POLITECNICO
MILANO 1863

ATD

Acceptance Test Deliverable

Authors:

David Leonardo Gomez, 10787907

Martin Anselmo, 10789796

Vinay Krishna Munnaluri, 10780972

Version: 1.0

Date: 10 February 2021

Professor: Elisabetta Di Nitto

1. Analyzed Project

1.1. Document objective

This document describes the acceptance testing process of a working prototype of CLup described in the ITD, RASD (Requirement Analysis and Specification Document) and DD (Design Document) documents. This document is intended to be a testing result compendium made by a team different from the developers one and explains the results of the testing and the completion of use cases described on earlier documents. It also provide input on how to perform integration testing between the implemented components

1.2. Group Information

The members of the project analyzed are:

- Andrea Franchini - 10560276
- Ian Di Dio Lavore - 10580652
- Luigi Fusco - 10601210

The code can be found here

1.3. Documents considered

For a reference of the CLup system implementation, there will be used 3 documents:

- CLup RASD
- CLup DD
- CLup ITD

1.4. Implemented Requirements

Accordingly to the previously referenced documents the requirements that were implemented for CLup are the next ones:

- R1 - Allow a User to sign up for an Account after providing a mobile phone number.
- R2 - Allow a Registered User to find Stores nearby a specified user or an hardcoded location. Note: This is because the Google Maps API that converts addresses to coordinates is paid.
- R4 - Allow a Registered User to get in the virtual line at a specified store.
- R5 - Allow a Totem User to get in the virtual line of the store where the totem is installed. Note: For the purposes of the demo, the user interface of the totem contains fields to specify the auth token and the store id. In the final systems these

would be hard-coded and dependent on where the totem is installed.

- R6 - Allow a Registered User to preview an estimate of the queue time.
- R8 - Allow a Registered User to cancel their reservation.
- R9 - Allow a Registered User to leave the virtual queue.
- R10 - Allow a Registered User and a Totem User to retrieve a scannable QR Code/Barcode that they must present in order to be granted access to a store.
- R12 - The System cancels User reservations in case of a major delay.
- R13 - The System enforces the limits on the allowed number of concurrent Customers inside a store by restricting the access at the entry points (for example, automatic doors or turnstile). Note: Although the physical barrier is not present, the functionality is implemented in software and a text message is shown indicating if access is granted or not.
- R14 - The System grants a User with a reservation access only within a short time (set by the manager) after the User's time of reservation.

2. Installation Setup

2.1. Steps taken

Due to the group's effort in *dockerizing* the whole system, installation was as easy as installing docker and docker-compose and running:

- docker-compose up

One problem we encountered when installing the software, was due to breaking change in a small change version of docker-compose. The syntax of the docker-compose.yml was slightly changed and what version was used was not specified.

3. Acceptance Test Cases

The following section focuses on testing different use scenarios of the system. Most of this use cases where extracted from the requirements given to us in the ITD. They consider the main flow of what the system should do, in addition to alternative flows, considering errors or simply other paths.

3.1. Use Case: Login into CLup

Name	Login
Description	This test assures that the system allows for users to sign in with their credentials.
Actors	Customer
Entry condition	User loads the main page of CLup, where he is prompted with a login screen.
Expected event flow	1. User clicks on the Login Button 2. User enters his phone number a. An SMS code is sent to his phone 3. The user enters the code into the field 4. The user clicks to confirm
Alternative flow	- An incorrect phone number is entered, prompting an error message. - The user is already authenticated when he loads the page
Exit conditions	The user is redirected to the /stores page or the /ticket page, in case he has an active ticket
Notes	The SMS code is not implemented, as stated in the ITD. To log in,

	the user currently has to go into the logs to get the code.
Testing description	To perform this test, the system was loaded and we simulated being an user performing the action. No other actor was involved.
Test result	The test was performed successfully. Every detail above was conducted with no problems.

3.2. Use Case: Select a store

Name	Select a Store
Description	The test case allows user to select a store from list of available stores
Actors	Customer
Entry condition	user must be logged in
Event flow	1.) User login to the application through home by providing phone number and sms key 2.) Once a customer clicks on the login button he will be navigated to the store's page if he doesn't have any reservation or ticket existing. 3.) he can either select from a list of stores shown on left side or from the map by clicking on store location
Alternative flow	No alternate flow
Exit conditions	Once the customer selects the store he will redirected to store details page where he can see queue details and enter time
Notes	the user can access stores only if doesn't have any active ticket or reservation
Testing description	To perform this test , the system was loaded and we simulated being as an actual user performing login and viewing stores. No other actor was involved
Test result	test successful

3.3. Use Case: See a preview for the waiting time

Name	preview waiting time

Description	The test case allows user to see waiting
Actors	Customer
Entry condition	User must be logged in
Event flow	1.) User login through home page 2.) select a store 3.) click on queue now 4.) if users are already in queue then you should get waiting time
Alternative flow	1.) login from home page 2.) you will be shown in your ticket info page
Exit conditions	null
Notes	This feature is not implemented
Testing description	logged in as real customer and click on store details and click on queue now
Test result	As it is not implemented its a failure

3.4. Use Case: Join Queue (Line up)

Name	Join Queue
Description	The test case allows us to verify user to join the queue
Actors	Customer
Entry condition	User must be logged in
Event flow	1.) User login through home page 2.) select store 3.) store details are displaced and queue now button exists 4.) user clicks on queue now button
Alternative flow	Already logged in and delete a queue ticket then again store list is displayed select a store and click on queue now button to enter queue again
Exit conditions	user is navigated to ticket details page
Notes	user can enter queue only if its not full
Testing description	Logged in as real customer and selected a store and clicked on queue now button if queue is empty it shows your turn else wait in the queue
Test result	successful

3.5. Use Case: Leave Queue

Name	Leave queue
Description	the test case allows use to verify user leaving the queue
Actors	Customer
Entry condition	User must be logged in to the application
Event flow	1.) User login through home page 2.) select store 3.) store details are displaced and queue now button exists 4.) user clicks on queue now button 5.) user clicks on leave queue button if wants to leave the queue
Alternative flow	1.) User login through home page 2.) Active ticket exists and ticket details are shown and at the end leave queue button is shown 3.) User clicks on it to leave the queue
Exit conditions	User will be redirected to stores page where he can see the list of stores available
Notes	leave queue is available only if you have an active ticket or if you are already in the queue
Testing description	logged in as an actual customer and selected a store and clicked on the queue button and then the leave queue button again.
Test result	test successful

3.6. Use Case: Book a ticket (make a reservation)

Name	Book a Ticket
Description	The process of booking a ticket
Actors	Authenticated user
Entry condition	Authenticated User selects a Supermarket in the application
Event flow	1. One of the following path: (a) The Authenticated User clicks the "Make a Reservation" button. (i) The user chooses a suitable time among the available ones. (b) The Authenticated User selects the less crowded slot proposed by the application. 2. The system provides the Reservation Receipt with the generated

	QR code.
Alternative flow	No alternative flow
Exit conditions	The customer has now a pending reservation.
Notes	
Testing description	Ticket will be booked like a real authenticated user
Test result	Test successful

3.7. Use Case: Cancel reservation

Name	Cancel Reservation
Description	The process of canceling a booking
Actors	Authenticated User
Entry condition	Authenticated user enters the home page of the application and has a pending Reservation.
Event flow	1. The user clicks on the button "My Reservation". 2. The Authenticated User clicks the "Cancel Current Reservation" button. 3. The Authenticated User confirms their choice.
Alternative flow	No alternative flow
Exit conditions	The customer has now canceled the reservation, and can make a new one
Notes	
Testing description	A ticket will be booked from an authenticated user perspective and later canceled after the booking time
Test result	Test successful

3.8. Use Case: Join local Queue (totem user)

Name	Join Local Queue
Description	Join to a virtual queue by getting the book at the physical store
Actors	Totem User

Entry condition	A user approaches the totem in front of a store while the store is open.
Event flow	1. The user clicks the button appearing on screen. 2. The totem prints a ticket containing a QR code and the estimated waiting time.
Alternative flow	No alternative flow
Exit conditions	The user is now in queue.
Notes	A QR code is not shown, only the ticket number.
Testing description	
Test result	Test passed with a note, no QR code is shown nor available for printing.

3.9. Use Case: Show QR code from ticket or booking

Name	Display a QR code so it can be scanned
Description	When a user wants to enter a store, he must show a QR code that can be scanned for entering the store.
Actors	Registered User
Entry condition	A user is in Queue or has a reservation.
Event flow	1. The user waits until a notification on the app appears. 2. The user enters the store by letting the totem read their QR code.
Alternative flow	There is an exception on Step 2 (Exceptions and handling in notes)
Exit conditions	The QR code is shown and can be scanned in the totem
Notes	Exceptions: • There is a major delay in the store. • The time remaining to wait in the queue exceeds the opening hours of the store. • The ticket is unvalid or can’t be read.
Testing description	A ticket must be booked from the perspective of the authenticated user, and later on, used for simulating a shopping entry. Then, we must open the main page for the ticket to show.
Test result	Test successful, as the QR code is visible from the user perspective

3.10. Use Case: Show QR code from totem ticket

Name	Show/print QR code generated by a totem
Description	When an user wants to get a ticket from a totem, he must ask for it directly there. The option to show or print the ticket must be available for the customer to keep
Actors	Totem user.
Entry condition	Totem user approaches the store to get a ticket
Event flow	1. The user approaches the totem and requests a ticket
Alternative flow	A ticket is not available because of an exception (described in notes)
Exit conditions	The user has now a printed QR code
Notes	Exceptions: • There is a major delay in the store. • The time remaining to wait in the queue exceeds the opening hours of the store. • The ticket is unvalid or can't be read.
Testing description	For testing, we loaded the totem.html file and generated a ticket from there.
Test result	Test failed, as only the ticket descriptor (QXX) is shown, but not a QR code nor the option for printing it.

3.11. Use Case: Reservation is automatically cancelled in case of delay

Name	Reservations are cancelled in case of delay
Description	When a customer makes a reservation, and he does not show up within a deadline of 5 minutes, the reservation is not valid anymore.
Actors	Customer
Entry condition	A customer has a reservation for a store, and the time to enter is up
Event flow	1. Customer has a reservation and approaches 5 minutes after the ticket is said to be ready. 2. Customer scans his reservation 3. The ticket is not valid anymore
Alternative flow	No alternative flows are considered for this use case
Exit conditions	The customer is not allowed inside the store

Notes	Testing was hard due to the timezone set in the server
Testing description	A reservation was created for a store. In the database, the ticket time to enter was changed to before 5 minutes of the current time, the we tried scanning it.
Test result	The test is passed as the ticket is not allowed inside.

3.12. Use Case: Queue ticket is automatically cancelled in case of delay

Name	Cancelation of ticket in queue in case of delay
Description	When a customer is late for his appointment, the rest of the queue should not be made to wait for him, so the system should automatically cancel the ticket.
Actors	Customer
Entry condition	User requests a ticket for the current queue. But then fails to enter the store within the allocated time slot (1 minute set up in database)
Event flow	1. User has the ticket to enter the store and is next in line 2. User takes more than 1 minute to enter the store since his time is up.
Alternative flow	- After the ticket is cancelled, the user requests a new ticket. Now the user is at the end of the queue.
Exit conditions	- The code for the ticket is now invalid (cannot be scanned at the totem) - The next user in line now is ready to enter (moves a place in the queue)
Notes	- The ticket for the next user in line doesn't display correctly the information indicating that he is next in line until another user requests a new ticket.
Testing description	To perform this test, we had two users request a ticket, with a delay of 40 seconds. Then, we waited 1 minute after the first one was requested and reloaded the page. We tried to scan the ticket and received an error. Lastly, we reloaded the second's user ticket and tried to scan it, being able to.
Test result	Apart from the display detail described on *Notes*, the test passes all criteria.

3.13. Use Case: Scan QR/ticket

Name	Scan a ticket for entering the store
Description	When an user wants to enter a store, he must show a valid ticket for entering it. The system will then decide if the user can enter or not.
Actors	Customer, Totem
Entry condition	A user has a ticket to scan
Event flow	1. User approaches the totem and enters the ticket code 2. The totem shows a message indicating success 3. The user enters the store and then wants to exit 4. The user then checks out from the store (with another totem)
Alternative flow	2.1. The totem shows an error message if the ticket is invalid for entering.
Exit conditions	- The next ticket in the queue is now ready for scanning - The number of people inside the store is counted correctly
Notes	- The scanning of QR codes was not implemented, the codes have to be entered by hand.
Testing description	Two tickets were generated for a store. First, the second in line tried to scan his ticket to enter and failed. Then, the first one scanned his ticket correctly. The number of people in the store was incremented. Now the second customer scans his ticket and gets to enter the store. Lastly, the first user exits the store, leaving the database status of that store with 1 customer inside.
Test result	The test was successful as everything worked as expected.

3.14. Use Case: Maximum store capacity is enforced

Name	Maximum store capacity enforcement
Description	Due to COVID-19 restrictions, the store's capacity cannot be over a defined amount.
Actors	Customers, Store
Entry condition	The store is full (current customers inside = maximum capacity) and a customer wants to enter, as it is his turn in the queue.
Event flow	1. One customer enters the store leaving it filled with customers 2. A new customer wants to enter the store, scanning his ticket

	3. The totem prompts a message saying that the store is full 4. A customer exits the store 5. The original customer reescans his ticket, and now he can enter the store
Alternative flow	3.1. The customer decides to not wait and leaves the queue 3.2. The next customer in line follows with the original path
Exit conditions	- The customer is not allowed in until someone exits the store
Notes	- If a customer can't enter the store due to it being full, and it takes longer than 1 minute for someone to leave the store, the ticket is rendered invalid. This shouldn't happen as it is not the customer's fault.
Testing description	The store state was set to be almost at full capacity, and we made one customer enter the store, leaving it full. We then created another ticket and tried to enter the store, but couldn't. After we simulated an exit, the customer was able to enter.
Test result	The test is successful, the limit of people inside the store is enforced.

3.15. Use case: Reservations work as intended

Name	Reservations work as intended
Description	When a customer makes a reservation, he is able to enter the store within a reasonable amount of time compared to his time.
Actors	Customer
Entry condition	A customer has a reservation for a store, and the time to enter is up
Event flow	4. Customer approaches the store and scans his reservation 5. The ticket is valid and is let in.
Alternative flow	- The store is full, in that case the user may have to wait until someone exits the store
Exit conditions	The ticket is valid independent of the current queue of the store
Notes	Testing was hard due to the timezone set in the server
Testing description	A reservation was created for a store. In the database, the ticket time to enter was changed to the current time and tested the entry.
Test result	Everything worked as expected

4. Extra Comments on the Project

Some positive comments about the project are

- The code is extremely well documented, including docstrings for function parameters and descriptions about functionalities.
- The code structure matches very well the DD, including all components, classes and inputs.
- The code is simple enough to be understandable and is separated well into logical components.
- The dockerization of the project is really useful for other people to quickly run it locally, without needing to install a ton of dependencies.
- The UI of the client is really good, and provides a good mobile experience.

Some negative comments about the project and the code are

- While the code matches very well the structure described in the DD, this may create a handicap while developing, because there are a lot of parts involved.
- While what was developed in enough for a prototype, there are certain features that would have been nice to have, like a proper interface for the totem. (without it, the display of QR codes is useless)