



POLITECNICO
MILANO 1863

- RASD -

Requirements Analysis and
Specification Document

COMPUTER SCIENCE AND ENGINEERING
SOFTWARE ENGINEERING II

A.A. 2020/2021

DANIELE MAMMONE - 10625264

GIANMARCO NARO - 10610374

MASSIMO PARISI - 10583470

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.2.1	World Phenomena	4
1.2.2	Shared Phenomena	4
1.2.3	Goals	5
1.3	Definitions, Acronyms, Abbreviations	5
1.3.1	Definitions	5
1.3.2	Acronyms	6
1.3.3	Abbreviations	6
1.4	Revision History	6
1.5	Reference Documents	6
1.6	Document Structure	7
2	Overall Description	8
2.1	Product Perspective	8
2.1.1	UML Description	8
2.1.2	State Charts	8
2.2	Product Functions	10
2.2.1	Book a visit	10
2.2.2	Cancel a reservation	11
2.2.3	Check-in/Check-out	11
2.2.4	Store Capacity	11
2.3	User Characteristics	11
2.4	Assumptions, Dependencies, Constraints	12
2.4.1	Domain Assumptions	12
3	Specific Requirements	12
3.1	External Interface Requirements	12
3.1.1	User Interfaces	12
3.1.2	Hardware Interfaces	12
3.1.3	Software Interfaces	12
3.1.4	Communication Interfaces	12
3.2	Functional Requirements	12
3.2.1	List of Requirements	12
3.2.2	Mapping	12
3.2.3	Use Cases	12
3.2.3.1	Registration of a customer	13
3.2.3.2	Registration of a store	13
3.2.3.3	Login of a customer	14
3.2.3.4	Login of a store manager	15
3.2.3.5	Customer makes a reservation	15
3.2.3.6	Customer visualizes reservations	17

3.2.3.7	Manager modifies store parameters	17
3.2.3.8	The store manager monitors the store situation .	18
3.2.3.9	The store manager manages customers bookings	19
3.2.3.10	Customers reservations management	21
3.2.4	Sequence Diagram	21
3.2.5	Scenarios	21
3.3	Performance Requirements	22
3.4	Design Costraints	22
3.4.1	Standards Compliance	22
3.4.2	Hardware Limitations	22
3.4.3	Any Other Constraint	22
3.5	Software System Attributes	22
3.5.1	Reliability	22
3.5.2	Availability	22
3.5.3	Security	22
3.5.4	Maintainability	22
3.5.5	Portability	22
3.6	Additional Specifications	22
4	Formal Analysis Using Alloy	22
4.1	Alloy	22
5	Effort Spent	22
6	References	22

1 Introduction

1.1 Purpose

The main target of this document is to describe the software through functional and non-functional requirement and is used as contractual basis between the customer and the developer. The structure of the document follows the one studied during lectures and aim to describe faithfully the software behaviour in all of its aspects.

The software in question is *CLup*, a mobile service usable through app, made both for store managers and customers. It facilitates customers to book a visit to a store and, on the other hand, to help store managers to observe the new strict rules due to *Covid-19*.

1.2 Scope

The main purpose of *CLup* is to facilitate customers to access at a store in **security**, both allowing them to reserve a spot on the queue for entering the store through the app and to book a visit at the store in a determined time window, selected by the user. Thanks to this, store managers can manage the **affluency** in their store more easilier, and moreover can reduce the crowd in front of the store, that is the main purpose of the application. The main idea is that when a person's number is called, he can enter the supermarket. Moreover, the app should generate a *QR Code* that the customer will scan at the entry and at the exit of the store, so that the system knows in real time how many people there are in the store. *CLup* should also estimates the *ETA* from the turn of a person. Moreover, it is able to suggest people other store options when there is a high waiting time to enter to the store requested by the user. The app also allows people to reserve their spot at the supermarket and, in order to optimize the waiting time, people can select specifics reparts where they want to go in the store.

Summing up, *CLup* has this main functionalities:

- **Manage of lining up of the store:** the app will manage the accesses to the store, based on numbered tickets released to people. When it's the turn of a person, it will be authorized to enter the shop. Futhermore, the store manager is able to manage the access and the affluency to the store.
- **Booking visit:** users can book a visit at the store, in a certain time frame decided in the booking process. For them, there is no requirement of ticket, since are able to access the store only scanning the *QR Code* at the store entrance. The system will grant access if the time of entering is correct.
- **Alternatives:** the app is able to suggest other stores options and time if some store is full, or comfortable times aren't available at the moment of the booking.

1.2.1 World Phenomena

WP1	A user enters a supermarket
WP2	A user waits in a lineup
WP3	A user exits the supermarket
WP4	A certain number of people is inside the supermarket
WP5	A certain number of people is at a specific repart of the supermarket

1.2.2 Shared Phenomena

SP1	The user gets a ticket/QR	M
SP2	The user books a visit to the store	M
SP3	The store generates in presence a ticket	M
SP4	The user comes to knows how much time they have to wait before entering	U
SP5	The users know how many people there are in a store at a certain moment	U
SP6	The user scan the QR code and enters in the supermarket	U
SP7	The user scan the QR code and exits from the supermarket	U
SP8	The user can indicate the categories of items that he intend to buy	U

1.2.3 Goals

G1	Allow customers to select a store and book a spot on the queue from <i>CLup</i> app
G2	Allow customers to book a spot on the queue from a physical ticket dispenser
G3	Allow store managers to regulate the influx of people in the building
G4	Allow people to select a better store option to avoid waisting a lot of time waiting to entry at a specific store.
G5	Allow users to depart from their location in time to avoid waiting too much, and to avoid losing their turn in lineup
G6	Allow users to decrease waiting times specifing reparts they want to visit

1.3 Definitions, Acronyms, Abbreviations**1.3.1 Definitions**

QR Code	Bidimensional bar code that allows the user to check-in/check-out
Customer	The clients of the store, that uses the application part reserved to bookings
Store manager	The user that access to stores' bookings and occupancy, in order to manage the flow of customers
QR Code Reader	Device used to scan customers' QR Code
Totem	Electronic device that allows customers to book a store visit, allowing them to specify the same parameters that can be inserted through the app
QR Code Printer	Device used to print QR Code at the stores
Repart	Part of the store that contains the same category of products

1.3.2 Acronyms

RASD	Requirement Analysis and Specification Document
ETA	Estimated Time of Arrival
GPS	Global Positioning System

1.3.3 Abbreviations

WP _n	World phenomena number n
SP _n	Shared phenomena number n
G _n	Goal number n
R _n	Requirement number n

1.4 Revision History

Version	Date	Changelog
1.0	10/11/2020	Overview of the specifications and initial draft of the document

1.5 Reference Documents

- Specification Document
- Slides of the lectures

1.6 Document Structure

The structure of the document is thought with the intention of allowing simple navigation throught it. Also, various abbreviations, highlighted in Abbreviations section, have been used to make the content smoother. Hence, the structure of the document is the following one:

- **Introduction:** The main purpose of this section is to introduce in a general way the scope of the application throught the analysis of the *World Phenomena*, *Shared Phenomena* and *Goals*. Moreover, the main functions of the software are illustrated and the abbreviations, acronyms and definitions are defined in order to allow an easy reading.
- **Overall Description:** The section starts with a summary description of the UML of the software, so as to have a general presentation of the operation of the application. Then, in order to clarify the behaviour of the application, there are state charts of the most important functions and the detailed description of all software functions. Subsequently, the section ends with the most important phenomena that cannot be managed by the system.
- **Specific Requirements:** The main focus of this section is to describe the essential hardware and software interfaces and requirements, indeed *CLup* uses this interfaces to provide its services to the external world. After this, there is the core of the section represented by use cases that provides detailed information about the relation with requirements.
- **Formal Analysis Using Alloy:** This section describes formally the model using Alloy language, highlighting the main problems of the software, solving them in a formal way.
- **Effort Spent:** The main focus of this section is to track the time spent to complete this project. In particular, is highlighted the subdivision of the working hours of the various sections
- **References:** This section is dedicated to all references used in this project.

2 Overall Description

2.1 Product Perspective

In Figure ??? is reported an UML Class Diagram that represents the domain of the application with main concepts and data involved, including their relationships.

The store managers registers to the application providing all necessary information and can decide at a later stage to modify the capability options (regarding each department of the store). The customer simply downloads the application on his device to be able to use it. Here we can identify the main aspects related to CLup:

- The customer can generate a reservation, choosing between the registered chain store (and one of their specific store) or a normal store, a time slot and, optionally, the departments that they want to access; CLup will retrieve a ticket containing the number of the reservation and a QR code.
- The customer can entry in the store where he has a reservation (on the right time slot) scanning the QR code with a totem/the help of a store manager.
- The customer exit the store reusing the QR code, notifying the application that a new spot is now free (on certain department).

The UML does not include every class of the actual implementation of the system.

2.1.1 UML Description

Inserire UML qui

2.1.2 State Charts

Now we are going to examine some essential aspects of the application, modelling their behaviours and showing the evolution over time of their states through adequate state diagrams, which are reported below.

When the store manager opens the store, the system initialize itself and waits for some ticket that can be called. For example, tickets related to the *FIFO* queue can be called as soon as the requested zones are available, and tickets related to bookings can be called only after the start time of the booked time slot, and will be called around this time, when all the booked zones become available (to avoid starvation on calling bookings, other tickets requesting at least one booked zone won't be called until the booked tickets enters the supermarket). After a ticket is available to be called, the system notify in some way (e.g. through a store employee) that now certain ticket is allowed to enter the store. At this point, the system waits for the scan of the associated *QR code*. If the

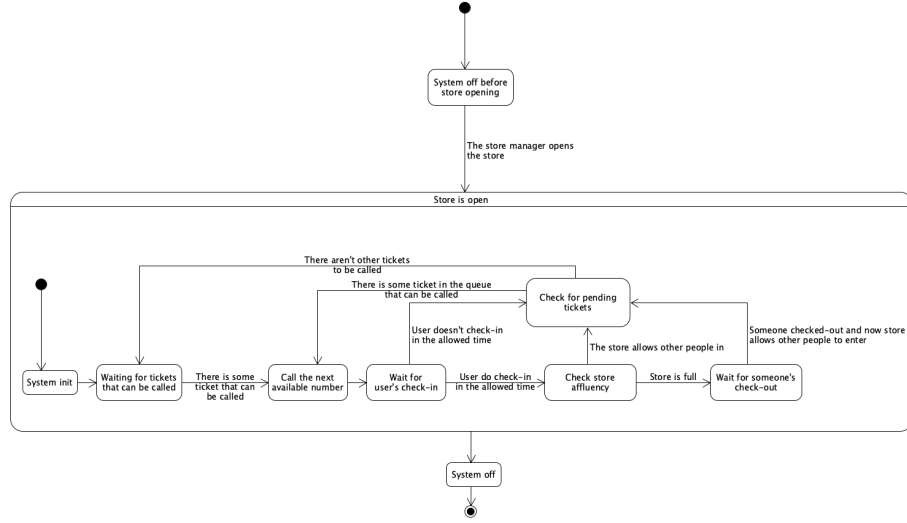


Figure 1: Number calling system

customer doesn't check-in in the assigned time, the system discard the ticket and checks if there are other available tickets. If so, it return in the state of calling the number; else, it will wait for an eligible ticket to be called. If the user, otherwise, scan his *QR code* in time, the system checks the affluency of the store. If it's full, the system will wait for someone's check out, in order to check if some ticket can be called. Else, if the store isn't full, the system doesn't have to wait for a check out to check if there is some ticket eligible to be called. When at some time the store manager will close the store, the system begins its shutdown procedure. It's assumed that at the closure there isn't any other uncalled ticket, since at a certain point the generation of tickets will be blocked, so that that the last ticket will be served around the time of closure.

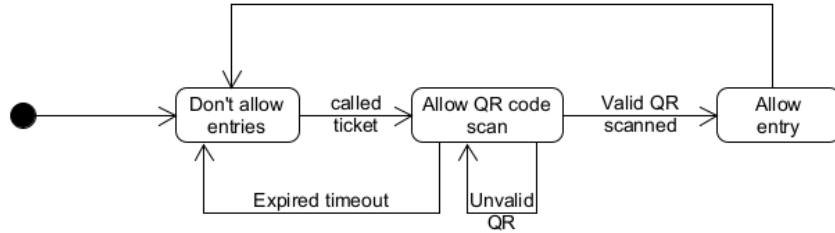


Figure 2: QR Code scanner state machine

The system doesn't allow entries since a ticket is called by the system described above. Then, the QR code reader waits for a QR code to be scanned. When a QR is scanned, the system checks if it's a valid one; if so, allows the entry, otherwise notify the wrong QR code and continue waiting for the right one. If the valid QR isn't scanned in time, the system block the entry and waits for the next called ticket. Otherwise, if the valid QR code is scanned in time, the system allows the entry and, detected the entry, or expired the timeout, block the entries and waits for the next ticket to be called.

2.2 Product Functions

In the previous chapter there were introduced, sketchily, the main features of the software in order to understand, in general, its functioning. Alternatively, in this section will be illustrated and described accurately all the functions that the software allows to do.

2.2.1 Getting a ticket

In order to manage the line of the supermarket, it's used a system based on the "call of numbers". Each user in the queue has a unique ID, and when his number is called (eg. by a store employee), the user is authorized to enter. This mechanism works also for the booking feature, since the system generates a number that won't be called until the beginning of the time slot selected by the user. Users can get a ticket both on the application and at a totem installed at the entry of the store (obviously, on the app users are required to select some specific store of their preference from a given list, sorted by the nearest one from the position obtained from the *GPS*). With the generation of the ticket, the user will get both an ID and a *QR Code* associate to it, to be scanned at the enter of the store. When getting a ticket on the application, users must be logged in with their credentials (if not, they must register at the services following the procedure that will be explained later), while at the Totem they can simply get the ticket without giving any data. On app, customers get a digital ticket with its associated *QR Code*; at totem, the ticket and the qr code will be printed. On both of them, there is displayed the expeted time of call. Getting the ticket, the user can indicate the repart in which he is interested, and using the app can decide both to have a ticket to enter asap in the store or to plan his visit. At the store's totem, he can only get a ticket to enter as soon as possible. Once completed the operation on the app, the user will be able to see at any moment the time needed to reach the store by the preferred mean of transport selected in the settings.

2.2.2 Calling process

The process to admit people in the store follows a *FIFO* logic: the first one that got a ticket, is the first to access the store. When called, a person is authorized to access the store. To register his entry, he is requested to check-in at the entry

in a certain amount of time.

2.2.3 Check-in/Check-out

At the store entries and exits, users have to scan their *QR Codes* to respectively Check-in and Check-Out in the supermarket. If someone doesn't check-in in a prefixed amount of time, definable by the store manager, he'll lost his turn in the queue. *QR Code* scan is required for two important reasons: register someone's entry in the store for contact tracing purposed, and to check that the person entering the store is the one called, to avoid stoles of turns.

2.2.4 Plan a visit

The app allows to brand's customers to plan a visit in a store of their preference. At the end of the process of getting a ticket, if the user select to plan his visit to the store, the software shows the user the time table highlighting the days and the time slots available, in such a way as to allow the user to select the best option for him. If the options do not satisfy the user, he can ask the application for help to find other supermarkets with more comfortable hours. If a reservation process is interrupted in the middle, the user will be able to resume it reopening the app.

2.2.5 Managing single reparts

In the process of getting a ticket, users can indicate the categories of items that they intend to buy in the store, so that it's possible to optimize the waiting time, or they may skip the process, booking a visit to the whole store, but this may lead to greater waiting time, because they can enter the store only when all the reparts are free at the same time. In fact, each repart has a fixed capacity of admitted people, so that each prenotation in line can be called when each selected repart is available. It's obvious that selecting the whole store option, the waiting times are greater.

2.2.6 Notify users

If some account have non completed visits, depending on the position of the client (retrieved by *GPS*), and the preferred option of reaching the selected store (eg. on foot, by car or public transport), the app will notify the client when he can exit home, to arrive in time to enter the supermarket, without losing its turn and avoiding long waiting times. Notice that not all of the moving options may be available in all the places, since it depends on the used map service.

2.2.7 Cancel a visit

If the user can not reach the store in time, he can decide to cancel the booked visit or the generated ticket. In this case the software delete the customer from the queue and rearrange the last one.

2.2.8 Store capacity

The app, also, allows store managers to see the store's affluency and to modify the store capacity in every single repart. When a repart capacity is changed, the latest overflowing bookings will be cancelled, and a notify is sent to the customer, proposing him other comfortable options for the same preferences.

2.2.9 Infer waiting time from users

The system can infer the average time spent by a user in the supermarket using the previous visits as data. When a user get a ticket for the first time without planning his visit, he's asked to enter a reasonable time that he will spend in the supermarket. For the next visits, app will infer this time by time spent in store between check in and check out, so that can estimate more accurately waiting times, and to suggest users better time options while planning their visit.

2.3 User Characteristics

CLup gives access to two different sets of functionalities based on the two different category of users:

- **Customer:** Allows user to book a visit to the supermarket. The software generates a *QR Code* that the user can use to enter in the supermarket. The user can indicate the exact list of items that he intends to purchase, or, at least, the categories of items that he intends to buy and, also, he can indicates the approximate expected duration of the visit. Moreover, the user can see the *ETA* to enter the store. *CLup* also allows clients to obtain their book at the store entry.
- **Store Manager:** The Store Manager can view information about the store, in particular he can access to the reservations made by users to predict the future affluency and check the level of affluency in the store (and also in specific zones of the store)

2.4 Assumptions, Dependencies, Constraints

2.4.1 Domain Assumptions

extra

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.2.1 List of Requirements

extra

3.2.2 Mapping

extra

3.2.3 Use Cases

3.2.3.1 Registration of a customer

Name	Registration of a customer
Actors	Customer
Entry Condition	Customer has the internet connection available and has accessed the application on its device
Event Flow	<ol style="list-style-type: none">1. Customer visualizes the initial page of the app2. Customer clicks on “Sign up as customer” button3. Customer inserts a username, a password, an e-mail, his name, his surname and his phone number as mandatory fields4. The system confirms the registration of the customer5. The system saves the information of the customer
Exit Conditions	Customer is successfully registered to the application
Exception	<ol style="list-style-type: none">1. The username is already present in the system2. The customer did not fill up all the mandatory fields with valid data <p>If one or more of the above situations occur, the application will throw an error message and will return to the registration form page</p>

3.2.3.2 Registration of a store

Name	Registration of a store
Actors	Store manager
Entry Condition	Store manager has the internet connection available and has accessed the application on its device
Event Flow	<ol style="list-style-type: none"> 1. Store manager visualizes the initial page of the app 2. Store manager clicks on “Sign up as store” button 3. Store manager compile all the mandatory fields concerning the store 4. Store manager loads a certification document which proves that it is a real store 5. The system validates the certification 6. The system confirms the registration of the store 7. The system saves the information of the store
Exit Conditions	The store is successfully registered to the application
Exception	<ol style="list-style-type: none"> 1. The store is already present in the system 2. The store manager did not fill up all the mandatory fields with valid data 3. The certification is invalid <p>If one or more of the above situations occur, the application will throw an error message and will return to the registration form page</p>

3.2.3.3 Login of a customer

Name	Login of a customer
Actors	Customer
Entry Condition	Customer is already registered to the application service
Event Flow	<ol style="list-style-type: none"> 1. Customer accesses the application through its device 2. Customer clicks on “Login as customer” button 3. The system opens the “Login as customer” page 4. Customer compiles the fields “Username” and “Password” 5. Customer clicks on “Login” button 6. The system opens the “Customer menu” page
Exit Conditions	Customer has successfully logged in
Exception	<ol style="list-style-type: none"> 1. Customer enters invalid Username 2. Customer enters invalid Password <p>If one or more of the above situations occur, the application will throw an error message and will return to the “Login as customer” page</p>

3.2.3.4 Login of a store manager

Name	Login of a store manager
Actors	Store manager
Entry Condition	Store manager’s store is already registered to the application service
Event Flow	<ol style="list-style-type: none"> 1. Store manager accesses the application through its device 2. Store manager clicks on “Login as store” button 3. The system opens the “Login as store” page 4. Store manager compiles the fields “ID” and “Password” 5. Store manager clicks on “Login” button 6. The system opens the “Store menu” page
Exit Conditions	Store manager has successfully logged in
Exception	<ol style="list-style-type: none"> 1. Store manager enters invalid ID 2. Store manager enters invalid Password <p>If one or more of the above situations occur, the application will throw an error message and will return to the “Login as store” page</p>

3.2.3.5 Customer makes a reservation

Name	Customer makes a reservation
Actors	Customer
Entry Condition	Customer is already logged in the application service
Event Flow	<ol style="list-style-type: none"> 1. Customer clicks on “Make a reservation” 2. The system opens the “Reservation” page 3. The customer can see the list of stores in his city and can filter this list by choosing a specific chain from a drop down menu 4. Customer selects a store and clicks on “Next” button 5. The system opens the “Objects’ category” page 6. Customer can see the list of all possible objects’ category and can select some of them (optional) 7. Customer click on “Next” button to confirm the selection 8. The system opens the “Time preference” page 9. Customer can see two button, “As soon as possible” and “Choose a time slot” <p>(a) if the customer clicks on “As soon as possible” button, the system will check if there is a free spot on the following hour</p> <ol style="list-style-type: none"> i. if there is a free spot, the system will generate a QR code, show it to the customer and save it in the database ii. otherwise, the system will inform the customer and suggest him stores less crowded <p>(b) if the customer clicks on “Choose a time slot”, the system will open the “Time slots” page, where he can see all the available time slot, and clicking on one of them will generate a QR code, that will be showed to the customer and saved in the database</p>
Exit Conditions	Customer has successfully made a reservation
Exception	<ol style="list-style-type: none"> 1. Customer click on a timeslot no longer available 2. Customer tries to book more than the set limit of reservations per week If the above situation occur, the application will throw an error message and will reload the “Time slots” page (updating it) 3. Customer doesn’t select any product category If the above situation occurs, the application will throw an error and ask the client to select at least some item. 4. Customer tries to get a ticket while he already have a unused one <p>If the above situation occurs, the application shows an error message and invite the client to use the pending ticket</p>

3.2.3.6 Customer visualizes reservations

Name	Customer visualizes reservations
Actors	Customer
Entry Condition	Customer is already logged in the application service and made a reservation
Event Flow	<ol style="list-style-type: none">1. Customer clicks on “Show requests” button2. The app show the list of bookings made and tickets requested3. The customer select the desired option4. The system opens the detail page of the selected option, that includes the “QR Code” and the number that should be called, among with the scheduled entry date and time
Exit Conditions	Customer can visualize the QR code
Exception	None

3.2.3.7 Manager modifies store parameters

Name	Store manager modifies store parameters
Actors	Store manager
Entry Condition	Store manager is already logged in the application service
Event Flow	<ol style="list-style-type: none"> 1. Store manager clicks on “Modify parameters” button 2. The system opens the “Modify parameters” page 3. Store manager can see the name of each parameter flanked by a modifiable text box containing the previous value of that parameter (eg. zone capacity, store capacity) <ol style="list-style-type: none"> (a) Store manager modify one, more or none of the parameters and clicks “Save changes” button (b) Store manager choose to add/delete (except for the default parameters “Store capacity” and “Maximum number of weekly reservations for client”) some parameter of the store 4. The system save the changes and bring him back to “Store menu” page
Exit Conditions	Store manager has successfully updated store parameters
Exception	<ol style="list-style-type: none"> 1. Store managers enters an invalid value for one or more parameters 2. The store manager tries to add an already existing parameter, or tries to delete the default “Store capacity” and “Maximum number of weekly reservations for client” parameters 3. The sum of the capacities of the single zones is greater than the store capacity <p>If the above situation occur, the application will throw an error message and will return to “Modify parameters” page</p>

3.2.3.8 The store manager monitors the store situation

Name	Store manager modifies store parameters
Actors	Store manager
Entry Condition	Store manager has opened the app and is already logged in
Event Flow	<ol style="list-style-type: none">1. Manager clicks on “Monitor store” button2. The app shows a page with statistics on the store, including number of people inside the store, percentage of store occupation and the number of daily access to the building<ol style="list-style-type: none">(a) If the manager wants, he can see the same statistics per store zone by clicking the “monitor zones” button. The system will show all the zones sorted by criticity
Exit Conditions	Store manager can see statistics on the store
Exception	None

3.2.3.9 The store manager manages customers bookings

Name	The store manager manages customers bookings
Actors	Store manager
Entry Condition	Store manager has opened the app and is already logged in
Event Flow	<ol style="list-style-type: none"> 1. Manager clicks on “Manage bookings” button 2. The app shows a list of reservations, with some of their details in preview (such as chosen date and time slot) 3. The manager can choose one of them, and the app will show some possibilities to the manager <ol style="list-style-type: none"> (a) The manager can press on the button “Contact client” to contact the client for some reasons. <ol style="list-style-type: none"> i. The manager can choose to contact the client via mail clicking on the button “Email option” <ol style="list-style-type: none"> A. The app will show an interface where the manager can insert the text B. The manager clicks on the “Send” button to send the message ii. The manager can choose to call the client clicking on the “Call customer” button <ol style="list-style-type: none"> A. The app will invoke mobile OS API to start a call iii. The app returns to the previous page (b) The manager can click on the button “Cancel booking” to cancel a reservation <ol style="list-style-type: none"> i. The app will show a dialog box where the manager can put-in an optional message, explaining the reasons of the cancel ii. The manager clicks on the “Delete button” iii. The app will show a confirmation box to ask if the manager is sure to proceed iv. The manager clicks on “Yes” to confirm the deletion v. The app closes the dialog box (c) The manager can choose to reschedule a booking <ol style="list-style-type: none"> i. The app will show a dialog box where the manager can choose a new time slot and insert a message explaining the reasons ii. The manager clicks on “Modify” to modify the booking iii. The app closes the dialog box
Exit Conditions	Store manager can manage the store and is able to edit reservations
Exception	None

3.2.3.10 Customers reservations management

Name	Customers reservations management
Actors	Customer
Entry Condition	The customer has the application opened, is logged in and has at least one pending request
Event Flow	<ol style="list-style-type: none"> 1. The user press on the “Show requests” button 2. The app shows a page with the requests made by the client 3. The user selects the desidered requests 4. The app will show the requests details 5. The user press the “edit” button <ol style="list-style-type: none"> (a) If the selected requests is a ticket, the customer can delete it pressing the “Delete button” <ol style="list-style-type: none"> i. The system will show a confirmation dialogue ii. The client press the yes button iii. The ticket is deleted and the app will return to the previous screen if there are other requests, or to the main menu otherwise (b) If the selected request is a booking, the client can both click the delete button, and follow the above procedure or can select the “modify button” <ol style="list-style-type: none"> i. If the modify button is selected, the app will restart the “Make a reservation process” described in Use Case n. 5. The customer can modify all the parameters of his reservation, including transform-ing it in a “As soon as possible ticket
Exit Conditions	The client can modify his reservation
Exception	None

3.2.4 Sequence Diagram

extra

3.2.5 Scenarios

extra

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

3.6 Additional Specifications

extra

4 Formal Analysis Using Alloy

4.1 Alloy

extra

5 Effort Spent

6 References