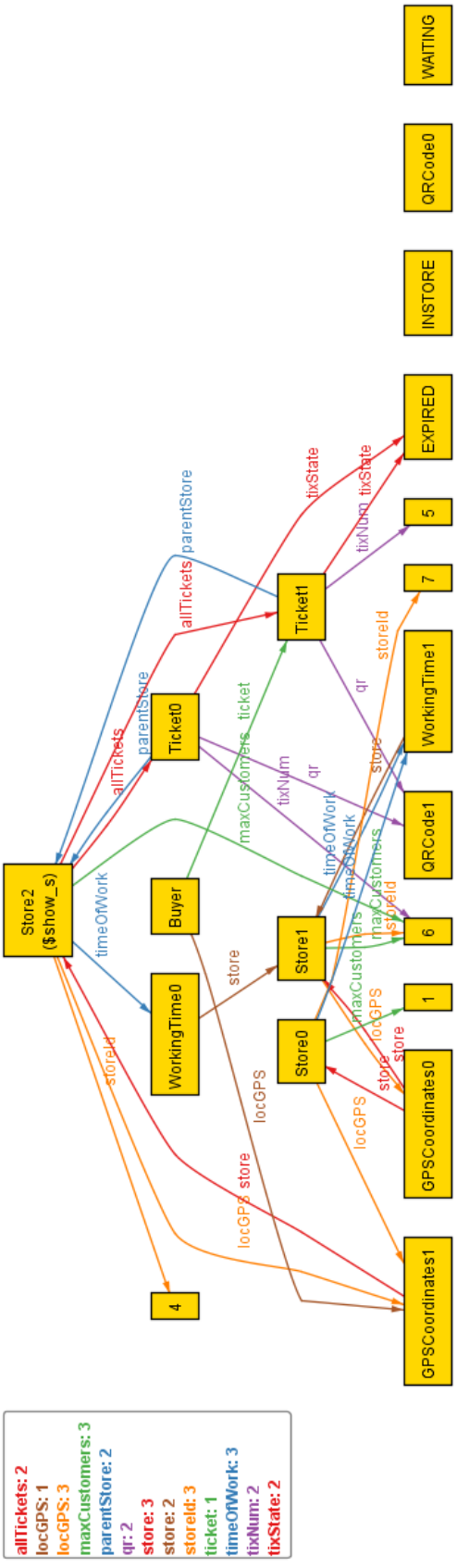

Figure 22: **Alloy graph 2**

5 Effort Spent

Task Description	Time spent[hours]
Introduction:	1
Overall description: Product perspective and functions, user characteristics, and constraints	5
Overall description: Goals, Assumptions, and Dependencies	2
Specific requirements: External interface Requirements	2
Specific requirements: Use Case diagrams	5
Formal analysis using Alloy:	6
LaTex document composition:	12
Total:	33

Table 11: **Effort spent - Robert Medvedec**

Task Description	Time spent[hours]
Introduction:	
Overall description: Product perspective and functions, user characteristics, and constraints	
Overall description: Goals, Assumptions, and Dependencies	
Specific requirements: External interface Requirements	
Specific requirements: Use Case diagrams	
Formal analysis using Alloy:	
LaTex document composition:	
Total:	

Table 12: **Effort spent - Toma Sikora**

