POLITECNICO
MILANO 1863

M.Sc. Automation and Control Engineering

Software Engineering (for Automation)

Academic year 2021/2022

Testing Document

CLup: Customers Line-up

Advisor: Prof. Matteo Giovanni Rossi

Students: Alessandro Nocentini alessandro.nocentini@mail.polimi.it
Alberto Valentini alberto.valentini@mail.polimi.it
Fausto Luca Pichierri faustoluca.pichierri@mail.polimi.it

Table of Contents

1. Introduction

1.1 Purpose

This document is intended to present the work done about the verification and the validation of the application, outlining the tests that have been conducted, during and after the implementation phase, aimed at verifying the proper functioning of the implemented requirements.
By comparing the list of requirements present in the RASD with the one tested here it can be noticed that all the core requirements have been implemented, as for almost all the medium priority ones, except for one, plus two belonging to the good to have set.

1.2 Scope

The coronavirus emergency has put a strain on society on many levels, due to many countries imposing lockdowns that allow people to exit their homes only for essential needs, and enforcing strict rules even when people are justified in going out (such as limiting the number of accesses to buildings and keeping a distance of at least one meter between people).
In particular, grocery shopping a most essential need can become a challenge in the presence of such strict rules. Indeed, supermarkets need to restrict access to their stores to avoid having crowds inside, which typically results in long lines forming outside, which are themselves a source of hazards. In these trying times, people turn to technology, and in particular to software applications,to help navigate the challenges created by the imposed restrictions.
The goal of this project is to develop an easy-to-use application that, on the one side, allows store managers to regulate the influx of people in the building and, on the other side, saves people from having to line up and stand outside of stores for hours on end.
Note: CLup covers only the user experience side.

1.3 Glossary

1.3.1 Definitions

- Customer: person that has to buy something at the supermarket.
- User: customer with a smartphone that has downloaded CLup app and uses it.
- Non-User: customer that does not use the application.
- Store manager: person that administrates the store.
- Store capacity: maximum number of customers allowed in the store at the same time.
- QR Code: type of matrix barcode machine-readable.
- System: sum of hardware and software units dedicated to provide services and features guaranteed by the application.
- Ticket: Element generated by the system containing the QR Code.
- User city: the city in which the user looks for a supermarket.
- Time slot: A time window of half of an hour.
- Full time slot: time slot that has reached the maximum number of acceptable reservations set by the store manager.
- Past time slot: time slot that is no more bookable because its time window is over at the

user time.

- Free time slot: time slot that is not past or full, so available for reservation.
- Reservation time: time window booked at the supermarket by the user.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- CLup: Customer Line-Up (name of the application).
- QR Code: Quick Response code.

1.3.3 Test case codes

- TC[test case n.]_[requirement name]_[requirement n.]

1.4 References

- Project proposal: link to the document
- Course slides on WeBeep.

2. Overall Description

2.1 Testing environment

The tests were conducted exploiting the “AI Companion” functionality present in the MIT App Inventor environment, which allows to perform live testing without the need to build and install a new application at each test. At the end, the final tests were executed on the real application instead.
For what concerns the databases, some Google Sheet documents were created and used, which even if they are not adapt for a real case scenario, they were suitable for a functioning prototype thanks to their ease of use.
The communication between the application and the databases instead is regulated by some Google App Script codes, which, when called by App Inventor, are able to read and write into the various documents, updating in this way the database.

2.2 Testing methodology

The chosen approach was a Black-Box testing conducted step by step during all the implementation phase, with the intention to verify at each step the right functioning of the application, also covering the cases of boundary values and wrong inputs. At each detected error we proceeded performing debugging through White-Box testing directly on the App Inventor interface or the AppScript code, and at the end again Black-Box testing as confirmation of correction.

2.3 Traceability matrix

In this document the tests presented are organized following the requirement list present in the RASD, to show their correct implementation, both in case of correct and wrong inputs.

The following table describes the relation mapping between requirements and corresponding test cases, which validate their implementation.

Requirement:	**Validated by:**
R1: The system must allow users to sign up if and only if they are not already registered	TC01_SignUp_1 TC02_SignUp_1 TC03_SignUp_1 TC04_SignUp_1
R2: The system must allow users to sign in if and only if already registered and the credentials inserted are valid	TC05_LogIn_2 TC06_LogIn_2 TC07_LogIn_2
R3: The system must allow the user to log out	TC08_LogOut_3
R4: The system must allow the user to search and select the supermarket	TC09_MarketSelection_4
R5: The system must allow the user to choose the city	TC10_CitySelection_5
R6: The system must restrict the supermarket research to the ones that are in the user city	TC11_MarketRestriction_6
R7: The system shows time slots of the selected market according to available timetable and crowding data	TC12_TimeTable_7
R8: The system must allow the user to select the day on which they want to book in the selected supermarket, among the ones that the store manager makes available	TC13_DaySelection_8
R9: The system allows the user to make a reservation only in free time slots	TC14_Reservation_9 TC15_Reservation_9 TC16_Reservation_9
R10: The system must generate one and only one ticket per reservation	TC17_TicketGeneration_10
R11: The system must update the time slot crowding of the supermarket for each ticket created/deleted in that store	TC18_Crowding Update_11 TC19_Crowding Update_11
R12: Each ticket generated by the system must have a unique QR code	TC20_QRCode_12

R13: The system allows the user to make a reservation over more time slots only if they are all free, consecutive and in the same day	TC21_MultipleTimeSlots_13 TC22_MultipleTimeSlots_13
R14: The system allows the user to make a reservation only if it does not contain time slots already involved in existing reservations at the selected market on the same day	TC23_OverlappingReservations_14
R15: the system must not allow the user to have concurrent reservations over a maximum number	TC24_MaxTickets_15
R16: The system must allow the user to check their pending tickets	TC25_TicketCheck_16
R17: The system must allow the user to delete their pending tickets	TC26_TicketElimination_17
R18: The system does not allow the user to change single settings of already generated tickets	TC27_TicketImmutability_18
R19: The system must automatically delete the ticket at the end of its reservation time	TC28_TicketAutoElimination_19
R20: The system should allow the user to update their data and delete their account	TC29_DataUpdate_20 TC30_DataUpdate_20 TC31_DataUpdate_20 TC32_DataUpdate_20 TC33_DataUpdate_20 TC34_DataUpdate_20
R21: The system should allow the user to auto-log in if they have previously ticked a “Remember me” box	TC35_AutoLogin_21
R22: The system should allow the user to recover their credentials	TC36_CredentialRecovery_22 TC37_CredentialRecovery_22
R23: The system should alert the user when their account is created/deleted	TC38_AccountEmail_23 TC39_AccountEmail_23
R24: The system should filter supermarkets by name	TC40_MarketFilter_24 TC41_MarketFilter_24
R27: The system should allow the user to contact the assistance	TC42_Assistace_27 TC43_Assistace_27
R28: The system should allow the user to download the QR ticket to use it offline	TC44_OfflineQR_28

3. Tests

Below are presented all and only the tests performed on the actual last version (v 1.53.187).

3.1 Core requirement tests

- **R1**: The system must allow users to sign up if and only if they are not already registered.

Test Case ID	**TC01_SignUp_1**
Description	Register a new account inserting valid data
Test Case Procedure	▪ The system shows the login page ▪ Press the “SIGN UP” button ▪ The system redirects the user to the signup page ▪ Fill in with valid name, surname, email and password ▪ Press the “SIGN UP” button
Expected Output	▪ The system shows an alert confirming the registration ▪ The system redirects the user to the login page ▪ The new account is successfully created
Result	Pass
Note	

Test Case ID	**TC02_SignUp_1**
Description	Attempt of registration with an already existing email
Test Case Procedure	▪ The system shows the login page ▪ Press the “SIGN UP” button ▪ The system redirects the user to the signup page ▪ Fill in with an already existing email ▪ Press the “SIGN UP” button
Expected Output	▪ The system shows a warning saying the email is already registered and remains in the sign-up page ▪ The new account is not registered
Result	Pass
Note	

Test Case ID	**TC03_SignUp_1**
Description	Attempt of registration leaving blank fields
Test Case Procedure	▪ The system shows the login page ▪ Press the “SIGN UP” button ▪ The system redirects the user to the signup page ▪ Fill in with valid data leaving at least one field blank ▪ Press the “SIGN UP” button
Expected Output	▪ The system shows a warning saying to fill in al the fields and remains in the sign-up page ▪ The new account is not registered
Result	Pass
Note	

Test Case ID	**TC04_SignUp_1**
Description	Attempt of registration with mismatching passwords
Test Case Procedure	▪ The system shows the login page ▪ Press the “SIGN UP” button ▪ The system redirects the user to the signup page ▪ Fill in with valid data but inserting the wrong password in the last field ▪ Press the “SIGN UP” button
Expected Output	▪ The system shows a warning saying there is a mismatch between the passwords and remains in the sign-up page ▪ The new account is not registered
Result	Pass
Note	

- **R2**: The system must allow users to sign in if and only if already registered and the credentials inserted are valid.

Test Case ID	**TC05_LogIn_2**
Description	Log in the account filling with the right data
Test Case Procedure	▪ The system shows the login page ▪ Fill in the fields with the correct email and password ▪ Press the “LOG IN” button
Expected Output	▪ The system shows a welcome alert ▪ The user is redirected to his home page
Result	Pass
Note	

Test Case ID	**TC06_LogIn_2**
Description	Try to log in with an unregistered email
Test Case Procedure	▪ The system shows the login page ▪ Fill in the fields inserting an unregistered email ▪ Press the "LOG IN" button
Expected Output	▪ The system shows a warning saying the email is not registered and remains in the log in page
Result	Pass
Note	A blank email field is interpreted as an unregistered email case

Test Case ID	**TC07_LogIn_2**
Description	Try to log in with the wrong password
Test Case Procedure	▪ The system shows the login page ▪ Fill in the fields inserting a wrong password ▪ Press the "LOG IN" button
Expected Output	▪ The system shows a warning of wrong password and remains in the log in page
Result	Pass
Note	A blank password field is interpreted as a wrong password case

- **R3**: The system must allow the user to log out.

Test Case ID	**TC08_LogOut_3**
Description	Log out from the account
Test Case Procedure	▪ The system shows the home page ▪ Press the "Logout" button
Expected Output	▪ The system logs out from the user account and redirects the user to the log in page
Result	Pass
Note	

- **R4**: The system must allow the user to search and select the supermarket.

Test Case ID	**TC09_MarketSelection_4**
Description	Search and select the desired supermarket
Test Case Procedure	▪ The system shows the search page ▪ Type the name of the desired supermarket ▪ Press the name of the searched supermarket
Expected Output	▪ The system redirects to the page of the supermarket timetable
Result	Pass
Note	

- **R5**: The system must allow the user to choose the city.

Test Case ID	**TC10_CitySelection_5**
Description	Select the desired city
Test Case Procedure	▪ The system shows the search page ▪ Press the name of the actual city ▪ The system shows a list of available cities ▪ Select the desired city
Expected Output	▪ The system redirects to the search page, displaying the name of the new selected city
Result	Pass
Note	

- **R6**: The system must restrict the supermarket research to the ones that are in the user city.

Test Case ID	**TC11_MarketRestriction_6**
Description	Restrict the research to only the supermarkets present in the chosen city
Test Case Procedure	▪ The system shows the search page ▪ Type the name of the desired supermarket
Expected Output	▪ The system shows the supermarkets according to the search bar restricting the research to those present in the chosen city
Result	Pass
Note	This is a default function, so it is not possible to search supermarkets regardless of the city

- **R7**: The system shows time slots of the selected market according to available timetable and crowding data.

Test Case ID	**TC12_TimeTable_7**
Description	Visualize the timetable and crowding data of a supermarket
Test Case Procedure	▪ The system shows the search page ▪ Search the desired supermarket and select it
Expected Output	▪ The system shows the supermarket time-table page ▪ The time slots show the crowding level through a color codification
Result	Pass
Note	Pressing the "?" button on the bottom right, a legend with some information will be displayed

- **R8**: The system must allow the user to select the day on which they want to book in the selected supermarket, among the ones that the store manager makes available.

Test Case ID	**TC13_DaySelection_8**
Description	Select the desired day on which to make the reservation
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the arrows near the date
Expected Output	▪ The system navigates through the available days according to the arrows pressed ▪ The system shows the right opening and closing times, based on the selected day (if not open it shows a "CLOSED" sign)
Result	Pass
Note	In the actual version is possible to choose only between the current day and the next one

- **R9**: The system allows the user to make a reservation only in free time slots.

Test Case ID	**TC14_Reservation_9**
Description	Make a reservation in a free (white or yellow) time slot
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the desired time slot (white or yellow) ▪ The system shows the buttons "CLEAR" and "BOOK" ▪ Press the "BOOK" button ▪ The system shows a summary alert asking for confirmation ▪ Press the "Confirm" button
Expected Output	▪ The new reservation is created, and the system redirects the user to the ticket page
Result	Pass
Note	

Test Case ID	**TC15_Reservation_9**
Description	Attempt to make a reservation in an unavailable (grey or red) time slot
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the desired time slot (grey or red)
Expected Output	▪ The system does not allow to select the time slot
Result	Pass
Note	

Test Case ID	**TC16_Reservation_9**
Description	Attempt to make a reservation in a free time slot, taken after the page has loaded the time slot colors
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the desired time slot (white or yellow) ▪ The system shows the “CLEAR” and “BOOK” buttons ▪ Press the “BOOK” button ▪ The system shows a summary alert asking for confirmation ▪ From the database change the crowding of that time slot to make it unavailable ▪ Press the “Confirm” button
Expected Output	▪ The system shows a warning saying the time slot is no more available ▪ The reservation is not created ▪ The time-table colors are refreshed with the updated crowding
Result	Pass
Note	

- **R10**: the system must generate one and only one ticket per reservation.

Test Case ID	**TC17_TicketGeneration_10**
Description	Generate a ticket for each new reservation
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the desired time slot (white or yellow) ▪ The system shows the “CLEAR” and “BOOK” buttons ▪ Press the “BOOK” button ▪ The system shows a summary alert asking for confirmation ▪ Press the “Confirm” button
Expected Output	▪ The system generates a ticket, and the user is redirected to its page
Result	Pass
Note	

- **R11**: The system must update the time slot crowding of the supermarket for each ticket created/deleted in that store.

Test Case ID	**TC18_Crowding Update_11**
Description	Update the time slots color when a certain number of tickets is created
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ From the database set the crowding level of a time slot such that the next created ticket will make the color change ▪ Book that time slot from the app ▪ The system shows the created ticket ▪ Click the “Back” button
Expected Output	▪ The system shows the supermarket timetable with the time slot color of the just booked ticket changed (from white to yellow or from yellow to red)
Result	Pass
Note	

Test Case ID	**TC19_Crowding Update_11**
Description	Update the time slots color when prior created tickets are deleted
Test Case Procedure	▪ The system shows the home page with at least one previously created ticket ▪ Press the “CHECK” button ▪ From the database set the crowding level of the booked time slot such that the next eliminated one will make the color change ▪ Press the “DELETE” button ▪ The system shows an alert asking for confirmation ▪ Press the “Yes” button ▪ The system redirects the user to the home page ▪ Go to the supermarket time-table page
Expected Output	▪ The system shows the supermarket timetable with the time slot color of the just deleted ticket changed (from red to yellow or from yellow to white)
Result	Pass
Note	This will work also with deleted tickets due to account cancellation

- **R12**: Each ticket generated by the system must have a unique QR code.

Test Case ID	**TC20_QRCode_12**
Description	Generate a unique QR code for each ticket
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Press the desired time slot (white or yellow) ▪ The system shows the "CLEAR" and "BOOK" buttons ▪ Press the "BOOK" button ▪ The system shows a summary alert asking for confirmation ▪ Press the "Confirm" button
Expected Output	▪ The system generates a ticket, and the user is redirected to its page ▪ The page contains a unique scannable QR code having the booking info
Result	Pass
Note	

- **R13**: The system allows the user to make a reservation over more time slots only if they are all free, consecutive and in the same day.

Test Case ID	**TC21_MultipleTimeSlots_13**
Description	Make a reservation in consecutive and free (white or yellow) time slots
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Select the desired consecutive and free (white or yellow) time slots ▪ Press the "BOOK" button ▪ The system shows a summary alert asking for confirmation ▪ Press the "Confirm" button
Expected Output	▪ The new ticket is created taking the start time of the first selected time slot and the end time of the last one ▪ The user is redirected to the ticket page
Result	Pass
Note	

Test Case ID	**TC22_MultipleTimeSlots_13**
Description	Attempt to make a reservation in non-consecutive time slots
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Select the desired nonconsecutive time slots
Expected Output	▪ The system does not allow to select nonconsecutive time slots
Result	Pass
Note	

- **R14**: The system allows the user to make a reservation only if it does not contain time slots already present in existing reservations at the selected market on the same day.

Test Case ID	**TC23_OverlappingReservations_14**
Description	Attempt to make a reservation that overlaps with another existing for the same supermarket in the same day
Test Case Procedure	▪ The system shows the supermarket time-table page ▪ Select the desired time slots, overlapping with the ones of an already existing reservation for the same supermarket in the same day ▪ Press the “BOOK” button ▪ The system shows a summary alert asking for confirmation ▪ Press the “Confirm” button
Expected Output	▪ The system shows a warning saying an overlapping occurred ▪ The new ticket is not generated
Result	Pass
Note	This works whether the overlapping is total or partial

- **R15**: the system must not allow the user to have concurrent reservations over a maximum number.

Test Case ID	**TC24_MaxTickets_15**
Description	Try to make more reservations than allowed (6)
Test Case Procedure	▪ Make 6 valid reservations ▪ Make another valid reservation
Expected Output	▪ The system shows a warning saying the max number of pending reservations has been reached ▪ The new reservation is not registered
Result	Pass
Note	

- **R16**: The system must allow the user to check their pending tickets.

Test Case ID	**TC25_TicketCheck_16**
Description	Check a previously generated ticket
Test Case Procedure	▪ The system shows the home page with at least one previously generated ticket ▪ Press the “CHECK” button of the desired reservation
Expected Output	▪ The system redirects the user to the ticket page, containing all the reservation info, the QR code and the “DELETE” button
Result	Pass
Note	

- **R17**: The system must allow the user to delete their pending tickets.

Test Case ID	**TC26_TicketElimination_17**
Description	Delete a previously generated ticket
Test Case Procedure	▪ The system shows the home page with at least one previously generated ticket ▪ Press the "CHECK" button of the desired reservation ▪ The system shows the ticket page ▪ Press the "DELETE" button ▪ The system shows an alert asking for confirmation ▪ Press the "Yes" button
Expected Output	▪ The system deletes the reservation and the relative ticket and redirects the user to the home page
Result	Pass
Note	

- **R18**: The system does not allow the user to change single settings of already generated tickets.

Test Case ID	**TC27_TicketImmutability_18**
Description	Attempt to change single settings of previously generated tickets
Test Case Procedure	▪ The system shows the home page with at least one previously generated ticket ▪ Press the "CHECK" button of the desired reservation ▪ The system shows the ticket page ▪ Try to change single settings of the ticket
Expected Output	▪ The system does not allow to modify the ticket, only to delete it
Result	Pass
Note	

- **R19**: The system must automatically delete the ticket at the end of its reservation time.

Test Case ID	**TC28_TicketAutoElimination_19**
Description	Automatic elimination of expired reservation tickets
Test Case Procedure	▪ Make a reservation ▪ Wait until the ticket is expired ▪ Go to the home page
Expected Output	▪ The system shows only the still valid tickets, and has eliminated the expired ones
Result	Pass
Note	This control and eventual elimination of expired tickets happens only when the user load his home page. Until then the tickets remain in the database

3.2 Medium priority requirement tests

- **R20**: The system should allow the user to update their data and delete their account.

Test Case ID	**TC29_DataUpdate_20**
Description	Update user info inserting valid data
Test Case Procedure	▪ The system shows the setting page ▪ Press the button near “Personal Info” ▪ The system shows the “Update Info” page ▪ Update the desired fields with valid data ▪ Press the “UPDATE” button
Expected Output	▪ The system redirects the user to the setting page ▪ The personal info, shown also in the setting page, are updated according to the user inputs
Result	Pass
Note	

Test Case ID	**TC30_DataUpdate_20**
Description	Attempt to update user info leaving blank fields
Test Case Procedure	▪ The system shows the setting page ▪ Press the button near “Personal Info” ▪ The system shows the “Update Info” page ▪ Update the desired fields leaving at least one blank field ▪ Press the “UPDATE” button
Expected Output	▪ The system shows a warning saying to fill all the fields ▪ The user info are not updated
Result	Pass
Note	

Test Case ID	**TC31_DataUpdate_20**
Description	Attempt to update user info with mismatching passwords
Test Case Procedure	▪ The system shows the setting page ▪ Press the button near “Personal Info” ▪ The system shows the “Update Info” page ▪ Update the desired fields inserting mismatching passwords ▪ Press the “UPDATE” button
Expected Output	▪ The system shows a warning saying that the passwords do not match ▪ The user info are not updated
Result	Pass
Note	

Test Case ID	**TC32_DataUpdate_20**
Description	Attempt to update user info with an already existing email
Test Case Procedure	▪ The system shows the setting page ▪ Press the button near “Personal Info” ▪ The system shows the “Update Info” page ▪ Update the desired fields inserting an already existing email ▪ Press the “UPDATE” button
Expected Output	▪ The system shows a warning saying that the email is already used by another account ▪ The user info are not updated
Result	Pass
Note	

Test Case ID	**TC33_DataUpdate_20**
Description	Elimination of the account inserting the right password
Test Case Procedure	▪ The system shows the setting page ▪ Press the “DELETE ACCOUNT” button ▪ The system shows a dialog box asking to confirm with the password ▪ Insert the right password ▪ Press the “Confirm” button
Expected Output	▪ The system redirects the user to the login page ▪ The new account is deleted
Result	Pass
Note	Deleting the account will cancel all the pending tickets and update the relative time slots crowding

Test Case ID	**TC34_DataUpdate_20**
Description	Attempt to delete the account inserting the wrong password
Test Case Procedure	▪ The system shows the setting page ▪ Press the “DELETE ACCOUNT” button ▪ The system shows a dialog box asking to confirm with the password ▪ Insert the wrong password ▪ Press the “Confirm” button
Expected Output	▪ The system shows a warning saying that the password is not correct ▪ The account is not deleted
Result	Pass
Note	A blank password field is interpreted as a wrong password case

- **R21**: The system should allow the user to auto-log in if they have previously ticked a "Remember me" box.

Test Case ID	**TC35_AutoLogin_21**
Description	Automatic login of the user
Test Case Procedure	▪ The system shows the login page ▪ Fill in the fields with correct data ▪ Check the "Remember me" box ▪ Press the "LOG IN" button ▪ The system redirects the user to the home page ▪ Close the application ▪ Open the application
Expected Output	▪ The system automatically logs in the user, and shows the home page
Result	Pass
Note	This procedure checks if the ip address is the same of when we checked the "Remember me" box, so changing the internet connection will disable the autologin

- **R22**: The system should allow the user to recover their credentials.

Test Case ID	**TC36_CredentialRecovery_22**
Description	Recovery of a forgotten password
Test Case Procedure	▪ The system shows the login page ▪ Insert the email in the proper field ▪ Press "Forgot Password?"
Expected Output	▪ The system sends you an email containing the forgotten password
Result	Pass
Note	

Test Case ID	**TC37_CredentialRecovery_22**
Description	Recovery of a forgotten password inserting an incorrect email
Test Case Procedure	▪ The system shows the login page ▪ Insert the email in the proper field ▪ Press "Forgot Password?"
Expected Output	▪ The system shows a warning saying the email is wrong ▪ The email with the password is not sent
Result	Pass
Note	If the email corresponding to someone else account is inserted, this person will receive the email with his password. This should not affect the security of his account

- **R23**: The system should alert the user when their account is created/deleted.

Test Case ID	**TC38_AccountEmail_23**
Description	Create a new account
Test Case Procedure	▪ The system shows the login page ▪ Go to the sign-up page ▪ Fill in with valid name, surname, email and password ▪ Press the "SIGN UP" button
Expected Output	▪ The user receives a welcome email in the registered address
Result	Pass
Note	

Test Case ID	**TC39_AccountEmail_23**
Description	Delete an account
Test Case Procedure	▪ The system shows the setting page ▪ Press the "DELETE ACCOUNT" button ▪ The system shows a dialog box asking to confirm with the password ▪ Insert the right password ▪ Press the "Confirm" button
Expected Output	▪ The user receives an email informing that his account has been deleted
Result	Pass
Note	

- **R24**: The system should filter supermarkets by name.

Test Case ID	**TC40_MarketFilter_24**
Description	Filter the supermarkets using an existing name
Test Case Procedure	▪ The system shows the search page ▪ Write the desired (existing) supermarket name on the search bar
Expected Output	▪ The system shows all and only the supermarkets with the searched name, present in the selected city
Result	Pass
Note	

Test Case ID	**TC41_MarketFilter_24**
Description	Filter the supermarkets using a non-existing name
Test Case Procedure	▪ The system shows the search page ▪ Write a non-existent supermarket name on the search bar
Expected Output	▪ The system does not show any supermarket below the search bar
Result	Pass
Note	

3.3 Good to have requirement tests

- **R27**: The system should allow the user to contact the assistance.

Test Case ID	**TC42_Assistace_27**
Description	Contact the support service
Test Case Procedure	▪ The system shows the account page ▪ Press the "Support" button ▪ Fill in both the subject and the problem description field ▪ Press the "SEND" button
Expected Output	▪ The system shows an alert saying the email has been sent ▪ The support service receives an email with the written subject and problem description ▪ The user is redirected to the account page
Result	Pass
Note	

Test Case ID	**TC43_Assistace_27**
Description	Attempt to contact the support service leaving blank fields
Test Case Procedure	▪ The system shows the account page ▪ Press the "Support" button ▪ Leave one or both the fields blank ▪ Press the "SEND" button
Expected Output	▪ The system shows a warning saying to fill all the fields ▪ The email to the support service is not sent
Result	Pass
Note	

- **R28**: The system should allow the user to download the QR ticket to use it offline.

Test Case ID	**TC44_OfflineQR_28**
Description	Save the QR code in the smartphone gallery
Test Case Procedure	▪ The system shows the home page with previously created tickets ▪ Press the "CHECK" button of the desired ticket ▪ Make a screenshot of the QR code
Expected Output	▪ The QR code is saved in the photo gallery of the user smartphone, and it is so available offline
Result	Pass
Note	There is not a specific implemented function to save QR codes since the user can easily take screenshots to save them