POLITECNICO MILANO 1863

Design Document
2020-2021

Alessandro Polidori (Codice persona 10573078)
Olimpia Rivera (Codice persona 10617517)

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to further detail the software system discussed in the RASD by providing more technical and precise informations. The RASD, indeed, describes the system in terms of requirements, assumptions and goals and gives a general overview of the main system's functionalities, whereas the DD goes deeper into software design and architecture. In particular, all the required components of the system and the interactions among them are presented, as well as run-time processes and motivated design choices. In addition, the document describes the implementation, integration and testing plans.
The relevant issues explored in the Design Document are the following:

- High-level architecture
- Main components and relative interfaces
- Runtime behavior
- Design patterns
- Implementation, integration and testing plan

This document represents a vehicle for stakeholders communication, as it defines a set of design decisions made to offer both the essential functionalities and the required quality attributes.

1.2 Scope

The CLup software application is designed to help both common users and store managers to deal with some relevant difficulties and challenges created by the coronavirus emergency. The CLup system, indeed, gives users the possibility to join virtual lines to stores and wait for their turn at home instead of standing in line outside stores. The system itself notifies users when it is time to leave to reach the store. Alternatively, thanks to the CLup application, users can choose to book visits to stores in advance, by selecting the preferred date and time slot. In order for the lining up mechanism to work effectively, all the users are asked to provide their expected shopping duration and the system will calculate, and eventually modify, the waiting times of the users in line by taking into account the following events: users currently in line, users currently in the store, users getting out of the line, cancelled reservations, crowdedness of stores' departments and users' visits' durations. Indeed, a QR code will be generated for each user and scanned at entrances and exits by smart turnstiles, so that the system can collect and store informations about visits' duration. The influx of people is managed by the system in an even finer way, by asking users to provide a list of product categories they intend to purchase. The system is also designed to regulate accesses to stores in order to respect restrictions imposed by the virus

emergency, by taking into account the stores' capacities. For this reason, store managers as well benefit from such a software application because they have the possibility to monitor the number of people in they stores updating in real time.
A more detailed overview of the functionalities offered by the system can be found in the RASD.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

1.3.2 Acronyms

- RASD: Requirements Analysis and Specifications Document
- DD: Design Document
- API: Application Programming Interface

1.3.3 Abbreviations

1.4 Revision History

1.5 Document Structure

The Design Document is structured in the following chapters:

- **Chapter 1** describes the document's purpose and scope and its differences with respect to the RASD. Useful specifications (definitions, acronyms, abbreviations) are included for a better understanding of the next sections of the document.
- **Chapter 2**
- **Chapter 3**
- **Chapter 4**
- **Chapter 5**
- **Chapter 6** presents the effort spent by the group members while working on this project.
- **Chapter 7** includes the reference documents.

2 Architectural Design

2.1 Overview

The application architecture is going to be developed as a three-tier architecture. The three logical al physical computing tiers are the following:

- **Presentation Tier:** manages the interaction between users and the application. It is responsible for collecting informations form the users and for displaying informations.

- **Business logic or Application Tier:** it is where all the communication goes though: it is in charge of processing information collected in the presentation tier and passing it to the data tier. It also manages the way functions will be presented to the users.

- **Data Access Tier:** it stores and manages the informations processed by the application by performing the necessary operations into the database.

More precisely, the architecture will be made as a three-tier client-server style, in which presentation, application logic and data processing layers will be divided across client and server devices and they will all have their own dedicated hardware. The advantages for choosing such an architecture will be further explained in the next sections.

Depending on the functionality requested by a user, the client requests a certain service from the server and the invocation of different methods takes place. The server itself provides the needed services through well defined APIs.

The functionalities offered bu CLup will work in the following way:

- **Line up:** the application server receives a request of lining up, it takes the required information about the current state of the line from the database server and it sends it back to the client. If the user decides to line up, the application server will communicate with the database server to store the information about the new user, so that the state of the line can be changed. Additionally, the application server will interact with the user by sending notification (asynchronously) when it is time to reach the store.

- **Book visit:** the application server receives a request for booking a visit and provides the user with the necessary forms to complete the reservation. The informations provided by the users will be sent to the application server and then stored in the database.

- **Check counter:** the application server will receive the request from the store manager's client, it takes the information about the counter from the database server and it sends it back to the client.

- **Print ticket:** whenever a store manager decides to print a physical ticket, the application server will inform the database server in order for the state of the line to be correctly updated.