POLITECNICO
MILANO 1863

Travlendar+ project Neroni, Pozzi, Vetere

Design Document

Deliverable:	DD
Title:	Design Document
Authors:	Cristiano Neroni, Davide Pozzi, Maurizio Vetere
Version:	1.0
Date:	10-January-2021
Download page:	https://github.com/pollo-fritto/PozziNeroniVetere
Copyright:	Copyright © 2021, Neroni| Pozzi| Vetere – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The purpose of this document is to provide more technical and detailed information about the software discussed in the RASD document. It will represent a strong guide for the programmers that will develop the application considering its different parts: the basic service and the two advanced functions. In this DD we present hardware and software architecture of the system in terms of components and interactions among those components. Furthermore, this document describes a set of design characteristics required for the implementation by introducing constraints and quality attributes. It also gives a detailed presentation of the implementation plan, integration plan and the testing plan. In general, the main different features listed in this document are:

- The high-level architecture of the system
- Main components of the system
- Interfaces provided by the components
- Design patterns adopted

Stakeholders are invited to read this document in order to understand the characteristics of the project being aware of the choices that have been made to offer all the functionalities also satisfying the quality requirements.

1.2 Scope

Clup is an application that aims to avoid users from crowding outside supermarkets when doing grocery shopping in pandemic times.
The application can be used both by store customers and store managers. On one hand users can virtually queue by Clup to enter the supermarket and they are provided with real time information about the line, in this way they can arrive at the entrance only when they are allowed to enter. On the other hand the application monitors and stores the information about people fluxes; this data is then provided to store managers who can take actions depending on the situation. The few paragraphs just read represent an overview of the main functionalities offered by the system: more detailed information can be found on the RASD document.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 Revision history

1.5 Reference Documents

1.6 Document Structure

- Chapter 1 describes the scope and purpose of the DD, including the structure of the document and the set of definitions, acronyms and abbreviations used.
- Chapter 2 contains the architectural design choice, it includes all the components, the interfaces, the technologies (both hardware and software) used for the development of the application. It also

includes the main functions of the interfaces and the processes in which they are utilised (Runtime view and component interfaces). Finally, there is the explanation of the architectural patterns chosen with the other design decisions.

- Chapter 3 shows how the user interface should be on the mobile and web application.
- Chapter 4 describes the connection between the RASD and the DD, showing the matching between the goals and requirements described previously with the elements which compose the architecture of the application.
- Chapter 5 traces a plan for the development of components to maximize the efficiency of the developer team and the quality controls team. It is divided in two sections: implementation and integration. It also includes the testing strategy.
- Chapter 6 shows the effort spent for each member of the group.
- Chapter 7 includes the reference documents.

2 Architectural Design

2.1 Overview

The architecture of the application is structured according to three logic layers:

- Presentation level (P) handles the interaction with users. It contains the interfaces able to communicate with them and it is responsible for rendering of the information. Its scope is to make understandable the functions of the application to the customers.
- Business logic or Application layer (A) takes care of the functions to be provided for the users. It also coordinates the work of the application, making logical decisions and moving data between the other two layers.
- Data access layer (D) cares for the management of the information, with the corresponding access to the databases. It picks up useful information for the users in the database and passes them along the other layers.

The architecture has to be made in client-server style. Client and server are being allocated into different physical machines and their communication takes place via other components and interfaces located in the middle of the structure, composed by hardware and software modules. The process begins with the invocation of a method to provide any functionality to the client, like sending a report or requiring some information about violation or accidents. Then, the invocation of a specific method is caught by the server and its behaviour depends on the required function.

2.2 Component view

2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patterns

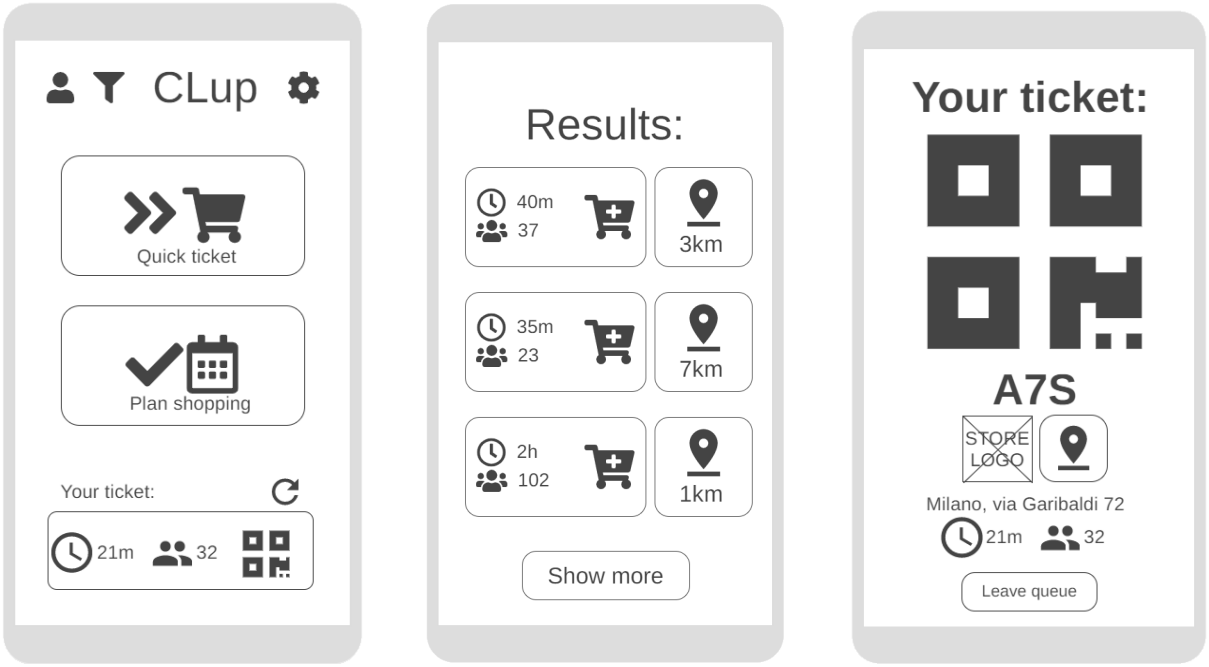
2.7 Other design decisions

3 User Interface Design

3.1 Overview

CLup is an application aimed at decreasing the probability of contracting COVID-19 (diseases in general) when going shopping to a supermarket. There are two fundamental components: the main one targets customers of supermarkets while the second one is available for store managers.

3.1.1 User Interfaces



(a) Main screen (b) Store list (c) Your ticket

Figure 1: Clup wireframes

End user functionalities Regarding the first target - customers - they will be required to register to the service the first time they use it by inserting their full name, email address, ID card, phone number and a password. The customer will also be requested to specify his physical address, or to enable the GPS, in order to allow CLup to find stores nearby; this last information can be changed anytime the user needs. If the customer is not willing to register or share his address, the service will not be available.

Once the setup is done, customers will be able to access the homepage of the application, where they can tap on a "Virtually queue" button that will allow them to see a list of stores inside a specified range from their current location: for each store a distance in kilometers from the user position will be outlined, as well as the number of people inside the store and its maximum capacity; whenever a store is full, the current number of people in line and an *EWT* are displayed.
It is also possible to visualize stores on a map and, by tapping on one of them, to see the same information displayed in the list. Now, if the user chooses to reserve a spot in the line, the application will open a confirm dialog specifying EWT and the expiration of the ticket. If the user refuses nothing happens, if he accepts instead CLup will process the request, show his ticket and the real time evolution of the line; the ticket is also visible from a home button. The process is shown in Figure **??**.
The distance range in which CLup will look for supermarkets is specified by the user through the filter

button in the homepage, this button will in fact open the filter screen in which, among other parameters, a sliding bar controls the distance and a drop-down list allows the user to filter the chains of supermarkets.

Another important feature is the possibility to book an entrance later in the day or in another day. The user can specify from the filters whether he prefers to choose the day or the store first and he can set the time range in which he wants to book. There is a dedicated button in the app's main screen that redirects the user to either the list/map of supermarkets or the calendar, and once the user chooses he will be respectively shown the calendar or the list/map, this time with colours to indicate the average crowdedness of stores/days given the set time range. When the user chooses the day and supermarket combination, a timetable spanning the chosen time range is shown, divided in 15 minutes time slots each one having again a colour to indicate the crowdedness. The user will be able to check his reservation on the home page and near the entrance time he will be provided an actual ticket.
The access at the supermarket is restricted by turnstiles with QR code readers, a staff member is expected to verify that nobody waits his turn in front of the entrance, jumps the turnstile or does anything irresponsible.

Customers who, for any reason, do not use the app will still be able to queue in CLup supermarkets by obtaining a printed ticket from a physical totem located near such stores; the functioning of the application will be similar to the "Quick ticket" app function with the difference that the user can only obtain a ticket by using a totem.
The tickets consist of a QR code and an easy to remember alphanumeric code alternative to enter the store. There will also be monitors that show the numbers allowed to enter and, eventually, delays.

Internal use functionalities Another component of CLup targets store managers: when the store decides to join the CLup network, ad-hoc credentials to access the web app will be given. Special staff-only functions will then be achieved by the use of a web application, accessible via internal-use terminals.

4 Requirements Traceability

5 Implementation, Integration and Test Plan

5.1 Implementation

5.2 Integration

5.3 Test Plan

6 Effort Spent

References

[1] S. Bernardi, J. Merseguer, and D. C. Petriu. A dependability profile within MARTE. *Software and Systems Modeling*, 10(3):313–336, 2011.

