



Requirement Analysis and Specification Document

Authors: Sergey Nabatov, Nikita Rozov, Olexandr Shchukhlyi
Date: 29 November 2020
Professor: Matteo Rossi

Deliverable: RASD

Title: Requirement Analysis and Verification Document

Authors: Sergey Nabatov, Nikita Rozov, Olexandr Shchukhlyi

Version: 1.1

Date: 29-November-2020

Download page: <https://github.com/serynabatov/NabatovRozovShchukhlyi>

Copyright: Copyrights 2017, S. Nabatov, N. Rozov, O. Shchukhlyi - All rights reserved

Contents

1	Introduction4
1.1	Purpose4
1.1.1	General Purpose4
1.1.2	Goals4
1.2	Scope4
1.2.1	World Phenomena:5
1.2.2	Shared phenomena:5
1.3	Definitions, Acronyms, Abbreviations.....	.6
1.3.1	Definitions.....	.6
1.3.2	Acronyms6
1.3.3	Abbreviations6
1.4	Revision History.....	.6
1.5	Reference Documents7
1.6	Document Structure.....	.7
2	Overall Description8
2.1	Product perspective8
2.2	Product functions.....	.11
2.2.1	Booking management.....	.11
2.2.2	Location management12
2.2.3	Arrival/departure management.....	.12
2.2.4	Others12
2.3	User characteristics12
2.4	Assumptions, dependencies and constraints.....	.12
3	Specific requirements.....	.14
3.1	External interface requirements.....	.14
3.1.1	User interfaces.....	.14
3.1.2	Hardware user interfaces17
3.1.3	Software interfaces17
3.2	Functional requirements.....	.18
3.2.1	Common User18
3.2.2	Prioritized User25
3.2.3	Privilege User.....	.27
3.2.4	Requirements.....	.32
4	Formal Analysis Using Alloy.....	.32
4.1	Alloy Code35
4.2	World generated0
4.3	Alloy results0
5	Effor spent.....	.1

5.1	Nikita Rozov	1
5.2	Oleksandr Shchukhlyi	1
5.3	Sergey Nabatov	1
6	References	Ошибка! Закладка не определена.

List of Figures

- Figure 1 – Diagram of world phenomena & shared phenomena
- Figure 2 – Class diagram
- Figure 3 - State diagram: creation of a connection session via SMS
- Figure 4 – State diagram: authorization of a Prioritized user
- Figure 5 – State diagram: authorization process by store employees
- Figure 6 – CLUp: customer verification
- Figure 7 – CLUp: store choosing
- Figure 8 – CLUp: date choosing
- Figure 9 – CLUp: time choosing
- Figure 10 – CLUp: department choosing
- Figure 11 – CLUp: QR code generation
- Figure 12 – CLUp: built route
- Figure 13 – Use case diagram: Common User
- Figure 14 – Sequence Diagram: store choosing
- Figure 15 – Sequence Diagram: booking visit
- Figure 16 – Use case diagram: Prioritized User
- Figure 17 - Sequence Diagram: Book A Visit via the machine
- Figure 18 – Use case diagram: Privilege User
- Figure 19 – Sequence Diagram: Register the store in the system
- Figure 20 – Sequence Diagram: Analyzing
- Figure 21 – Alloy: World 1 generated
- Figure 22 – Alloy: World 2 generated
- Figure 23 – Alloy: World 3 generated
- Figure 24 – Alloy – Executing result

List of Tables

- 1.3.1 Definitions
- 1.3.2 Acronyms
- 1.3.3 Abbreviations
- 1.4 Revision History
- 5.1 Nikita Rozov
- 5.2 Oleksandr Shchukhlyi
- 5.3 Sergey Nabatov

1 Introduction

1.1 Purpose

1.1.1 General Purpose

The current world was faced with a pandemic of COVID-19. That virus is extremely infectious and is transmitted by airdrome droplets. So, the government and people should organize some restrictions to prevent the mass spreading of coronavirus.

One of the most visited places in the real-world is shops and stores. They cannot be closed because people have to purchase essential goods, such as food. So, there should be a way to provide a low chance of transmitting the virus to new people. As the World Health Organization established the maximum range of the effective virus spreading (1.5 meters) it could be used to estimate the optimal flow in the store concerning the security. However, it makes another trouble: people who are waiting outside a store making a crowd enhancing the chance of obtaining and spreading the virus.

1.1.2 Goals

G1: 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;

G2: To give people opportunity to stay in queue remotely to avoid crowds in stores and in real queues.

G3: Stores should have the possibility to hand out “tickets” on the spot;

G4: The system should allow customers to book a visit to the supermarket to optimize quantity of people in it by time of their visit, and category of products which they want to purchase;

Additional goals for the group of 3 members:

G5: The system (application and “tickets” on the spot) should include alternative slots (for another day), suggest to the customer the location of the nearest “safe” store based on his location;

G6: The system should allow the third party to get the statistical information to perform better management of the store.

1.2 Scope

The service “Clup” (stands for Customers line up) gives the possibility to plan your visit to a shop with coordination with the management of a store (call them the third side). The application creates a virtual queue instead of a real one, helping to avoid crowds. Also, the system enables the personnel of the management to monitor a situation in a store. For including a store into the application, the manager should accommodate the service with information about areas of a store and departments (to calculate the maximum people flow).

Clup allows customer three options: to simply stand in the queue (the shortest waiting time), to book your visit for the time/day chosen by you, and to accept a token from a machine right outside a store (of course it is not recommended, but in case of not knowing about the application it is the only way). After picking any option users could enter a supposed time of a visit and choose the goods categories, to balance human flow between departments to get the best efficiency.

The location of a customer is getting by GPS, the location of the store is provided by the manager of the shop.

If the user is choosing the first option, the service shows him/her an estimated time of waiting and an estimated time of walking/driving to a store. After coming to a shop customer scans the QR code and if the time for a visit is not expired (came not more than some minutes late) user could go in and start to buy what he/she needs. The application also has an additional feature which will give the possibility to detect customers who not wearing masks it happens when the server gets the data from entering the store. If the user does not have a mask, he will be banned for some time. After exiting from the store user is popping out from the virtual queue.

The second option adding the notification feature to the first one. So, a customer gets the message when the planned visit time is coming. The rest functionality is inherited from the first option.

If a person is using a mobile device to stand in queue/book a visit he can see a load of stores to choose the one with the minimum waiting time/people flow. Also, it is possible to cancel a booking if a user changed his/her mind.

The third option presumes that a customer does not know about mandatory using the application to get in a store. So, when a person comes, he got a ticket with the time of a visit. This request is adding to the queue such as it will be a request from the first option. To visit/leave the store a customer has to scan his ticket before arrival/departure.

1.2.1 World Phenomena:

- Stores are located at different places;
- Departments are located differently at each store;
- Stores and departments come in different sizes;
- Users can visit different stores and various departments in every store;
- Users purchase various products;
- Users choose basket based on the number of products they want to purchase;
- Users go from home to the store.

1.2.2 Shared phenomena:

- Booking;
- User arrival time;
- User departure time;
- The wearing mask fact;
- Store and department area;
- Customer queue

A diagram containing world phenomena and shared phenomena is shown in Figure 1:

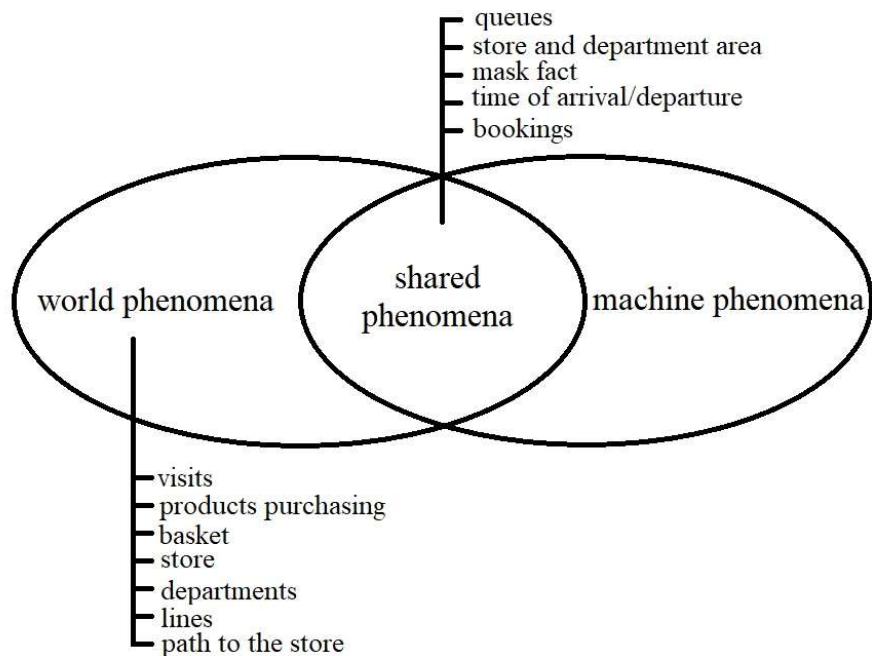


Figure 1 – Diagram of world phenomena & shared phenomena

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Third Party	Stands for “store owners” a.k.a. administration and for “Yandex Maps”
CLup	“Customer Line-Up”
Booking	The entity which includes the date and time of the visit and estimated time calculated by the software
Common User	The customer who booked a visit via a smartphone
Prioritized User	The customer who booked a visit via a machine at the store
Privilege User	Staff personnel

1.3.2 Acronyms

UML	Unified Modeling Language
API	Application Programming Interface
SMS	Short Message Service
NMS	Network Management System
STB	Software To Be
GPS	Global Positioning System
QR code	Quick Response code
UI	User Interface
PC	Personal Computer

1.3.3 Abbreviations

Gn	nth Goal
Dn	nth domain assumption
Rn	nth requirement

1.4 Revision History

Date	Modifications
29/11/2020	First Version
19/12/2020	Second Version: <ul style="list-style-type: none">○ Update UML○ Update signatures and constraints in Alloy code

1.5 Reference Documents

- Specification Document: “A.Y. 2020-2021 Software Engineering 2”
- Alloy doc: <http://alloy.lcs.mit.edu/alloy/documentation/quickguide/seq.html>
- ISO/IEC/IEEE 29148-2011 – Systems and software engineering – Life cycle processes – Requirements engineering
 - Yandex.maps API - <https://yandex.ru/dev/maps/>
 - Project’s diagrams were made with - <https://app.diagrams.net/>

1.6 Document Structure

Chapter 1 is an introduction that contains an informal description of the purpose of both the system and the service being developed, as well as more specific goals. For a better understanding of the document, this chapter points out the specifications and the root logic of the world and shared phenomena of the software.

Chapter 2 provides the overall description in other words a more detailed design of the project, starting with its model using class diagrams, continuing with the behavior through state diagrams. The product function will help to explain the main functionality of the application. In addition, assumptions, dependencies, and constraints set the initial conditions for building a system.

Chapter 3 consists of interface requirements, such as user interface - presentation of the design of the mobile application, hardware user interfaces - a necessary minimum of physical things for the system to function, software interfaces - the required set of pre-installed programs, utilities, etc. This section is key, as it displays both requirements of functional and non-functional, a step-by-step description of processes, their stages, exceptions, and test scenarios with corresponding sequence diagrams.

Chapter 4 includes an explanation of the main entities of the system and consideration of their critical areas using the alloy annotation language. This tool can check the metamodels and the correctness of assertions, which is demonstrated in the World generated and Alloy results sections.

Chapter 5 contains tables of human resources of each of the members involved in the creation of the document.

2 Overall Description

2.1 Product perspective

As a result, an entirely new mobile app will be developed that uses Yandex maps as additional service. Tracking the user's location will be performed with the help using the built-in sensors of the mobile device. Yandex Map API is responsible for displaying available stores and building the path to them.

The detailed architecture of the program and its interaction with third-party API will be described below.

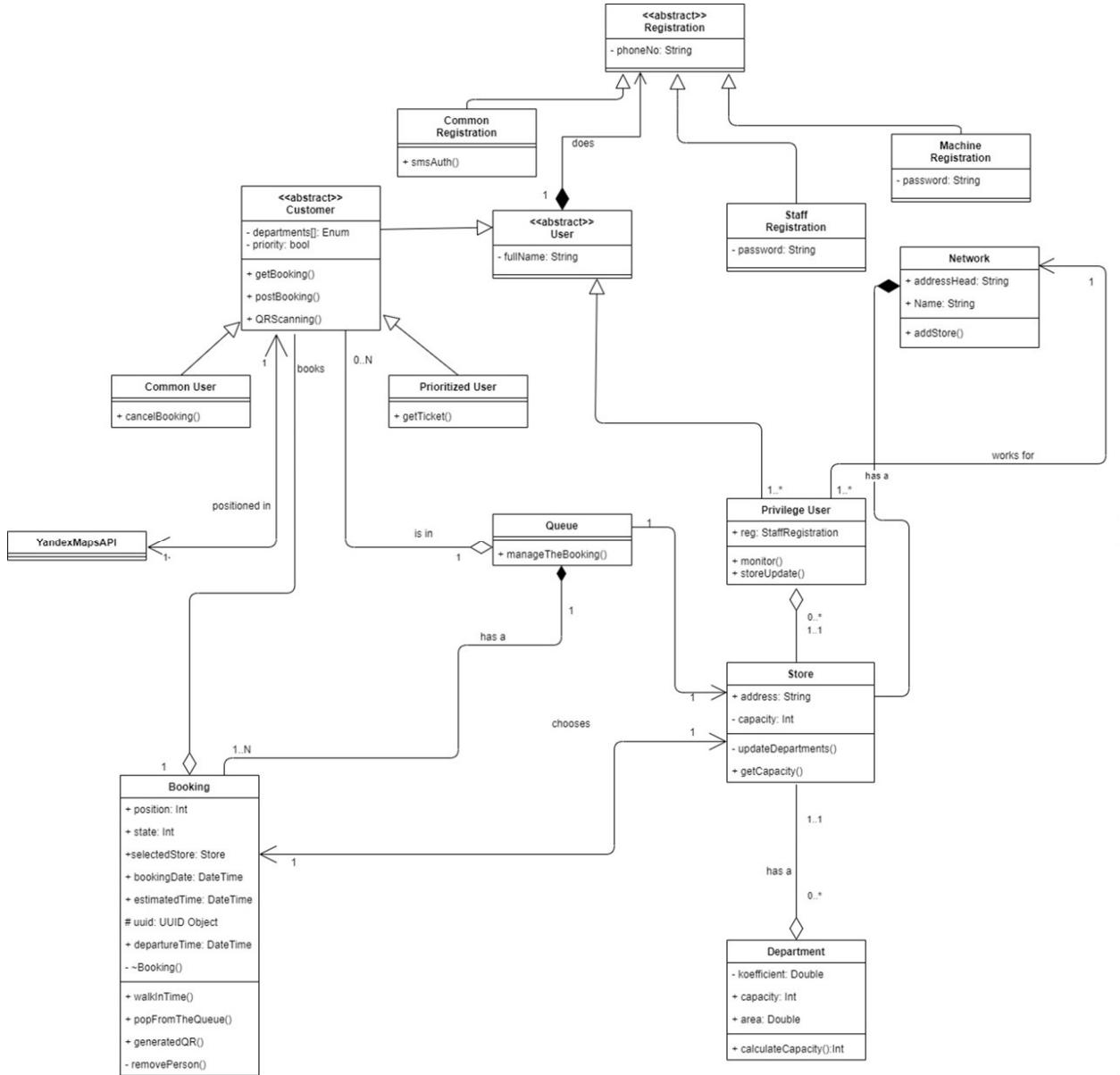


Figure 2 – Class diagram

As can be seen in the screenshot, the entry point of the program can be considered as the *Registration*, which is divided into two branches: the end-user registration and staff registration. Class *Customer*, after inheriting the qualities of the *User* Class, also can be divided into two parts: the one group of people who will use the application - *CommonUser*, with the consequent possibility of canceling reservations online, and the second group *PrioritizedUser*, which will receive its ticket to the queue upon arrival at the store. The *Customer* class, which could be from 0 to N *Queue* class in general, also uses the *GeoController*

class, which provides functionality for tracking the user position and building a route to the selected store. The *YandexAPI* library is responsible for the rest of the map functionality. The *Customer* class contains a key *getBooking* method that implements the logic of introducing a new user to the queue, storing more detailed information, such as the selected store and the capacity of its departments, the user's state (going to the store, making purchases, leaving the store), and generating a QR code that is necessary needed for going to the store. User's position data is transmitted via a mobile device; therefore, it is important to grant the appropriate rights to use the app. The *Queue* class stores an array of all bookings for the store. The purpose of the *Store* and *Department* classes is to prevent congestion and create a more efficient distribution of incoming people, to do this, we need to have additional information about the store itself that the owner provides. Let us say the user selects priority departments, and we know their capacity, which will help us calculate the approximate number of customers for each department. The *PrivilegeUser* class is the store staff that can access the app and its advanced settings. The *Network* class groups stores of a single registered trademark. It is worth noting that the above functionality is performed on the server-side.

Using a state diagram, it is possible to describe system behavior more accurately. For example, *Figure 3* shows the creation of a connection session to authenticate users via SMS.

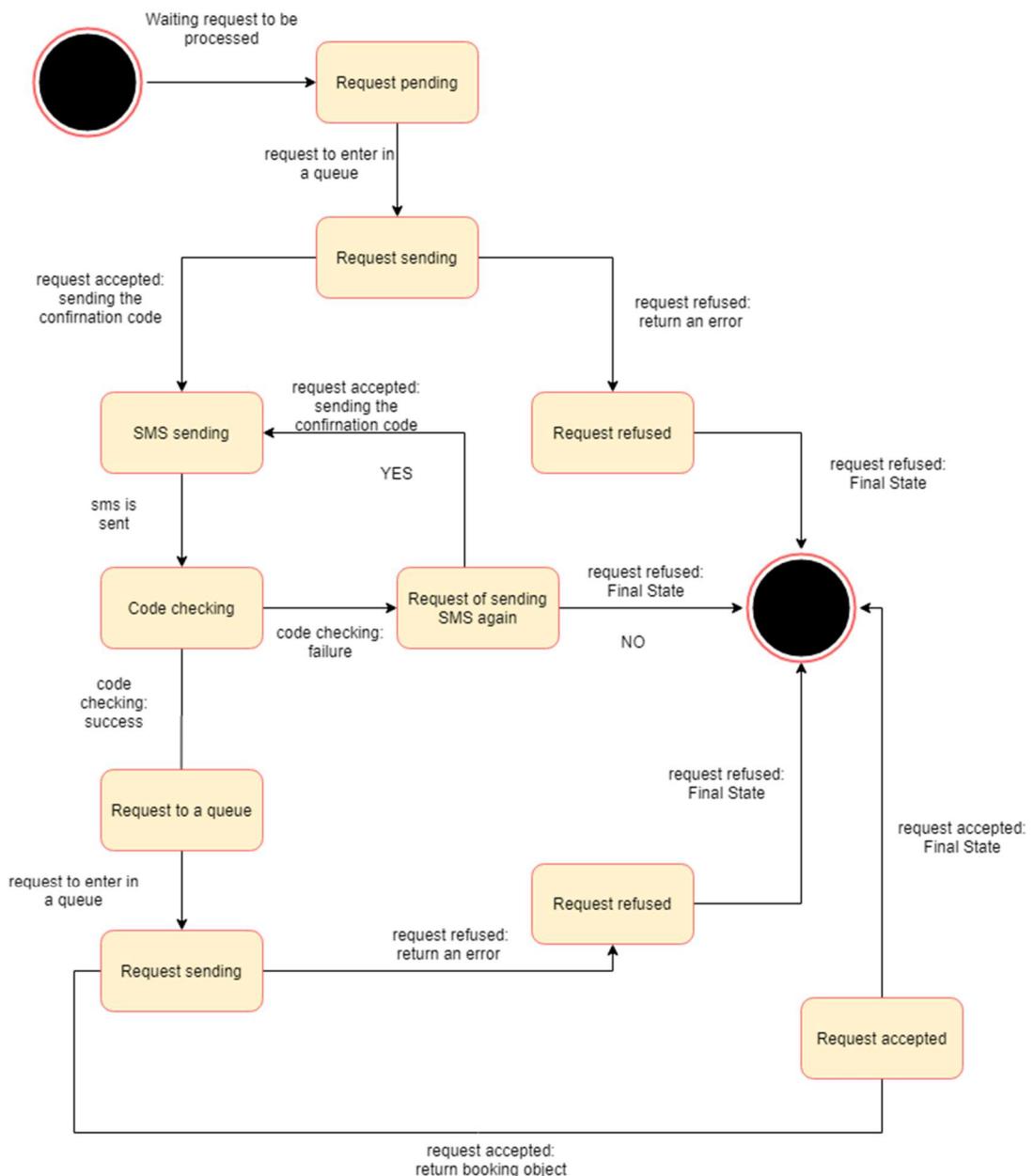


Figure 3 - State diagram: creation of a connection session via SMS

Figure 4 shows the step-by-step authorization of a Prioritized user (who books the queue, being physically in the store).

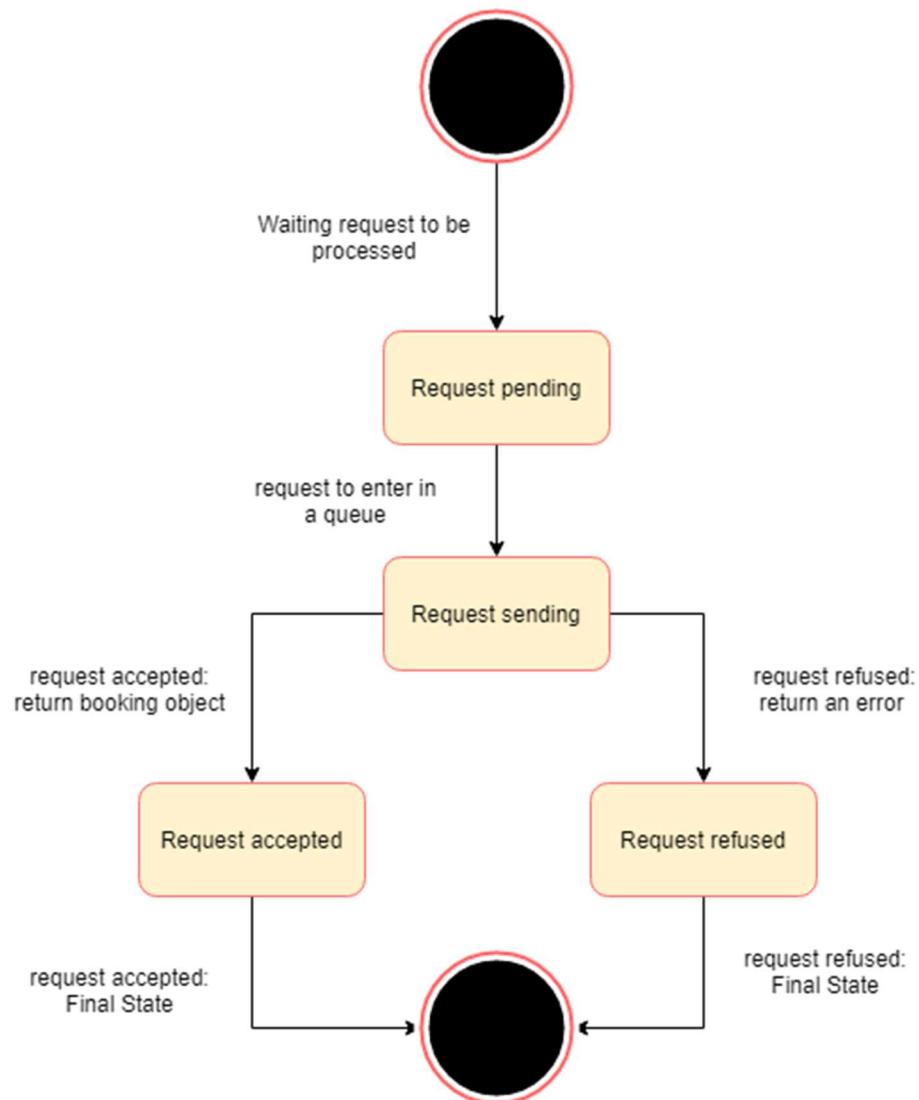


Figure 4 – State diagram: authorization of a Prioritized user

It is important to show the authorization process by store employees, what is observed in *Figure 5*.

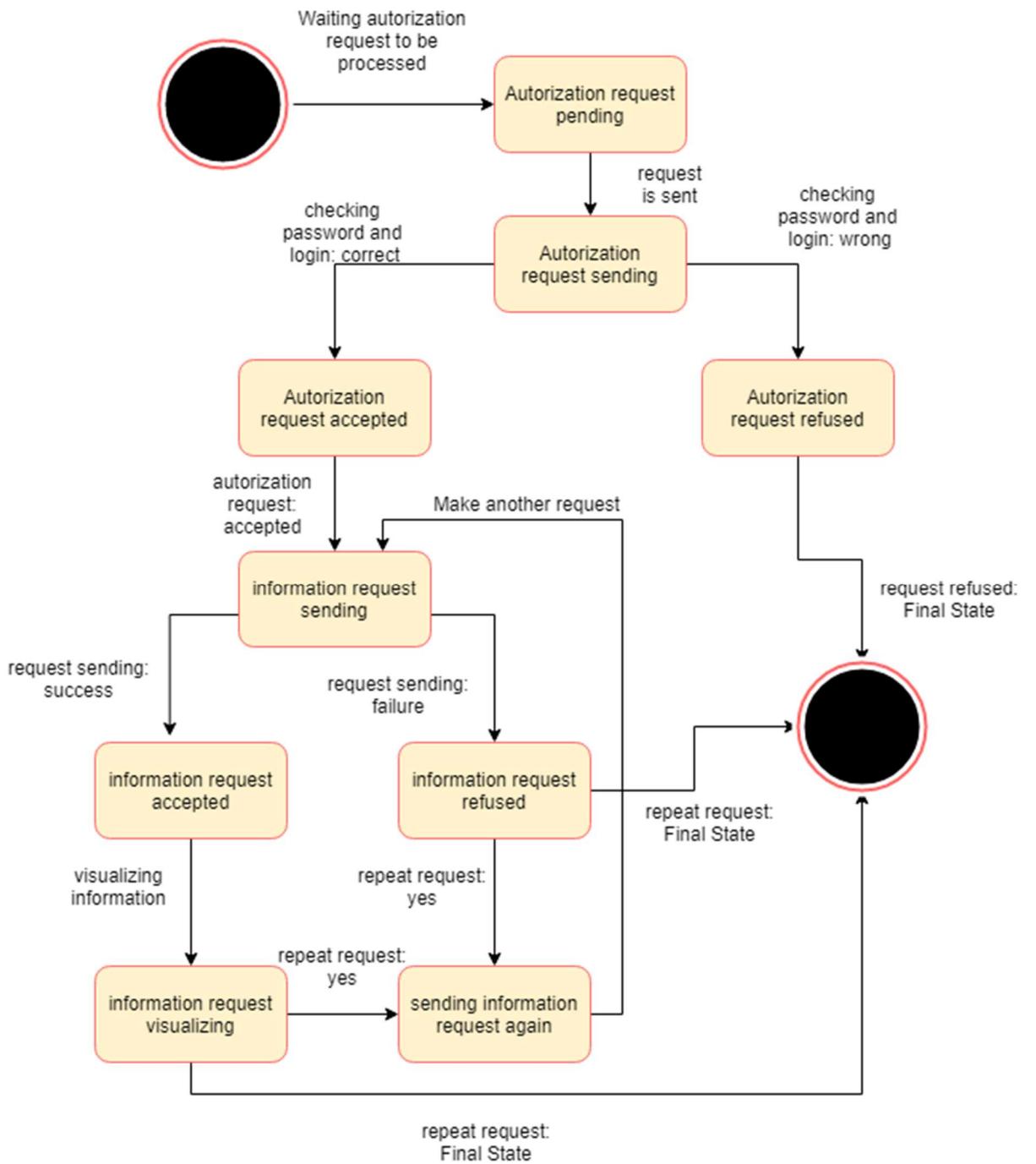


Figure 5 – State diagram: authorization process by store employees

2.2 Product functions

The following section contains the main functions that software will perform. As said earlier, there are additional features available for the personnel staff, but here will be displayed just the general functions.

2.2.1 Booking management

The most exciting thing about this product function is the opportunity of the customer - to not register. The system will provide SMS authentication and GPS location that will be enough to unambiguously define the user. The STB will not collect any personal data about users other than their phone number and their names (this feature is optional, if the user doesn't like to be called by his name via notifications, it's his choice). The system also will provide the tickets in place because not every human got his phone or could install an app. If the user could not get to the store in the estimated time, he will get the notification

reminding him that either he is just late but he goes to the store or he forgot about the store and he had to get a new booking. By the way, the user can plan his booking, he can choose every available slot in the timetable, or he can book as quickly as possible. The user can pick any store that is available. Every user can cancel the booking without any sanctions given to him. While performing a booking the user could also enter the data about departments he is going to visit the store, this action will be very helpful to preserve his own health and the others. The STB will process the data given by the user and return the solution that will help him not to be infected or not infect the others.

2.2.2 Location management

To reduce the possibility of a queue, the system will monitor the location of the user and will provide him the shortest path to the selected store. It can be obtained by using GPS to retrieve the current location of the user. The STB uses the shortest path but the maximum estimated time because the system couldn't know about the personal characteristics of the users.

2.2.3 Arrival/departure management

The arrival and departure of the user will be checked via QR code. When the user arrives at the store and his booking is still active then his QR code will be valid. The customer scan QR code and the STB starts the countdown because some users have entered the information about the departments they are going to visit, otherwise the system supposes that the time they will spend at the store will be distributed uniformly. When user time comes to an end, there will be a notification either via mobile phone. At the departure moment, the user scans the QR code again and the system supposes that the customer visit ends.

2.2.4 Others

The administration of the store must enter the system the store specifications i.e. its size, the departments' location, etc. On-site tickets are also handled by the store staff. The tickets given by the staff will have priority over the booking via app since the users are already in the store and they are more likely to get sick than the users that use a booking option. The administration of the store will be able to collect information about a human flow which is going through a store, to evaluate human load in different periods.

2.3 User characteristics

The following points describe the general characteristics that may influence usability. Hence, the actors are the following:

1. Common User: a person that book a visit from home. To do that, this person must enter the system his phone number and enable geolocation for the app. This user has access to the basic functionality i.e. the booking, choosing a store, and departments in its store.
2. Privilege User: a personnel staff of the store. It has access to the technical information and can enter the metadata about the store i.e. store size, its departments, and their sizes.
3. Prioritized User: a person that book a visit via machine. This person doesn't need to provide his phone number or anything, but he has to get a ticket from the machine. He will get a higher priority than other users in the queue.

2.4 Assumptions, dependencies and constraints

D1	Registration is via the phone number
D2	The device must have stable access to the Internet to establish a user status and queue calculation
D3	The user goes to the selected shop according to the notification on the mobile device

D4	The user should follow the rules that will be notified on the device (i.e. social distance, the presence of a mask, allocated time for purchase)
D5	People who cannot use the device should easily integrate into the queue by registering on the spot
D6	The customer will enter/exit the store using a QR code
D7	Store owners will have access to an expanded version of the app for more control and tracking of the flow of customers.
D8	The user should denote the supposed departments to visit
D9	Store owner must specify the area of the shop and departments
D10	The ticket given by machine will contain information about the customer arrival/departure time and QR code
D11	The user is going to the store by shortest path
D12	If the user is late more than 10 minutes (according to his/her appointed time) he/she is removing from the queue
D13	It is possible to cancel the booking if user plans are changed
D14	The location of the user is obtained by GPS. The locations of the stores are retrieved by API

For the personal use of the application, the user must have a mobile phone or a tablet, if he doesn't want to install the app or he doesn't have the phone/tablet, then he can use the store machine. Again, in case of personal use the device must match the following characteristics: 2G/3G/4G/802.11 (a/b/g/n/ac) Internet connection, it is welcomed if this device had GPS function enabled.

The device of common use must be connected to a permanent power supply. Device of common use must have the touch screen to provide a communication interface, also it must have facilities to print tickets.

For the personnel staff access, there must be any device (a PC, a laptop, a tablet, etc.) that has a stable connection to the Internet. Also, the application uses the Yandex Maps API that provides more lightweight architecture.

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

To represent the approximate concept and interface of the program, the first mockups are shown below.

3.1.1.1 User verification by sending an SMS to the specified number

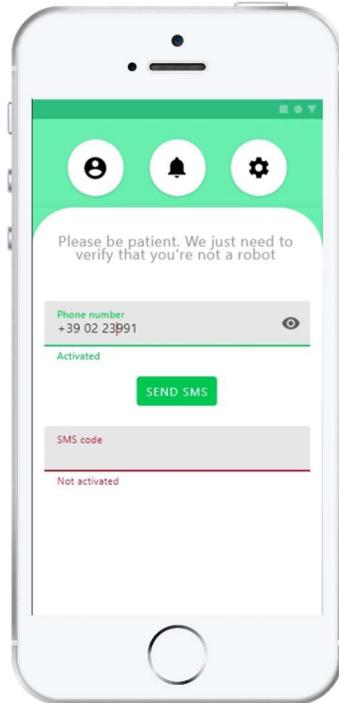


Figure 6 – CLUp: customer verification

3.1.1.2 Display and selection of the nearest available stores of the user

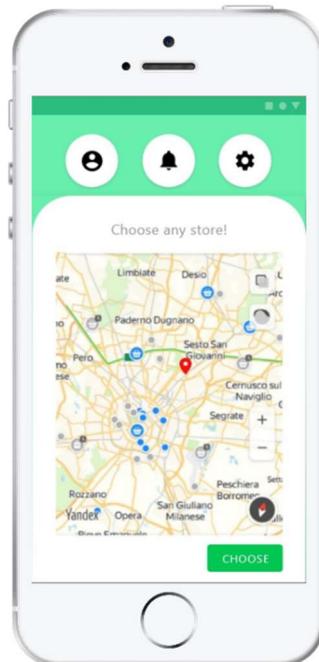


Figure 7 – CLUp: store choosing

3.1.1.3 Selecting the appropriate store visit dates



Figure 8 – CLup: date choosing

3.1.1.4 Selecting the appropriate store visit time

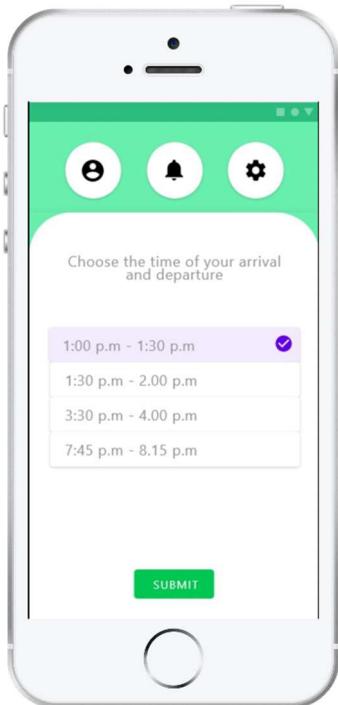


Figure 9 – CLup: time choosing

3.1.1.5 Selection of prospective departments to visit

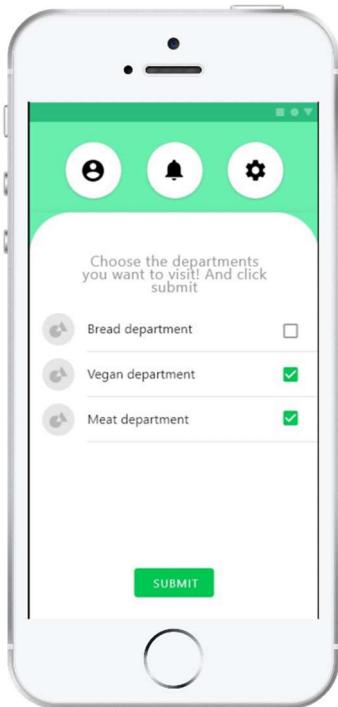


Figure 10 – CLup: department choosing

3.1.1.6 Generate the QR code required for entering and exiting



Figure 11 – CLup: QR code generation

3.1.1.7 Building a route to the selected store

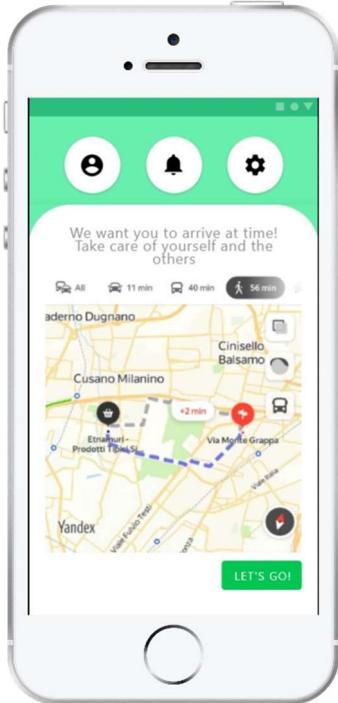


Figure 12 – CLup: built route

3.1.2 Hardware user interfaces

For third parties, it is important to have the following equipment:

- registration terminal for entering the queue and receiving a ticket with a QR code
- General information Board that displays all the numbers of customers waiting in line and their estimated time to enter
- a tableau at the employee's workplace (can be a computer that is authorized with the store's account), which contains all information about queue management

3.1.3 Software interfaces

The system does not provide any API to external applications, but it uses some proven services and ready-made functionality for stable performance.

- Yandex Maps API

Yandex Maps API is a set of services that allow to use Yandex map up-to-date information such as public transport schedules, traffic jams, reconstruction in a project. In our case, as mentioned, it is used to lay out the route and display the stores available in the area and their additional information such as opening hours, load, approximate waiting time.

- Calendar and alert system

Planning a trip to the store is not possible without a calendar and specifying the exact time of arrival. After viewing the available dates and time of booking, the user assigns the most comfortable one for them. For convenience, it is recommended using a notification system about planned visit

- QR code reading/generation

An important attribute of the program is the QR code. It is generated on the server side after the request to visit the store is confirmed. It is used to enter and exit the store after scanning by the staff

3.2 Functional requirements

3.2.1 Common User

Scenario 1

Because of the pandemic Hazel does not want to endanger her own life and lives of the others, and she does not leave the home. But using food delivery services soon became too expensive for her. And she started to look for the other ways to get food supplies, so as an active app user she downloaded and installed CLup. This service helped her to get to the store, buy all the supplies she needed and reduce contact with people to a minimum.

Use case diagram

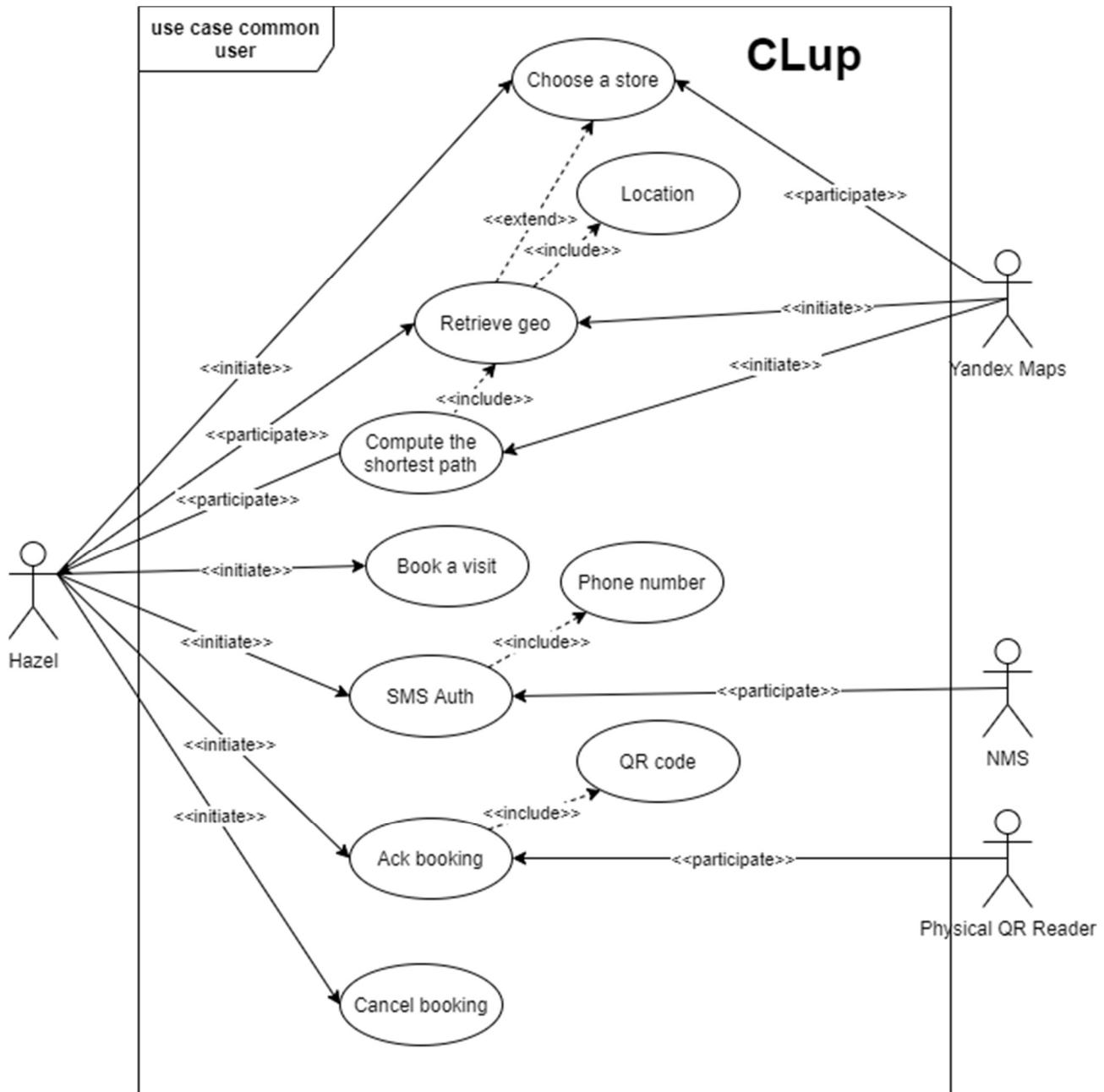


Figure 13 – Use case diagram: Common User

Use cases

Name	Choose a Store
Actor	Hazel, Yandex Maps
Entry conditions	1. Hazel has opened the application on her device 2. (Optionally) She could turn on the geo location
Events flow	1. Hazel chooses a store on the map 2. Hazel clicks on the “Choose” button 3. The app saves the data
Exit conditions	Hazel has chosen the store where she would buy the supplies
Exceptions	1. Hazel forgot to turn on the Internet connection, so she would get the Connection Error 2. Hazel has chosen the store that is closed this time, then the app would display the closed store status

Name	Location
Actor	Hazel
Entry conditions	Hazel has opened the application on her device
Events flow	1. The app displays the geolocation banner that asks the user to turn on the geolocation 2. Hazel approves or rejects 3. The app saves the data
Exit conditions	Hazel approved or rejected the usage of geolocation by this service
Exceptions	1. Unavailability of Yandex Maps API Servers

Name	Retrieve geo
Actor	Yandex maps, Hazel
Entry conditions	1. Hazel has opened the application on her device 2. The location must be approved
Events flow	1. Yandex maps retrieve Hazel geo position 2. Yandex maps display Hazel geo position on the map
Exit conditions	1. Yandex Maps API has retrieved geo position
Exceptions	1. Unavailability of Yandex Maps API Servers

Name	Compute the shortest path
Actor	Yandex maps, Hazel
Entry conditions	<ul style="list-style-type: none"> 1. Hazel has opened the application on her device 2. The location must be approved 3. Yandex maps must retrieve Hazel geo position
Events flow	<ul style="list-style-type: none"> 1. Yandex maps retrieve Hazel geo position 2. Hazel has chosen the store 3. Yandex maps return the shortest path to this store
Exit conditions	<ul style="list-style-type: none"> 1. Yandex Maps API calculated the shortest path
Exceptions	<ul style="list-style-type: none"> 1. Unavailability of Yandex Maps API Servers 2. Unexpected road works or accidents

Name	Book a visit
Actor	Hazel
Entry conditions	<ul style="list-style-type: none"> 1. Hazel has opened the application on her device 2. Hazel has chosen the store
Events flow	<ul style="list-style-type: none"> 1. Hazel chooses the departments she wants to visit 2. Hazel chooses the date and the time of the visit 3. Hazel enters the phone number 4. Hazel clicks “Send SMS” button
Exit conditions	<ul style="list-style-type: none"> 1. The booking has been done
Exceptions	<ul style="list-style-type: none"> 1. The Internet Connection is lost 2. The occurrence of “Nonrepeatable Read”

Name	SMS Auth
Actor	Hazel, NMS
Entry conditions	<ul style="list-style-type: none"> 1. Hazel has opened the application on her device 2. Hazel has chosen the store 3. Hazel has finished choosing the booking entry data 4. Hazel has entered her phone number
Events flow	<ul style="list-style-type: none"> 1. Hazel verifies the SMS code 2. Hazel gets the QR code acting as a ticket to the store 3. Hazel clicks “Ok” button
Exit conditions	<ul style="list-style-type: none"> 1. The user successfully authenticated
Exceptions	<ul style="list-style-type: none"> 1. The Internet Connection is lost 2. Hazel has not got the QR code 3. Hazel has not got the SMS code
Name	Ack booking
Actor	Hazel, Physical QR reader

Entry conditions	1. Hazel has booked a visit 2. Hazel has arrived to the store
Events flow	1. Hazel scans the QR code when she arrives 2. Hazel scans the QR code when she leaves
Exit conditions	1. Hazel leaves the store
Exceptions	1. Hazel did not arrive at the store 2. Hazel accidentally scans the QR code twice (maybe even thrice)
Special Requirements	1. Hazel receives a reminder notification 15 minutes before the estimated arrival at the store

Name	Cancel booking
Actor	Hazel
Entry conditions	1. Hazel has booked a visit
Events flow	1. Hazel goes to the account tab 2. Hazel cancels the visit
Exit conditions	1. The booking has been canceled
Exceptions	1. The Internet connection is lost

Sequence diagrams

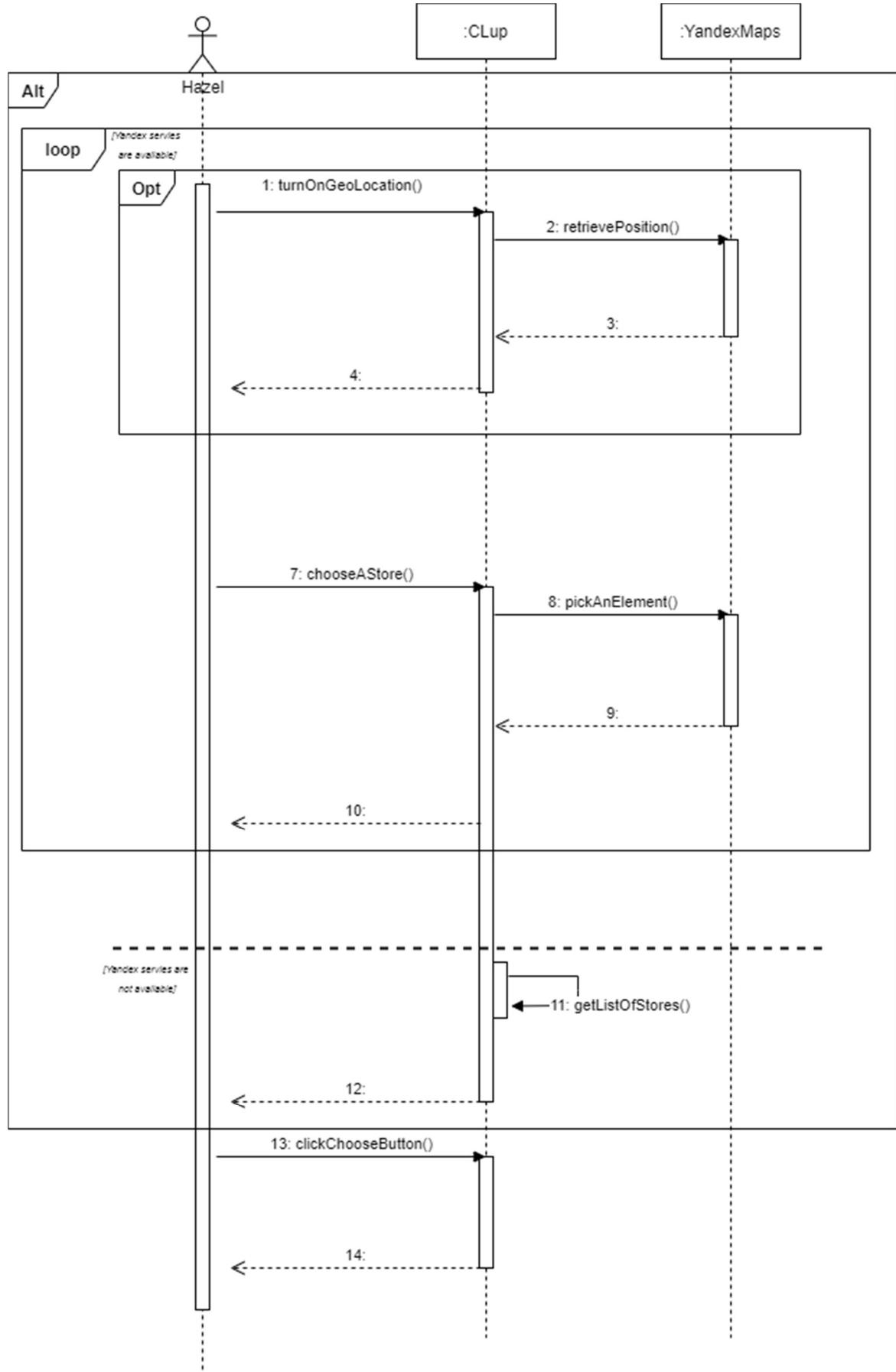


Figure 14 – Sequence Diagram: store choosing

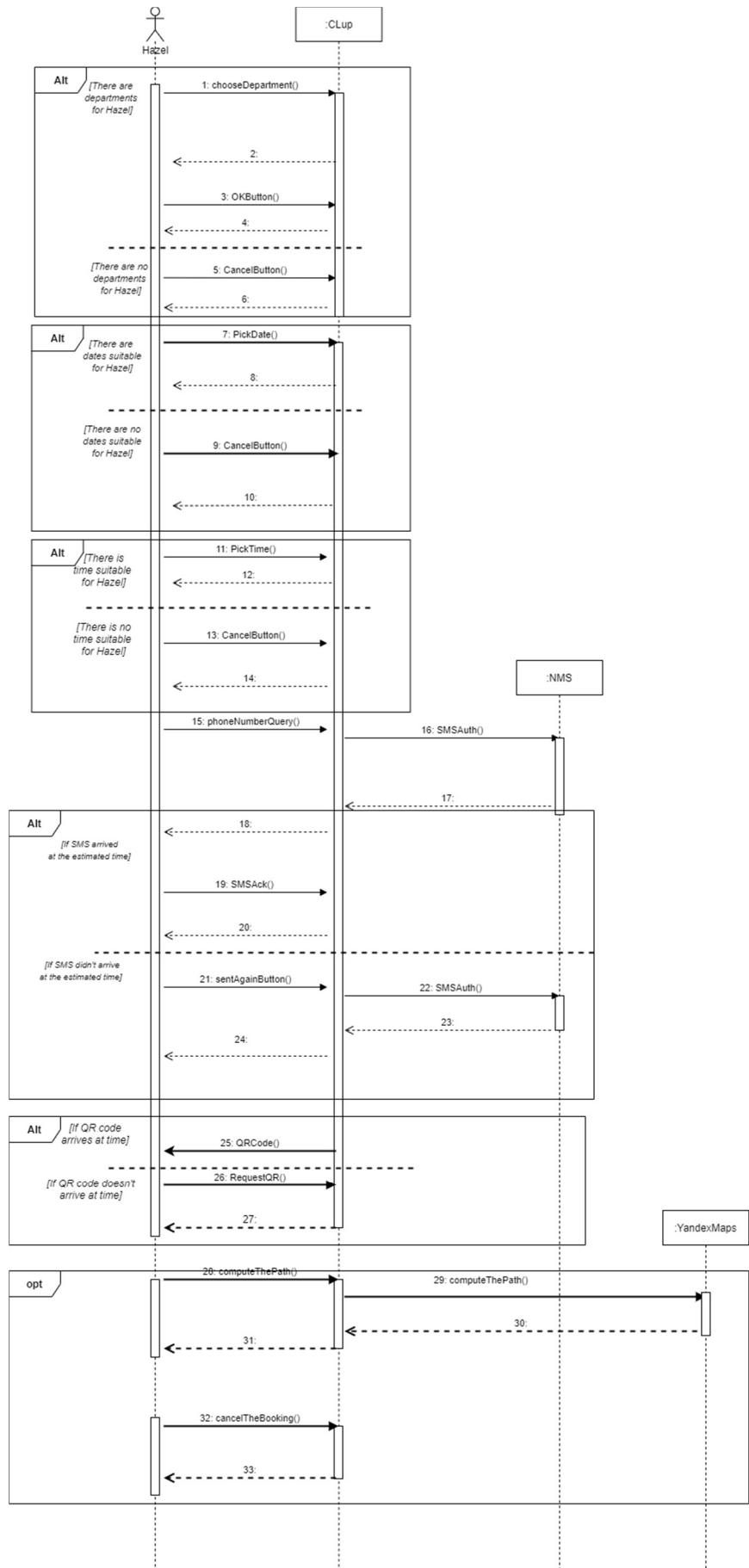


Figure 15 – Sequence Diagram: booking visit

3.2.2 Prioritized User

Scenario 2

Alexander is an elderly Afghanistan veteran, but the pandemic knocked him down. He has lost all the joys of his life: he could not see his relatives and he couldn't even go to the store without the risk of getting sick. He does not have a smartphone, and new technologies seem too complicated for him. So, Alexander goes to the store and he sees some machine giving tickets. Thanks to the presence of this machine the risks of getting sick for him have decreased.

Scenario 3

Julie is a student and as every young person, she does not like planning. Walking along the embankment (yes, she does not want to observe self-isolation) she realized that she would like to have a snack. So, she goes to the nearest store and books a visit via the machine using the "CLUp" service. This service will help the people concerning their health to minimize the risk of illness.

Use case diagram

The machine & QR scanner is the one device. This division was made in the diagram just to make it clear that the QR scanner is embedded into the machine.

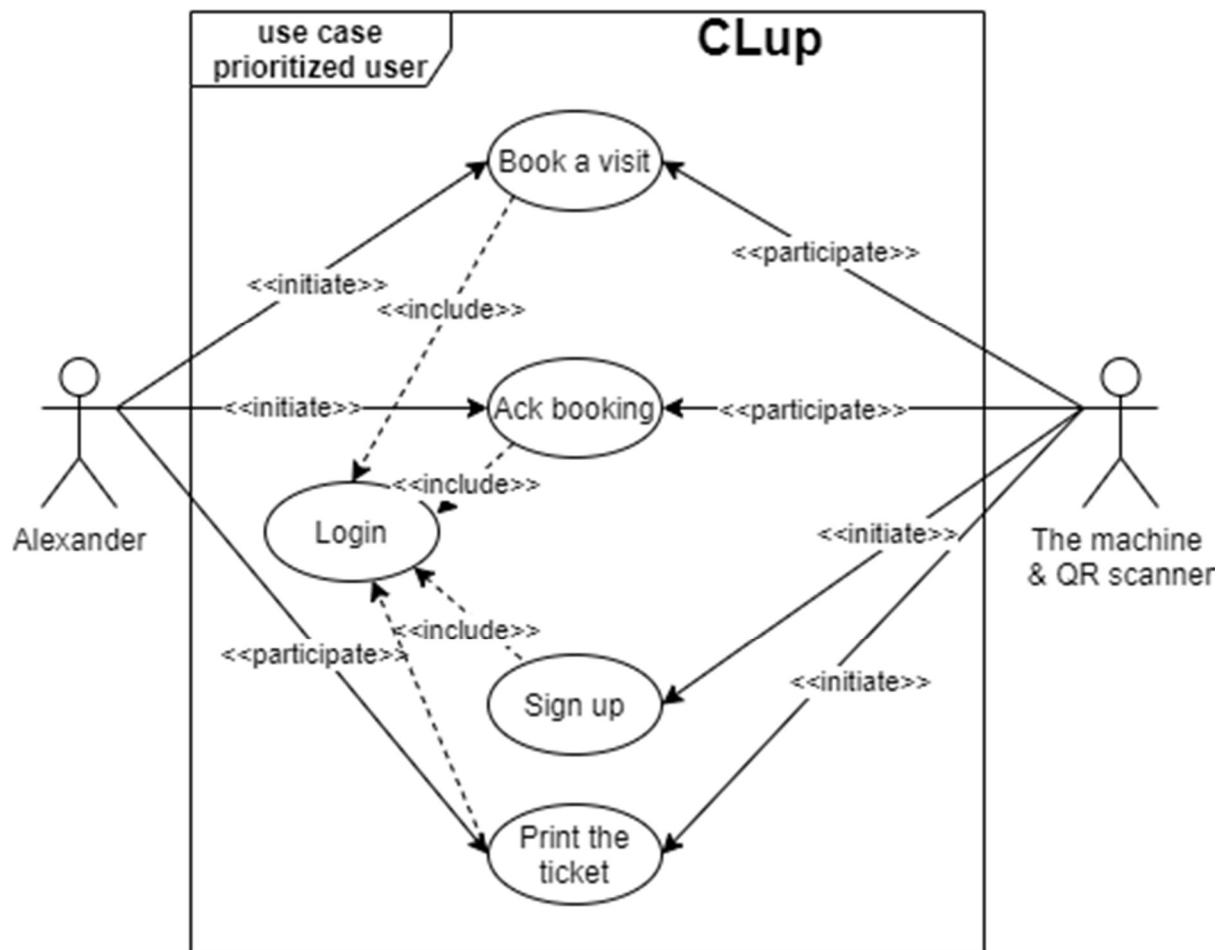


Figure 16 – Use case diagram: Prioritized User

Name	Book a visit
Actor	Alexander, the machine & QR scanner
Entry conditions	1. Alexander is in the store and near the machine 2. Machine has been registered in the system
Events flow	1. Alexander clicks the button “book a visit” 2. The machine shows the QR code and Alexander click “Ok”
Exit conditions	The booking has been done
Exceptions	1. There is no available slots the next hour
Special Requirements	1. Availability of a consultant to help and control the flow of people

Name	Ack booking
Actor	Alexander, the machine & QR scanner
Entry conditions	1. Alexander has booked a visit 2. Alexander has arrived at the store
Events flow	1. Alexander scans the QR code when he arrives 2. Alexander scans the QR code when he leaves
Exit conditions	1. Alexander leaves the store
Exceptions	1. Alexander did not arrive at the store 2. Alexander accidentally scans the QR code twice (maybe even thrice)

Name	Sign up
Actor	The machine & QR scanner, staff personal
Entry conditions	1. The machine has turned on
Events flow	1. Personnel staff enters the required data to register the machine in the system
Exit conditions	1. The machine has been registered
Exceptions	1. The Internet connection is lost 2. The serial number doesn't exist

Name	Login
Actor	The machine & QR scanner, staff personal
Entry conditions	1. The machine has turned on
Events flow	1. Personnel staff inputs the serial number of the machine
Exit conditions	1. The machine has been logged in
Exceptions	1. The Internet connection is lost 2. The serial number doesn't exist 3. Power failure

Name	Print the ticket
Actor	The machine & QR scanner, Alexander
Entry conditions	1. The visit has been booked
Events flow	1. The machine prints the ticket
Exit conditions	1. Alexander has got the ticket
Exceptions	1. The ink is out 2. The Internet connection is lost 3. Out of print paper 4. The user loses the received ticket

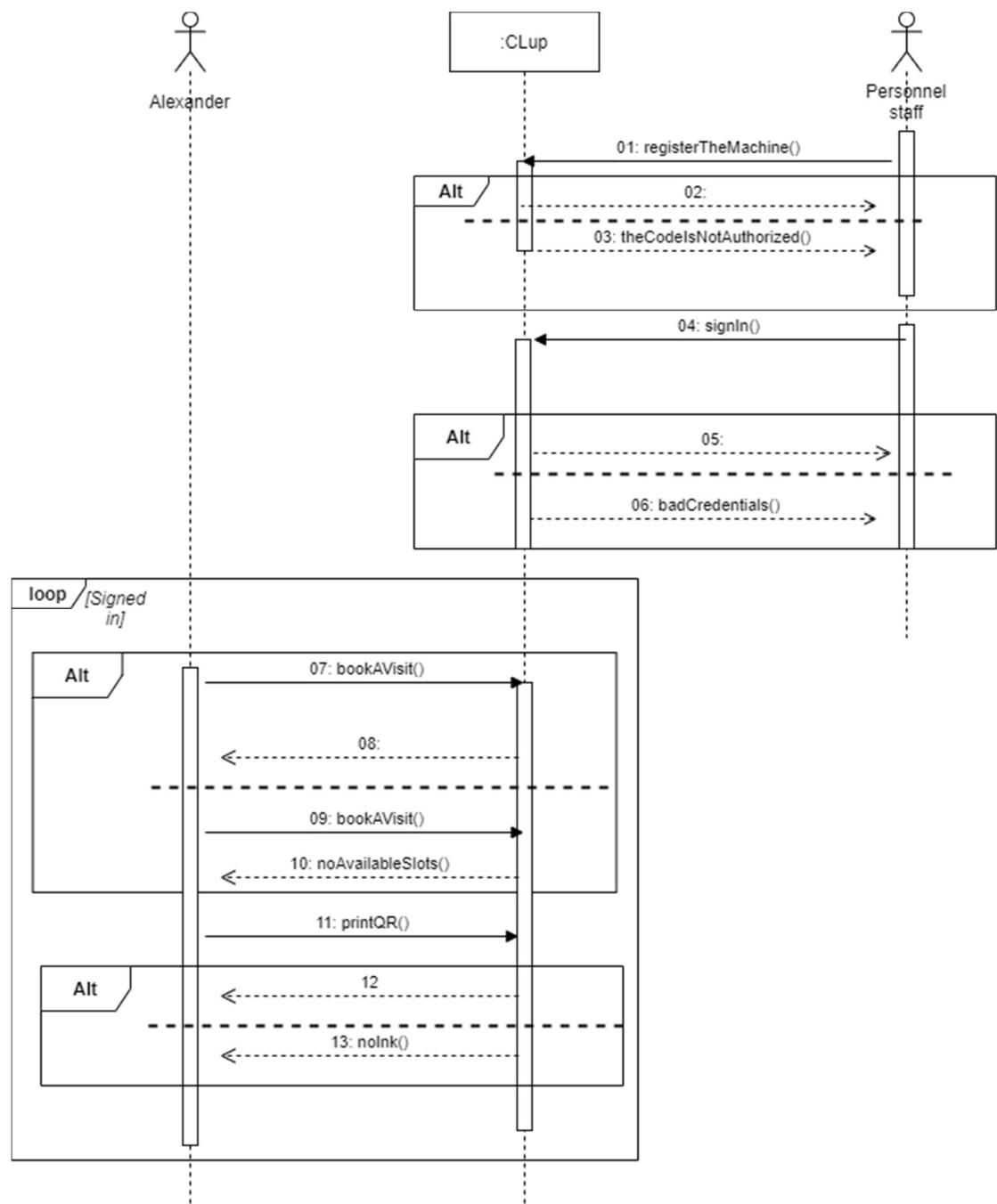


Figure 17 - Sequence Diagram: Book A Visit via the machine

3.2.3 Privilege User

Scenario 4

Working in a store during the pandemic is a big risk. Dave figured it out the hard way. He had COVID-19, and when he got on his feet, he decided to secure the store in which he works as a manager. So, his choice fell on the CLUp because he does not need to recruit couriers, it will help him to save some money at this difficult time. All he needs to do is to estimate the number of people for each department and in line at the cash register and provide this data to the system.

Scenario 5

Jimmy is the store owner, and as every businessman he wants to get profit. But in nowadays the profit is not high, because of the pandemic. So, Jimmy thought about getting his store safer for every customer. But he wants to analyze the possibilities to increase profit without endangering his customers. The CLUp provides the analytic services such safe way.

Use case diagram

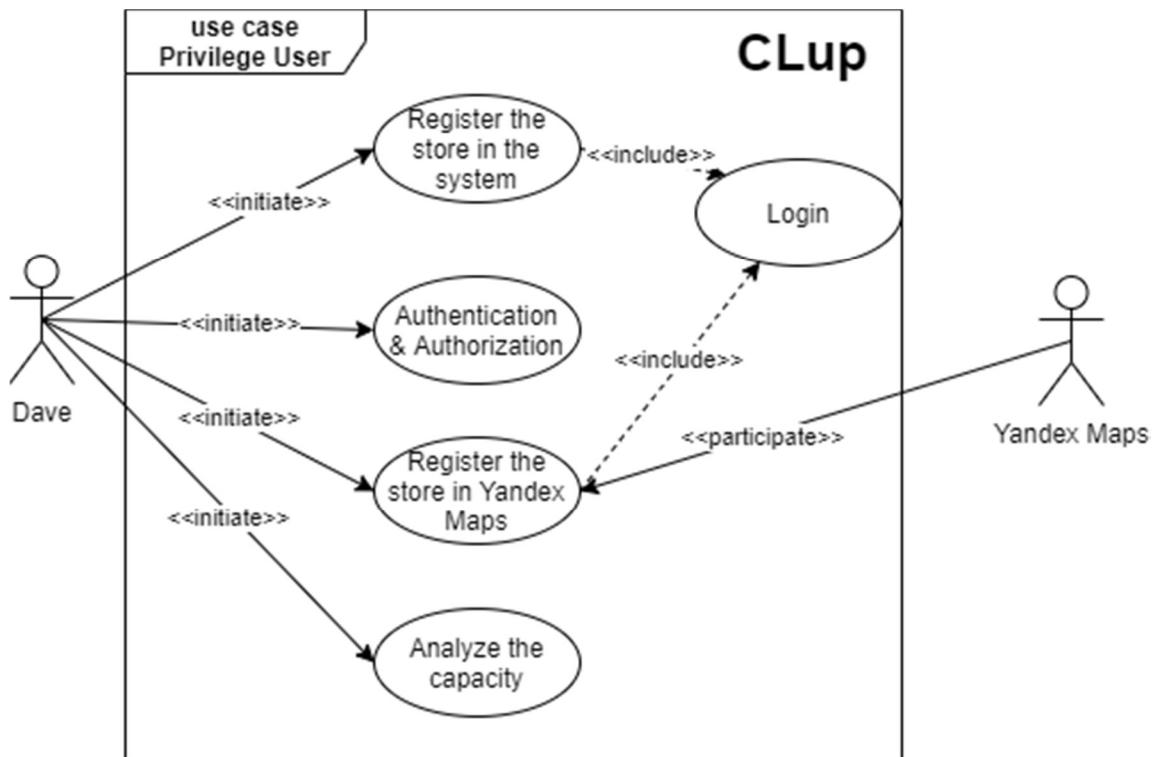


Figure 18 – Use case diagram: Privilege User

Name	Authentication & Authorization
Actor	Dave
Entry conditions	Dave has opened PC and started the app
Events flow	<ol style="list-style-type: none"> 1. Dave chooses the “Create” option 2. Dave enters login, password and phone number 3. Dave chooses the “OK” button 4. The system saves it
Exit conditions	<ol style="list-style-type: none"> 1. Dave is authenticated and authorized as a manager 2. System saves the data
Exceptions	<ol style="list-style-type: none"> 1. He inputs incorrect data (not correct phone number, long login, not strong password) 2. The session was interrupted for technical reasons

Name	Login
Actor	Dave
Entry conditions	Dave has signed in
Events flow	<ol style="list-style-type: none"> 1. Dave chooses the “Sign in” option 2. Dave enters login, password 3. Dave chooses the “OK” button
Exit conditions	Dave is logged in
Exceptions	<ol style="list-style-type: none"> 1. He inputs incorrect data (a login or a password) 2. The session was interrupted for technical reasons

Name	Register the store in the store system
Actor	Dave
Entry conditions	Dave has logged in
Events flow	<ol style="list-style-type: none"> 1. Dave enters the name of the store 2. Dave enters the address of the store 3. Dave inputs the personnel staff of the store and authorizes them to register the machines 4. Dave inputs the area of the store 5. Dave inputs the number of the departments and their area 6. Dave inputs the number of cash registers 7. Dave clicks “Ok” button 8. The system saves the information about this store
Exit conditions	The system has the information about the store. Dave logs out
Exceptions	<ol style="list-style-type: none"> 1. He inputs invalid data 2. He misprints some data and wants to correct it 3. The session was interrupted for technical reasons

Name	Register the store in Yandex Maps
Actor	Dave, Yandex Maps
Entry conditions	Dave has logged in
Events flow	<ol style="list-style-type: none"> 1. Dave creates the mark on Yandex Maps 2. Dave enters the name, address and the description of the store 3. Dave marks the store as active in the system 4. The system saves the information
Exit conditions	The store and its description are displayed on Yandex maps
Exceptions	<ol style="list-style-type: none"> 1. He inputs invalid mark 2. He misprints some data and wants to correct it 3. He enter invalid name or address 4. He forgot to mark the store as active 5. The session was interrupted for technical reasons

Name	Analyze the capacity (flow)
Actor	Jimmy
Entry conditions	Jimmy has logged in
Events flow	<ol style="list-style-type: none"> 1. Jimmy chooses the store in the list of his stores. 2. Jimmy clicks the button “Analyze” 3. He could do every manipulation with data
Exit conditions	Jimmy has observed the necessary information
Exceptions	<ol style="list-style-type: none"> 1. There is no data for the chosen store yet. 2. He entered the incorrect settings for modelling

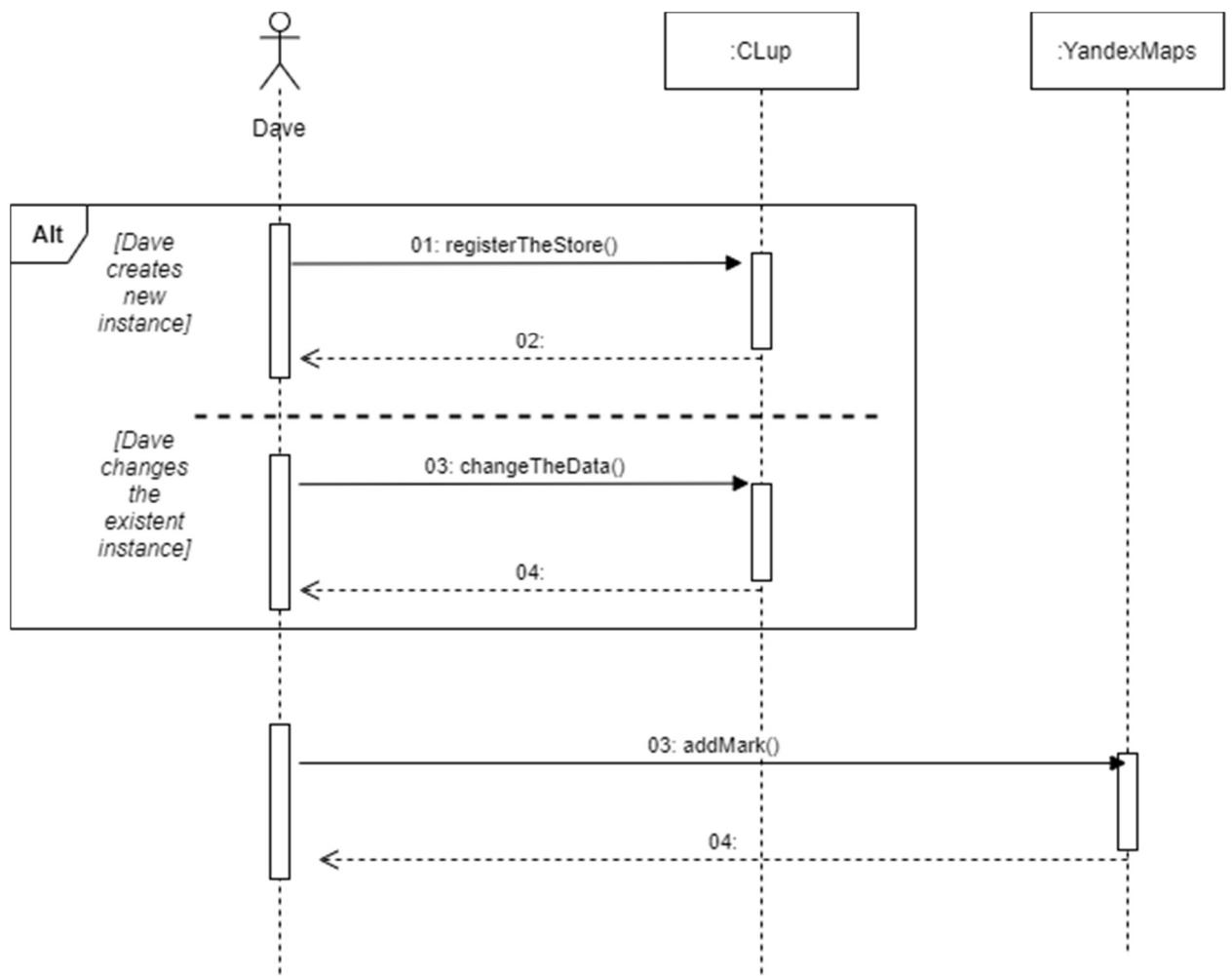


Figure 19 – Sequence Diagram: Register the store in the system

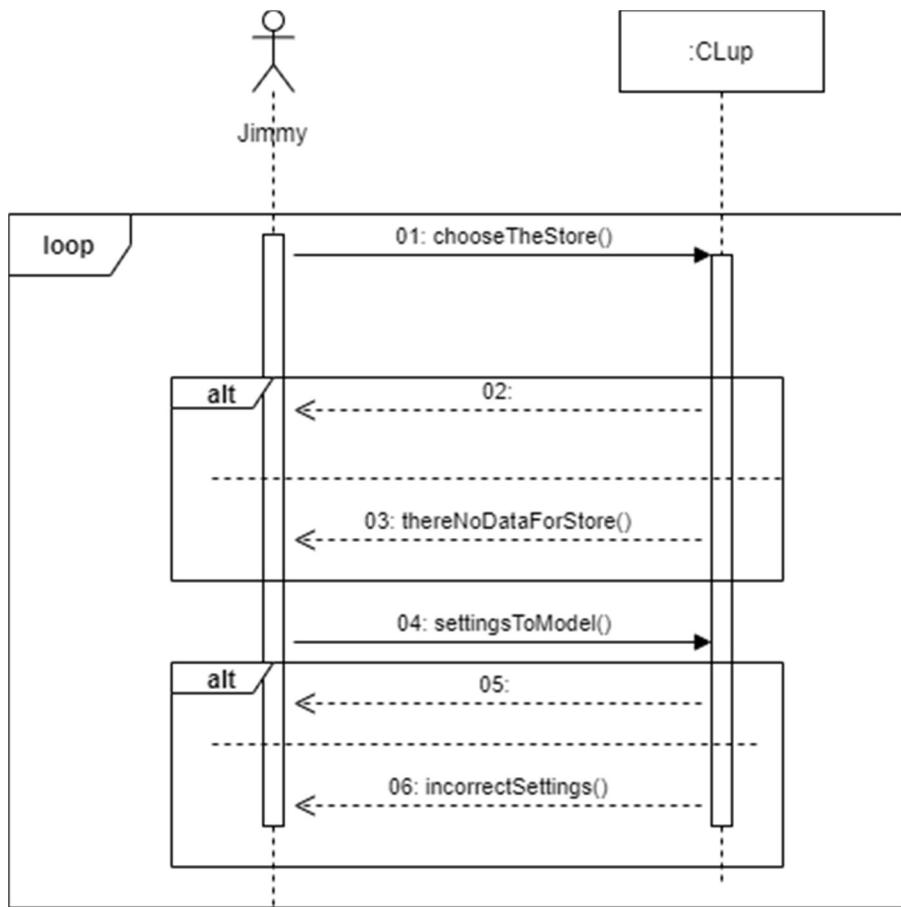


Figure 20 – Sequence Diagram: Analyzing

3.2.4 Requirements

G1	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
D2	The device must have stable access to the Internet to establish a user status and queue calculation.
D3	The user goes to the selected shop according to the notification on the mobile device.
D11	The user is going to the store by shortest path.
D12	If the user is late more than 10 minutes (according to his/her appointed time) he/she is removing from the queue.
D14	The location of the user is obtained by GPS. The locations of the stores are retrieved by API.
R1	The system should allow users to go to the store by shortest path calculated by Yandex Maps.
R2	All information about visit (arrival/departure time, the selected departments) should be analyzed by system, to estimate the time of waiting more precisely.

G2	To give people opportunity to stay in queue remotely to avoid crowds in stores and in real queues.
D1	Registration is via the phone number.
D4	The user should follow the rules that will be notified on the device (i.e. social distance, the presence of a mask, allocated time for purchase).
D6	The customer will enter/exit the store using a QR code

D8	The user should denote the supposed departments to visit.
D9	Store owner must specify the area of the shop and departments.
D12	If the user is late more than 10 minutes (according to his/her appointed time) he/she is removing from the queue.
D13	It is possible to cancel the booking if user plans are changed.
R3	The system should send the data of the visit to the server to organize the queue correctly.
R4	The system should provide concurrency to insert multiple request in short period of time in correct order.
R5	The system should balance the human flow in the store by analyzing the departments and store capacity.
R6	The system generates QR codes by UUID, telephone number/device id and suggested time of the visit.

G3	Stores should have the possibility to hand out “tickets” on the spot.
D2	The device must have stable access to the Internet to establish a user status and queue calculation.
D5	People who cannot use the device should easily integrate into the queue by registering on the spot.
D6	The customer will enter/exit the store using a QR code.
D8	The user should denote the supposed departments to visit.
D10	The ticket given by machine will contain information about the customer arrival/departure time.
D12	If the user is late more than 10 minutes (according to his/her appointed time) he/she is removing from the queue.
R7	The system should distinguish users of mobile phones and local devices to prioritize the second ones in the queue.
R8	QR readers should be installed at the entrance and exit of the store sending data of arrival and departure time to system.
R9	The machine should be authorized and logged in the system by the manager of the store.

G4	The system should allow customers to book a visit to the supermarket to optimize quantity of people in it by time of their visit, and category of products which they want to purchase.
D1	Registration is via the phone number.
D3	The user goes to the selected shop according to the notification on the mobile device.
D6	The customer will enter/exit the store using a QR code.
D8	The user should denote the supposed departments to visit.
D9	Store owner must specify the area of the shop and departments.
D12	If the user is late more than 10 minutes (according to his/her appointed time) he/she is removing from the queue.
D13	It is possible to cancel the booking if user plans are changed.
R10	The system automatically extends the queue if the user books a visit for particular date and time.
R11	CLup service appeals to NMS to provide the authentication of the user.

G5	The system (application and “tickets” on the spot) should include alternative slots (for another day), suggest to the customer the location of the nearest “safe” store based on his location.
D3	The user goes to the selected shop according to the notification on the mobile device.
D5	People who cannot use the device should easily integrate into the queue by registering on the spot.
D6	The customer will enter/exit the store using a QR code.
D9	Store owner must specify the area of the shop and departments.
D14	The location of the user is obtained by GPS. The locations of the stores are retrieved by API.
R12	The system should provide algorithms to assess the stores human flow.
R13	The system should process the given location and base on it provide the list of accessible stores for the customer.

G6	The system should allow the third party to get the statistical information to perform better management of the store.
D5	People who cannot use the device should easily integrate into the queue by registering on the spot.
D6	The customer will enter/exit the store using a QR code.
D7	Store owners will have access to an expanded version of the app for more control and tracking of the flow of customers.
D9	Store owner must specify the area of the shop and departments.
R14	The staff of the store have to login into the system to obtain statistic to manage the store in a better way.
R15	The system provides to the store owner only statistical information with embedded in the system instruments for visualization.

4 Formal Analysis Using Alloy

4.1 Alloy Code

This section is devoted to the critical moments of the system described with the help of Alloy annotation. In addition to system entities, special attention is paid to structural constraints such as:

- Every store must have a Network
- There is no two Networks have the same store
- It cannot happen that some users have identical Phone no
- A queue can't hang

To make sure that the constraints are correct, the alloy compiler analyzes the world models and the corresponding assertion.

Also, the worlds are represented in the following stories. The first world tells us about the queue with three users in it: two are common users and one is prioritized. As you can see if the user chooses the particular departments, these departments are there and not in the second store. The second world is about when we've got registered the stores but there are not any customers yet. Every staff personnel has unique registration as the machines in the store. The last world shows us the case when two persons go to different stores and booked the visit at the same time. Every queue is different for every store.

```
abstract sig Bool {}

one sig True extends Bool {}
one sig False extends Bool {}

// MAC-address
one sig MAC {}

sig DateTime {
    year: one Int,
    month: one Int,
    day: one Int,
    hour: one Int,
    minute: one Int
} {
    // these values are just for modeling
    year >= 0 and
    month >= 0 and month <= 4 and
    day >= 1 and day <= 6 and
    hour >= 0 and
    minute >= 0
}

// in our case we suppose that UUID version 1 defined as in RFC 4122
// so model it as the combination of MAC and datetime
sig UUID {
    dt: one DateTime,
    mac: one MAC
}

sig Location {
    lat: one Int,
    lon: one Int
} {
    // the values are for modeling
    lat >= -3 and lat <= 3
    and lon >= -6 and lon <= 6
}
```

```

}

sig PhoneNo {}

sig Password {}

sig FullName {}

sig Address {}

sig Brand {}

sig RegCode {}

abstract sig Registration {
    phoneNo: one PhoneNo,
}

sig UserRegistration extends Registration {
}

sig StaffRegistration extends Registration {
    password: one Password,
}

sig MachineRegistration extends Registration {
    regCode: one RegCode
}

abstract sig User {
    fullName: one FullName,
}

sig PrivilegeUser extends User {
    reg: one StaffRegistration,
    store: some Store
}

abstract sig Customer extends User {
    departments: set Department,
    queue: some Queue
}

sig CommonUser extends Customer {
    priority: one False,
    regC: one UserRegistration,
    location: some Location,
    b: some Booking
}

sig PrioritizedUser extends Customer {
    priority: one True,
    b: one Booking
}

sig Network {
    name: one Brand,
    addressHead: one Address,
}

```

```

stores: some Store,
staff: some PrivilegeUser
}

sig Store {
    address: one Address,
    capacity: one Int,
    deps: some Department,
    regM: some MachineRegistration
} {
    // the value for modeling
    capacity >= 1 and capacity <= 6
}

sig Department {
    koef: one Int,
    capacity: one Int,
    area: one Int
} {
    // the value for modeling
    koef >= 0 and koef <= 3 and
    capacity >= 1 and capacity <= 2 and
    area >= 0 and area <= 3
}

sig Queue {
    books: some Booking,
    store: one Store
}

sig Booking {
    position: one Int,
    // state indicates delete, create, enter, left
    state: one Int,
    selectedStore: one Store,
    bookingDate: DateTime,
    estimatedTime: DateTime,
    uuid: UUID,
    departureTime: DateTime,
} {
    // estimatedTime in booking must be equal to difference between
    departureTime
    // and bookingDate
    // departureTime = bookingDate + estimatedTime
    position >= 0 and position <= 3 and
    state >= 0 and state <= 3
}

-----<<facts>>-----

// fact. Networks HQ are not in the same address
// we suppose we work with big shop networks like Auchan or 7-11
fact NoNetworkSameHQAddress {
    all disj n1, n2 : Network | n1 != n2 implies n1.addressHead != n2.addressHead
}

```

```

// There is no two Networks have the same store
// but two or more stores in different networks may have the same address
fact NoNetworkHasTheSameStore {
    all disj n1, n2: Network | n1 != n2 implies n1.stores != n2.stores
}

// fact. Every store must have a network
fact EveryStoreMustHaveANetwork {
    all s : Store | one n: Network | s in n.stores
}

// fact. We think it's not good to have the same Network stores share the
// same address
fact NoStoreNetworkShareTheSameAddress {
    all disj s, s': Store | one n: Network | s in n.stores and s' in
n.stores implies s.address != s'.address
}

// fact. PhoneNo are unique if user are not staff
fact PhoneNoUnique {
    all r, r': Registration | r.phoneNo != r'.phoneNo
}

// fact Passwords are unique
fact PassUnique {
    all r, r': StaffRegistration | r.password != r'.password
}

// fact two staff can't have the same registration
fact RegistrationStaffUnique {
    all disj st, st': PrivilegeUser | st.reg != st'.reg
}

// fact CommonUser can't have the same registration
fact CommonUserUnique {
    all disj c, c': CommonUser | c.regC != c'.regC
}

// fact MachineReg is unique
fact MachineUnique {
    all disj s, s': Store | s.regM not in s'.regM
}

// fact users can't have the same username
fact UsernameUnique {
    all u, u' : User | u != u' implies u.fullName != u'.fullName
}

// fact PrivilegeUser must work in the Network
fact NetworkStaff {
    all n : Network | all p:PrivilegeUser | p in n.staff
}

// fact every store must have departments (delete the hanging departmens)

```

```

fact DepartmentStore {
    all d : Department | one s: Store | d in s.deps
}

// fact booking can't hang
fact BookToQueue {
    all b: Booking | one q: Queue | b in q.books
}

// fact Queues are unique
fact UniqueQueue {
    all disj q, q': Queue | q != q' implies q.store != q'.store
}

// uuid are unique
fact UUIDUnique {
    all disj b1, b2 : Booking | b1 != b2 implies b1.uuid != b2.uuid
}

// fact Store the same for Queue and for Booking
fact UniqueStoreQueue {
    all b: Booking | some q: Queue | b in q.books implies b.selectedStore =
q.store
}

// fact regCode unique
fact RegCodeUnique {
    all disj m1, m2: MachineRegistration | m1.regCode != m2.regCode
}

-----<<predicates>>-----

pred w1 {
    #Department = 5
    #CommonUser = 2
    #PrioritizedUser = 1
    #Booking = 2
    #Store = 2
    #Queue = 1
    (some disj b1, b2: Booking | b1.bookingDate != b2.bookingDate)
}

run w1 for 5 but 2 MachineRegistration, 1 PrioritizedUser, 1 PrivilegeUser

pred w2 {
    #PrivilegeUser = 5
    #Store = 2
}

run w2 for 7

pred w3 {
    #Department = 2

```

```
#CommonUser = 2
#Booking = 2
#Store = 2
#Queue = 2
(some disj b1, b2: Booking | b1.selectedStore != b2.selectedStore
    implies b1.bookingDate = b2.bookingDate)
}

run w3 for 5 but 2 MachineRegistration, 1 PrioritizedUser, 1 PrivilegeUser
```

4.2 World generated

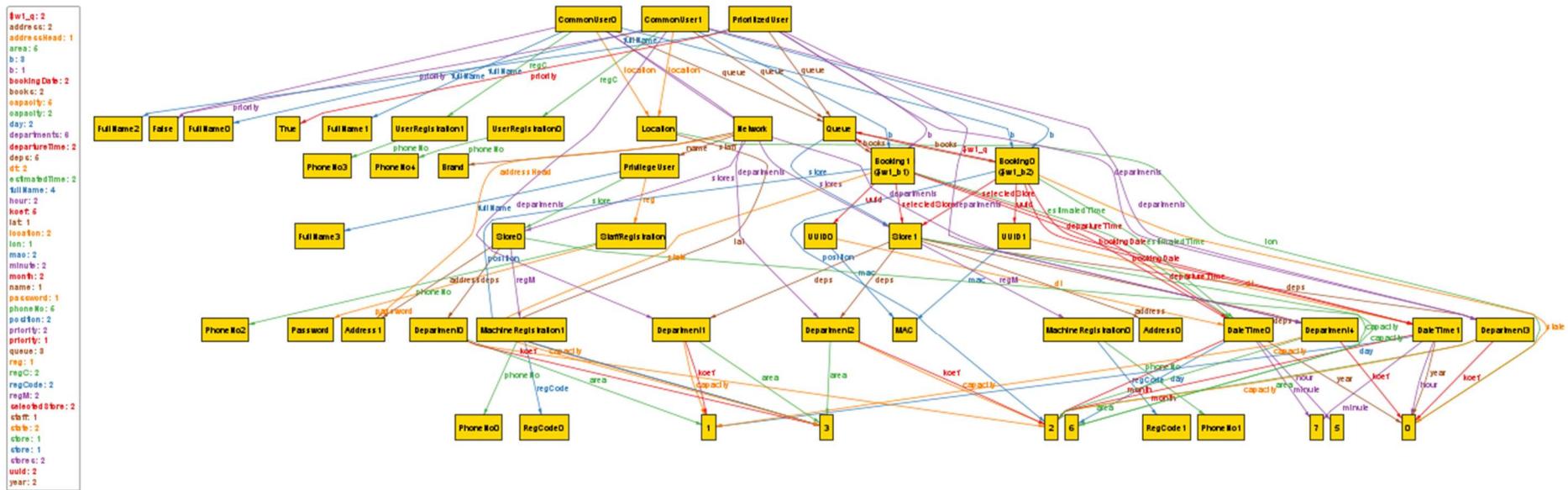


Figure 21 – Alloy: World 1 generated

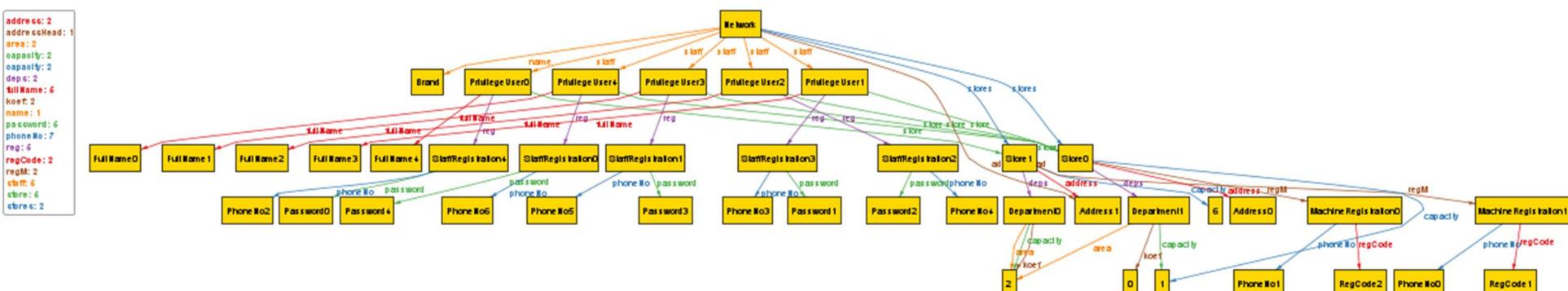


Figure 22 – Alloy: World 2 generated

```

$w3_q: 2
address: 2
addressHead: 1
area: 2
b: 2
b: 1
bookingDate: 2
books: 2
capacity: 2
capacity: 2
day: 1
departments: 3
departureTime: 2
deps: 2
dt: 2
estimatedTime: 2
fullName: 4
hour: 1
koef: 2
lat: 1
location: 2
lon: 1
mac: 2
minute: 1
month: 1
name: 1
password: 1
phoneNo: 5
position: 2
priority: 2
priority: 1
queue: 3
reg: 1
regC: 2
regCode: 2
regM: 2
selectedStore: 2
staff: 1
state: 2
store: 1
stores: 2
uuid: 2
year: 1

```

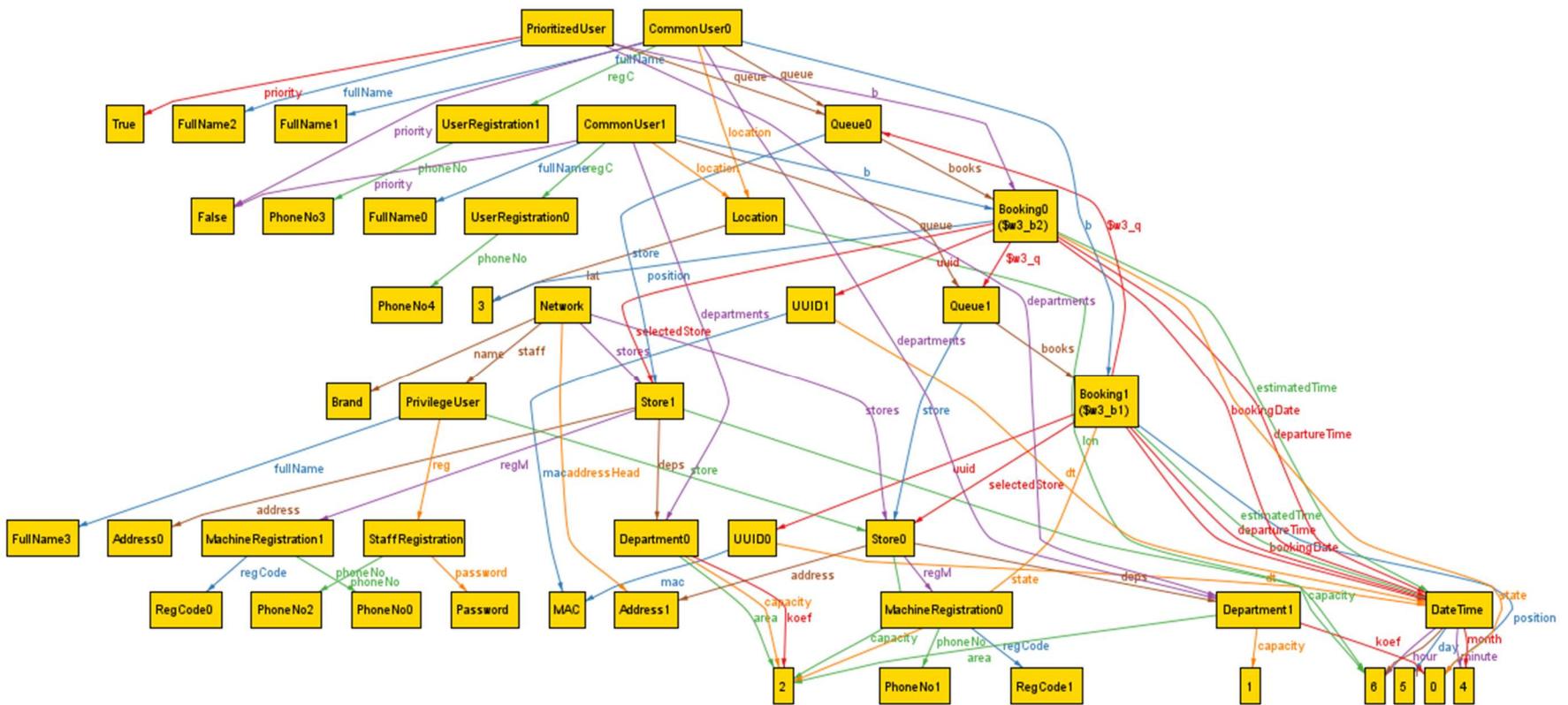


Figure 23 – Alloy: World 3 generated

4.3 Alloy results

The following is the result of the above code:

```
Option Verbosity changed to low
Executing "Run w1 for 5 but 2 MachineRegistration, 1 PrioritizedUser, 1 PrivilegeUser"
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20
22796 vars. 1865 primary vars. 58634 clauses. 63ms.
Instance found. Predicate is consistent. 175ms.
```

```
Executing "Run w2 for 7"
Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
36637 vars. 2989 primary vars. 90816 clauses. 120ms.
Instance found. Predicate is consistent. 311ms.
```

```
Executing "Run w3 for 5 but 2 MachineRegistration, 1 PrioritizedUser, 1 PrivilegeUser"
Solver=sat4j Bitwidth=4 MaxSeq=5 SkolemDepth=1 Symmetry=20
22852 vars. 1865 primary vars. 58797 clauses. 62ms.
Instance found. Predicate is consistent. 156ms.
```

Figure 24 – Alloy – Executing result

5 Effort spent

5.1 Nikita Rozov

Date	Task	Hours
25/10/2020	Introduction, scope and goals	2
29/10/2020	Product perspective	0.5
1/11/2020	Product functions	1
4/11/2020	Domain assumptions	1.5
10/11/2020	External interface requirements	2
13/11/2020	Functional requirements	16
23/11/2020	Non-functional requirements	1.5
6/12/2020	Formal analysis using Alloy	14

5.2 Oleksandr Shchukhlyi

Date	Task	Hours
25/10/2020	Introduction, scope and goals	0.5
29/10/2020	Product perspective	3
1/11/2020	Product functions	0.5
4/11/2020	Domain assumptions	1.5
10/11/2020	External interface requirements	3
13/11/2020	Functional requirements	13
23/11/2020	Non-functional requirements	1
6/12/2020	Formal analysis using Alloy	15

5.3 Sergey Nabatov

Date	Task	Hours
25/10/2020	Introduction, scope and goals	0.5
29/10/2020	Product perspective	1
1/11/2020	Product functions	4
4/11/2020	Domain assumptions	1.5
10/11/2020	External interface requirements	4
13/11/2020	Functional requirements	7
23/11/2020	Non-functional requirements	0.5
6/12/2020	Formal analysis using Alloy	21