CLup: Customers Line-up

Alessandro Nocentini
Alberto Valentini

Requirements Analysis and Specification Document

Version: n°1 **Date: 06/07/2021**

Table of Contents

1. Introduction

1.1 Purpose

The coronavirus emergency has put a strain on society on many levels, due to many countries imposing lockdowns that allow people to exit their homes only for essential needs, and enforcing strict rules even when people are justified in going out (such as limiting the number of accesses to buildings and keeping a distance of at least one meter between people).

In particular, grocery shopping---a most essential need---can become a challenge in the presence of such strict rules. Indeed, supermarkets need to restrict access to their stores to avoid having crowds inside, which typically results in long lines forming outside, which are themselves a source of hazards. In these trying times, people turn to technology, and in particular to software applications,to help navigate the challenges created by the imposed restrictions.

The goal of this project is to develop an easy-to-use application that, on the one side, allows store managers to regulate the influx of people in the building and, on the other side, saves people from having to line up and stand outside of stores for hours on end.

1.2 Scope

First of all, the application allows the user to look for a supermarket and check the actual accessibility based on the capacity of the store. In case of reservation, it will provide the user a QR code to be scanned at the entrance of the supermarket.

In addition, it shows the timetable of the supermarket with the possibility of booking a future available time slot (at most till the day after). The reservation can always be canceled later.

In doing this, the app takes into account the effective customer’s access, monitoring in real time the number of customers in the store.

A notification will send when the booked time is coming (30 minutes in advance) and, if the access does not happen within 10 minutes from the start instant of the reservation, the booking will be automatically deleted by the system (with related notification).

1.2.1 Goals

- G1: Allow the user to enter in the store.
- G2: Allow the user to book a ticket.
- G2: Allow the user to check a ticket.
- G3: Allow the user to delete a ticket.
- G4: Send notification to user to alert about incoming reservation.
- G5: Send notification to warn the user about deleted reservation.

1.3 Definitions, Acronyms, and Abbreviations.

1.3.1 Definitions

- Customer: A person that has to buy something at the supermarket.
- User: A person with a smartphone that downloaded the application and uses it.
- Non-User: A person that does not use the application.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document
- QR Code: Quick Response code

1.3.3 Abbreviations

- CLup: Customer Line-Up (name od the application)
- Gn: Goal number n
- Dn: Domain assumption number n
- Rn: Requirement number n
- Pn: Performance requirement number n

1.4 References

Project proposal: https://conf.researchr.org/home/icse-2021/score-2021#clup-customers-line-up

2. The Overall Description

2.1 Product Perspective

2.1.1 UML Description

2.2 Product Functions

- **Booking management**: Allows the user to look for a supermarket from a search bar, among which of them that are supported by the application and are in the user's city, and then choose the desired store. Then, the application asks the user to choose one of the available time slots (the ones that are not in red) from the supermarket timetable (at most till the day after) and a virtual ticket, composed by a QR code, will be generated accordingly. The QR code will be scanned at the entrance of the supermarket and it contains some useful infos: name, surname and the expected arival time. Once the ticket is created, it can be checked or deleted. If after 10 minutes of the booked time the user does not have scanned the QR code, the booking will be automatically canceled by the system. In order to allow the access to the store also to who does not have downloaded the application, the store capacity dedicated to the booking by app is equal to X% (X parameter decided by the manager).

- **Sign Up & Login**: The first time the user uses the application he must sign up filling all the mandatory fields (name, surname, e-mail address) and choosing an username and a password. The next times, the user accesses to CLup through the login procedure, so inserting the username or email address and the password and then clicking on the login button. If the user also ticks the "Remember me" box, next times he will access the application directly.
  If the user does not remember the password, he can click on "forgot password?", so the system will send an email to the user's email address containing the instructions to set another password.

- **Send notification**: if the user books a ticket when miss more than 30 minutes at the beginning of the reservation, 30 minutes before the start of the time slot the system sends a reminder. The system also sends a notification when the reservation is automatically canceled in case the QR code would not scanned within 10 minutes the start of time slot.

2.3 User Characteristics

The user is someone who wants to avoid spending a lot of time in long queues and at the same time wants to stay safe avoiding gatherings. The application was designed to be able to be used by any range of ages and without distinction of any kind.

2.4 Domain Assumptions

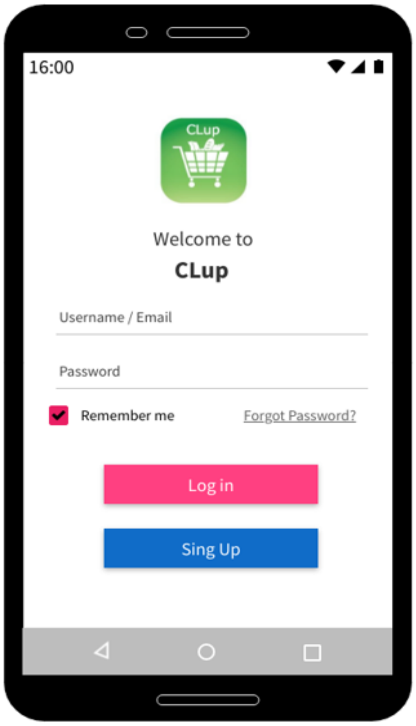
- D1: Each reservation owns one inique QR code.
- D2: When the ticket is generated before the actual start of the selected time slot, its status is “WAITING”, otherwise it is directly “VALID”.
- D3: When the selected time slot starts, the status of the ticket changes from “WAITING” to “VALID”
- D4: When the QR code is scanned, its status will change from “VALID” to “USED” and at the end of the shift it will be automatically changed to “EXPIRED”.
- D5: If the QR code is not scanned within 10 minutes the expected entrance. Its status will change from “VALID” to “EXPIRED”.
- D6: QR codes are readable and scannable.
- D7: The internet connection works properly.
- D8: Each user has a smartphone and he has downloaded CLup.
- D9: CLup users are correctly registered.
- D10: Timetables of supermarkets are known.
- D11: Each supermarket has limited capacity.
- D12: Each supermarket accepts a limited numer of users (the remaining capacity is assigned to the non-users).

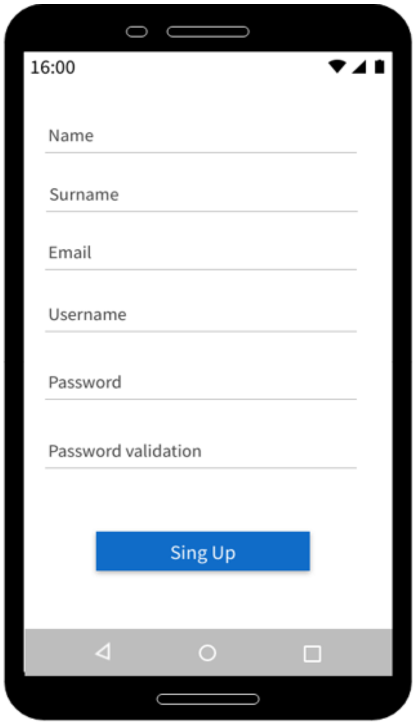
3. Specific Requirements

3.1 User Interface

The application must be able to be used by any range of ages, so the user interface has to be simple and intuitive. The following pictures represent how the main features of the app should look to a user.

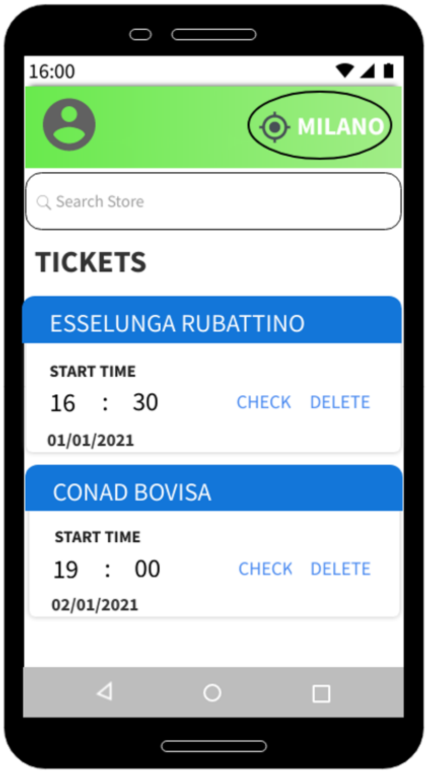
1. Log-in & Sign up





2. Home

3. Check Ticket



ESSELUNGA RUBATTINO

START TIME

16 : 30

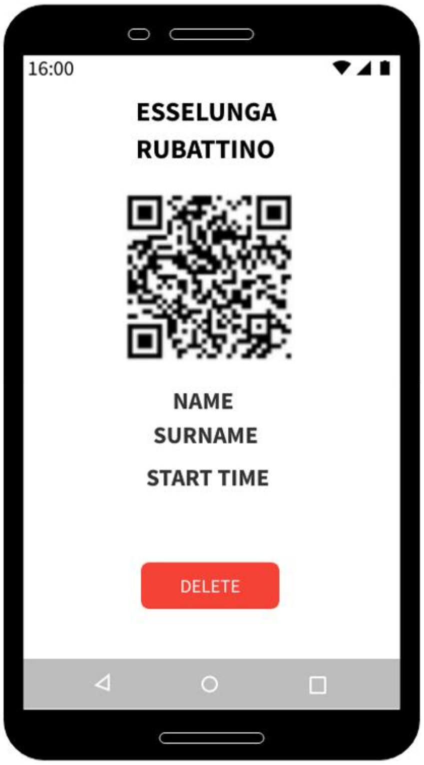
01/01/2021

CONAD BOVISA

START TIME

19 : 00

02/01/2021

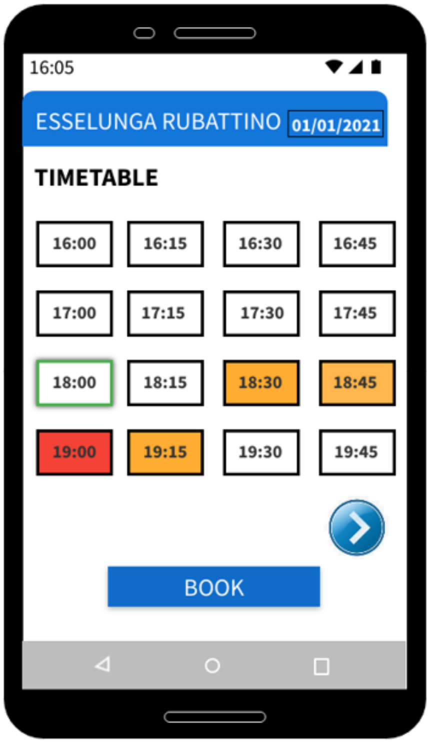


ESSELUNGA RUBATTINO

NAME
SURNAME
START TIME

4. Create ticket

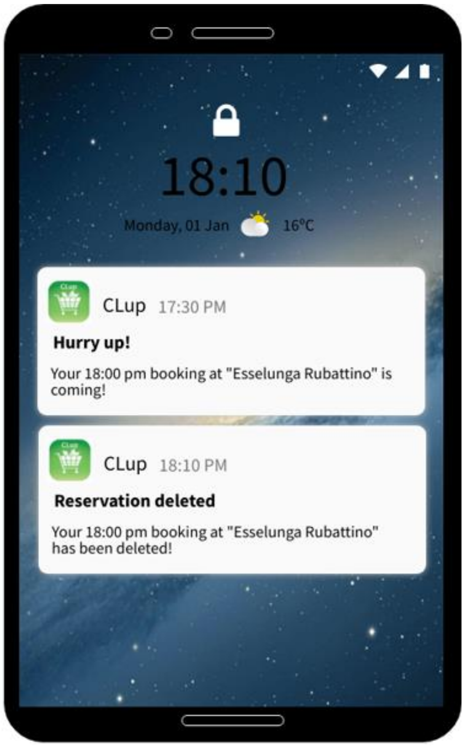
5. Notification



ESSELUNGA RUBATTINO 01/01/2021

TIMETABLE

16:00	16:15	16:30	16:45
17:00	17:15	17:30	17:45
18:00	18:15	18:30	18:45
19:00	19:15	19:30	19:45



CLup 17:30 PM

Hurry up!

Your 18:00 pm booking at "Esselunga Rubattino" is coming!

CLup 18:10 PM

Reservation deleted

Your 18:00 pm booking at "Esselunga Rubattino" has been deleted!

3.2 Functional Requirements

- R1: User is registered and the user’s data are stored in a database.
- R2: User has to search and select the supermarket from the search bar.
- R3: User has to choose the city. The default city set by the system is “MILANO”.
- R4: The system shall restrict the research of supermarkets to the ones that are in the user's city.
- R5: User has to select the day in which he wants to book a ticket: the current day or the day after.
- R6: User has to select one of the time slots from the supermarket timetable.
- R7: The system shall show in white the free time-slots, in orange the ones whose reservations exceed Y% of capacity but are still available, and in red the ones that are full and so not available.
- R8: The system shall allow the user to book a time slot only if it is not full.
- R9: The system allows the user to create only one ticket per day.
- R10: Customer has to scan the QR code to enter.
- R11: The system shall allow the entry in the store only if the ticket status is “VALID”.
- R12: The system shall allow the user to check the ticket.
- R13; The system shall allow the user to delete the ticket .
- R11: The system shall automatically remove the ticket when its status changes to “EXPIRED”.
- R12: The system automatically delete the reservation if the QR code will be not scanned within 10 minutes the beginning of the booked time slot.
- R13: The system shall generate a “WAITING” or “VALID” ticket based on the situation (view D2).
- R14: The QR code generated by the system must contain the user data, the start time of reservation and its status.
- R15: The system should be able to receive information from the supermarket.
- R16: Internet connection must stable in order to properly use the application.
- R17: The system shall allow the user to update the data inserted in the registration phase.
- R18: The system shall allow the user to log out.

3.2.1 Scenarios

1. Classic Use of CLup

   After the login, the user can look for the desired supermarket from the search bar and then choose it. So, the user can see the available time-slots for the day selected according to the supermarket timetable (he can eventually use the arrows to scroll the timetable). The system shows with a white background the boxes corresponding to the free time slots, while it uses an orange background to mark the ones thar are near to saturation and a red colour to means “sold out” . Once the user has chosen the desired time slot (marked in green) and clicked on the “BOOK” button, the QR code will be generated. When his reservation starts, the user can scan his ticket at the entrance of the supermarket and so enter to start his shopping. The ticket will be automatically deleted by the system at the end of the turn.

2. The utility of the reminder

   Riccardo is a very busy businessman, so he tends to manage his appointments the day before. Due to his high number of daily briefings, can happen that Riccardo forgets the supermarket reservation booked with CLup the day before. In this case, the CLup notification 30 minutes before the start of the turn allows him to get in motion in order to try to arrive in time at the supermarket, or decide to delete the reservation.

3. The straggler

   By referring to the "utility of the reminder" section, Riccardo has eventually decided to try to arrive in time at the supermarket but, unfortunately, he didn't arrive within 10 minutes the start of the reservation, so the system has automatically deleted his ticket and a notification is sent to warn him about that.

4. There is always a first time!
   The first time the user uses the application, he must create an account through the registration procedure. From the login page, the user has to click on the “Sign Up” button and then CLup asks the user to insert his name, surname, email address and to set an username and a password. Once did that, he can login filling the fields “Username/Email” and “Password” and clicking on the “Log In” button. If the user ticks the box “Remember me”, the system will not ask to repeat the login procedure next times until he will log out from the account section of the application. Once the user entered the homepage, he must set the city in which he wants to search the supermarket, otherwise, it will remain “MILANO” by default.

5. Forgot password?
   If the user is not able to access to CLup because he has forgotten his password, he can click on "forgot password?": the system will send an email to the user's email address containing the instructions to set another password that will be valid starting from the next access on the app.

3.3 Performance Requirements

- P1: The system shall send the QR code to the user within 5 seconds the request has been received.
- P2: If the user books a ticket when miss more than 30 minutes at the beginning of the reservation, 30 minutes before the start of the time slot the system shall sends a notification.
- P3: The system shall send a notification to the user not after 5 seconds the system itself has canceled the reservation due to the user late (10 minutes from the beginning of the reservation).

- P4: If in the login procedure the user clicks on "forgot password?", the system shall send an email to the user's email address within 5 seconds.
- P5: The application must able to handle the simultaneous connection of a minimum amount of 10.000 users.

3.4 Design Constraints

The application requires a smartphone with a stable internet connection.
The system requires a remote server for each supermarket supported by the application.

3.5 Software System Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes by specified so that their achievement can be objectively verified. The following items provide a partial list of examples. These are also known as non-functional requirements or quality attributes.

These are characteristics the system must possess, but that pervade (or cross-cut) the design. These requirements have to be testable just like the functional requirements. Its easy to start philosophizing here, but keep it specific.

3.5.1 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery. If you have MTBF requirements, express them here. This doesn't refer to just having a program that does not crash. This has a specific engineering meaning.

3.5.2 Availability

Specify the factors required to guarantee a defined availability level for the entire system such as checkpoint, recovery, and restart. This is somewhat related to reliability. Some systems run only infrequently on-demand (like MS Word). Some systems have to run 24/7 (like an e-commerce web site). The required availability will greatly impact the design. What are the requirements for system recovery from a failure? "The system shall allow users to restart the application after failure with the loss of at most 12 characters of input".

3.5.3 Security

Specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:

- *Utilize certain cryptographic techniques*
- *Keep specific log or history data sets*
- *Assign certain functions to different modules*
- *Restrict communications between some areas of the program*
- *Check data integrity for critical variables*

3.5.4 Maintainability

Specify attributes of software that relate to the ease of maintenance of the software itself. There may be some requirement for certain modularity, interfaces, complexity, etc. Requirements should not be placed here just because they are thought to be good design practices. If someone else will maintain the system

3.5.5 Portability

Specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems. This may include:

- *Percentage of components with host-dependent code*
- *Percentage of code that is host dependent*
- *Use of a proven portable language*
- *Use of a particular compiler or language subset*
- *Use of a particular operating system*

Once the relevant characteristics are selected, a subsection should be written for each, explaining the rationale for including this characteristic and how it will be tested and measured. A chart like this might be used to identify the key characteristics (rating them High or Medium), then identifying which are preferred when trading off design or implementation decisions (with the ID of the preferred one indicated in the chart to the right). The chart below is optional (it can be confusing) and is for demonstrating tradeoff analysis between different non-functional requirements. H/M/L is the relative priority of that non-functional requirement.

ID	**Characteristic**	**H/M/L**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	Correctness													
2	Efficiency													
3	Flexibility													
4	Integrity/Security													
5	Interoperability													
6	Maintainability													
7	Portability													

8	Reliability		▨	▨	▨	▨	▨	▨	▨	▨				
9	Reusability		▨	▨	▨	▨	▨	▨	▨	▨	▨			
10	Testability		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨		
11	Usability		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
12	Availability		▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Definitions of the quality characteristics not defined in the paragraphs above follow.

- *Correctness - extent to which program satisfies specifications, fulfills user's mission objectives*
- *Efficiency - amount of computing resources and code required to perform function*
- *Flexibility - effort needed to modify operational program*
- *Interoperability - effort needed to couple one system with another*
- *Reliability - extent to which program performs with required precision*
- *Reusability - extent to which it can be reused in another application*
- *Testability - effort needed to test to ensure performs as intended*
- *Usability - effort required to learn, operate, prepare input, and interpret output*

THE FOLLOWING (3.7) is not really a section, it is talking about how to organize requirements you write in section 3.2.   At the end of this template there are a bunch of alternative organizations for section 3.2. Choose the ONE best for the system you are writing the requirements for.

3.6 Organizing the Specific Requirements

For anything but trivial systems the detailed requirements tend to be extensive. For this reason, it is recommended that careful consideration be given to organizing these in a manner optimal for understanding. There is no one optimal organization for all systems. Different classes of systems lend themselves to different organizations of requirements in section 3. Some of these organizations are described in the following subclasses.

3.6.1 System Mode

Some systems behave quite differently depending on the mode of operation. When organizing by mode there are two possible outlines. The choice depends on whether interfaces and performance are dependent on mode.

3.6.2 User Class

Some systems provide different sets of functions to different classes of users.

3.6.3 Objects

Objects are real-world entities that have a counterpart within the system. Associated with each object is a set of attributes and functions. These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

3.6.4 Feature

A feature is an externally desired service by the system that may require a sequence of inputs to effect the desired result. Each feature is generally described in as sequence eof stimulus-response pairs.

3.6.5 Stimulus

Some systems can be best organized by describing their functions in terms of stimuli.

3.6.6 Response

Some systems can be best organized by describing their functions in support of the generation of a response.

3.6.7 Functional Hierarchy

When none of he above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by either common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be use dot show the relationships between and among the functions and data.