CLup

Customer Line-up

1) INTRODUCTION

A) *PURPOSE*

The purpose of this document is to build a more concrete foundation of what the system-to-be will be. It will also define the general behaviour and specific limitations of the system. This document is primarily addressed to the programmers and mostly includes technical language.

B) *SCOPE*

The scope of the design document is to define the system's behaviour in general cases and some critical scenarios, and to design the architecture of the system-to-be so as to provide a time-efficient, logical allocation of the components and the interaction between these components.

The document is not only limited to the architecture and behaviour of the components, but it also extends in part to the implementation and testing plan, where one possible course of action is explained, user interface design of user applications and requirements traceability relating to the Requirements and Specifications Document (RASD).

C) *DEFINITIONS, ACRONYMS, ABBREVIATIONS*

D) *REVISION HISTORY*

E) *REFERENCE DOCUMENTS*

F) *DOCUMENT STRUCTURE*

2) ARCHITECTURAL DESIGN

A) *Overview: high-level components and their interaction*

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

- *Presentation Layer (P)* manages the presentation logic, namely the interaction with the user. It comprises a GUI (Graphic User Interface) that makes the application's functionalities more understandable to the user.
- *Business Logic* or *Application Layer* (*A*) handles all the functions to provide to the user and manages the exchange of information between the user interface and the data source.
- *Data Access Layer* (*D*) provides access to the stored data. The implementation of the access logic should be both easy and structurally robust to guarantee a correct abstraction from the specific database and provide a model easy to use.

In order to guarantee as much flexibility and scalability as possible, the system is based on a 4-tier architecture (Client, Web Server, Application Server, Database Server) with a thin client. Since the application should be easy to use and executable in several different devices, the use of a thin client prevents a heavy computation load client side, carrying out all the heavy operation at server side. As represented in figure 2.1 the user, through a smartphone or a tablet, and the physical dispenser can directly communicate with the application server, while the store manager can access the functionalities devoted to him through a web application communicating with the web server. The use of a web server for the store manager's functionalities is based on the fact that, in general, the web apps are quicker and easier to build, maintain and update and less expensive than the mobile apps, even if slower. The application server communicates with the data server to store the needed information. A more detailed description of the architectural design is given in the next section.

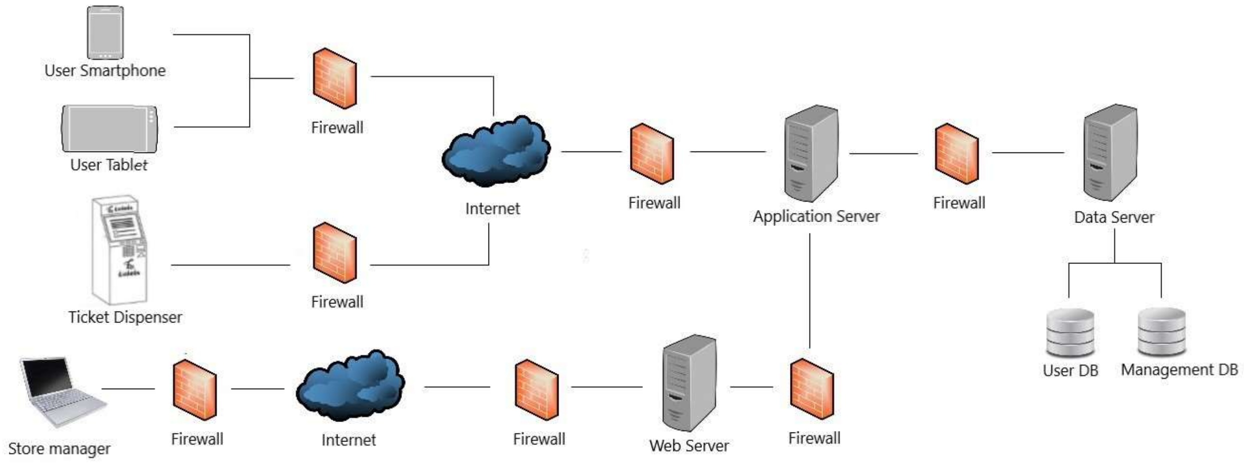
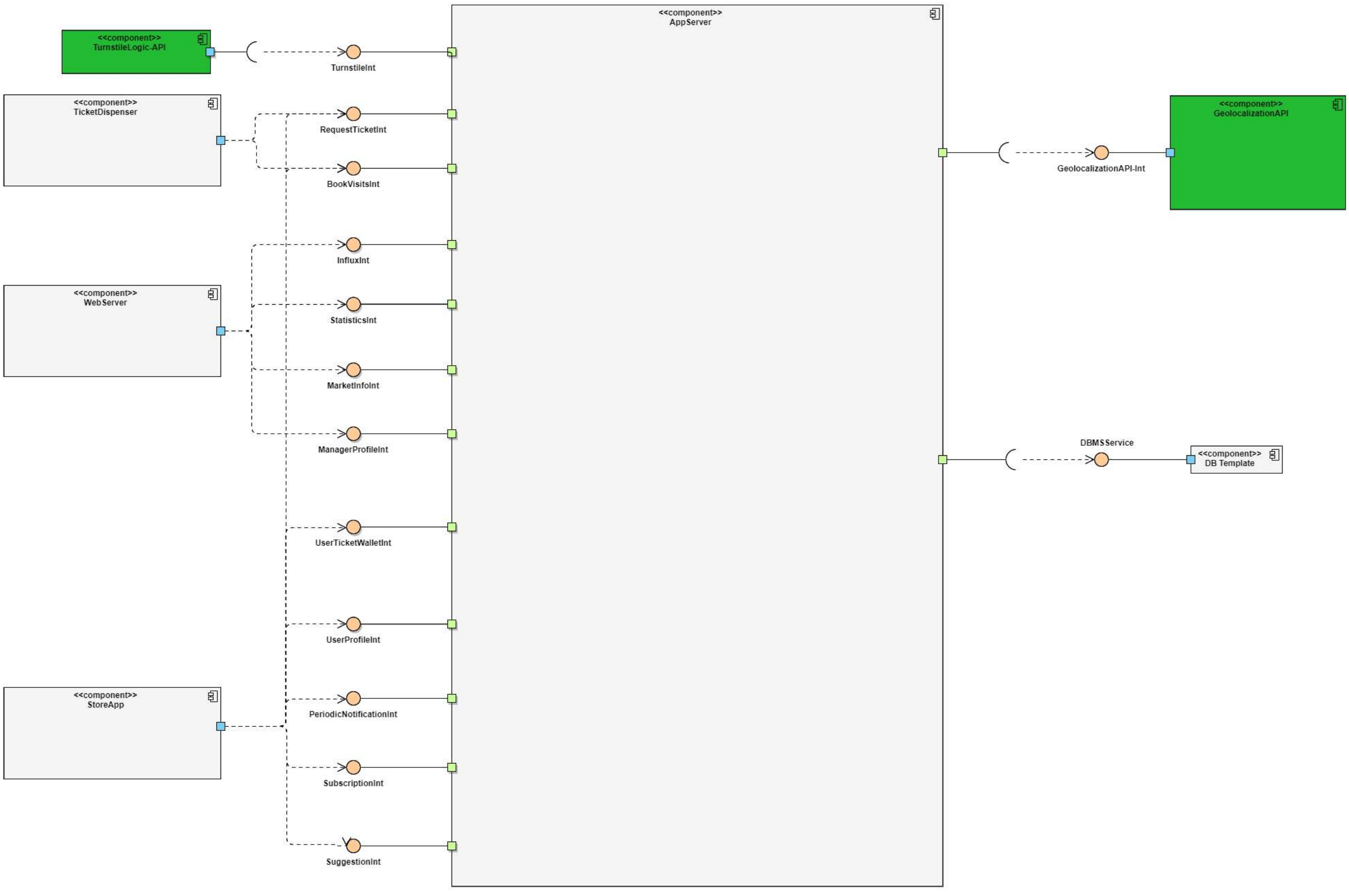


FIGURE 2.1 – HIGH LEVEL ARCHITECTURE

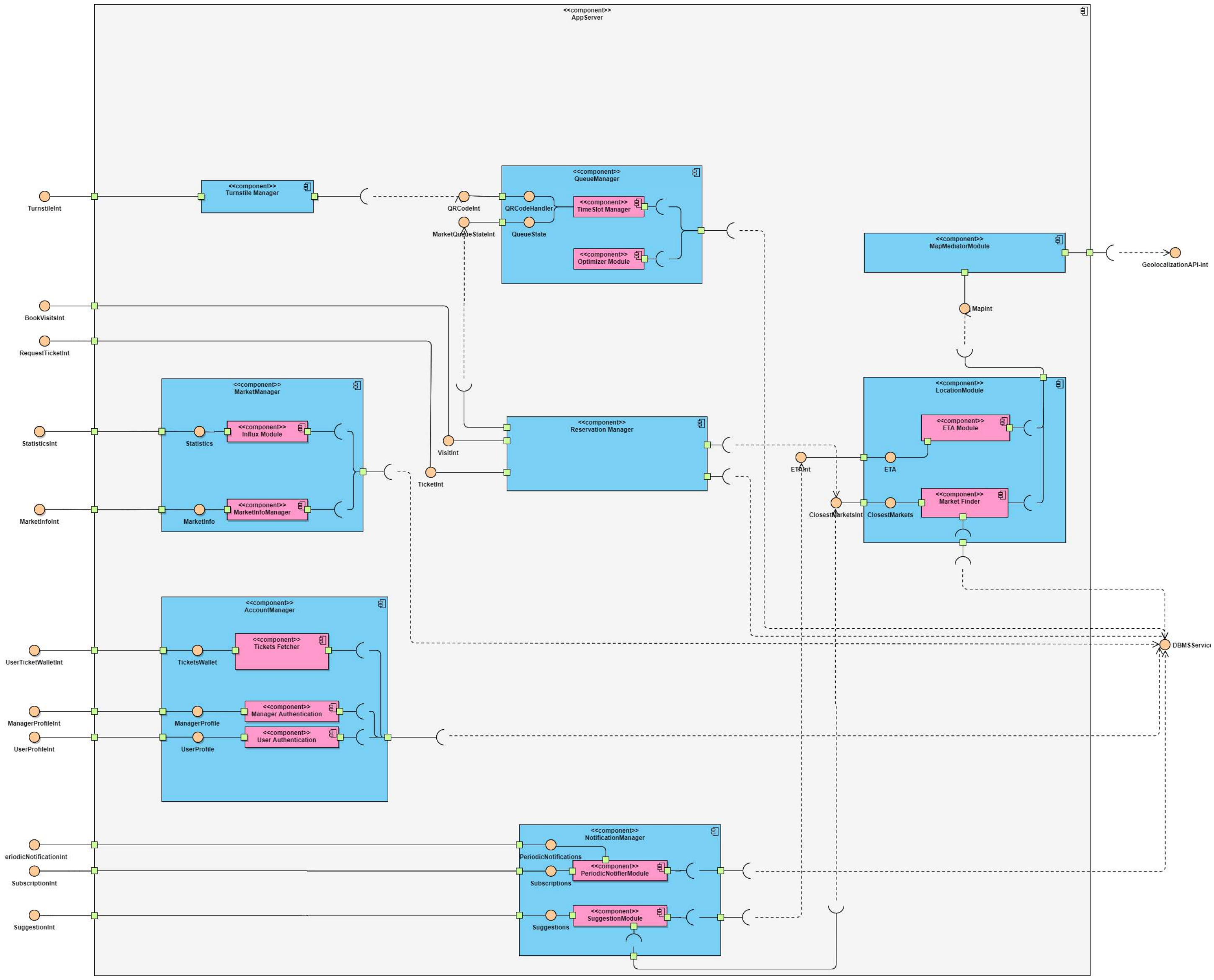
B) *Component view*

B.1) *General component view*



In the above image represents the general component representation of the system. Here are shown the interfaces with the application server needed by the other components of the system: turnstiles have an interface to handle QR codes checking; the physical dispenser interacts with the application server as it was a normal device getting a ticket, but with “restricted” degrees of freedom; Web Server will interface with application server to get all information on statistics it needs to allow the correct usage of the web app; Mobile App, having a thin client, will interface with the system to do basically everything it needs: account and ticket managing, subscribing to notifications, getting new tickets and visits. It is also shown that the application server communicates with external DBMS to maintain permanent data and with a geolocation service, that must provide services allowing retrieving distances between locations (both in terms of space and time) and retrieving info on nearby places given a type of place (in our case, shops). Google Maps is a good candidate, as it provides APIs as “Places API” that allow lookups for specific types of places, with useful parameters as “radius”.

B.2) Application Server component view



The image above shows the internal architecture of the Application Server and all the interfaces that are provided to the outside world, and all interfaces required.

Starting from what are conceptually two mediators, the system will contain two components called:

- *Map Mediator Module*
  This component's job is to communicate with the external API, by modifying the API's information so that it can be comprehensible by the Application server, and adapting the requests to the API's protocol, required by other modules of the server. It provides information regarding the local map with all the relevant stores in the map as well as an estimation of the time it takes to reach one store from a certain location.

- *Turnstile manager*
  The turnstile manager, as opposed to the previous component, provides an interface for the turnstiles so that they can be opened or closed through the scans of the right QR codes.

For the authentication of an account, be it user or store manager, the following component will be of help:

- *Account manager*
  This is the component which is responsible for the authentication of any user that wants to log in or ***register***, so it saves all the relevant information about that account and furthermore, provides to all users the possibility to fetch all upcoming tickets that they have an appointment for.

Once a user is logged in the system, this must provide functionalities to the user that allow him to book visits, get tickets, be reminded of upcoming visits and be suggested different stores in case the requested ones are full. The following components allow users to:

- *Reservation manager*
  It provides the interfaces that allow booking of tickets and visits. It communicates with different components to provide the functionalities required by tickets and visits requests i.e., check the relevant, available timeslots, gets the info that will allow the presentation layer to build map of the available shops around the user's current location, handle visits additional info…(See interface diagram for a more detailed list)
- *Notification manager*
  This component has the job of handling notifications of two types:
  -reminding users when it is time to leave their current location so to reach in time their destination; to do this, the component interfaces with the maps provider, getting info on ETA.
  -provide suggestions of relevant available stores and timeslots when the requested one is full or the user is inactive.

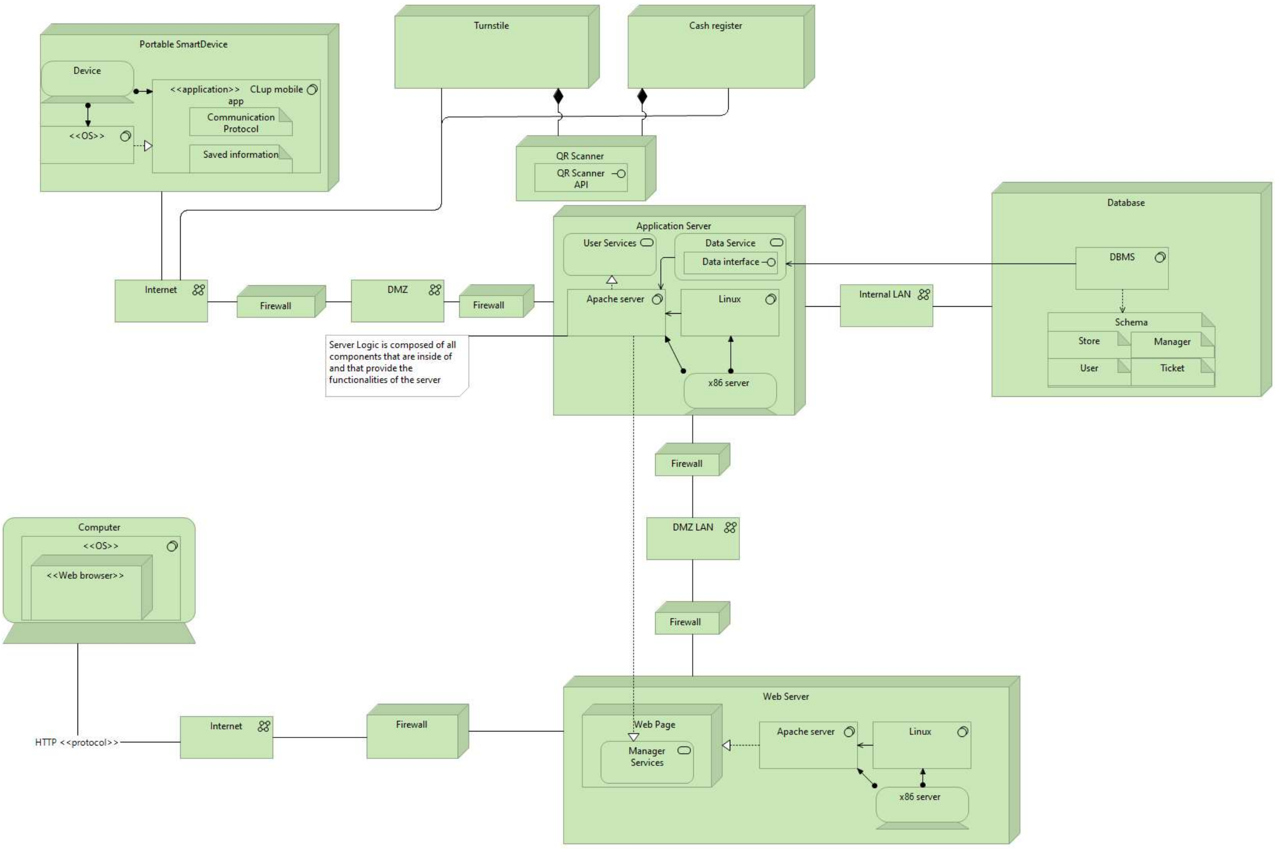
Instead, functionalities related to the store managers are provided by the following component:

- *Market Manager*
  Issues, through its interfaces, the functionalities of controlling how many customers can enter in the store, how many people are inside of said store and the statistics regarding all entrances within a certain period.

The components which connect everything together and provide the logic of the dispensing of tickets are the following:

- *Queue Manager*
  This is the most critical and important component of the system, as it defines and handles timeslots and the queue. Through the timeslot manager subcomponent, timeslots are searched to see or update their availability on new tickets requests. It also generates QRCodes for new tickets and visits, assigning them to available timeslots, and checks QRCodes on scan, so to correctly update the state of the queue in that timeslot.
  The other components is a timeslot optimizer, that is a periodic activated component, which handles aspects of the optimization of timeslots, as the maximum number of people that can enter considering info about user visits that declared their shopping list or departments they will visit; another possible optimization is the duration of timeslots, performed once in a greater period of time w.r.t. to the previous one, that could optimize the duration of timeslots based on the average shopping duration.
- *Location Module*
  It provides interfaces that allow the search of the closest stores in a range around a given location, filtering on the stores that are registered and use CLup; it also provides the ETA services, needed by other components to have an estimation of the user distance (quantified in time) from the given location. To provide these services, it interfaces with the Mediator, that will then process and forward requests to the external API.
- *Data Service*
  It provides access to the external interface of the database, doing queries and generally interacting with the database for every operation needed.

C) *Deployment view*



Some important aspects to highlight are the following:

- With Portable SmartDevice we mean any mobile device such as tablets and smartphones. With OS we intend either IOS or Android.
  In computer, any OS and Web browser can communicate with the web server as long as the HTTP protocol is being used.
- The CLup mobile app will store information in the client side such as upcoming tickets data, for faster data retrieval and user information for identification. It furthermore uses a specific communication protocol that the programmers might see fit.
- The installation of the CLup mobile app is done through the mobile phone's application store and is not been shown in the deployment view.
- The application server is connected to the database through an internal LAN with no firewalls, for increased connection speed.
- Turnstiles and cash registers need to forward information regarding QR codes through the internet.

D) *Runtime view*: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases

E) Component interfaces

F) *Selected architectural styles and patterns*: Please explain which styles/patterns you used, why, and how

G) *Other design decisions*

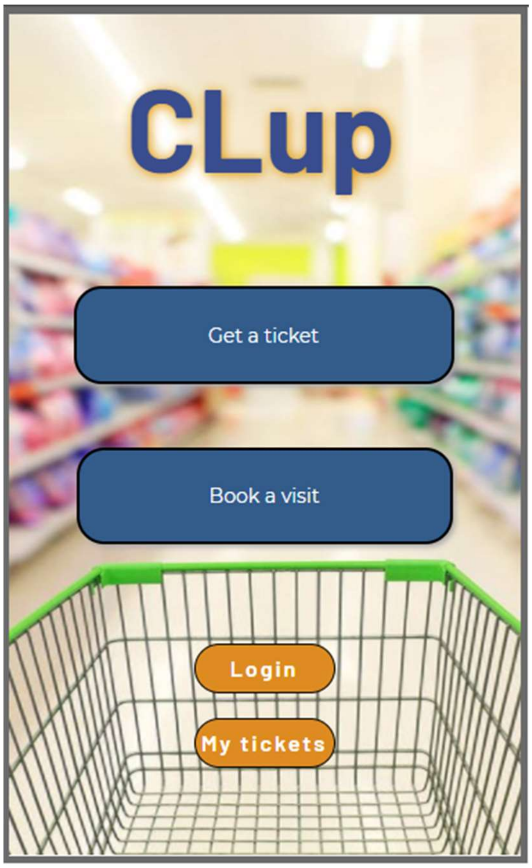
3) *USER INTERFACE DESIGN*:

Here we include some draft mockups about how the mobile applications and the web application should look like. As they are drafts, these are not to be intended as strictly constraining designs: backgrounds, styles, text formatting and page structure are simplified and rough, as we want just to give an idea of the general schema of the pages.

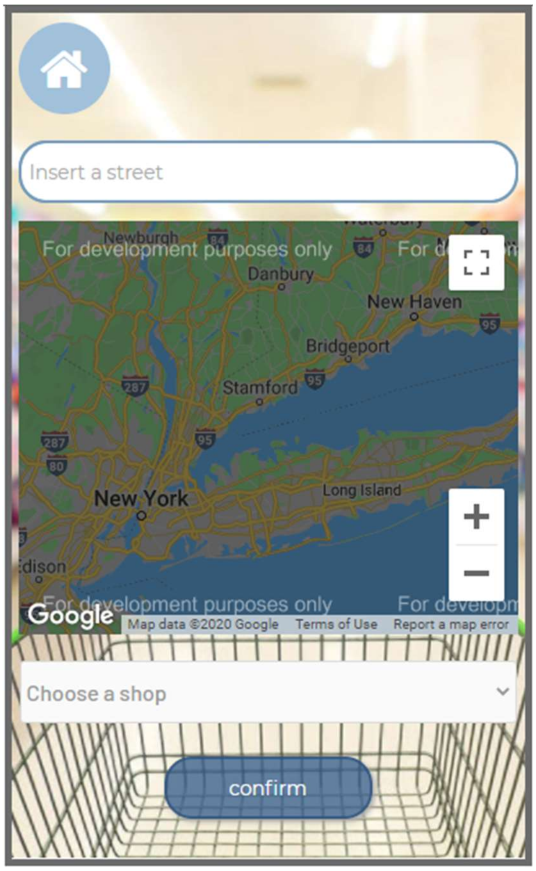
3.1) Mobile App Mockups

As already mentioned in RASD, the mobile app must have simple and intuitive interfaces, as to be easily usable by users that may not be practical with technology (as may be elderly people).

The app will allow user to define first either the shop or the date and time, in the case of "Book a visit", it is not shown in this demo because it is similar and irrelevant.

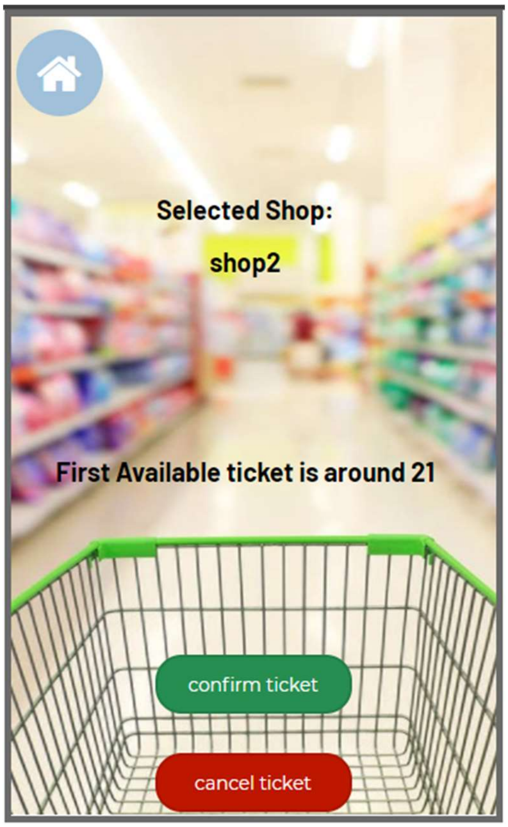


HOME PAGE OF THE APP



SHOP SELECTION

**Mockups are taken from an interactive graphic demo with graphic-driving purposes only, linked in the "reference documents" paragraph.*

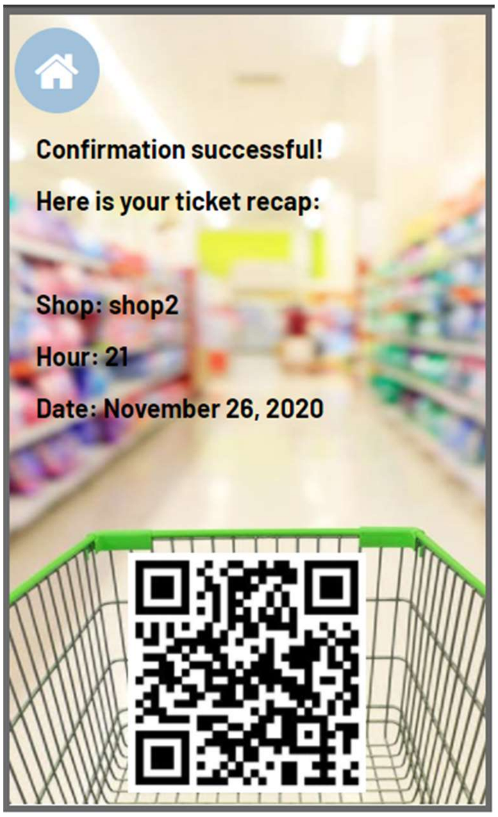


Selected Shop:

shop2

First Available ticket is around 21

TICKET PROPOSAL



Confirmation successful!

Here is your ticket recap:

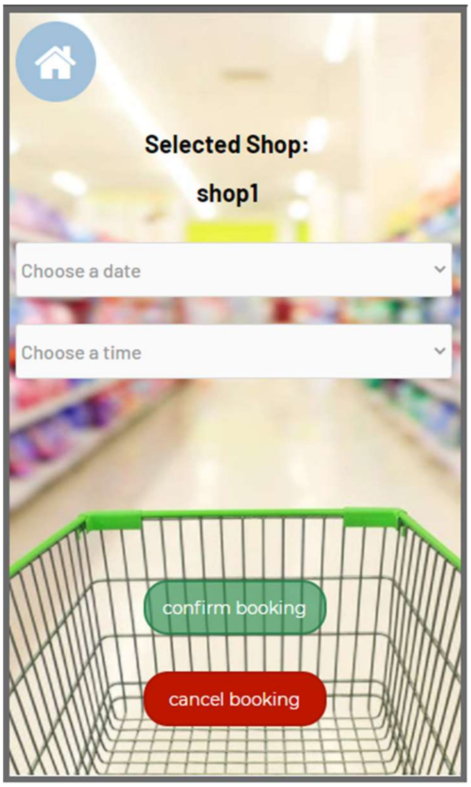
Shop: shop2

Hour: 21

Date: November 26, 2020

TICKET CONFIRMATION

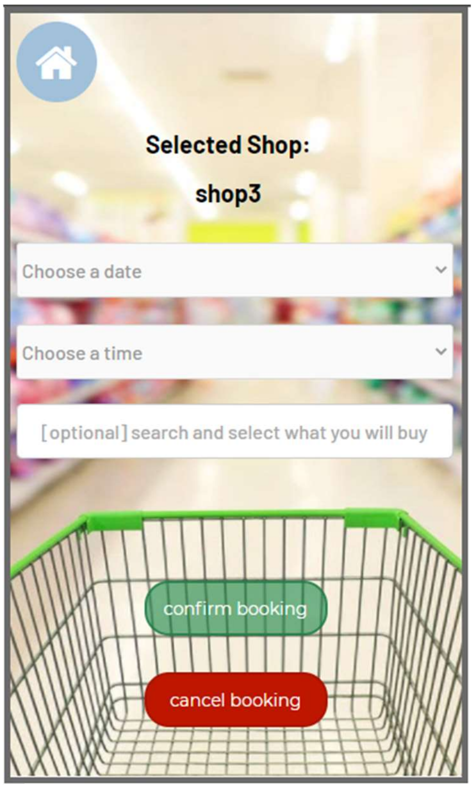
In the case of "book a visit", after the shop selection, the app will allow registered users to select their shopping list:



Selected Shop:

shop1

BOOK A VISIT AS GUEST



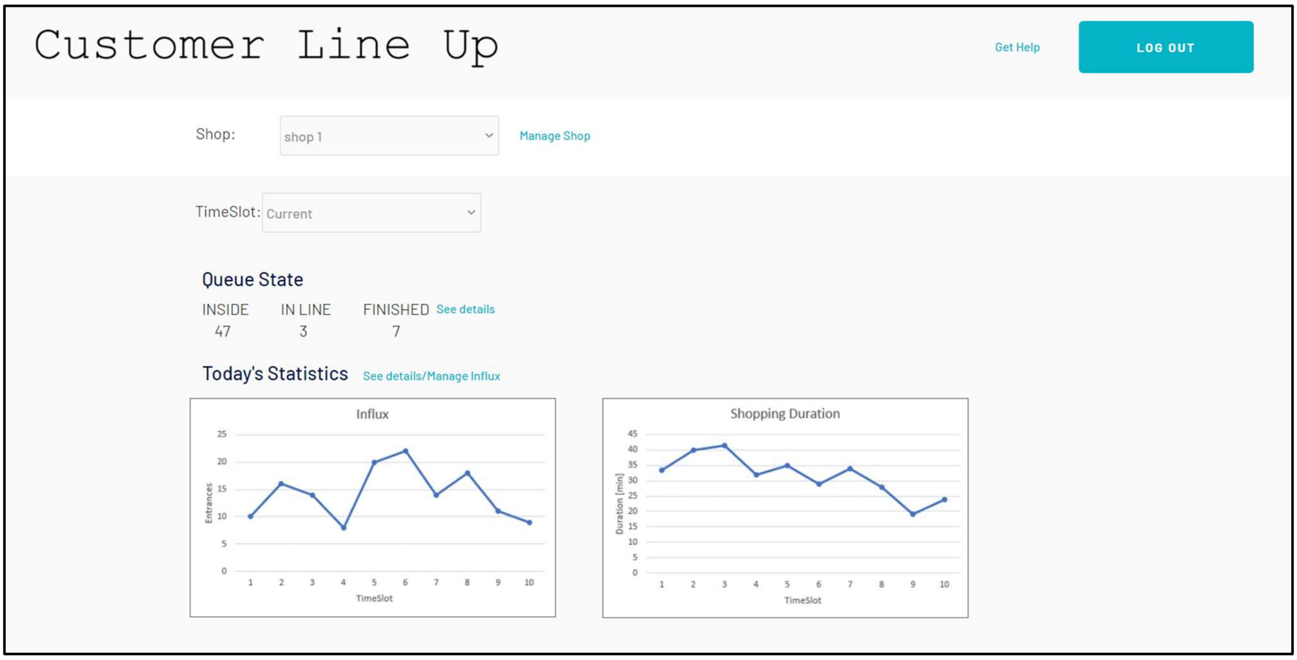
Selected Shop:

shop3

BOOK A VISIT AS REGISTERED USER

3.2) Shop Manager Web App mockups

The web app for managers will allow only to login with an authorized account. For security purposes, such account cannot be registered directly through the Web App, but it has to be set up by CLup System staff through certified communication with the market/market chain.
The web app must allow managers to trace entrances, to see the queue state of any timeslot, and to see statistics as average time of shopping and the average influx in a given period.
In the real app other statistics may be implemented, as the most declared shopping items or departments (given by user that specified these elements in their visits). Graphics in the real app may be interactive, allowing more interactivity, usability and utility.


Customer Line Up

Shop: shop 1

TimeSlot: Current

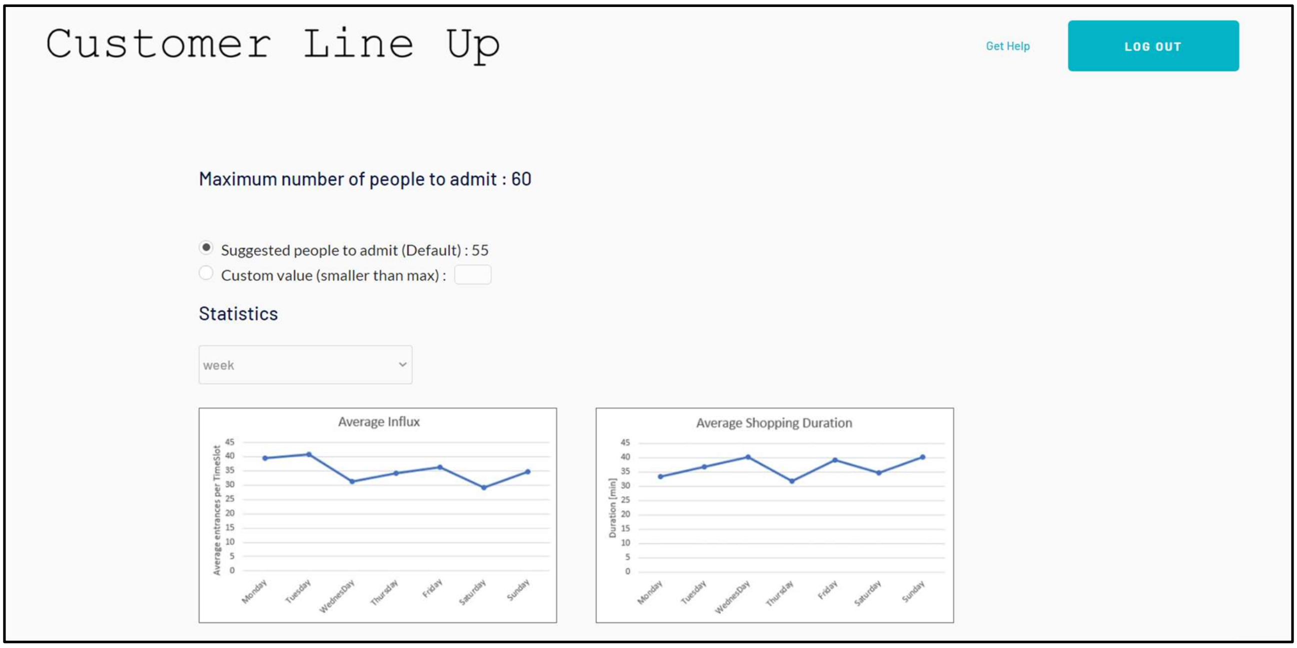
Queue State

INSIDE	IN LINE	FINISHED
47	3	7

Today's Statistics

WEB APP HOMEPAGE

The "Manage Shop" page is not shown here, it simply allows managers to add or update logistics info for the shop, as product positions or departments. It just lists departments and their items.


Customer Line Up

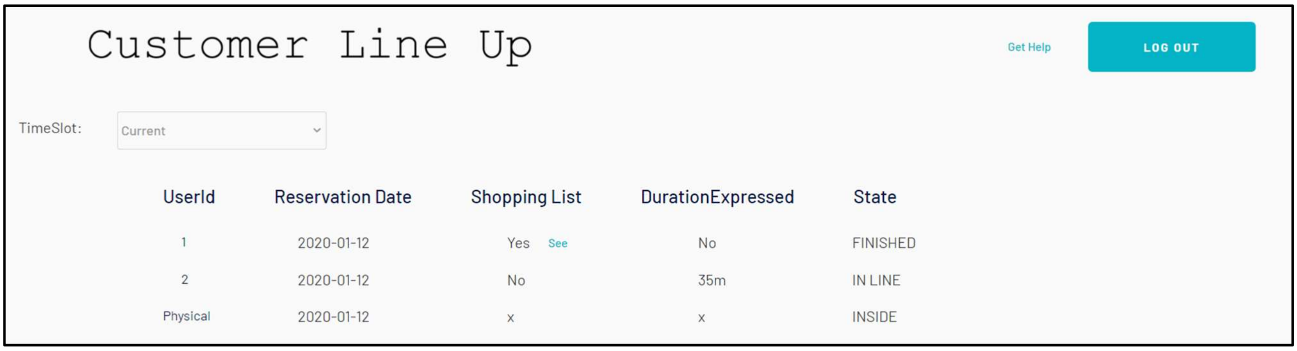
Maximum number of people to admit : 60

● Suggested people to admit (Default) : 55
○ Custom value (smaller than max) :

Statistics

week

WEB APP STATISTICS PAGE



TimeSlot: Current

UserId	Reservation Date	Shopping List	DurationExpressed	State
1	2020-01-12	Yes See	No	FINISHED
2	2020-01-12	No	35m	IN LINE
Physical	2020-01-12	x	x	INSIDE

Web App Queue State

The page shown upon clicking the "See" button next to the declared shopping list is not shown here, since it is unnecessary as it simply shows the list of declared items.

As shown in the image, physical users have no Id, but they still are tracked and considered correctly in the queue state.

4) REQUIREMENTS TRACEABILITY:

R1	The system shall allow users to get a ticket with a date and time that shows when to go to a certain store, virtually
R3	The system shall allow users to book a visit virtually with their desired store.
	Components: • Reservation Manager • Queue Manager • Location Module • Map Mediator Module
R2	The system shall allow users to get a ticket with a date and time that shows when to go to a certain store, physically
	Components: • Reservation Manager • Queue Manager
R4	The system shall allow users to look up on a map available registered stores where to go to
	Components: • Map Mediator • Location Module
R5	The system shall ask users how much he or she thinks the trip to the store will last
	Components: • Reservation Manager

R6	The system shall allow users to be identified by their device unique ID
R7	The system shall allow users to be identified by a username of their choosing
	Components: • Account Manager
R8.1	The system shall notify the user who is inactive while on the confirmation page of booking a visit on other available stores he could go to and other timeslots of the same store
R8.2	The system shall notify the user who is inactive while on the confirmation page of getting a ticket on other available stores he could go to
	Components: • Notification Manager: Suggestion Module • Location Module: Market Finder • Map Mediator • Queue Manager: Timeslot Manager
R9	The system shall allow its users to insert information about which categories or items they want to buy
	Components: • Reservation Manager
R10	The system shall infer how long it will take for “Book a Visit” customers to buy the expressed shopping list
	Components: • Queue Manager: Optimizer Module
R11	The system shall store data about registered virtual users' visits durations and expressed shopping lists
	Components: • Reservation Manager • Turnstile Manager • Queue Manager: Timeslot Manager
R12.1	The system shall inform users periodically of the available time slots in the store for which he subscribed to the service
R12.2	The system shall allow its users to select which store(s) to get informed about
R12.3	The system shall allow its users to select which time slots he is interested to get informed about
R12.4	The system shall allow its users to select how often to get notified
	Components: • Notification Manager: Periodic Notifier Module
R13	The system shall allow the user to scan its QR code in entrance through the turnstiles
R14	The system shall allow the user to scan its QR code in exit through the turnstiles or cash register

R15.1	The system shall unlock turnstiles after a unique QR code scan in entrance
R15.2	The system shall unlock turnstiles after a unique QR code scan in exit
R16	The system shall map the QR code of a scan to the virtual user who owns the QR code
	Components: • Turnstile Manager • Queue Manager: Timeslot Manager
R15.3	The system shall lock the turnstiles after a push has occurred
	Components: • Turnstile Manager
R17	The system shall send a reminder to the user when it is time for him to leave so that he can arrive at the store in time
	Components: • Suggestion Module
R18	The system shall calculate the time it takes the user to go to a shop in which he has an appointment
	Components: • Suggestion Module • Location Module: ETA Module • Map Mediator Module
R19	The system shall manage how many people can enter inside the store for each timeslot, as to not exceed the maximum number of people allowed inside the store
R20	The system shall calculate the maximum number of people allowed inside each store, as to allow for social distancing to take place
	Components: • Queue Manager: Optimizer Module
R21	The system shall count the number of entrances and exits each day for each market
R22	The system shall store the number of daily entrances and exits for each market
	Components: • Queue Manager: Timeslot Manager
R23	The system shall manage how many people can enter inside the store for each timeslot, as to not exceed the maximum number of people allowed inside the store hourly, without exceeding the maximum calculated by the system
R24	The system shall allow store managers to see how many customers have entered the store in any day.
	Components: • Market Manager: Influx Module • Web Server
R25	The system shall allow store managers to register their stores
R26	The system shall allow store managers to input the store's location

R27	The system shall allow store managers to input what categories of goods and what products are contained in the store
R28	The system shall allow store managers to input the dimensions of the store
	Components: - Market Manager: Market Info Manager

Note that, as shown in the component and in the sequence diagrams, each component interacts with the database through Data Service component, that was omitted here to avoid repeating it in every requirement.

5) IMPLEMENTATION, INTEGRATION AND TEST PLAN: Identify here the order in which you plan to implement the subcomponents of your system and the order in which you plan to integrate such subcomponents and test the integration.

6) EFFORT SPENT: In this section you will include information about the number of hours each group member has worked for this document.

7) REFERENCES