POLITECNICO
MILANO 1863

CLup project Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu

The Implementation and Test deliverable

Deliverable:	ITD
Title:	The Implementation and Test Deliverable
Authors:	Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu
Version:	1.0
Date:	8-2-2021
Download page:	https://github.com/Furcanzo/BurattiIncesuluHrvoj
Copyright:	Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

Contents

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

List of Figures

List of Tables

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

1 Introduction

1.1 Purpose

1.1.1 Description of the Proposed System

The project "CLup - Customer Line up" is a line spot reservation system planned to be used by managers, clerks, and customers of many local vendors and chains. The system proposes a handy solution for the ongoing issue of proper social distancing management, particularly in grocery shopping, by providing assistance to cope with the customer load for managers and helping customers access stores in a safe and controlled manner.

In particular, users will be able to see the stores, get a line number, and book in advance for the grocery stores they would like to visit. Once assigned a line number, the customer will track the estimated time of arrival of the line and wait for the notification that informs about his or her line's forthcoming arrival; hence waiting time in the line in the crowd is minimum. Also, "CLup" provides uniquely generated QR codes per the line number, which can be utilized by the store managers as a proper monitoring tool in the entrances and exits of the locations. The product's general purpose is to keep the congestion levels in line with the stores at a minimum via providing useful features for all the users.

1.1.2 Goals

- G_1 Regulate the influx of people in the stores.
- G_2 Avoid people lining up in front of the stores in order to reduce the spreading of COVID-19.

1.2 Scope

The system is expected to implement the following features, which are detailed in the future sections of this document:

- **Generate Line Number**: The customers of the store can generate a line number for themselves as soon as possible to visit a store.
- **Schedule Line Number**: The customers can generate a line number for a future planned visit. The system thus will allocate them a line number according to the time slot the customer has selected and it will send an email to the customer if the line number is canceled for a reason. From the system, the customers can view the store's location and plan their route accordingly. While scheduling their line numbers, the customers can see an occupancy forecast for the store during time slots so that they can further decide on visiting the store. If added by the manager, the customers can instead prefer to visit a partner store.
- **Manage Store**: The managers in the store can update the information regarding the store, which are the name and the location of the store. Furthermore, the managers can use this feature to activate or deactivate additional features provided by the system to manage customers more effectively. They can update the timeout interval for a ticket to expire, to manage the queue in front of the store. They can assign more managers and clerks if necessary to use the system to its full extent. They can administer the time slots for their customers by setting the opening and closing hours, with possible breaks in between. They can administer the partner stores of their stores if they exist, for their customers to select instead. The stores are to be created by the Back Office User, while granting access to the first manager.

The system is expected to be delivered in two parts, a client application, and a server. Given its functionality is depending on the current ongoing pandemic, the app is expected to be implemented

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

before the COVID-19 crisis ends, which is estimated as the date general public receives vaccination as the end of July 2021. For this, the client application is expected to be implemented as an app running on a browser, increasing the range of devices being able to run the app, without increasing the development costs, which can then be ported to a mobile application if needed. The server is expected to serve the client application to its users and handle any transaction incoming. To implement the location-specific features of the system, due to the complexity of those, namely location finding, addressing, and route planning tasks, an external Maps API is used.

1.2.1 Targeted Users

"CLup" aims to resolve the problem of Customers queueing up in front of a store. It aims to provide this by automating the availability for visits and improve contact tracing by the managers. It is aimed towards users of all demographics.

Customer:
Customers will be able to obtain specific line numbers for various stores using CLup, which they can track the estimated time available with and also view the location on a map application to plan their visit. Customers can further obtain line numbers for future visits, based on the system's info about the availability of free spots on the specific time intervals. They may prefer to visit a different branch of the same chain. Furthermore, they can provide specific product categories they intend to purchase or set an estimated time for their visit to allow finer granularity. Some customers may also prefer to obtain a line number upon visiting the store physically. To plan their visit in a time slot where the store will be less crowded, the users can see the store's occupancy based on already taken line numbers and forecasts provided by the system.

Clerk:
The clerk (which can be a shopping assistant or a security detail) can monitor customers' flow and manually intervene in case of missing line numbers via performing manual checkout for a specific customer or by printing line numbers physically for some customers.

Manager:
The store manager (which could be an actual manager or someone responsible for handling customer management) can provide details regarding the availability of products and the store in general, by setting the opening hours, maximum allowed customers in the shop, in-shop location and maximum customer capacity of different product categories, the maximum amount of reservations that can be made per customer, line number timeout and the location. Also, for chains and relevant stores, the manager can add chain members for the store. **Back Office User:**
In order to use the CLup application, store or business owners should contact CLup Back Office User who will give them permission to use the application and create a Manager account for them.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- *Store*: The physical location of the business that uses the line reservation system
- *Manager*: A person in charge of executive action within the store
- *Customer*: A person to visit the store
- *Clerk*: A person in charge of handling the entrance and exit of customers. This user can be replaced by a QR code totem or a similar device for customers to directly interact with instead.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

- *Visit Time*: The time interval in which a customer performs a visit to the store
- *Back Office User*: A user assigned by the company hosting the system to grant manager access to certain users.
- *Line Number*: A number that indicates the ordering of a specific customer in the line
- *Line Number Ticket*: A physical ticket printed that features the line number and the QR code.
- *Time Slot*: Specific intervals of time in which a customer can schedule their visits determined by the opening hours.
- *Partner Store*: A different store included in the same beneficiary chain of command (such as another member of the franchise or store chain) or a mutual agreement with the specific store
- *Product*: Any item, items, service or services demanded by the customer, and provided by the store to the customer.
- *In-store Location*: A location of a specific product category inside the store.
- *Working hours*: The time intervals that the store is open during each day.
- *Maps API*: A third-party mapping service implementation used for location tracking.
- *SSO Provider*: A third-party service providing SSO authentication.
- *ID Token*: A token provided by the SSO provider to verify the user's identity from the server.

1.3.2 Acronyms

- **RASD**: Requirement Analysis and Specification Document
- **QR Code**: Quick Response Code
- **API**: Application Programming Interface
- **SMTP**: Simple Mail Transfer Protocol
- **HTTP**: Hyper-Text Transfer Protocol
- **REST**: REpresentational State Transfer
- **JDBC**: Java DataBase Connectivity
- **DBMS**: DataBase Management System
- **JWT**: JSON Web Token
- **ACID**: Atomicity, Consistency, Isolation, Durability
- **ORM**: Object-Relational Mapping

1.3.3 Abbreviations

- G_n: n^{th} goal
- D_n: n^{th} domain assumption
- R_n: n^{th} functional requirement
- F_n: n^{th} function

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

1.4 Revision history

Table 1: Revision History

1.5 Reference Documents

- Specification Document: Implementation Assignment AY 2020-2021

1.6 Document Structure

This document is composed of seven sections, each with the purpose described below:

- **Introduction**: This section provides an introduction of the problem, the scope of the project with details regarding the goals, target users and phenomena. The goals of the project are formulated in accordance with the description of actions and actors in the Design Document. Within the project's scope, properties and duties of different users and user groups are described in Target Users section. Furthermore, under the scope of the project, the relevant phenomena of the project is presented through their relevance to the world and the machine.
- **Implemented Features**: This section provides a list of implemented functions with motivations for including them.
- **Adopted Development Frameworks**: This section provides a list with used frameworks, programing languages as well as additional APIs used in the implementation of the CLup application.
- **Source Code**: This section provides the short overview of the source code, and shows how the code is structured.
- **Testing**: This section describes functional tests that were used for testing the CLup, such as unit tests, integration tests and end to end tests.
- **Installation Instructions**: This section describes how to install the CLup application.
- **Effort Spent**: This section features the effort table, in which all team members provide a rough estimation of the time spent on the creation of the various sections of the document.
- **References**: This section features different reference materials referred to inside this document.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

2 Implemented Features

This section explains all implemented requirements of the CLup application alongside with motivations for including these specific featurs.

2.1 Create Store

Create store function is used by the Back Office User to create the first instance of a new store that joins the CLup application. It is a part of the BackOfficeService class which is used for generating new stores and for generating employee with the store. The store cannot exist without the employee in charge of it. The store is generated with the related pieces of information such as its name, description, location in a form of longitude and latitude, number of maximal customers and with a timeout for the line number tickets. Methods that are used to create and generate a store with an employee are createStore, generateStore, and generateEmployeeWithStore. The creation of the store is a vital function of the CLup application since the main purpose of the application is to create a line spot reservation system.

2.2 Update Store Information

Update store information is a function used by the manager to update the store information. Through the editing form, a manager can update general information about the store, with the current values already pre-filled. Besides all the information's that a store has since it was generated in the first place, it is now possible to add partner stores and working hours. Update store is also a vital function for the CLup application to work as intended. A store needs to have working hours to be able to generate line number tickets.

2.3 Book Future Line Number

Book future line number is used by the customer to book their visit to the store. It belongs to the CustomerService class and uses the method called generateLineNumber to generate new line numbers. To simplify the process of generating new line numbers, product categories have been omitted. Product categories are nice to have feature but aren't vital for the CLup application to work. Book future line number is an important function of the CLup application. Without it, the customer couldn't book a visit to the desired store.

2.4 Retrieve Line Number

Retrieve line number is a function used by the clerk or by the customer to retrieve the first available line number from the system. It's a part of a CustomerService class and it is somewhat similar to the book future line number function. Retrieve line number is used by the clerk when the customer approaches the store without a valid line number ticket and asks for the ticket to be generated by the clerk. Customers may as well use the retrieve line number function if they want to reserve the first available spot at chosen store. Retrieve line number is an important function of the CLup application as well as book future line number. They are both the core features of the CLup application in order to

2.5 Add Staff Member

Add staff member is used by the manager to generate new staff members. The manager needs to fill the relevant information such as email, storeID and the role of the new staff member. The new role of a member can either be a manager or a clerk. New staff member is created with the generateEmployee method. Generation of the new employee is important for the CLup application, but not vital. A store can function with only having one manager created by the back office user if it doesn't need clerks for scanning and creating line number tickets at the entrance of the store.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

2.6 Monitor Customers

Monitor Customers is a simple but yet effective function which can be used by the manager to simply retrieve the number of customers in the store for the moment. It uses monitorLive method from the EmployeeService class. It is not a vital function for the CLup application to work but it was easy to implement and can be handy for the manager. Since all the customers must scan their line number ticket when entering and existing store, number of customers in a moment is easily calculated.

2.7 Grant Access

Grant Access is used for scanning customers QR codes while entering and exiting the store. It has two methods defined in the EmployeeServices class. Checkin is used when the customer wants to enter the store. Customers line number status is then changed from “WAITING” to “VISITING”. When the customer exits the store, checkout method is used and customers line number status is changed from “VISITING” to “VISITED”. Grant Access is a vital function for CLup application because it ensures that the store isn’t overcrowded.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

3 Adopted Development Frameworks

3.1 Programming languages

Java language has been used for the backend implementation. One of the main reasons to use Java was it being platform independent and it being mainly object oriented. We have decided that Java is good for building web application such as CLup, given that it provides common interfaces like JPA with minimal configuration necessary.

3.2 Frameworks

We used the Java Spring framework because it offers a lot of modules that can help with the implementation. One of those is Spring Web that offers annotations for exposing rest endpoints, as well as Spring Data that offers the JPA repositories which are used to handle the JPA mapping.

For the frontend part, we have used Bootstrap framework to have a basic set of widgets to build our application. Also, we have used Hyperapp tooling to perform state management on the whole app.

3.3 Additional APIs

Google Maps API has been used in order to set the store location. Instasan library is used to automatically retrieve QR codes from scanned input streams.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

4 Source Code

The source code is divided into "src" directory containing source code related to the backend of CLup application and into "frontend" directory which contains source code for the frontend of the CLup application.

4.0.1 Backend

Under the "src" directory, there are 5 important directories to mention alongside their usage in the application itself. The controllers directory contains controllers that are responsible for processing incoming requests. They invoke a business logic class to process business-related tasks. The model directory is in charge of modeling the data and contains "dtos" and "entities" subdirectories. DTOs (Data transfer objects) are used only to transfer data from one process or context to another. Entities contain all objects that represent persistent data maintained in a database. Repositories are used to decouple the business logic and the data access layers in the CLup application. Services are used to implement the business logic of the CLup application. Each type of user has its service class. Another important file in the src directory is the ApplicationConfiguration file in which different configurations have been defined.

4.0.2 Frontend

The frontend includes a lib directory where library instascan is provided. Under src directory there exist a directory for each user type, which includes the functions to generate the view and to update the application state. Inside the directory also there are files that define update and view logic of the whole app (actions such as login, logout are handled in those). requests.ts include all the specifications for interacting with backend, while effects.ts include all the side-effects needed by the app.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

5 Testing

5.1 Functional Testing

5.1.1 Unit Testing

A unit test case is a part of code designed mainly for testing, which ensures that another part of code works as expected. To get the desired results quickly, a test framework was required. JUnit is a perfect unit test framework for the Java programming language. A formal written unit test case is characterized by a known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition. There are at least two unit test cases for each requirement – one positive test and one negative test. If a requirement contains sub-requirements, each sub-requirement has at least two test cases as positive and negative. Automatic Junit tests have been proven to be more efficient than the manual testing process. Junit tests have been created with the help of the Java Unit Test Generator for IntelliJ IDEA in order to speed up the testing process.

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved

6 Installation Instructions

In order to install and run this app:

1. Provision a MySQL DB.
2. Export the necessary environment variables by creating a .env file on project root (touch .env if you use POSIX), and adapt the values accordingly:

```
CLUP_DB_HOST="localhost"
CLUP_DB_PORT="3306"
CLUP_DB_NAME="clup"
CLUP_DB_USER="root"
CLUP_DB_PASSWORD="root"
CLUP_BASE_URL="localhost:8080"
```

3. Execute the SQL queries given in file "database.sql"
4. Run *mvnw spring-boot:run* (or mvnv.cmd if you are using Windows)

Copyright © 2021, Roberto Buratti, Hrvoje Hrvoj, Ozan Incesulu – All rights reserved