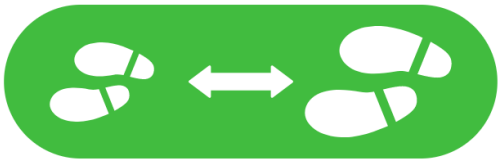
POLITECNICO
MILANO 1863

Computer Science and Engineering

Software Engineering 2 Project

Acceptance Testing Document

Version 1.0 - 14/02/2021



Customers Line-up

STAY SAFE

Authors

Enrico Gherardi
Ludovico Righi

Professor

Elisabetta Di Nitto

Tutor

Annalisa Binato

Contents

Chapter 1

Introduction

1.1 Scope

The scope of this document is to present the acceptance testing of the Software produced by the team BianchiCampiDeSantis and described in section 1.2.

1.2 Project

The project to discuss is SafeStreets from BianchiCampiDeSantis team, link to Github repository: https://github.com/teobia98/BianchiCampiDeSantis. In the team's folder there are all the previous documents (RASD, DD and ITD), so first the RASD was analyzed to understand the project with an high view, then the ITD to proceed with the installation and finally the DD was needed to execute better the testing on the software.

1.3 Definitions, Acronyms, Abbreviations

This section gives some definitions in 1.3.1, acronyms in 1.3.2 and abbreviation in 1.3.3 which will be use in the document, in order to explain some concept and help the general understanding.

1.3.1 Definitions

- Booked ticket: it is a reservation ticket for a specific date and time
- Queue ticket: it is a ticket for customer who lined up in a store's virtual queue

- Operator: it is a physical entity (supermarket personnel, gate, or ticket machine)
- Maximum capacity: maximum number of people allowed at the same time in his store, defined by the store manager
- Reservation limit: it is a restriction on the number of people that can request a booked ticket in the same time slot
- Reserved time slot: it is defined as the date and time of a reservation, and the expected duration of the visit stated by the customer
- Long-term customer: it is a customer who booked many visits
- Physical line: it is a line or sequence of people awaiting their turn to enter the store
- Real-time queue status: it contains the number of people in the queue and the approximate waiting time a customer must wait if he lines-up. If referred to an existing ticket, it contains the information about the relative position in the queue and the remaining waiting time

1.3.2 Acronyms

- *RASD*: Requirement Analysis and Specification Document.
- *DD*: Design Document.
- *UI*: User Interface
- *ITD*: Implementation and Testing Document.

1.3.3 Abbreviations

- RN: Requirement number N.

1.4 Revision History

- Version 1.0 : 14/02/2021

1.5 Reference Documents

All the file names are referread to the ones in the delivery folder.

- "RASD 1.3.pdf"
- "DD 1.2.pdf".
- "I&TD 1.0.pdf"

1.6 Document Structure

The document is structured in the following way:

- Chapter1 : Introduction of the document and the analyzed software.
- Chapter2 : Description of installation setup and issues.
- Chapter3 : Description of the Acceptance testing.
- Chapter4 : Additional notes on the code.

Chapter 2

Installation Setup

The installation guide is extremely precise and detailed: it provides also the screenshots of what the tester has to do in order to set up all the environment.
In particular, our installation was straightforward, taking us only a few minutes having Java and MySQL Worbench already installed on our computers.
However, we are sure that the installation process would have been successfully carried out without any problems even by a tester with less knowledge of this environment.

Chapter 3

Acceptance Testing

This chapter presents the acceptance testing on the software, in section 3.1 there is the list of tested requirements implemented by the other team and in section 3.2 there are some test cases and features extracted by the RASD and the user experience.

3.1 Developed Requirements Checking

- **R1**: The system shall allow store managers to identify themselves (login as Manager).

  Test Case: login with valid store manager credentials.
  Input: username="manager1", password="password".
  Output: Successful login.

  Test Case: login with valid store manager username and wrong password.
  Input: username="manager1", password="wrong".
  Output: Refused login.

  Test Case: login with invalid store manager credentials.
  Input: username="manag", password="wrong".
  Output: Refused login.

- **R2**: The system shall allow the store manager to register stores to the CLup administration

Test Case: register store with all valid fields.
Input:chain="ExampleStore", name="ExampleStore market", address="via Mons. G. Cavazzuti 24", openingTime="08:30", closingTime="18:45", capacity="100", reservationLimit="20", waitMultiplier="5".
Output: Successful store registration.

Test Case: register store with a missing field (the same test case is valid for any missing field).
Input:chain=(null), name="ExampleStore market", address="via Mons. G. Cavazzuti 24", openingTime="08:30", closingTime="18:45", capacity="100", reservationLimit="20", waitMultiplier="5".
Output: Missing field error; store not registered.

Test Case: register store with reservation limit greater than maximum capacity.
Input:chain="ExampleStore", name="ExampleStore market", address="via Mons. G. Cavazzuti 24", openingTime="08:30", closingTime="18:45", capacity="100", reservationLimit="200", waitMultiplier="5".
Output: Reservation limit error; store not registered.

- **R3**: The system shall allow the store manager to insert parameters for his stores

  Test Case: implicitely tested through the previous test case.

- **R5**: The system shall allow the store manager to register operators for ticket evaluation.

  Test Case: register operator with all valid fields.
  Input: username="op3car", password="password", repeatPassword="password", store="ExampleStore market".
  Output: Successful operator registration.

  Test Case: register operator with a missing field
  Input: username=(null), password="password", repeatPassword="password", store="ExampleStore market".
  Output: Refused operator registration.

Test Case: register operator with code lenght different from 6.
Input: username=operato16, password="password", repeatPassword="password", store="ExampleStore market"..
Output: Refused operator registration.

Test Case: register operator wrong password confirmation.
Input: username="op3car", password="password", repeatPassword="passwordWrong", store="ExampleStore market".
Output: Refused operator registration.

- **R6**: The system shall let in the store only customers with a valid ticket.

  Test Case: insert valid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de06895" .
  Output: Successfull entrance.

  Test Case: scan invalid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de54268" .
  Output: Refused entrance.

- **R7**: The system must invalidate a ticket after 180 minutes since the entrance evaluation.

  Test Case: check ticket status after 180 minutes or more.
  Input: /.
  Output: Invalidated ticket.

- **R8**: The system must reset the number of people in the queue every time it closes.

  Test Case: check queue status after closing time.

Input: /.
Output: Number of people in the queue resetted.

- **R9**: The system must show the number of people inside the store to operators.

  Test Case: click on "Current Information" on operator menu
  Input: /.
  Output: People inside and maximum capacity are shown.

- **R10**: The system shall allow operators to scan tickets when a customer enters the store.

  Test Case: insert valid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de06895" .
  Output: Successfull entrance.

  Test Case: insert invalid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de54268" .
  Output: Refused entrance.

- **R11**: The system shall allow operators to scan tickets when a customer exits the store.

  Test Case: insert valid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de06895" .
  Output: Successfull exit.

  Test Case: scan invalid QR Code.
  Input: QR Code="Ba697f6be-14fb-4378-bd9d-08a56de54268" .
  Output: Refused exit.

- **R12**: The system shall allow lined-up customers to withdraw their ticket.

  Test Case: click on withdraw button.
  Input:/.
  Output: Successfull ticket withdraw.

- **R16**: The system shall provide to lined-up customers their real time approximate waiting time.

  Test Case: check an active lined-up ticket.
  Input:/.
  Output: Waiting time shown.

- **R17**: The system shall provide to lined-up customers their real time position in the queue.

  Test Case: check an active lined-up ticket.
  Input:/.
  Output: Position shown.

- **R18**: The system shall show a real-time queue status to customers who wants to line-up.

  Test Case: check an active lined-up ticket.
  Input:/.
  Output: Status shown.

- **R19**: The system must invalidate a booked ticket after 15 min if a customer does not show up.

  Test Case: check booked ticket 15 minutes after the selected time.
  Input:/.
  Output: Booked ticket deleted.

- **R20**: The system shall allow operators to call the next one in the queue if the called customer does not show up.

  Test Case: click on "Call next" on the operator interface.
  Input:/.
  Output: Next customer is called.

- **R23**: The system shall allow people to line-up in a virtual queue (no more than once at the same time).

  Test Case: create a new line-up for a customer that has not another active one.
  Input: store="ExampleStore market".
  Output: Line-up successfully sent.

  Test Case: create a new line-up for a customer that has already another active one.
  Input: store="ExampleStore market".
  Output: Impossible to send the line-up.

- **R24**: The system shall provide a queue ticket to customers who line-up.

Test Case: check customer interface.
Input:/.
Output: Line-up information are shown.

- **R27**:The system shall allow the customer to identify himself (login/sign-in).

  Test Case: login with valid customer credentials.
  Input: username="teobia98", password="password".
  Output: Successful login.

  Test Case: login with valid customer username and wrong password.
  Input: username="teobia98", password="wrong".
  Output: Refused login.

  Test Case: login with invalid customer credentials.
  Input: username="teob", password="wrong".
  Output: Refused login.

  Test Case: register customer with all valid fields.
  Input: username=ludorighi, email="example@gmail.com", name="Ludovico", lastName="Righi", password="password", repeatPassword="password".
  Output: Successful customer registration.

  Test Case: register customer with a missing field (the same test case is valid for any missing field).
  Input: username=(null), email="example@gmail.com", name="Ludovico", lastName="Righi", password="password", repeatPassword="password".
  Output: Missing field error; customer not registered.

  Test Case: register customer with wrong password repetition.
  Input: username=ludorighi, email="example@gmail.com", name="Ludovico", lastName="Righi", password="password", repeatPassword="passw".

Output: Passwords not matching error; customer not registered.

- **R28**: The system shall allow customers to book a visit for a specific date/time and duration (once per day and not for the current day).

  Test Case: send new booking for the current day.
  Input: store="ExampleStore market" , date=(today), time="08:00", duration="30 minutes".
  Output: Impossible to select current day; booking not sent.

  Test Case: send new booking for the same date of another booking.
  Input: store="ExampleStore market" , date=(repeatedDate), time="08:00", duration="30 minutes".
  Output: Booking successfully sent (requirement violated).

  Test Case: send new booking with all valid fields.
  Input: store="ExampleStore market" , date="03/03/2021", time="08:00", duration="30 minutes".
  Output: Booking successfully sent.

- **R30**: The system shall provide a booked ticket to customers who reserve a visit.

  Test Case: check customer interface.
  Input:/.
  Output: Booked ticket information are shown.

- **R31**: The system shall allow customers who booked a visit to withdraw their tickets.

  Test Case: click on withdraw button.

Input:/.
Output: Successfull booked ticket withdraw.

- **R32**: The system shall provide real-time free time slots for customers who wants to book a visit.

  Test Case: check free time slots when sending a booked ticket.
  Input:/.
  Output: Free time slots provided.

3.2 Additional Features Testing

- **Test Case:** Testing the update of the waiting time after the deletion of a preview Line-Up request.
  Result: The waiting time is correctly updated.
- **Test Case:** Testing the update of the waiting time after the entrance of its customer.
  Result: The waiting time is static and not updated.
- **Test Case:** Testing the usability of the Web Application.
  Result: The graphical interface is user-friendly and well understandable.

Chapter 4

Additional Notes

The code is well structured and it is coherent with the structure presented in the I&T document.
Furthermore, the code is also appropriately commented and this permit an easy understanding in a very efficient way. Also, they provided the JavaDocumentation, which, even if it is not fully detaild, it represents a further tool for better understanding the whole software artifacts.