

M.Sc. Computer Science and Engineering

Software Engineering 2 Project

**CLup - Customers Line-Up**

**Design Document**

Leoni Luca, Locarno Silvia, Minotti Luca

10th January 2021

Version 1.0

GitHub Repository: https://github.com/lucagrammer/LeoniLocarnoMinotti

**Contents**

[1 Introduction 3](#_Toc58154785)

[1.1 Purpose 3](#_Toc58154786)

[1.2 Scope 3](#_Toc58154787)

[1.3 Definitions, Acronyms, Abbreviations 3](#_Toc58154788)

[1.3.1 Definitions 3](#_Toc58154789)

[1.3.2 Acronyms 4](#_Toc58154790)

[1.3.3 Abbreviations 4](#_Toc58154791)

[1.1 Revision History 4](#_Toc58154792)

[1.2 Reference Documents 4](#_Toc58154793)

[1.5 Document Structure 5](#_Toc58154794)

[2 Architectural Design 6](#_Toc58154795)

[2.1 Overview 6](#_Toc58154796)

[2.2 Component View 8](#_Toc58154797)

[2.2.1 Application Component 8](#_Toc58154798)

[2.2.2 Web Server Component 9](#_Toc58154799)

[2.2.3 Application Server Component 9](#_Toc58154800)

[2.2.4 Data Components 12](#_Toc58154801)

[2.2.5 External services and infrastructures 14](#_Toc58154802)

[2.3 Deployment View 14](#_Toc58154803)

[2.4 Runtime View 14](#_Toc58154804)

[2.5 Component interfaces 14](#_Toc58154805)

[2.6 Selected Architectural Styles and Patterns 14](#_Toc58154806)

[2.7 Other design decisions 14](#_Toc58154807)

[3 User Interface Design 15](#_Toc58154808)

[3.1 UX Diagrams 15](#_Toc58154809)

[3.2 User Interface Mockups 15](#_Toc58154810)

[4 Requirements Traceability 16](#_Toc58154811)

[5 Implementation, Integration and Test Plan 17](#_Toc58154812)

[5.1 Development Process 17](#_Toc58154813)

[5.2 Implementation 17](#_Toc58154814)

[5.2.1 Implementation Plan 17](#_Toc58154815)

[5.2.2 Implementation Choices 17](#_Toc58154816)

[5.3 Integration 17](#_Toc58154817)

[5.3.1 Entry Criteria 17](#_Toc58154818)

[5.3.2 Elements to be integrated 17](#_Toc58154819)

[5.3.3 Integration Sequence 17](#_Toc58154820)

[5.4 Test Plan 17](#_Toc58154821)

[6 Effort Spent 18](#_Toc58154822)

[6.1 Leoni Luca 18](#_Toc58154823)

[6.2 Locarno Silvia 18](#_Toc58154824)

[6.3 Minotti Luca 18](#_Toc58154825)

[7 References 19](#_Toc58154826)

**Chapter 1**

# 1 Introduction

## Purpose

This document constitutes the Design Document (DD). It provides a more technical overview of the Requirement Analysis and Specification Document (RASD) of the system-to-be, describing the main architectural components, their communication interfaces and their interactions.

It will also present the implementation, integration and testing plan. This type of document is mainly addressed to developers since it provides a guide during the development process through an accurate vision of all parts of the software-to-be.

## Scope

As explained in the RASD, CLup aims to manage the queues to access stores in the Coronavirus era in a simple but effective way. Avoiding dangerous gatherings and unnecessary waste of time, CLup allows you to queue at a store directly from the application and receive a notification when it is time to reach the store, taking into account the time required to get to the shop using the indicated means of transport. Of course, fall-back options are available for people who don't have access to the required technology.

Moreover, the CLup Application also allows you to book a visit to a store in advance by indicating the date, time and product categories you are interested in. In addition, in case the desired slot is full, CLup provides useful suggestions for other slots available at the selected store or similar less crowded stores at the indicated date and time.

Furthermore, CLup can periodically notify you of the availability of slots in the day/time range in which you usually shop.

On the other hand, supermarket managers can easily keep access data under control and effortlessly ensure compliance with safety regulations.

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

* **CLup System (or “The System”)**: refers to the whole system to be developed.
* **CLup Services (or “Services”)**: refers to the functionalities offered by the CLup System, such as the queue management mechanism and the booking service.
* **CLup Application (or “The Application”)**: refers to the application that makes CLup Services available everywhere.
* **QR** **Code**: quick response code, a type of matrix barcode.
* **Reservation ID**: a code that unequivocally represents either a position in the queue or a slot reservation.
* **Physical Ticket Dispenser**: a computer connected to the CLup System that distributes paper tickets. It acts as a proxy for Guests.
* **Guest**: a person who has not access to the CLup Application but still uses the CLup Service to access stores through Physical Ticket Dispensers.
* **Customer**: a person that uses the CLup Application and its services to access stores and book visits.
* **Client (or Store Client)**: either a Customer or a Guest.

### 1.3.2 Acronyms

* **RASD**: Requirement Analysis and Specification Document.
* **DD**: Design Document
* **UML**: Unified Modelling Language.
* **API**: Application Programming Interface.
* **PTD**: Physical Ticket Dispenser.
* **GPS**: Global Positioning System.
* **API**: Application Programming Interface.
* **ETA**: Estimated Time of Arrival

### 1.3.3 Abbreviations

* **[R.i]**: i-th requirement.

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Summary** |
| 1.0 | --/--/2020 | Leoni Luca  Locarno Silvia  Minotti Luca | First release |

## Reference Documents

* Specification document: Project Assignment A.Y. 2020-2021.pdf
* RASD of CLup
* Software Engineering 2 course slides
* Previous project examples:
* Specification document: Project Assignment A.Y. 2019-2020.pdf
* DD to be analyzed.pdf
* IEEE Standard on Requirement Engineering (ISO/IEC/IEEE 29148)

## 1.5 Document Structure

This document is structured as follows:

1. ***Introduction*** - A general introduction of the system-to-be. It aims at giving general, but exhaustive, information about what this document is going to explain.
2. ***Architectural Design -*** An overview of the high-level components and their interactions, with a focus on both static and dynamic view, helped by diagrams.
3. ***User Interface Design -*** A representation of how the User Interface will look like.
4. ***Requirements Traceability* *-*** An explanation about how the requirements defined in the RASD map to the design elements defined in this document.
5. ***Implementation, Integration and Test Plan* *-*** Identification of the order in which the sub-components of the system should be implemented, integrated and tested.
6. ***Effort spent*** - Effort spent by all team members shown as the list of all the activities done during the realization of this document.
7. ***References*** - References to documents that this project was developed upon.

**Chapter 2**

# Architectural Design

## Overview

The figure below represents a high-level description of the main components which constitute the System. They are organized in a 4-tier architecture thus facilitating maintainability and scalability. In particular, the *Web Server* and the *Application Server* have been separated mainly for security reasons. Further details about the architectural choices can be found in sections 2.6 and 2.7.

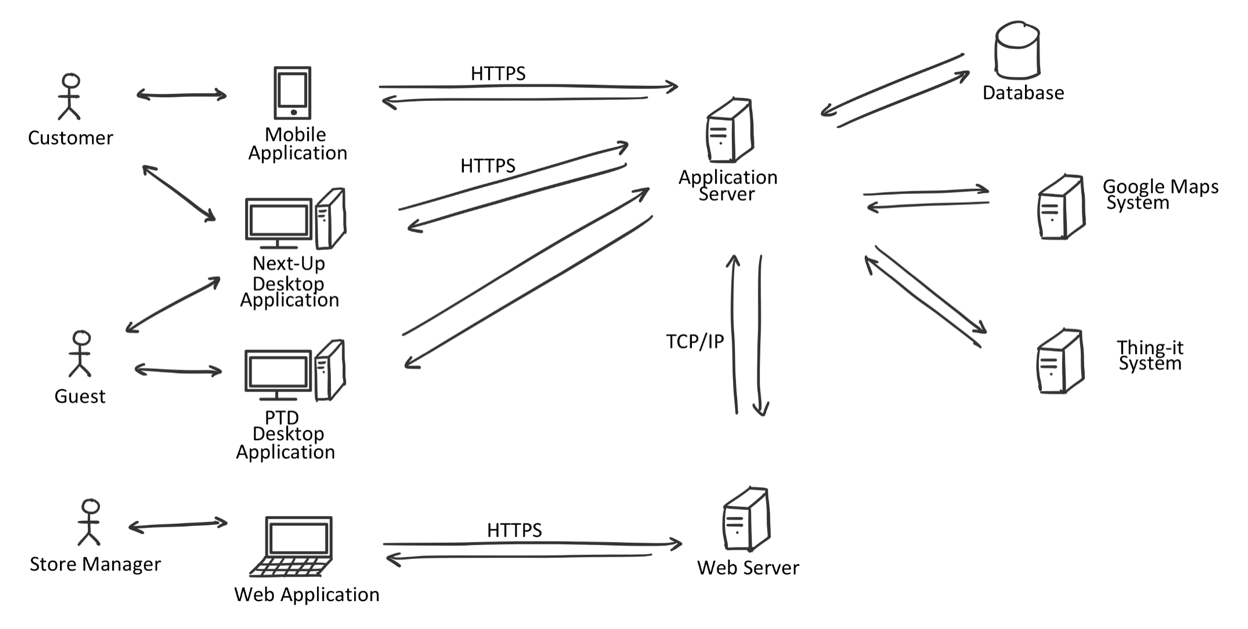


Figure 1: Overall architecture of the system

The main components of the Systems are the following:

* **Web Application**

A web application accessible through the Store Manager’s browser that allows him to access CLup Services. The web app will work with the most modern internet browsers, which communicate with the System by sending requests to the *Web Server*.

* **Customer Mobile Application**

An application installed on the Customer’s smartphone that allows him to use CLup Services anywhere. It communicates with the System forwarding every request to the *Application Server*. The mobile application will be available for both iOS and Android devices.

* **PTD Desktop Application**

An application installed on a touchscreen device located near the entrance to each store that allows Guests to use the “Line Up” Service. It communicates with the System forwarding every request to the *Application Server*. This application will be available for both macOS and Windows devices.

* **Next Up Desktop Application**

An application installed on a device located near the entrances of each store that allows both Customer and Guests to know which Reservation IDs are allowed to access the store. It communicates with the System through the *Application Server*. This application will be available for both macOS and Windows.

* **Web Server**

It is the back-end component of the *Web Application* that communicates with the Store Manager’s browser on one hand and with the components of the *Application Server* on the other (i.e. the Business logic tier).

* **Application Server**

It is the main back-end component of the System on which the business logic of the application takes place: it elaborates the requests coming from the end-user applications, it interacts with the data layer and the web layer and it communicates with the various external systems.

* **Database**

It is the component responsible for data storage. It can only be accessed by the Application Server.

* **External Systems**

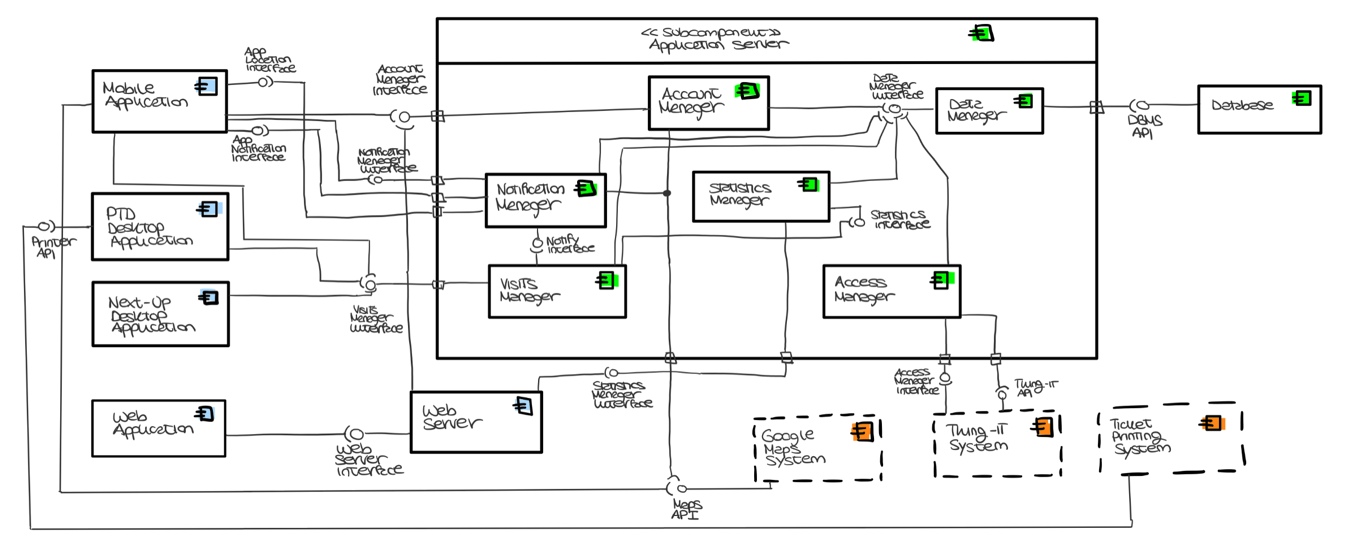
These are systems that provide functionalities that are not internally developed. *External Systems* interact mainly with the *Application Server*.

* **Google Maps System**: this system is responsible for providing the Map Services necessary for computing the ETA and supporting the identification of stores close to a customer.
* **Thing-it System**: this Smart-Office System allows CLup to remotely control the sliding doors of the store and communicate with the QR code readers.
* **Ticket Printing System**: this system is responsible for printing paper tickets. It’s directly connected to the device running the *PTD Desktop Application.*

## Component View

In this section, every high-level component is analysed in terms of its subcomponents. External Systems, such as *Google* *Maps System, Thing-it System* and *Ticket Printing System*, are presented as black-boxes that expose only the interfaces used by CLup.

Further details about the component interfaces can be found in section 2.5.



Component Diagram 1

### 2.2.1 Application Component

The *Application Component* is the front-end of the system. Its purpose is to render the graphical interfaces (detailed in section 2.3) that allow the end-user to take advantage of the CLup Services.

The only logic incorporated in the *Application Component* is the ability to perform basic checks, such as the detection of incomplete forms. Indeed, all the computational tasks are carried out by the *Application Server* and, for this reason, the *Application Component* must be able to continuously communicate with it.

As shown in the figure above, the *Application Component* consists of the following sub-elements, already presented in section 2.1:

* **Customer Mobile Application**

It is the application dedicated to Customers that allows them to line up directly from their smartphone, book visits, access their QR Codes and receive periodic notifications.

* **Next-Up Desktop Application**

It is the application that runs on a device whose screen is visible from outside each store. It shows which Reservation IDs are allowed to access that store at any given time. Both Guests and Customers can verify whether it is their turn to enter the store via this application.

* **PTD Desktop Application**

It is the application dedicated to Guests that allows them to line up directly from the store and receive a paper ticket.

* **Web Application**

It is the application dedicated to Store Managers that allows them to keep access data under control and possibly modify the store information.

### 2.2.2 Web Server Component

A *Web Server* is required in order to provide the *Web Application* for the Store Managers. This component receives HTTPS requests from the Store Manager’s browser and forwards them to the *Application Server* to collect all data required for generating the dynamic web pages.

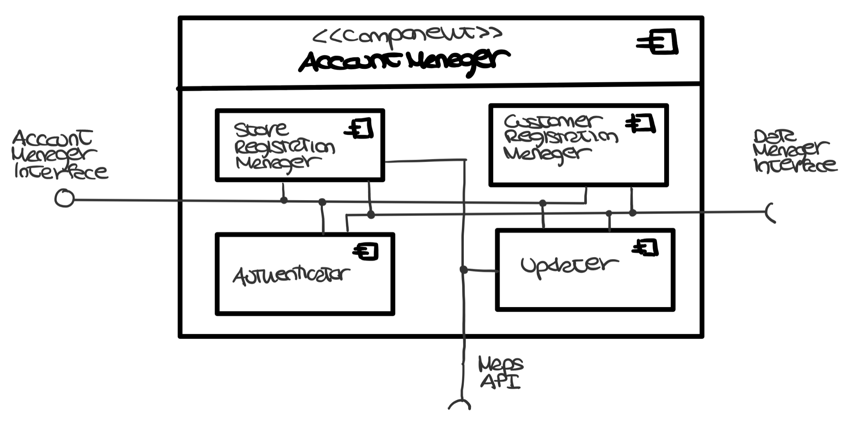
### 2.2.3 Application Server Component

The *Application Server* realizes the business logic, its role is to compute all the needed data and coordinate the flow of information between the application layer and the data layer.

As shown in the component view, the *Application Server Component* consists of the following sub-elements:

* **Account Manager**

It handles all the operations related to accounts of Store Managers and Customers. It communicates with the *Data Manager* in order to verify, access, store and delete account information. Indeed, this component is responsible for account creation and for the authentication process.

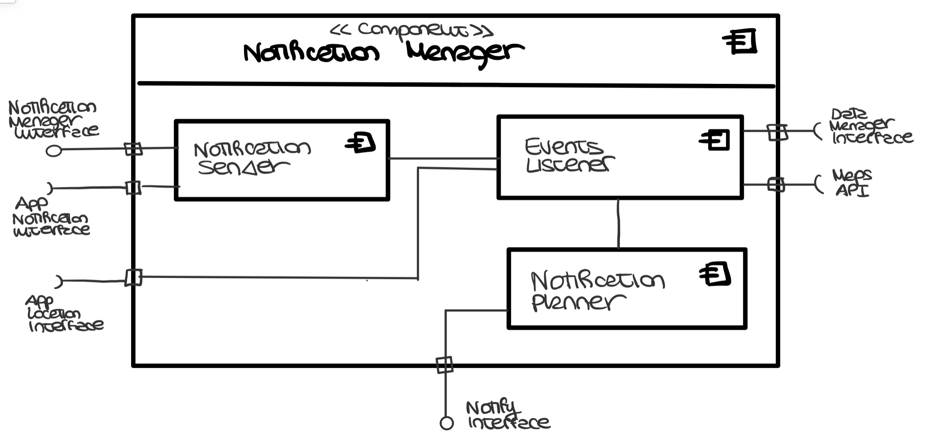


Sub-Component Diagram 1: Account Manager

It consists of the following subcomponents:

* **Store Registration Manager***:* it manages the registration of new stores and, in particular, it checks that the indicated Store ID is not already taken. It also interacts with the *Google Maps System* to obtain the coordinates of a store from its address.
* **Customer Registration Manager***:* it manages the registration of new Customers.
* **Authenticator**: this subcomponent handles the authentication by checking the user’s credentials.
* **Updater:** it is responsible for account data updates and account deletion.
* **Notification Manager**

It handles notifications to customers. Specifically, it exposes an interface to the *Mobile Application* that allows changing notification preferences. A further interface is made available to the *Visit Manager* to allow the registration of new events that will trigger the sending of notifications. To manage these events, the component needs to interface with the *Data Manager* and the *Google Maps System.*

****

Sub-Component Diagram 2: Notification Manager

It consists of the following subcomponents:

* + **Notification Planner***:* it manages the registration of the events that will trigger the sending of notifications.
  + **Notification Sender:** it takes care of actually sending notifications to the *Mobile Application.*
  + **Events Listener:** it takes care of detecting the occurrence of the events that trigger the sending. It also manages the functionality of periodic notifications.
* **Visits Manager**

It manages all the operations related to visit-scheduling. It communicates with the *Data Manager* in order to access, store and delete visit information. This component also interacts with the *Notification Manager* to notify the Customer when it is time to reach the store.



Sub-Component Diagram 3: Visits Manager

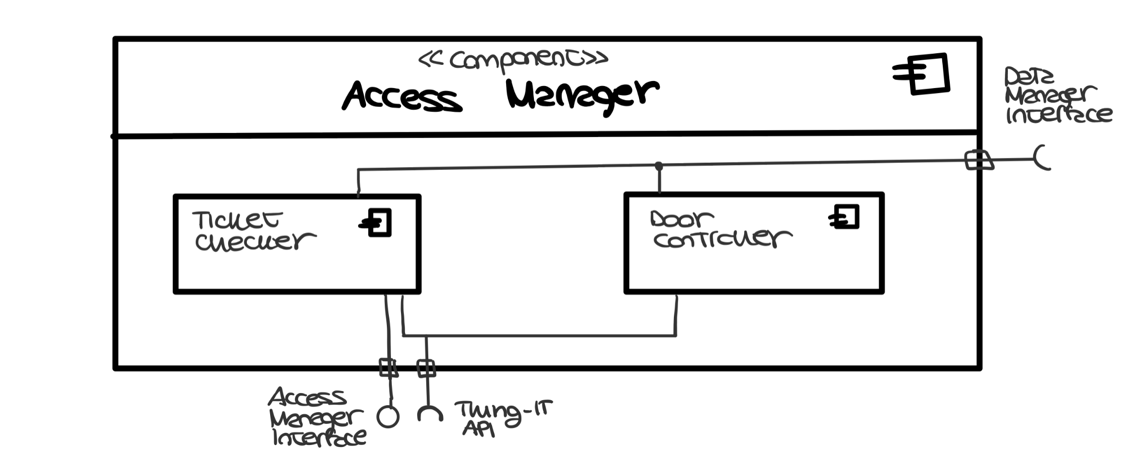
It consists of the following subcomponents:

* **Visit Scheduler:** it plans all the visits possibly assigning them an expected duration calculated by the *Statistics Manager*.
* **Waiting Time Estimator:** it computes the estimated waiting time based on the scheduled visits stored in the database
* **Statistics Manager**

It is responsible for computing the access statistics based on the data stored in the database, which is accessed through the *Data Manager.*

* **Access Manager**

It manages all the access requests to the stores. In particular, it receives the data coming from the QR Code readers and verifies whether the ticket is allowed to access or not. Depending on this, it interacts again with the *Thing-it System* to control the status of the sliding doors.

****

Sub-Component Diagram 4: Access Manager

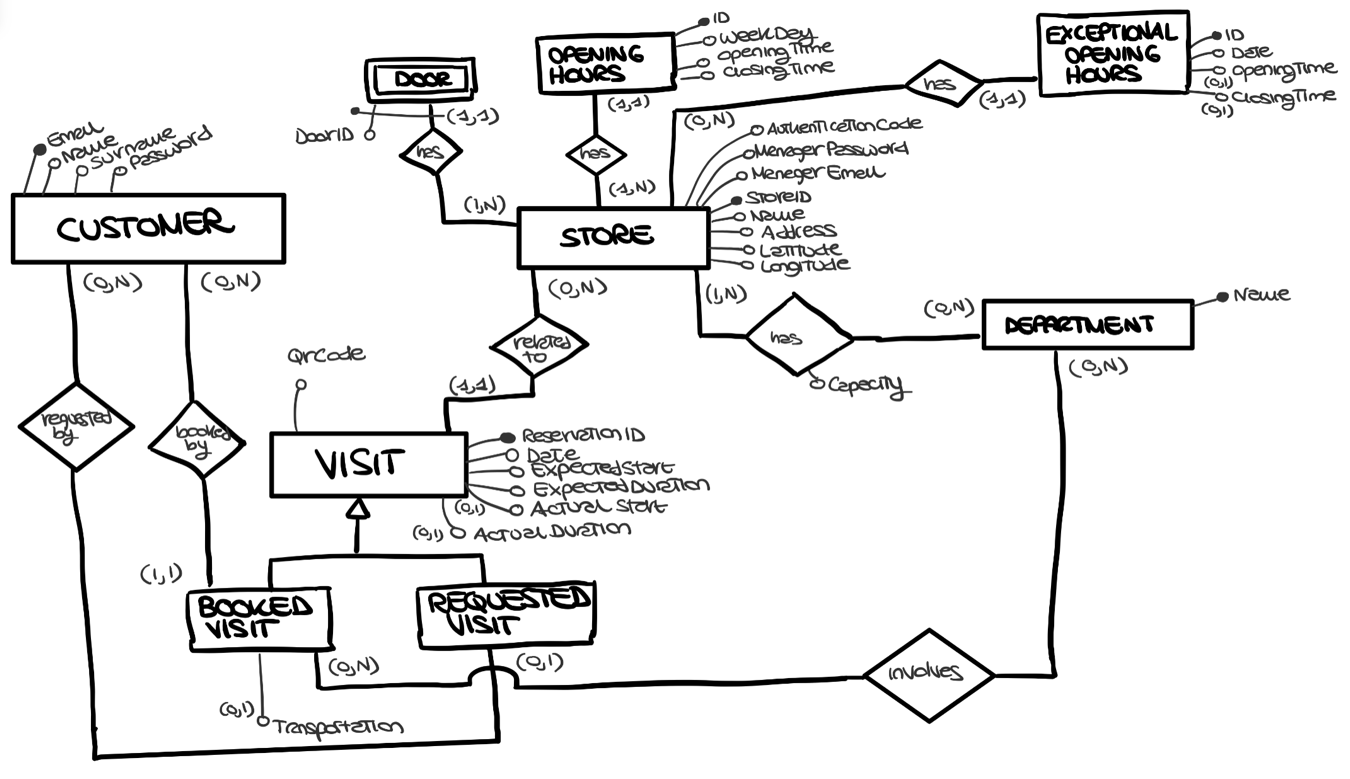
It consists of the following subcomponents:

* **Ticket Checker**: it checks whether the read QR Code is associated with a visit scheduled for the current date and time.
* **Door Controller**: it manages the requests to the Smart-Office System for opening and closing the doors
* **Data Manager**

It is the only component that interacts with the data layer. Indeed, it receives requests coming from the other components for reading, storing and deleting data from the database.

### 2.2.4 Data Components

The data layer is composed of a single relational database, which is managed by a DBMS that optimizes and performs queries. The image below represents the Entity-Relationship (ER) diagram of the database.



CLup Entity-Relationship Diagram

Relevant aspects of the ER schema are the following:

* The *Visit* entity refers to all the scheduled visits, both those booked in advance and those requested through the “Line Up” Service. Both are assigned a Reservation ID, a QR Code and a time slot possibly estimated by the system. The actual entry and exit time data are also stored to allow the computation of statistics. Finally, any information relating to the product categories in which the Customer is interested is also stored. If this information has not been specified, the System will assume during the computation that all departments are visited. Note that the visits requested by the Guests will not be connected to any Customer.
* The *Store* entity refers to the stores registered to CLup. For each of them, the data relating to opening hours and departments are saved. Other information stored are the coordinates, to allow CLup to retrieve the stores close to the Customer, and an authentication code provided by the Store Manager. The latter allows CLup to take control of doors and code readers remotely via the *Thing-it System* (further details will be provided in subsection 2.2.5). The data relating to these devices are also stored in the CLup database.

From the previous ER schema, it is possible to derive the following logical model:

**Customers** **(** Email, Name, Surname, Password **)**

**Store** **(** StoreID, Name, Address, Latitude, Longitude, ManagerEmail,

ManagerPassword, AuthenticationCode **)**

**Doors** **(** StoreID, DoorID**)**

**OpeningHours (** ID, WeekDay, OpeningTime, ClosingTime, StoreID **)**

**ExceptionalOpeningHours (** ID, Date, OpeningTime\*, ClosingTime\*, StoreID **)**

**Departments (** Name, Keywords **)**

**StoreDepartments** **(**StoreID, DepartmentName, Capacity**)**

**Visits** **(** ReservationID, QrCode, Date, ExpectedStart, ExpectedDuration, ActualStart\*,

ActualDuration\*, Transportation\*, StoreID, CustomerEmail\***)**

**BookedDepartments** **(** DepartmentName, ReservationID **)**

### 2.2.5 External services and infrastructures

## 2.3 Deployment View

## 2.4 Runtime View

## 2.5 Component interfaces

## 2.6 Selected Architectural Styles and Patterns

## 2.7 Other design decisions

**Chapter 3**

# User Interface Design

## 3.1 UX Diagrams

## 3.2 User Interface Mockups

**Chapter 4**

# Requirements Traceability

**Chapter 5**

# Implementation, Integration and Test Plan

## 5.1 Development Process

## 5.2 Implementation

### 5.2.1 Implementation Plan

### 5.2.2 Implementation Choices

## 5.3 Integration

### 5.3.1 Entry Criteria

### 5.3.2 Elements to be integrated

### 5.3.3 Integration Sequence

## 5.4 Test Plan

**Chapter 6**

# Effort Spent

The following tables summarize the effort spent by each member of the team to create the DD document.

## 6.1 Leoni Luca

|  |  |
| --- | --- |
| **Description of the task** | **Hours** |
|  |  |
|  |  |
|  |  |

## 6.2 Locarno Silvia

|  |  |
| --- | --- |
| **Description of the task** | **Hours** |
|  |  |

## 6.3 Minotti Luca

|  |  |
| --- | --- |
| **Description of the task** | **Hours** |
|  |  |

**Chapter 7**

# 7 References

* E. Di Nitto. Lecture Slides. Politecnico di Milano.
* E. Di Nitto. Project Assignment AY 2019-2020. Politecnico di Milano.
* ISO/IEC/IEEE 29148:2011. Standard on requirement engineering. https://standards.ieee.org/standard/29148-2011.html.