POLITECNICO DI MILANO

Computer Science and Engineering

Acceptance Test Deliverable

Customers Line-up

Software Engineering 2 Project

Academic year 2020 - 2021

14 February 2021

Version 1.0

Authors:
Samuele NEGRINI
Giorgio PIAZZA

Professor:
Matteo Giovanni ROSSI

Revision history

Date	Revision	Notes
14/02/2021	v.1.0	First release.

Contents

Chapter 1

Introduction

1.1 Purpose

This document is the Acceptance Test Deliverable (ATD) for the *Customers Line-Up* project. Its purpose is to analyse and test the project developed by another group, to point out any eventual bug of the system and inconsistencies with the requirements described in the RASD and DD documents.

1.2 Project analysed

The project analysed is the one developed by **Davide Merli** and **Dario Passarello**.
The source code, as well as the RASD, DD, ITD and every other document concerning the project, can be found on their GitHub repository:

https://github.com/davidemerli/MerliPassarello

Chapter 2

Installation Instruction

We followed the instructions given in the ITD. The group provided both the backend and frontend of their application already deployed. Though we also built and tested the application by running the backend and frontend ourselves on our devices.

To actually run the test code, a few, small adjustment were applied to the given instructions: for example the installation of some python dependencies.

The application was also run from the source code, no installation of the Flutter framework was necessary, as we already developed our project using Flutter.
We followed these steps:

1. clone the repository;
2. import the project in Android Studio;
3. build the project on our device.

Chapter 3

Acceptance Test Cases

In this chapter we analysed the implemented requirements of the project and tested them.
By comparing the list of requirements presented in RASD and the ones inside the ITD we can conclude that only a few requirements where actually implemented.
For instance, some requirements that we consider vital such as R4 and R12 are not implemented.

Moreover we noticed that in the second chapter of the ITD (Requirements and Functionalities Implemented) is stated that "*The management side is kept at a bare minimum providing only the store operator functions to call customer to the entrance, scan tickets and view real time data about the store crowdedness*". This is partially true because there is not the possibility to check the store crowdedness.

First thing we ran the unit and integration tests provided: all of them passes.
Then we concentrated on testing the app in a real use case scenario, just like stated in the RASD. The majority of the test performed are listed below.

3.1 Test Cases

ID	AT001_SignIn
Description	Login as a customer in the application.
Input	• The system shows the home page; • Fill in the form with the following data: Email: test@example.com Password: Password123@ • Press the Login button;
Expected Output	User correctly logged in and redirected to the map page.
Test Result	Success

ID	AT002_SignIn
Description	Login as a operator in the application.
Input	• The system shows the home page; • Fill in the form with the following data: Email: operator1@CLup.com Password: operator1@CLup.com • Press the Login button;
Expected Output	User correctly logged in and redirected to a page where the operator can call tickets.
Test Result	Success

ID	AT003_SignUp
Description	Register as a customer in the application.
Input	• The system shows the home page; • Go to the sign up page; • Fill the form with the following data: *Email: test@example.com* *Password: Password123@* *Confirm Password: Password123@* *Name: Mario* *Surname: Rossi* • Press the Sign Up button; • The system shows a page with a summary of the user data and a confirm button; • Press the confirm button;
Expected Output	User correctly registered and redirected to the home page.
Test Result	Success

ID	AT004_JoinQueue
Description	Join the queue of a store as a customer.
Preconditions	The customer is logged in.
Input	• The system shows the map page; • Select a store from the map; • Press the "Get a Ticket" button; • The system shows the store, press the "Get a Ticket" button and then "OK" on the popup message;
Expected Output	The ticket is retrieved successfully and the system shows a page which shows the retrieved ticket.
Test Result	Success

ID	AT005_JoinAnotherQueue
Description	Join the queue of a store as a customer who has already a valid ticket for another store.
Preconditions	The customer is logged in and has already a valid ticket for another store.
Input	• The system shows the map page; • Select a store from the map; • Press the "Get a Ticket" button; • The system shows the store, press the "Get a Ticket" button and then "OK" on the popup message;
Expected Output	The ticket is not retrieved and the system shows an error message.
Test Result	Success

ID	AT006_DeleteTicket
Description	Delete a ticket for a store as a customer.
Preconditions	The customer is logged in and has a valid ticket for a store.
Input	• The system shows the ticket page; • Press the "Cancel Ticket" button; • The system ask for confirmation, press the "Delete Ticket" button;
Expected Output	The ticket is deleted and the system redirects to the home page.
Test Result	Success

ID	AT007_CallPerson
Description	A store operator call the next person in queue.
Preconditions	The store operator is logged in and a customer is inside the store queue.
Input	• Press the "Call Next Person" button;
Expected Output	The person is called and on the customer app, a notification is displayed.
Test Result	Success

ID	AT008_ScanTicket
Description	A store operator scans a called customer’s ticket.
Preconditions	The store operator is logged in and a customer is called.
Input	• Press the "Scan QR Code" button; • The system open the camera and the operator scans the QR of the customer’s ticket.
Expected Output	The ticket is validated and the person is inside the store.
Test Result	Success

ID	AT009_ScanTicketOnStoreFull
Description	A store operator scans a called customer's ticket when the store is already full.
Preconditions	The store operator is logged in, the store is full and a customer is called.
Input	• Press the "Scan QR Code" button; • The system open the camera and the operator scans the QR of the customer's ticket.
Expected Output	The customer is scanned but the system refuse the ticket because there is no more space in the store.
Test Result	Success

We noticed that there is a design flaw about the ticket call when the store is full. An operator can always call all the tickets in the queue without any check about the store capacity.
This results in the following bad behaviour:

1. The store reaches the maximum of his capacity with the scan of a ticket.
2. The operator calls the next ticket or more.
3. If the store remains full the scan of a ticket fails.
4. Then, if no one exits the store in 5 minutes (expiration time of a called ticket) the called tickets expire and this results in the deletion of the ticket.
5. The customer who owned the ticket need to get in queue again with a new one.

Chapter 4

Additional Notes

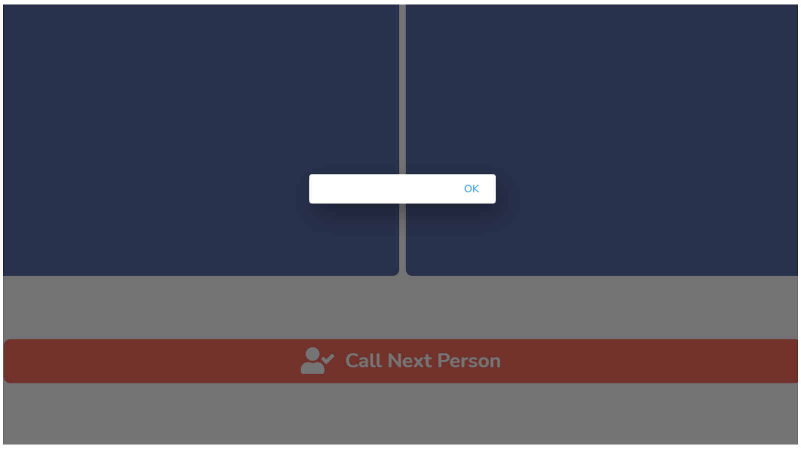
4.1 Source code

The source code is well developed both in the flutter app and the server. The general structure of the code follows the MVC pattern as stated in the DD.
Some parts of the backend could be better documented but nothing too serious, nevertheless the client side is very well documented.

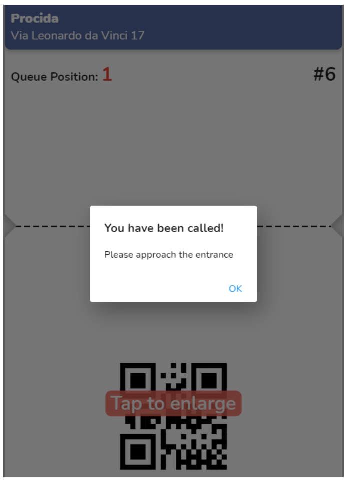
4.2 User Interface Concerns

During the testing of the User Interface of the flutter application we encountered some UI bad behaviours.
We make this evident just because on the ITD it was stated that the development of the application was focused on the user experience.

- There are some pop up messages which don't tell nothing about what is happening. For example, when an operator calls a customer, an empty pop up is shown.



- When a customer is called, the app shows to him an alert telling that he was called. This happens every 10 seconds causing the stacking of several messages.



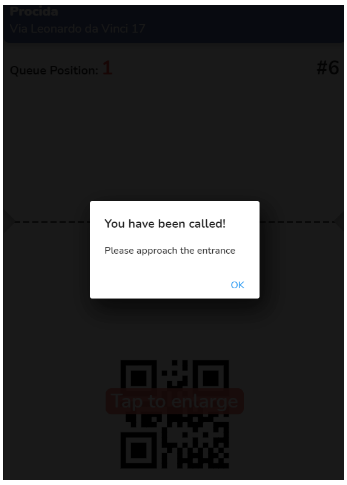
Procida
Via Leonardo da Vinci 17

Queue Position: 1

#6

You have been called!

Please approach the entrance



Procida
Via Leonardo da Vinci 17

Queue Position: 1

#6

You have been called!

Please approach the entrance

Chapter 5

Effort Spent

5.1 Teamwork

Task	Hours
Initial briefing	2

5.2 Samuele Negrini

Task	Hours
Testing acceptance	3
Document writing	4

5.3 Giorgio Piazza

Task	Hours
Setup docker tools	1
Build android app	1
Testing acceptance	3
Document writing	1