

**CLup project Roberto Buratti, Hrvoje  
Hrvoj, Ozan Incesulu**



**POLITECNICO**  
MILANO 1863

# **Requirement Analysis and Specification Document**

---

<b>Deliverable:</b>	RASD
<b>Title:</b>	Requirement Analysis and Verification Document
<b>Authors:</b>	Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu
<b>Version:</b>	1.0
<b>Date:</b>	28-11-2020
<b>Download page:</b>	<a href="https://github.com/Furcanzo/BurattiIncesuluHrvoj">https://github.com/Furcanzo/BurattiIncesuluHrvoj</a>
<b>Copyright:</b>	Copyright © 2020, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

---

# Contents

<b>Table of Contents</b>	<b>3</b>
<b>List of Figures</b>	<b>5</b>
<b>List of Tables</b>	<b>5</b>
<b>1 Introduction</b>	<b>7</b>
1.1 Purpose	7
1.1.1 Description of the Proposed System	7
1.1.2 Goals	7
1.2 Scope	7
1.2.1 Targeted Users	7
1.2.2 Relevant Phenomena	8
1.3 Definitions, Acronyms, Abbreviations	8
1.3.1 Definitions	8
1.3.2 Acronyms	9
1.3.3 Abbreviations	9
1.4 Revision history	9
1.5 Reference Documents	9
1.6 Document Structure	9
<b>2 Overall Description</b>	<b>11</b>
2.1 Product Perspective	11
2.1.1 Book Line Number	12
2.1.2 Get Line Number	12
2.1.3 Print Line Number	13
2.1.4 System Stop	14
2.1.5 Schedule System Stop	14
2.2 Product functions	15
2.3 User characteristics	15
2.4 Assumptions,dependencies and constraints	15
2.4.1 Domain Assumptions	15
2.4.2 Dependencies	16
2.4.3 Constraints	16
<b>3 Specific Requirements</b>	<b>17</b>
3.1 External Interface Requirements	17
3.1.1 User Interfaces	17
3.1.2 Hardware Interfaces	20
3.1.3 Software Interfaces	20
3.1.4 Communication Interfaces	20
3.2 Functional Requirements	20
3.2.1 Customer	20
3.2.2 Clerk	27
3.2.3 Manager	30
3.2.4 Requirements	36
3.2.5 Summary Table	37
3.2.6 Tracability Matrix	38
3.3 Performance Requirements	39

3.4	Design Constraints	39
3.4.1	Standards compliance	39
3.4.2	Hardware limitations	39
3.4.3	Any other constraint	39
3.5	Software System Attributes	39
3.5.1	Reliability	39
3.5.2	Availability	39
3.5.3	Security	39
3.5.4	Maintainability	39
3.5.5	Portability	39
4	Formal Analysis Using Alloy	40
5	Effort Spent	41

## List of Figures

1	Class Diagram	11
2	State Diagram for feature Book Line Number	12
3	State Diagram for feature Get Line Number	13
4	State Diagram for feature Print Line Number	13
5	State Diagram for feature System Stop	14
6	State Diagram for feature Schedule System Stop	15
7	Screenshot: List tickets	17
8	Screenshot: Booking a new ticket	17
9	Screenshot: Edit location screen of manager	18
10	Screenshot: Ticket generate screen of clerk	18
11	Screenshot: Login screen	19
12	Screenshot: Register screen	19
13	Use Case Diagram for Customer	20
14	Sequence Diagram for Use Case: Book future line number	21
15	Sequence Diagram for Use Case: See amount of customers in the store	22
16	Sequence Diagram for Use Case: See store location	23
17	Sequence Diagram for Use Case: Sign up	24
18	Sequence Diagram for Use Case: Login	25
19	Sequence Diagram for Use Case: Notify ticket delete	26
20	Sequence Diagram for Use Case: Update user information	27
21	Use Case Diagram for Clerk	27
22	Sequence Diagram for Use Case: Grant Access	28
23	Sequence Diagram for Use Case: Print guest ticket	29
24	Use Case Diagram for Manager	30
25	Sequence Diagram for Use Case: Initialize	31
26	Sequence Diagram for Use Case: Monitor state	32
27	Sequence Diagram for Use Case: Schedule a stop	33
28	Sequence Diagram for Use Case: Stop for emergency	34
29	Sequence Diagram for Use Case: Add Clerk	35
30	Sequence Diagram for Use Case: Add Manager	36

## List of Tables

1	Relevant Phenomena of CLup	8
2	Use Case: Book future line number	21
3	Use Case: See amount of customers in the store	22
4	Use Case: See store location	23
5	Use Case: Sign up	24
6	Use Case: Login	25
7	Use Case: Notify ticket delete	25
8	Use Case: Update user information	26
9	Use Case: Grant access	28
10	Use Case: Print guest ticket	29
11	Use Case: Initialize	30
12	Use Case: Monitor state	31
13	Use Case: Schedule a stop	32
14	Use Case: Stop for emergency	33
15	Use Case: Add Clerk	34

16	Use Case: Add Manager . . . . .	35
17	Summary Table . . . . .	37
18	Tracability Matrix . . . . .	38

# 1 Introduction

## 1.1 Purpose

### 1.1.1 Description of the Proposed System

The project ĆLup - Customer Line up is a line spot reservation system that is planned to be used by managers, clerks and customers of many local vendors and chains. The system aims to come up with a handy solution for the ongoing issue of proper social distancing management particularly in the matter of grocery shopping, by providing assistance to cope with the customer load for managers and helping customers access to products in a safe and controlled manner.

In particular, users will be able to see the locations, get a line number and book in advance for the grocery stores they would like to visit. Once assigned a line number, the customer will be able to track the estimated time of arrival of the line as well as will wait for the notification that informs about his or her line's forthcoming arrival, hence waiting time in the line in the crowd is minimum. Also, ĆLup provides uniquely generated QR codes per the line number, which can be utilized by the store managers as a proper monitoring tool in the entrances and exits of the locations. All in all, the general purpose of the product is to keep the congestion levels in lines of the locations at minimum via providing useful features for all the users.

### 1.1.2 Goals

- $G_1$  Customers can issue a line number for a location.
- $G_2$  Customers can issue line numbers for their future visits.
- $G_3$  Customers can detail their visits by category and/or product.
- $G_4$  Customers can plan their visit to the store.
- $G_5$  Customers may prefer to use alternative time slots or partner stores for their visit.
- $G_6$  Managers can prevent customers from issuing line numbers.
- $G_7$  Managers can customize the system to allow optimizations for increased granularity, flow control and time slot forecasting.
- $G_8$  Clerks and Managers can monitor the customers through their entrance and exits.
- $G_9$  Customers can obtain printed line number tickets.

## 1.2 Scope

### 1.2.1 Targeted Users

CLup aims to resolve the problem of Customers queueing up in front of a location, without any control over the availability of place in the location and further contact tracing by the managers.

#### Customer:

Customers will be able to obtain specific line numbers for various locations using CLup, which they can track the estimated time available with and also view the location on a map application to plan their visit. Customers can further obtain line numbers for future visits, based on the info provided by the system about the availability of free spots on the specific time intervals. They may prefer to visit a different branch of the same chain. Furthermore, they can provide specific products they intend to purchase or set

an estimated time for their visit to allow finer granularity. Some customers may also prefer to obtain a line number upon visiting the store physically. To plan their visit in a time slot where the location will be less crowded, the users can see the occupancy of the location based on already taken line numbers and forecasts provided by the system.

#### **Clerk:**

The clerk ( which can be a shopping assistant or a security detail) can monitor the flow of customers and manually intervene in case of missing line numbers via performing manual checkout for a specific customer, or by printing line numbers physically for some customers.

#### **Manager:**

The location manager (which could be an actual manager, or someone responsible for handling customer management) can provide details regarding the availability of products and the location in general, by setting the opening hours, maximum allowed customers in shop, in-shop location of different products and categories, maximum amount of reservations that can be made per customer, line number timeout and the location. Also, for chains and relevant stores, the manager can add chain members for the location.

### **1.2.2 Relevant Phenomena**

<b>Phenomenon</b>	<b>World / Machine</b>	<b>Shared</b>
Line Number	Machine	Yes
Line Number Ticket	World	No
Product	World	No
Product Category	World	No
In-store Location	World	Yes
Occupancy Forecast	Machine	No
Store	World	Yes
Time Slot	Machine	Yes
Ticket Printer	World	Yes
Line Number Timer	Machine	No
Customer Scheduling Algorithm	Machine	No

Table 1: Relevant Phenomena of CLup

## **1.3 Definitions, Acronyms, Abbreviations**

### **1.3.1 Definitions**

- *Location*: the physical location of the business that operates the line reservation system
- *Manager*: the user in charge of executive action within the location
- *Customer*: the user with the goal of making a visit to the location
- *Clerk*: the user in charge of handling the entrance and exit of customers
- *Hard-coded Super User*: a single user that is used to create managers of locations administered by the software vendor
- *Visit Time*: the time interval in which a customer performs a visit to the location



- *Line Number*: A number that indicates the ordering of a specific customer in the line
- *Line Number Ticket*: A physical ticket printed that features the line number and the QR code.
- *Time Slot*: Specific intervals of time that are determined by the opening hours and average visit time per customer.
- *Partner Store*: A different location that is included in the same beneficiary chain of command (such as another member of the franchise or store chain) or in a mutual agreement with the specific location
- *Product*: Any item, items, service or services demanded by the customer, and provided by the location to the customer.
- *In-store Location*: A location of a specific product or a product category inside the location.
- *Working hours*: The time intervals that the store is open during each day.
- *Maps API*: A third-party mapping service implementation used for location tracking

### 1.3.2 Acronyms

- **RASD**: Requirement Analysis and Specification Document
- **QR Code**: Quick Response Code
- **API**: Application Programming Interface

### 1.3.3 Abbreviations

- $G_n$ :  $n^{th}$  goal
- $D_n$ :  $n^{th}$  domain assumption
- $R_n$ :  $n^{th}$  functional requirement

## 1.4 Revision history

## 1.5 Reference Documents

- **Specification Document: R&DD Assignment AY 2020-2021**
- **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**
- **ISO/IEC 18004:2015 QR Code bar code symbology specification**

## 1.6 Document Structure

This document is composed of six sections, each with the purpose described below:

- **Introduction**: This section provides an introduction of the problem, the scope of the project with details regarding the goals, target users and phenomena. The goals of the project are formulated in accordance with the description of actions and actors in the Specification Document. Within the project's scope, properties and duties of different users and user groups are described in Target Users section. Furthermore, under the scope of the project, the relevant phenomena of the project is presented through their relevance to the world and the machine.

- **Overall Description:** This section builds upon the introduction section by providing expansion of the scope and relevant functionalities of the system. The basic perspective is provided considering the shared phenomena between the world and the system and the integrations of the system with other parties and products. The overall actions that can be conducted over the system is described using a detailed class diagram, portraying all required components of the system, their inner state and provided functionalities for other components to facilitate their purpose. All core features of the system is listed and explained using user scenarios to provide a better understanding of the real-world condition on which the use case should apply to, a state chart for demonstrating the relationship between various states the system can be in during the execution of the provided core feature scenario. Next, the requirements for the system to archive the aforementioned goals and allow the execution of the provided scenarios. The user characteristics, based on their roles in the system is further evaluated in this section, through the means of the needs that they have, how the system archives those needs via its requirements and goals. No system can be designed without a broad idea on what domain it exists in and how much the environment limits its ability to perform its features. Ergo, we provide the expected assumptions of the domain, the dependencies of the system to external components and the limits imposed by the world on the system.
- **Specific Requirements:** This section is the main section of the document. The external interfaces that the system requires to function correctly is presented with details and additional mockups if needed. The interfaces section is mainly concentrated on different views of user interfaces as the user-facing part is the main way to integrate with the system. In this section, functional requirements of the system are presented in detail, a common use case diagram is provided, and each use-case of the system is analyzed with details on a use case description table and a sequence diagram. The performance requirements of the system, with a focus on The design constraints of the system, with a focus on different standard compliances, hardware limitations of the system to function and other constraints, like GDPR are also elaborated on this section. The vertical aspects that needs consideration are evaluated under the Software System Attributes subsection of this document, with an emphasis on Reliability, Availability, Security, Maintainability and Portability.
- **Formal Analysis Using Alloy:** This section features various models built and hypothesis verified using Alloy. This section demonstrates that some aspects of the requirements document can be formally proven to be correct and provides additional information about the proposed system to the engineers.
- **Effort Spent:** This section features the effort table, in which all team members provide a rough estimation on the time spent on creation of the various sections of the document.
- **References:** This section features different reference materials referred inside this document.

## 2 Overall Description

### 2.1 Product Perspective

The CLup system is expected to be developed ground-up, without any core components being shared with other services, and is expected to integrate with few external services. The system will be using a mapping API to allow its various requirements regarding navigation. The system will be used to control, verify and schedule customer visits to different stores.

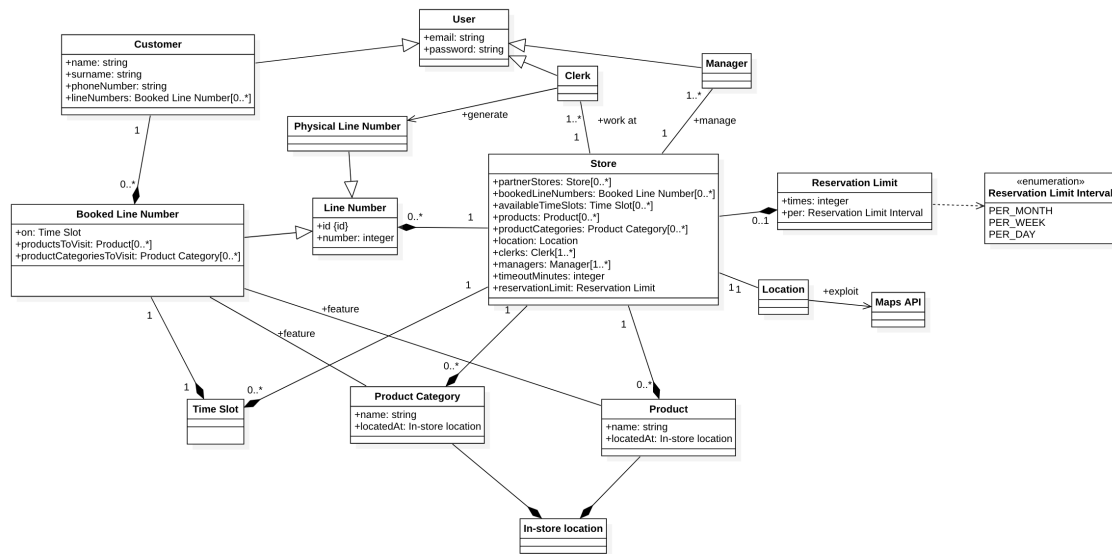


Figure 1: Class Diagram

The **Class Diagram** provided above illustrates the core high-level components of the system and their interactions.

The system is designed to be used by different types of users, the **Customers** whom visit or plan to visit the **Store**, the **Managers** who are responsible for handling management related tasks like providing store details and handling system stops, and the **clerks** whom are on active duty for monitoring the entrance and exit of customers. All users on the system are identified with their *e-mails* and require their *passwords* to authenticate. The **Customers** need to register to the system with their *name*, *surname* and *phoneNumber*, in addition to the information required for authentication. All **Line Numbers** feature a unique *id* and an actual *number* that indicate their order in line. Customers can have many **Booked Line Numbers**, limited by the *reservationLimit*, from various stores, which provide many **Time Slots** that the customers can select from under *availableTimeSlots*, which then becomes the time slot the booking is *on*. The line number is to be invalidated after *timeoutMinutes* from its target time interval. The **Store's Location** is set by the manager and observed by customers through a **Maps API**. The **Reservation Limit** imposes a customer can book a line number at most for some *times* *per* a specific **Reservation Limit Interval** (one of *PER\_MONTH*, *PER\_WEEK* or *PER\_DAY*). Customers' visits can also feature different **Products** or **Product Categories** that they might want to visit under their line number's *productsToVisit* and *productCategoriesToVisit*, each having a *name* and *locatedAt* their own **In-store Location**. **Clerks** can generate **Physical Line Numbers** for customers that don't have the app available for them.

Below some scenarios, state charts and their brief description are enumerated for some core and critical functionalities of the system. It is important to note that the app is targeted for users of all age ranges, thus the scenarios feature different users of different age. The state charts apply to many other scenarios that might occur out of interaction with various types of users.

### 2.1.1 Book Line Number

#### Scenario

Ozan wants to visit the new hamburger place that has recently opened just around the corner of his house. However, Ozan doesn't want to wait on a long line to purchase a burger and some fries. The fast food joint has incorporated the CLup system in their customer service portfolio, making it possible to Ozan to reserve his ticket beforehand without visiting the location. Ozan downloads the CLup application and books his line number from his house. Using the app, he can plan his route and schedule to the location beforehand and place his order directly upon his arrival.

#### State Chart

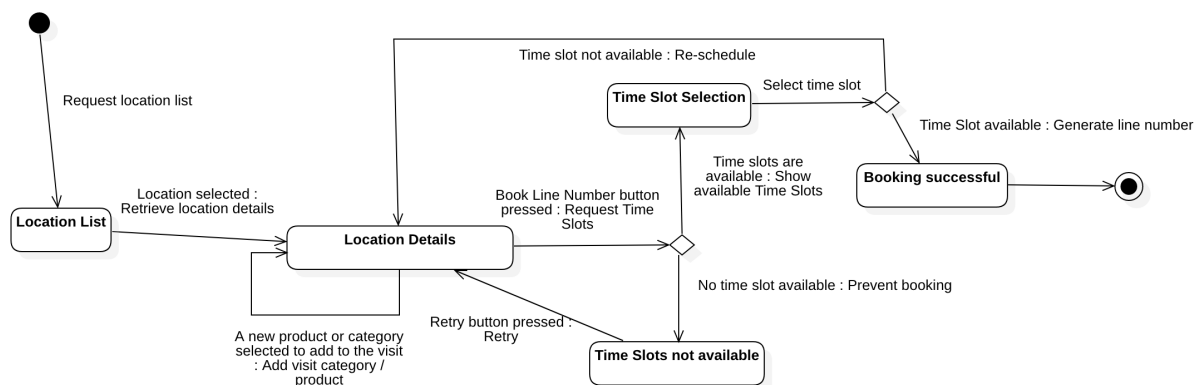


Figure 2: State Diagram for feature Book Line Number

The **State Diagram for feature Book Line Number** represents the execution flow of one of the core use cases, namely booking a line number for the future. The customer books the line number by first listing what locations are available for them to book. Then, when customer selects a location they view the details of the location, where they can add products or product categories to their target visit. The customer can then view the time slots that the system will generate specifically filtered for their visit based on the provided data and availability of the location. When the customer selects a specific time slot the system tries to book that specific location for the customer. If the system successfully allocates the desired time slot, it displays the success message along the customer's ticket. Else the user is notified of the failure and brought back to the scheduling screen for selection of another slot.

### 2.1.2 Get Line Number

#### Scenario

Roberto is living in Milan during the COVID lockdown and wants to visit a grocery store near his house. He wants to obtain a line number for his visit through his phone to not waste time while waiting in the line for other customers to be done with their affairs. He hears that the store he plans to visit has been on CLup, so he downloads the application and opens it. Using the app, Roberto can now see when exactly he should take off to reach the shop without waiting in front of the store.

#### State Chart

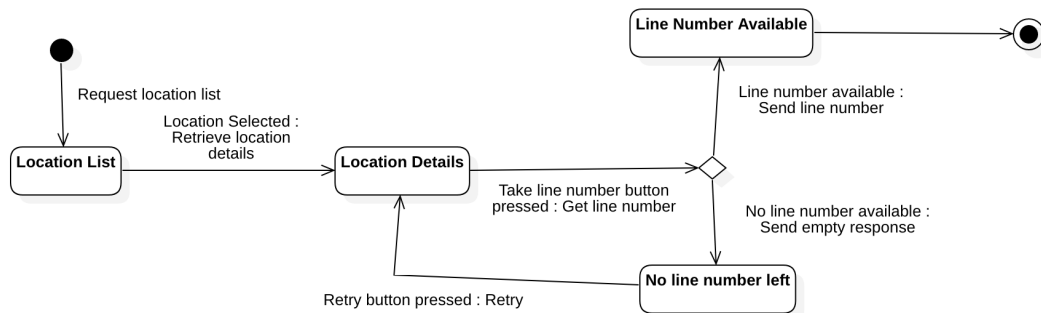


Figure 3: State Diagram for feature Get Line Number

The **State Diagram for feature Get Line Number** represents the execution flow of another core use case, booking a line number for a recent visit. The customer generates a line number by going over the list of available locations, similar to the Book Line Number feature, however this time they press the "Take Line Number" button directly. Then, the system tries to generate a line number for them, and if it is successful, the user retrieves the resulting line number. If the system can not fulfill the user's request, it communicates the problem to the user and allows a retry.

### 2.1.3 Print Line Number

#### Scenario

Hrvoje, a Croatian recently arrived in Italy, is not aware of the popularity of CLup app, however wants to visit an electronics store to purchase a new phone, because his current phone has died out of battery failure. Upon arrival, since he doesn't have a phone, he can not register to CLup system, however the security, that is also in charge of validating line numbers, prints him a ticket that he should keep during his whole visit. Therefore, Hrvoje is allowed in the store, even if he doesn't have the application and the shop manager can monitor his entrance and exit.

#### State Chart

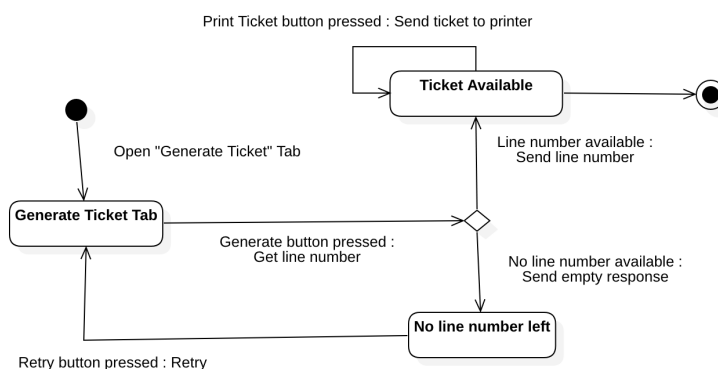


Figure 4: State Diagram for feature Print Line Number

The **State Diagram for feature Print Line Number** represents the execution flow of a similar use

case, printing an actual ticket for on foot visitors. Upon arrival of a new customer that doesn't have a line number, the clerk opens the generate ticket tab and presses the "Generate" button to create a new line number. Then, if all the constraints are satisfied the system allocates a new ticket number for the customer. The clerk can then print the ticket out using a printer. In case there are no line numbers left, the system notifies the clerk as such, with an option to retry the operation.

## 2.1.4 System Stop

### Scenario

Gianfranco is a team lead in a shoe factory. The shoe factory also has an outlet store, where the local brand sells its shoes directly to customers in a discounted price. One day, a fire erupts due to a malfunction in the automated sewing machine and it starts to spread all over the building. Gianfranco coordinates the evacuation of the factory and the store. He also uses the CLup application to issue a system stop to prevent further customers from flooding in. All the customers of the store receive notification regarding this unfortunate event, and don't arrive at the location.

### State Chart

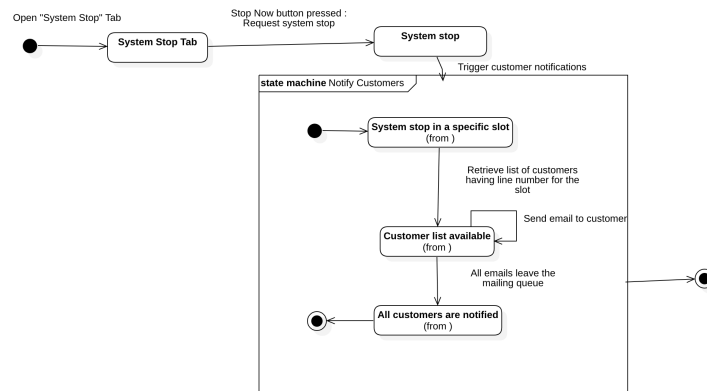


Figure 5: State Diagram for feature System Stop

The **State Diagram for feature System Stop** represents the execution flow for an important use case, the ability of the manager to halt the system, so that all line numbers gets cancelled, all users notified and no further line numbers are issued. The manager first opens the System Stop tab on their phone, and afterwards request a system stop. This action stops the system from issuing more tickets and triggers distribution of notifications for the time zones the system is stopped in. In the notification part of the state diagram, the system starts with a request for stop for a specific time slot, and using that retrieves all the customers that have already scheduled for that slot. The system, then starts sending emails to all customers that are in the list provided. When all emails leave the mailing queue, the system has successfully notified all the customers regarding the stop. The state machine of Notifying Customers is re-used to describe the notification mechanism in the **State Diagram for feature Schedule System Stop**.

## 2.1.5 Schedule System Stop

### Scenario

Giuseppe is an owner of a famous local pizza place. He is using the CLup system in his store to manage the customer lines. He is also preparing a new radio advertisement for his store, with an advertisement agency. They have a meeting next week during work hours and he doesn't have anyone to give the authority to manage the store. Therefore, he has to shut the store down for half the day. He

schedules a system stop from the CLup system to prevent any ticket numbers being issued in that day for his store.

### State Chart

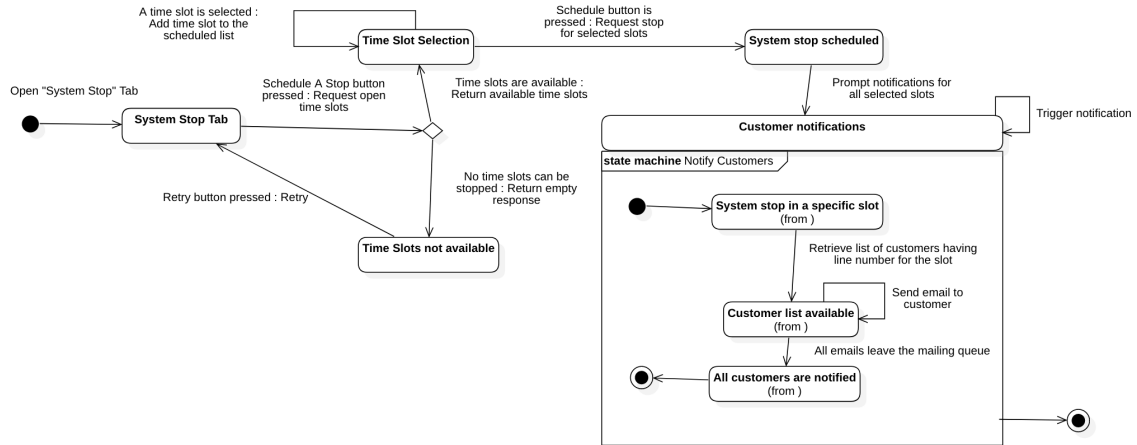


Figure 6: State Diagram for feature Schedule System Stop

The **State Diagram for feature Schedule System Stop** represents the final core use case of the system that is considered as significant, scheduling the system to stop in the future. To start this flow, the manager first opens the System Stop tab and presses the schedule a stop button. The system then pools all the available slots that the manager can issue a stop on, and returns the list to the time slot selection screen. If no such slot is available, a screen indicating such a case is visible, with an opportunity to retry the action. From the time slot selection list, the manager selects specific time slots that the system shall stop for, and press Schedule button to schedule the system stop. A scheduled system stop also sends a notification to all customers that have booked a time slot for that shop before the system stop. The mechanism of customer notification is detailed in the description of the **State Diagram for feature System Stop**, so it will not be repeated here.

## 2.2 Product functions

## 2.3 User characteristics

## 2.4 Assumptions, dependencies and constraints

### 2.4.1 Domain Assumptions

- $D_1$  %80 of the customers and all clerks and managers have basic ICT skills, has an email address that they are willing to use to authenticate to the system and has a smart phone or equivalent device that can connect to the Internet, have a browser that supports UTF-8, display QR codes and has a mapping application.
- $D_2$  Locations will not be visited by no more than 1000 people in any time slot.
- $D_3$  %98 of the customers will arrive at the given location either without a ticket or with a ticket that has not timed out.
- $D_4$  E-mail addresses are not shared by multiple users of the system.
- $D_5$  Clerks' mobile devices are equipped with at least one camera that the system can use.

- $D_6$  All users have a basic understanding of how the line numbering system works and respects the ordering provided by the system.
- $D_7$  Managers have an estimate for the amount of reservations that their location can at most have.
- $D_8$  Managers' device has location services that has a location acquisition error for no more than 20 meters.
- $D_9$  Clerks are constantly monitoring the locations entrances and exits.
- $D_{10}$  Locations have printing equipments that are in 5 meters range of all the entrances that can print QR codes and line numbers.
- $D_{11}$  At least one manager is available in the location during the working hours.
- $D_{12}$  The customer's entry and exit to the store is determined by whether the clerks have checked them in and out.
- $D_{13}$  The customer has their line number or line number ticket available with them through their visit, including their exit from the store

#### **2.4.2 Dependencies**

#### **2.4.3 Constraints**



## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

Below are some sample screenshots from the project's user interfaces with Samsung Galaxy S9 viewport:

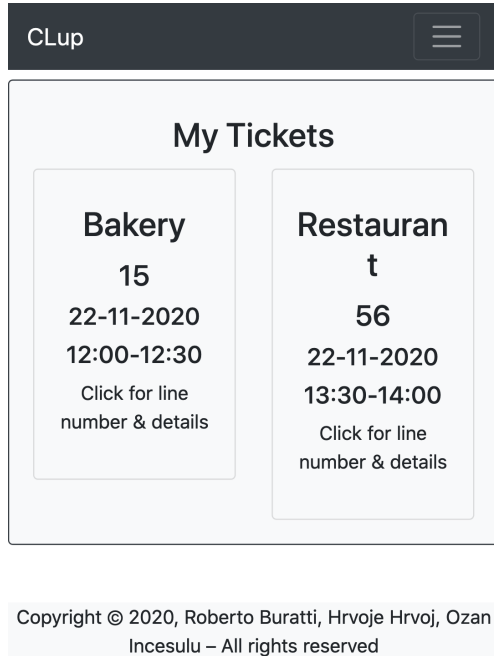


Figure 7: Screenshot: List tickets

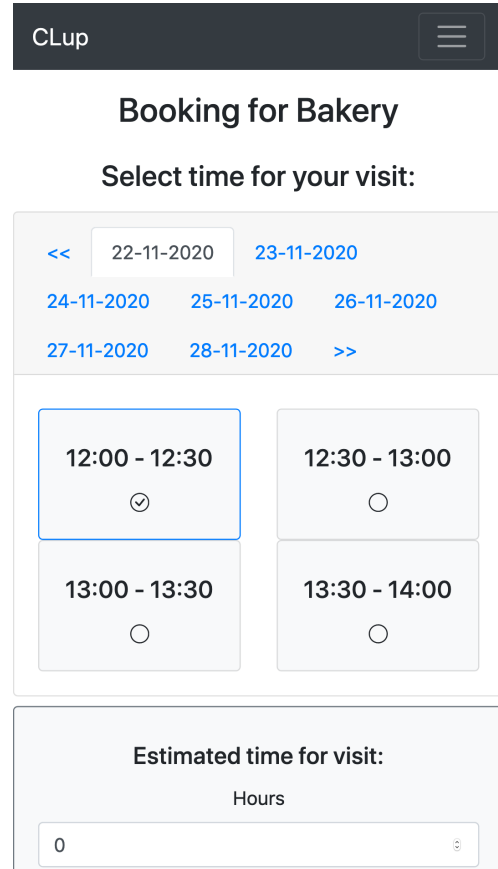


Figure 8: Screenshot: Booking a new ticket

CLup

## Edit Location

Name

Bakery

Customer Ticket Timeout (minutes)

30

### Products

Bread	Update	x
Sweets	Update	x
Sandwiches	Update	x
Coffee & Tea	Update	x

### New Product

Name:

Add

Figure 9: Screenshot: Edit location screen of manager

CLup

## Generated ticket:

15

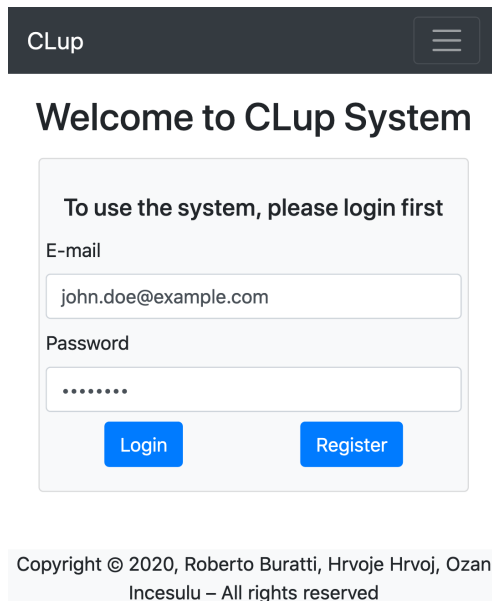
QR Code

Print Ticket

Create New Ticket

Copyright © 2020, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

Figure 10: Screenshot: Ticket generate screen of clerk



The screenshot shows the CLup login interface. At the top is a dark header with the 'CLup' logo and a hamburger menu icon. Below the header, the title 'Welcome to CLup System' is centered. The main content area is a light gray box containing the instruction 'To use the system, please login first'. Below this are two input fields: 'E-mail' with the value 'john.doe@example.com' and 'Password' with masked characters '.....'. At the bottom of the form are two blue buttons: 'Login' and 'Register'. A footer bar at the bottom contains the copyright notice: 'Copyright © 2020, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved'.

CLup

## Welcome to CLup System

To use the system, please login first

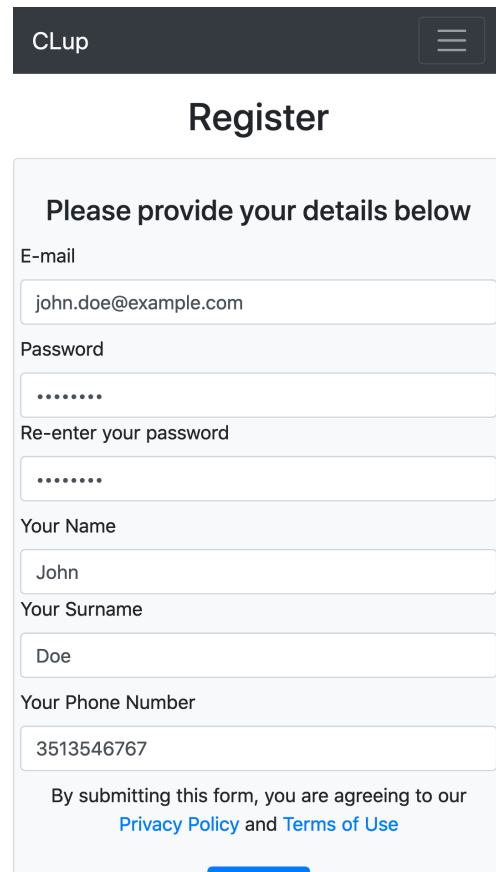
E-mail

Password

Login Register

Copyright © 2020, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

Figure 11: Screenshot: Login screen



The screenshot shows the CLup registration interface. It has the same dark header with 'CLup' logo and hamburger menu icon. The title 'Register' is centered below the header. The main content area is a light gray box with the instruction 'Please provide your details below'. It contains several input fields: 'E-mail' (john.doe@example.com), 'Password' (masked with '.....'), 'Re-enter your password' (masked with '.....'), 'Your Name' (John), 'Your Surname' (Doe), and 'Your Phone Number' (3513546767). At the bottom, there is a line of text: 'By submitting this form, you are agreeing to our [Privacy Policy](#) and [Terms of Use](#)'. A blue button is partially visible at the very bottom.

CLup

## Register

Please provide your details below

E-mail

Password

Re-enter your password

Your Name

Your Surname

Your Phone Number

By submitting this form, you are agreeing to our [Privacy Policy](#) and [Terms of Use](#)

Figure 12: Screenshot: Register screen

### 3.1.2 Hardware Interfaces

### 3.1.3 Software Interfaces

### 3.1.4 Communication Interfaces

## 3.2 Functional Requirements

### 3.2.1 Customer

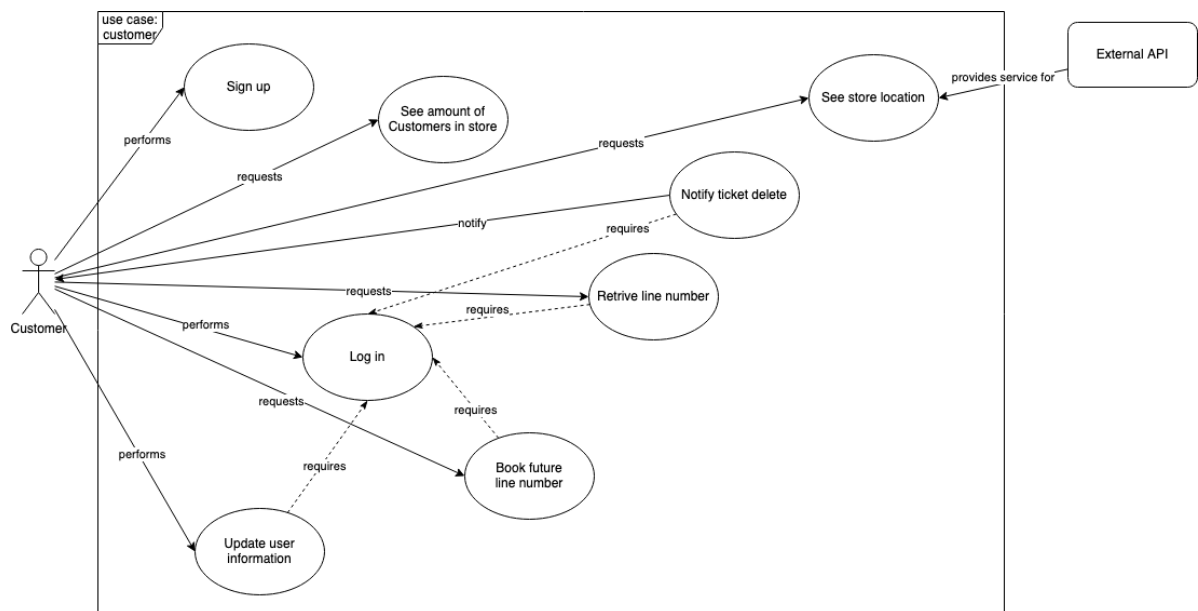


Figure 13: Use Case Diagram for Customer

## Use cases

<i>Name</i>	<b>Book future line number</b>
<i>Actors</i>	Customer
<i>Entry conditions</i>	The customer is logged in the application and wants to book a visit to the store.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>The customer clicks on the "Book a Visit" button in the application.</li> <li>The application asks the time slot and the estimated time of the visit presenting as default value the average of the previous times of visit of the same user.</li> <li>The customer sets the time slot and the estimated time of their visit.</li> <li>The application asks what category of products the customer wants to buy.</li> <li>The customer set the products categories.</li> <li>The application requests the line number from the server.</li> <li>The application generates the QR code on the response from the server.</li> </ul>
<i>Exit conditions</i>	The customer has booked a visit for the store.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>The server cannot retrieve the line number since the time slot is full.</li> </ul>

Table 2: Use Case: Book future line number

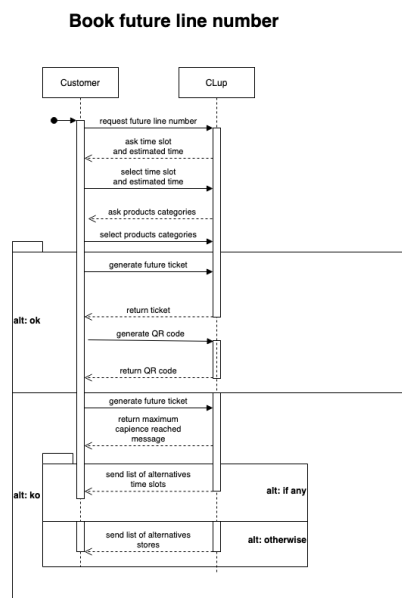


Figure 14: Sequence Diagram for Use Case: Book future line number

<i>Name</i>	<b>See amount of customers in the store</b>
<i>Actors</i>	Customer
<i>Entry conditions</i>	The customer is logged in the application and wants to know how many customers are in the store in order to decide whether to book a visit or retrieve a line number
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The customer clicks on the "Live Store Info" button.</li> <li>• The application sends a count request to the server.</li> <li>• The server returns the live data for the amount of customers in the store for each time slot.</li> </ul>
<i>Exit conditions</i>	The customer knows the amount of customers in the store and can plan their visit.
<i>Exceptions</i>	

Table 3: Use Case: See amount of customers in the store

## See amount of customers in store

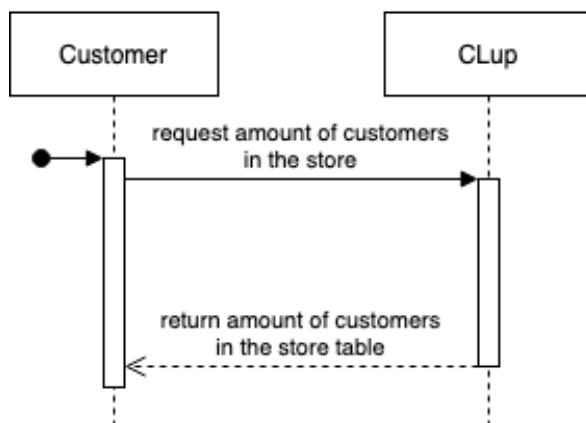


Figure 15: Sequence Diagram for Use Case: See amount of customers in the store

<i>Name</i>	<b>See store location</b>
<i>Actors</i>	Customer, Maps API
<i>Entry conditions</i>	The customer needs to know where the store is located and the travel time
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The customer clicks on the "Store Location" button.</li> <li>• The application contacts the Maps API with the store location.</li> <li>• The Maps API returns a map from the customer position to the store with a time estimation.</li> </ul>
<i>Exit conditions</i>	The customer knows how to go to the store and the travel time needed.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The customer's smartphone doesn't provide or allow access to location services.</li> </ul>

Table 4: Use Case: See store location

## See store location

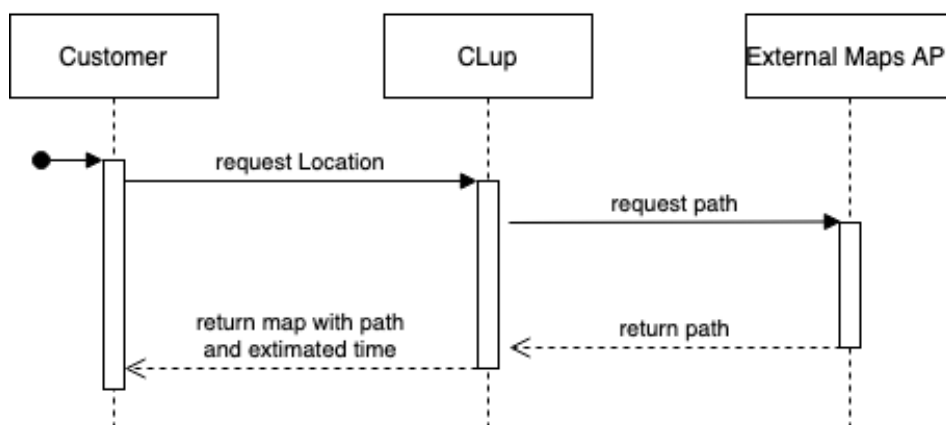


Figure 16: Sequence Diagram for Use Case: See store location

<i>Name</i>	<b>Sign up</b>
<i>Actors</i>	Customer
<i>Entry conditions</i>	The customer opened the application and they are not registered yet, and they want to register.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The customer clicks on the "Sign Up" button.</li> <li>• The customer inserts their credentials.</li> <li>• The application sends the information to the server.</li> <li>• The server stores the data related to the user.</li> <li>• The server returns an acknowledgement to the app.</li> </ul>
<i>Exit conditions</i>	The customer is now registered and can use all the functionalities of the app.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The e-mail provided in the registration form is already used by another user.</li> <li>• The credentials provided in the registration form are ill-formatted.</li> </ul>

Table 5: Use Case: Sign up

### Sign Up

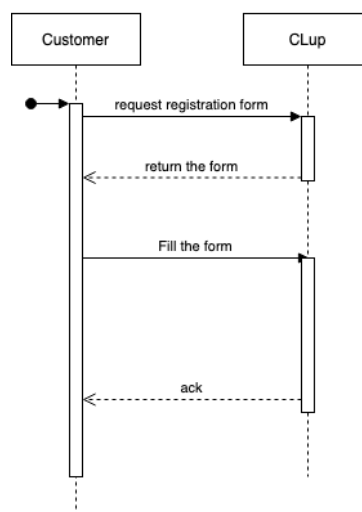


Figure 17: Sequence Diagram for Use Case: Sign up



<i>Name</i>	<b>Login</b>
<i>Actors</i>	User
<i>Entry conditions</i>	The user has already sign up and wants to login to use CLup functionalities
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The customer clicks on the "Login" button</li> <li>• The user provides their login credentials.</li> <li>• The system authenticates the user</li> </ul>
<i>Exit conditions</i>	The user is logged in and can now use all the CLup functionalities
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The credentials are wrong</li> </ul>

Table 6: Use Case: Login

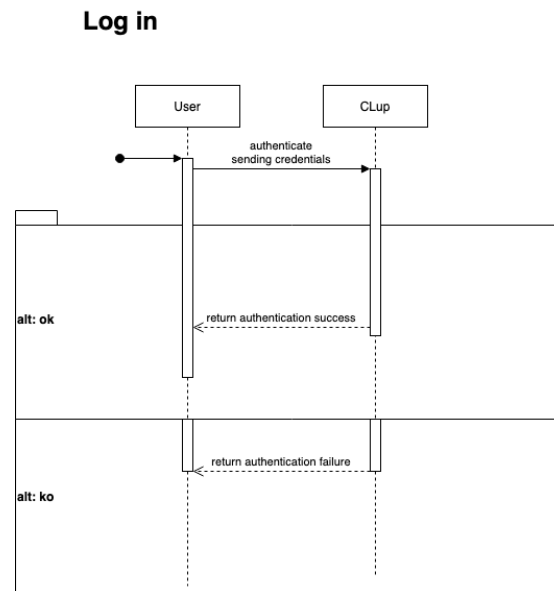


Figure 18: Sequence Diagram for Use Case: Login

<i>Name</i>	<b>Notify ticket delete</b>
<i>Actors</i>	Customer
<i>Entry conditions</i>	A ticket of the Customer was deleted from the server.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The system sends an email to the customer to notify the deletion of their ticket</li> </ul>
<i>Exit conditions</i>	The customer is informed about the deletion of their ticket.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>•</li> </ul>

Table 7: Use Case: Notify ticket delete

## Notify ticket delete

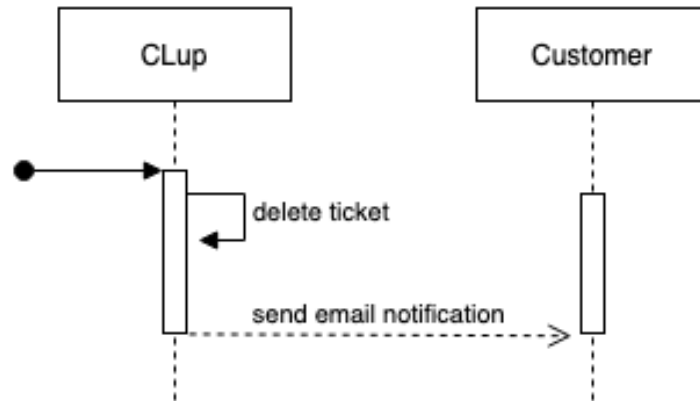


Figure 19: Sequence Diagram for Use Case: Notify ticket delete

<i>Name</i>	<b>Update user information</b>
<i>Actors</i>	User
<i>Entry conditions</i>	The user wants to change an information of themselves.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The user clicks on the "Update User Information" button.</li> <li>• The application shows a form precompiled with user's previous information.</li> <li>• The user changes the information on the respective fields and presses the "Submit" button.</li> <li>• The system saves the changes.</li> </ul>
<i>Exit conditions</i>	The user successfully updated their information.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The new information provided is ill-formatted.</li> </ul>

Table 8: Use Case: Update user information

### Update user information

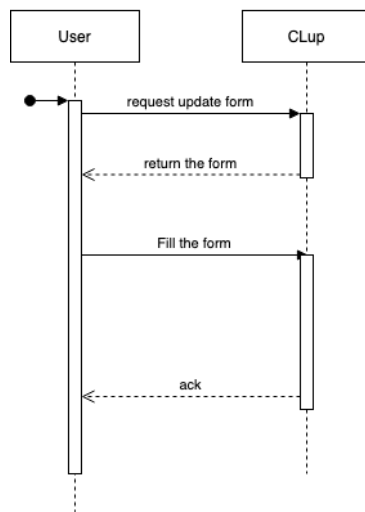


Figure 20: Sequence Diagram for Use Case: Update user information

### 3.2.2 Clerk

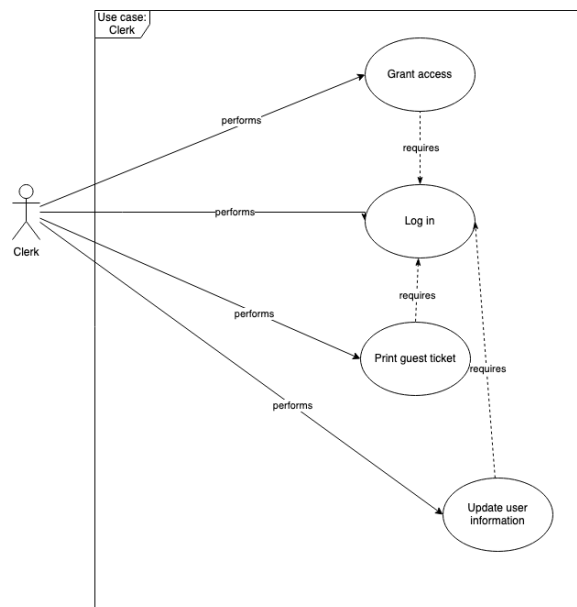


Figure 21: Use Case Diagram for Clerk

### Use cases

<i>Name</i>	<b>Grant access</b>
<i>Actors</i>	Clerk, Customer
<i>Entry conditions</i>	The customer has already obtained the QR code and they plan on entering the store.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The Clerk scans the QR code from the customer's smartphone or printed ticket using the application</li> <li>• The application analyzes the QR code and contacts to the Server</li> <li>• The server decides if the customer can enter based on the information received in the QR code</li> <li>• The server responds to the clerk application with the result</li> <li>• The clerk let the customer enter the store</li> </ul>
<i>Exit conditions</i>	The customer enters the store
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The server communicates to the clerk that the customer cannot enter because their line number is not available.</li> </ul>

Table 9: Use Case: Grant access

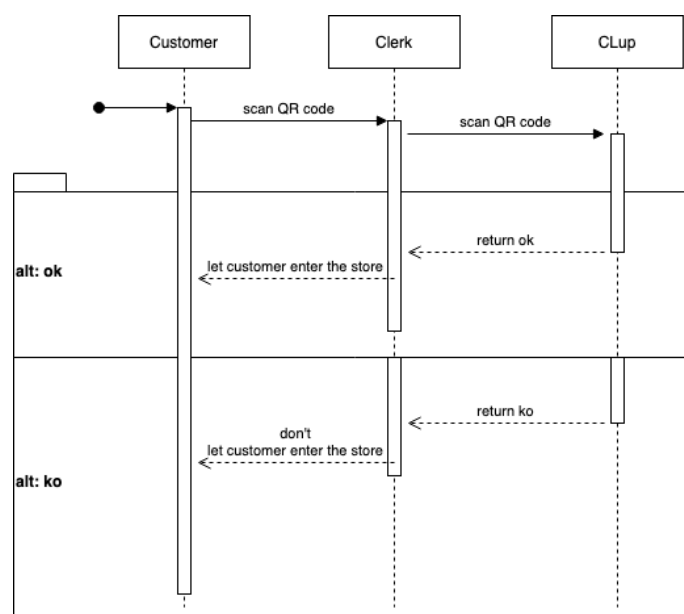


Figure 22: Sequence Diagram for Use Case: Grant Access

<i>Name</i>	<b>Print guest ticket</b>
<i>Actors</i>	Clerk, Customer
<i>Entry conditions</i>	The customer has arrived to the location, doesn't have a ticket on their smartphone, and needs a physical ticket to enter.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The customer asks the clerk for a ticket.</li> <li>• The clerk generate a ticket using the app.</li> <li>• The application sends a request to the server to generate a ticket.</li> <li>• The server generates a line number and a ticket.</li> <li>• The server sends the ticket back to the app.</li> <li>• The clerk prints the ticket.</li> <li>• The clerk gives the ticket to the customer.</li> </ul>
<i>Exit conditions</i>	The customer has a ticket.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The server cannot generate a line number due to capacity constraints.</li> </ul>

Table 10: Use Case: Print guest ticket

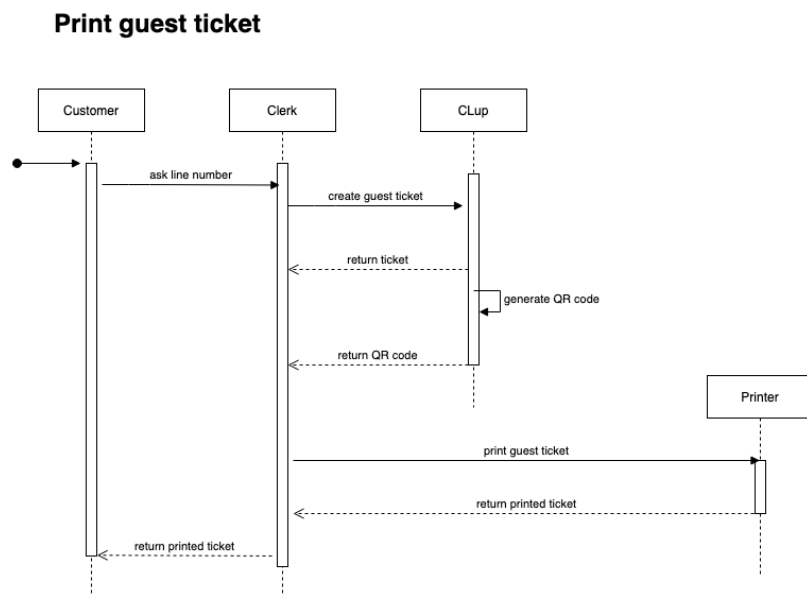


Figure 23: Sequence Diagram for Use Case: Print guest ticket

### 3.2.3 Manager

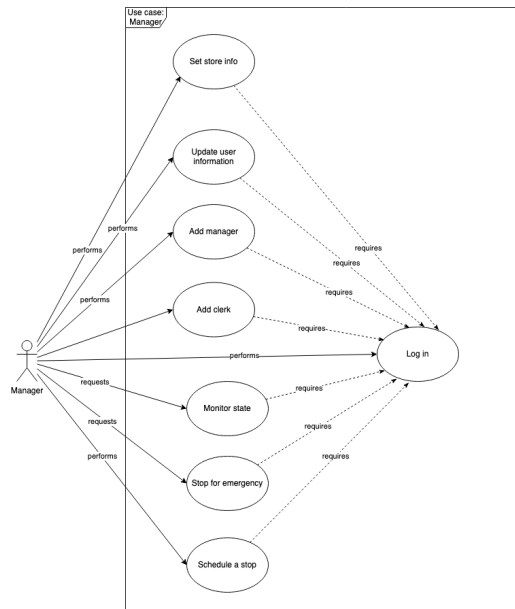


Figure 24: Use Case Diagram for Manager

### Use cases

<i>Name</i>	<b>Initialize</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	The manager of the store needs to set the basic information of the store in order to start the service
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager clicks on the initialize button</li> <li>• The application shows the location form to the manager</li> <li>• The manager fills the form and sends it to the server via the app</li> <li>• The server registers the information and acknowledges</li> </ul>
<i>Exit conditions</i>	The system is initialized and can offer all its functions
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• Some mandatory parts of the form are not filled.</li> </ul>

Table 11: Use Case: Initialize

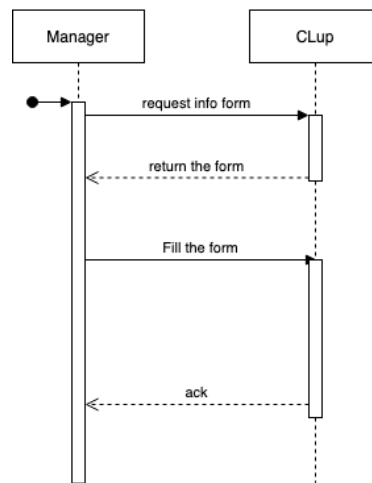
**Set store info**

Figure 25: Sequence Diagram for Use Case: Initialize

<i>Name</i>	<b>Monitor state</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	The manager wants to monitor the number of customers in the store in real time.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager clicks on the "Monitoring" button.</li> <li>• The application sends the number request to the server.</li> <li>• The server returns the number of customers in the store.</li> <li>• The application repeats the process periodically as long as the monitoring page is open.</li> </ul>
<i>Exit conditions</i>	The manager is informed on the number of the customers in the store in real time.
<i>Exceptions</i>	•

Table 12: Use Case: Monitor state

## Monitor state

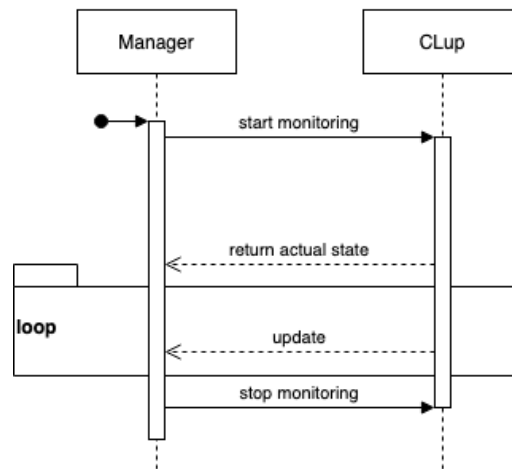


Figure 26: Sequence Diagram for Use Case: Monitor state

<i>Name</i>	<b>Schedule a stop</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	The manager wants to schedule a period of time in which the store will be closed so the users can not book a visit for that time.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager clicks on the "Schedule a Stop" button.</li> <li>• The application shows a form for the time of the stop.</li> <li>• The manager fills the form and submits it to the server through the app.</li> <li>• The server stores the information in the database and returns an acknowledgement.</li> </ul>
<i>Exit conditions</i>	The system has a scheduled stop stored in its database and will use it to prevent customers from booking a visit in that time period.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• There is already a planned stop in that period.</li> </ul>

Table 13: Use Case: Schedule a stop



## Schedule a stop

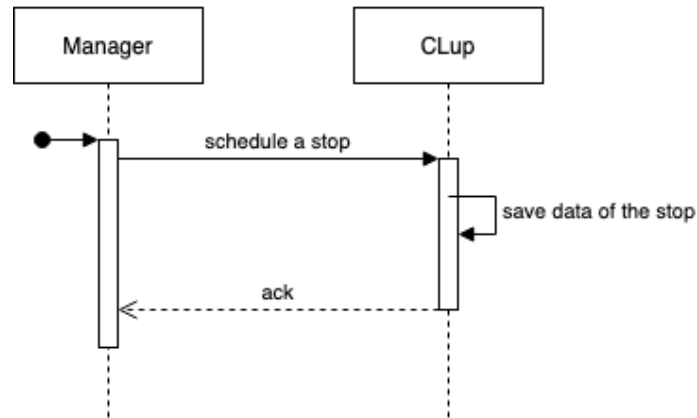


Figure 27: Sequence Diagram for Use Case: Schedule a stop

<i>Name</i>	<b>Stop for emergency</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	An emergency occurred and the manager wants to immediately stop the system from distributing line numbers.
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager click on the "Emergency Stop" button</li> <li>• The application asks for confirmation from the manager.</li> <li>• The manager confirms the stop of the system.</li> <li>• The application sends the system stop request to the server.</li> <li>• The server stops the service and return an acknowledgement.</li> </ul>
<i>Exit conditions</i>	The system has interrupted the service.
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The manager aborts the operation.</li> </ul>

Table 14: Use Case: Stop for emergency

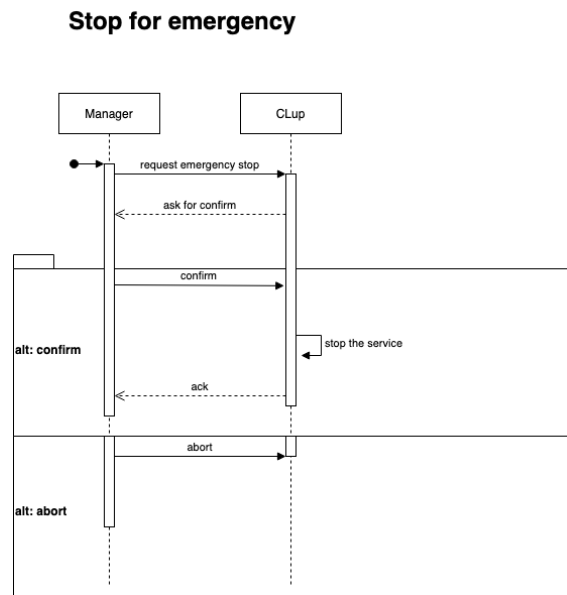


Figure 28: Sequence Diagram for Use Case: Stop for emergency

<i>Name</i>	<b>Add Clerk</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	The manager wants to add a new clerk to the system
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager click on the "add clerk" button</li> <li>• The system asks the data of the new clerk (e.g. the credentials)</li> <li>• The manager inserts the data and press "submit"</li> </ul>
<i>Exit conditions</i>	The new clerk is added to the system
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The data of the clerk are incomplete or incorrect</li> </ul>

Table 15: Use Case: Add Clerk

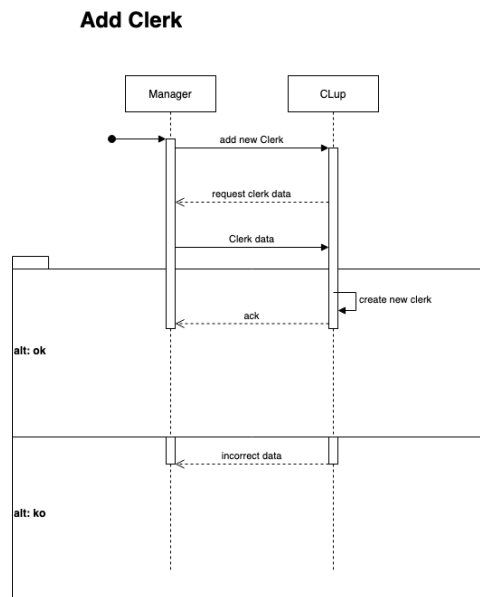


Figure 29: Sequence Diagram for Use Case: Add Clerk

<i>Name</i>	<b>Add Manager</b>
<i>Actors</i>	Manager
<i>Entry conditions</i>	The manager wants to add a new manager to the system
<i>Event flows</i>	<ul style="list-style-type: none"> <li>• The manager click on the "Add Manager" button</li> <li>• The system asks the data of the new manager (e.g. the credentials)</li> <li>• The manager inserts the data and press "Submit"</li> </ul>
<i>Exit conditions</i>	The new Manager is added to the system
<i>Exceptions</i>	<ul style="list-style-type: none"> <li>• The data of the Manager are incomplete or incorrect</li> </ul>

Table 16: Use Case: Add Manager

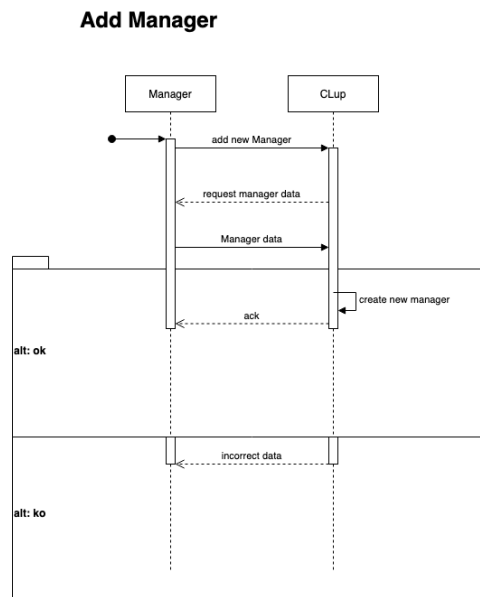


Figure 30: Sequence Diagram for Use Case: Add Manager

### 3.2.4 Requirements

- $R_1$  The system must allow users to authenticate using their e-mail address and password.
- $R_2$  The system must allow customers to register using their e-mail address, their name, surname, phone number and a new password.
- $R_3$  The system must provide a hard-coded super user to allow addition of locations and managers of locations.
- $R_4$  Managers must be able to add additional managers and clerks as users.
- $R_5$  Managers must be able to set and update location specific information, that are maximum number of customers in the location at any given time, opening and closing hours of the store per each day, line number timeout, limit of reservation per customer on a predetermined time interval that is one of month, week or day, and location of the place
- $R_6$  Managers can add any other location as a partner store.
- $R_7$  Managers can stop the system from issuing any more tickets for a given day
- $R_8$  Managers can schedule the system stop for a future time.
- $R_9$  Managers can set the in-shop locations for different categories and product items.
- $R_{10}$  In case of a system stop, no further line numbers can be issued for the given time slots.
- $R_{11}$  In case of a system stop, all line numbers in the stop time slots has to be cancelled.
- $R_{12}$  The system must cancel those line numbers that the customer didn't arrive to the location for more than the set timeout interval.
- $R_{13}$  In case of a ticket cancel, customer must be notified with an e-mail notification.
- $R_{14}$  Clerks must register the entrance and exit of customers via scanning the QR code for their line number.

- $R_{15}$  Clerks must be able to generate line number tickets in a printer compatible format.
- $R_{16}$  Customers must be able to obtain a line number, except when the system is stopped or the store is full.
- $R_{17}$  Customers must be able to obtain line numbers for different time slots in the future.
- $R_{18}$  Customers can not obtain line numbers that exceed the quantity per time interval limits.
- $R_{19}$  Customers can not obtain line numbers for time intervals that the system is stopped by a manager.
- $R_{20}$  Customers must be able to see the estimated time available for their line number.
- $R_{21}$  Customers must be able to set or update their phone number, password, name and surname.
- $R_{22}$  Customers can select specific product and/or product categories they plan to visit in the location while obtaining a line number.
- $R_{23}$  Customers can set an estimated time for their visit while obtaining a line number.
- $R_{24}$  Customers must be able to view the shop location
- $R_{25}$  Customers can view the occupation forecasts for the location at different time slots.
- $R_{26}$  Customers can see the alternative suggestions for time slots while obtaining a line number for the future.
- $R_{27}$  Customers can view the occupancy for the partner stores, if preferred time slot is not available while obtaining a line number.
- $R_{28}$  Customers can view their line numbers with the number and the QR code.
- $R_{29}$  The system must be able to provide a forecast for the occupancy of each location for any given time based on past visits.

### 3.2.5 Summary Table

Goal	Requirements	Domain Assumptions
$G_1$	$R_1, R_2, R_{12}, R_{16}, R_{18}, R_{19}$	$D_1, D_{13}$
$G_2$	$R_1, R_2, R_{12}, R_{17}, R_{18}, R_{19}, R_{27}$	$D_1$
$G_3$	$R_1, R_2, R_9, R_{22}, R_{23}$	$D_1$
$G_4$	$R_1, R_2, R_{13}, R_{20}, R_{24}, R_{25}, R_{28}, R_{29}$	$D_1, D_3, D_6, D_8$
$G_5$	$R_1, R_2, R_6, R_{25}, R_{26}, R_{27}$	$D_1$
$G_6$	$R_1, R_3, R_4, R_7, R_8, R_{10}, R_{11}, R_{13}, R_{19}$	$D_1, D_{11}$
$G_7$	$R_1, R_3, R_4, R_5, R_{12}, R_{18}, R_{29}$	$D_1, D_7$
$G_8$	$R_1, R_3, R_4, R_{14}, R_{21}, R_{28}$	$D_1, D_3, D_4, D_5, D_6, D_9, D_{11}, D_{12}, D_{13}$
$G_9$	$R_4, R_{15}$	$D_2, D_6, D_{10}, D_{13}$

Table 17: Summary Table

### 3.2.6 Tracability Matrix

Requirement	Use case
$R_1$	Login
$R_2$	Sign Up
$R_3$	
$R_4$	Add Clerk Add Manager
$R_5$	Initialize
$R_6$	Initialize
$R_7$	Stop for Emergency
$R_8$	Schedule a Stop
$R_9$	Initialize
$R_{10}$	Schedule a Stop Stop for Emergency
$R_{11}$	Schedule a Stop Stop for Emergency
$R_{12}$	
$R_{13}$	Notify Ticket Delete
$R_{14}$	Grant Access
$R_{15}$	Print Guest Ticket
$R_{16}$	
$R_{17}$	Book Future Line Number
$R_{18}$	Book Future Line Number Initialize
$R_{19}$	Book Future Line Number Schedule a Stop Stop for Emergency
$R_{20}$	
$R_{21}$	Update User Information
$R_{22}$	Book Future Line Number
$R_{23}$	Book Future Line Number
$R_{24}$	See Store Location
$R_{25}$	See Amount of Customers in the Store
$R_{26}$	Book Future Line Number
$R_{27}$	Book Future Line Number
$R_{28}$	
$R_{29}$	See Number of Customers in the Store

Table 18: Tracability Matrix

### **3.3 Performance Requirements**

### **3.4 Design Constraints**

#### **3.4.1 Standards compliance**

#### **3.4.2 Hardware limitations**

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

### **3.5 Software System Attributes**

#### **3.5.1 Reliability**

#### **3.5.2 Availability**

#### **3.5.3 Security**

#### **3.5.4 Maintainability**

#### **3.5.5 Portability**

## **4 Formal Analysis Using Alloy**



## 5 Effort Spent

Date:	Person:	Part:	Time (in hours):	Description:
18/10/2020	Ozan Incesulu	General Structure	0.75	Imported and built the general document structure, switched LF -> CRLF for Windows, replaced template parts with names, year and project title
18/10/2020	Ozan Incesulu	Introduction	1.5	Start writing introduction by adding comments for draft goals, adding further subsections, writing basic definition of the system, some abbreviations, definitions, acronyms and references
24/10/2020	Ozan Incesulu	Document Structure	1.5	Write the document structure of the introduction, provide some comments and assumptions regarding how other parts of the document shall be structured.
26/10/2020	Ozan Incesulu	Scope	1.25	Write the Scope section, by defining different users of the system and define different phenomena with the categories they belong. Also add additional definitions.
26/10/2020	Aydin Javadov	Scope	1	Extend the goal test to provide details about system function
29/10/2020	Ozan Incesulu	Introduction	0.25	Add the clerk role to the system instead of using door automation
31/10/2020	Ozan Incesulu	Domain Assumptions	1	Write domain assumptions for the system, considering different users and phenomena
31/10/2020	Ozan Incesulu	Requirements	1.5	Write system requirements for the system based on agreed goals of the system.
06/11/2020	Ozan Incesulu	Goals	1	Write system goals, merge changes from previous team member, some extra housekeeping tasks.
14/11/2020	Ozan Incesulu	Use-case	1.5	Export diagrams to images, try to align everything, fix grammar and text
15/11/2020	Ozan Incesulu	Product Perspective	3	Create state diagrams, write scenarios and explanation
22/11/2020	Ozan Incesulu	Product Perspective	2	Create class diagram for the system.
23/11/2020	Ozan Incesulu	Product Perspective	1.5	Add description section, add line number difference to class diagram
28/11/2020	Ozan Incesulu	Functional Requirements	1.5	Add traceability matrix and summary table