

A.Y. 2020-2021

Politecnico di Milano, Software Engineering 2 project



POLITECNICO
MILANO 1863

ATD

Acceptance Testing Document

version 1.0

Cirino Duilio - 968466

Cocchia Lorenzo - 968139

Table of contents

1. <u>INTRODUCTION</u>	<u>3</u>
1.1 Structure of the document	3
2. <u>INSTALLATION SETUP PROCESS</u>	<u>5</u>
3. <u>TESTS SECTION</u>	<u>6</u>
3.1 Technical information about the tests	6
3.2 Acceptance tests	6
3.3 Requirements satisfied	13
3.4 More notes and final thought on the tests conducted	18
4. <u>REFERENCES</u>	<u>19</u>
5. <u>EFFORT SPENT</u>	<u>20</u>

1. INTRODUCTION

This document is produced to document the acceptance tests done to the prototype of the project assigned, including how they have been done and their rationale. The tested project assigned is:

Name of the authors	Roberto Buratti , Hrvoje Hrvoj , Ozan Incesulu
Link to the Git repository	https://github.com/Furcanzo/BurattiIncesuluHrvoj

In order to do acceptance tests, we used the following documents from the delivery folder:

- Implementation & Testing Document v. 1.0: in order to install and set up the project, understand what are the functionalities implemented and what is the rationale behind the implementation of the prototype;
- Requirements Analysis & Specification Document v. 2.0.1: in order to understand what are the goals and the requirements expected to the accomplishment of them and the domain assumption that have been done;
- Design Document v. 2.0.0: in order to understand what is the expected dynamic behaviour of the listed use cases and the overall architecture.

1.1. Structure of the document

This document, other than this introductory section contains:

- Installation setup process. A section based on the assessment of the installation phase described in the ITD;
- Tests section. Aimed at describing how the tests have been done, through which framework, which were their aim, the expected result and the outcome, even related to the requirements stated in the RASD;
- References. A section in which we have listed all the referenced documents used for the purpose of this document;
- Effort spent. Describing how much effort it has been paid by each member of the team.

2. INSTALLATION SETUP PROCESS

The installation has been accompanied by Maven. This resulted in a great simplicity while installing the prototype, without giving any particular problem and being very fast. The only little detail that we did not find on the ITD was that, at least in our case, we could not figure out how to deploy the artifact if not with the 1.8 JDK version, it did not give us any particular problem in any case.

3. TESTS SECTION

This section will describe the technical background on which we relied in order to accomplish the acceptance tests, what were them and their outcome. In the first section we introduce the software that we used to accomplish the tests, on the second we described the details of the tests.

3.1. Technical information about the tests

In order to produce replicable and clear tests we used Selenium, a software that can be used even as a browser plugin that automatically creates test cases directly basing on the behaviour that we had in our browsers. When running a selenium test it automatically behaves as a user accomplishing the actions specified in the test such as clicking on some button, or in some HTML form, insert data specified in the test, submit and so on. Moreover, it has a system so that it can automatically verify some facts such as the presence of some banner (we used it with the feedback banners). In any case you can find the description of each test that we made through Selenium IDE inside the ATD folder in our repository in some file. The files are in an extension .SIDE which is readable through the Selenium IDE, a plug-in available for Chrome and Firefox browsers. When not specified, consider that we had to recreate the state of the dates for each test in order to satisfy the input conditions for each test case.

3.2. Acceptance tests

First of all, we read which were the functionalities implemented. Additionally, we considered even the login and registration phases (represented as test case number 8) as a matter of test because we believe that these functionalities were important to the functionalities of the application. Indeed, without being registered no user or manager, or employee, can use the application. Based on these functionalities we tried the tests with different inputs, fixing an expected outcome of the test and verifying which is the actual outcome. We started to create the table below, fixing the test cases, the input conditions and the outcome expected. Then, we verified the outcome given the input conditions. We referred even to the dynamic behaviours modelled through UML

sequence diagrams found in the DD [2.4] in order to know what to expect by the outcome of each test case. The reader can find even some additional notes about every test case attached.

To give a more schematic look on the test procedures we followed, we can divide them as:

- Correct behaviour tests
- Bad input tests
- Empty input tests
- Extreme cases tests

Test case number	Input parameters	Expected outcome	Actual outcome	Notes
1	Create store (use case 2.4.13)			
	Input conditions Login is made through the Super User credentials			
1.1	The input parameters are right	The store and the manager are inserted into the system	The store and the manager are inserted into the system	Actually, the UI did not allow the insert of all the information about the store, but only the manager's name and store's name
1.2	The input is repetitive → store's name	The store is not inserted into the system, the manager is inserted into the system	The store and the manager are inserted into the system	This can actually be considered as a design decision, but from the ITD it should be clearer
1.3	The input is repetitive → manager's email	The store is inserted into the system, but the manager is not inserted	The store and the manager are inserted into the system	

		into the system		
1.4	The input is incomplete	The operation is not accomplished, an error message appears	The operation is not accomplished, an error message appears	The email without the domain and the '@' character produces the same outcome of test 1.1
2	Update store information (use case 2.4.9)			
	Input conditions Manager's login and access to Update Store page, the store is already existent			
2.1	The input parameters are right	The operation is succeeded	The operation is succeeded	
2.2	Input is repetitive → The name of the store is already existent	The operation is not accomplished, an error message appears	The operation is succeeded	
2.3	Input is semantically wrong → The capacity is equal or less than 0	The operation is not accomplished, an error message appears	The operation is succeeded	
2.4	Input is semantically wrong → The ticket timespan is equal or less than 0	The operation is not accomplished, an error message appears	The operation is succeeded	

2.5	Input is semantically wrong → The partner store does not exist	The operation is not accomplished, an error message appears	The operation is not accomplished, an error message appears	
2.6	Input is semantically wrong → Opening or closing hour are numbers less than 0 or greater than 24	The operation is not accomplished, an error message appears	The operation is succeeded	
2.7	Input is semantically wrong → Opening and closing hour are the same number	The operation is not accomplished, an error message appears	The operation is succeeded	This can actually be considered as a design decision
3	Book Future Line Number (use case 2.4.5)			
	Input conditions The customer is logged with her credentials and at least one store is available for a future booking			
3.1	Input parameters are right	The booking is successful	The booking is successful	It is not possible to book tickets for after 8-2-2021 (tested the 11-2-2021)
3.2	Input is semantically wrong → the date is preceding the actual one	The booking is not accomplished, an error message appears	The booking is successful	

3.3	Input is semantically wrong → a reservation for a certain date has been already done by the customer	The booking is not accomplished, an error message appears	The booking is successful	This can actually be considered as a design decision
3.4	Input is semantically wrong → the booking date is in an hour in which the store is closed	The application does not give the possibility to reserve for that hour	The application does not give the possibility to reserve for that hour	
4	Retrieve a Line Number (use case 2.4.5)			
	Input conditions A store is actually available to retrieve a number			
4.1	Customer and clerk ask to get a number	The number is retrieved	The number is retrieved	In the clerk case, it seems that Generate QR Code and Scan QR code tabs are switched in the UI
4.2	The store is full	The number is not available at the moment	The number is not available at the moment	The clerk can in any case generate the number, while the customer is informed of the unavailability

5	Add Staff Member			
	Input conditions A manager is registered correctly and opens the Manage Staff view			
5.1	The email is inserted correctly	The system adds the new employee to the system	The system adds the new employee to the system	
5.2	The email is empty	The system prompts us an alert saying that the field must not be empty	The system adds the new employee to the system	This is wrong because you should not let an empty user register in the system, more so if you don't let him log in.
5.3	The email is correct but a employee with that email exists	The system prompts an error	The system prompts an error	
5.4	An employee is already in the database, we change his role	The system changes correctly the role	The system changes correctly the role	
5.5	An employee is added, with no errors, to the system and tries to login	The system lets him login and redirects him to the correct view	The system lets him login and redirects him to the correct view	For this the initial input condition of the product function must not be met, but we decided to put this test in this section because it is strictly related
6	Monitor Customers			

	Input conditions A manager is registered correctly			
6.1	The manager is logged in	We can see the number of people inside the store	We can see the number of people inside the store	There's no much to test regarding this functionality, it simply retrieves the number of actually VISITING people, his correct functioning relies on the other product functionality
7	Grant Access			
	Input conditions A store is actually available to retrieve a number			
				We did not manage to use this function because a popup is not shown to our web app, it should, but it doesn't after several tries and changes, even in different browsers and environments.
8	Login and Registration			
	Input conditions CLup services are up			
8.1.1	The input parameters are correct	The system allow the login	The system allow the login	
8.1.2	The input is incorrect → Email not existing inserted	The system doesn't allow the login	The system doesn't allow the login	
8.1.3	The input is incorrect →	The system doesn't	The system	

	Email field empty	allow the login	doesn't allow the login	
8.2.1	The input parameters are correct	The system allow the login	The system allow the login	
8.2.2	The input is incorrect → Telephone number is not a number	The system prompts an error and says to retry	The system allow the login	This is a logical problem more than a structural one, the system should not permit a String as input, it would be acceptable if stated in the ITD
8.2.3	The input is incorrect → Two different emails are inserted	The system prompts an error and says to retry	The system prompts an error and says to retry	
8.2.4	The input is incorrect → A field is missing from the form	The system prompts an error and says to retry	The system prompts an error and says to retry	
8.2.5	The input is incorrect → An empty form is sent	The system prompts an error and says to retry	The system prompts an error and says to retry	
8.2.6	The input parameters are correct but a user with that email exists	The system prompts an error and says to retry	The system prompts an error and says to retry	

3.3. Requirements satisfied

In this section we are going to assess which are functionalities that are satisfying each of the requirements declared in the RASD [3.2.5]. We will refer to the numeration of the functionalities as F_n where n is the number of functionality declared on the preceding section. If we think that some additional note is needed, the reader can find an additional column attached to the table.

Requirement identifier	Description	F_n	Notes
R_1	The system must allow users to authenticate using an external SSO provider.	F_8	The SSO is not a third-party software, but this being a prototype it is ok.
R_2	The system must allow customers to register using their email address, name, surname, and phone number after authenticating through SSO.	F_8	
R_3	Managers must be able to add additional managers and clerks as users.	F_5	
R_4	Managers must be able to set and update location-specific information, that is, the maximum number of customers in the location at any given time, opening and closing hours of the store per each day, line number timeout, the limit of reservation per customer on a predetermined time interval that is one of the month, week or day, and location of the place	F_2	
R_5	Managers can add any other location as a partner store.	F_2	
R_6	Managers can stop the system from issuing any more tickets for a given day	---	The stop feature is not listed as the ones implemented

R ₇	Managers can schedule the system stop for a future time	---	The stop feature is not listed as the ones implemented
R ₈	Managers can set in-shop locations for different product categories.	---	The feature consisting on setting the in-shop locations is not listed as the ones implemented
R ₉	In case of a system stop, no other line numbers can be issued for the given time slots.	---	The stop feature is not listed as the ones implemented
R ₁₀	In case of a system stop, all line numbers in the stop time slots have to be canceled.	---	The stop feature is not listed as the ones implemented
R ₁₁	The system must cancel those line numbers that the customer did not arrive at the store for more than the set timeout interval.	---	We were not able to test it, and it is not very clear if this requirement can be included in one of the functionalities implemented
R ₁₂	In case of ticket cancellation, the customer must be notified with an e-mail notification.	---	The e-mail service is not listed as the ones implemented
R ₁₃	Clerks must register the customers' entrance and exit via scanning the QR code for their line number.	F ₇	We were not able to test the functionality
R ₁₄	Clerks must be able to generate line number tickets in a compatible printer format.	F ₄	
R ₁₅	Customers must be able to obtain a line number, except when the system is stopped or the	F ₄	The stop feature is not listed as the ones implemented

	store is full.		
R ₁₆	Customers must be able to obtain line numbers for different time slots in the future.	F ₃	
R ₁₇	Customers can not obtain line numbers for time intervals that the system is stopped by a manager.	---	The stop feature is not listed as the ones implemented
R ₁₈	Customers must be able to see the estimated time available for their line number.	F ₃	
R ₁₉	Customers must be able to set or update their phone number, name, and surname.	F ₈	
R ₂₀	Customers can select a specific product categories they plan to visit in the location while obtaining a line number.	---	The feature consisting on setting the in-shop locations is not listed as the ones implemented
R ₂₁	Customers can set an estimated time for their visit while obtaining a line number.	F ₃	
R ₂₂	Customers must be able to view the shop location	F ₃	
R ₂₃	Customers can view the occupation forecasts for the location at different time slots.	---	
R ₂₄	Customers can see the alternative suggestions for time slots while obtaining a line number for the future	F ₃	We were not able to see actually "future" time slots but only past ones

R ₂₅	Customers can view the partner stores occupancies' if the preferred time slot is not available while obtaining a line number.	---	
R ₂₆	The system must be able to provide a forecast for each location's occupancy for any given time based on past visits.	---	
R ₂₇	The system must allow its supervisors to create stores and managers through super users.	F ₁	

There is no reference of the F₆ functionality in the requirements. To sum up:

Functionality	Requirements satisfied
F ₁	R ₂₇
F ₂	R ₅
F ₃	R ₁₆ , R ₁₈ , R ₂₁ , R ₂₂ , R ₂₄
F ₄	R ₁₄ , R ₁₅
F ₅	R ₃
F ₆	---
F ₇	R ₁₃
F ₈	R ₁ , R ₂ , R ₁₉

3.4. More notes and final thoughts on the tests conducted

We found it difficult to extract some information on the usage of the program, the ITD in fact seems to leave some crucial aspects of the usage

implicit, like they would be of easy understanding for everyone, which is not the case.

One example of this can be identified in the Back Office User, which is a crucial part of the program, because without it you cannot create groceries and the related managers, but inside the implementation document there was never a reference on how to use it, so we had to figure it out. Back Office User that was introduced as a name change for the SuperUser mentioned in the RASD.

If it was not clear from the test notes, we encountered a strange bug for which the calendar of available reservations was “shifted” of four weeks behind the current date, so that if I wanted to make a reservation on the 11-02-2021, the first available date proposed by the system was 11-01-2021.

Another issue regarded the performance of the program that struggled by just adding one store and trying to manage it, and very few reservations. This has something to do with the reservation system that is really slow because of some really CPU intensive computation, that really affects the overall experience.

Finally in general as it can be assumed by the tests we performed the main issue is the lack of control over user input that many times can lead to unexpected behaviours and exceptions being thrown at runtime. In overall the functionalities implemented work if used correctly, even if sometimes there can be some performance issue or random errors caused by the lack of input checks.

4. REFERENCES

- Git repositories:
<https://github.com/duiliocirino/CirinoCocchia/>
<https://github.com/Furcanzo/BurattiIncesuluHrvoj/>

5. EFFORT SPENT

Duilio Cirino:

11-2	6	Installation of the project to test and setting of the ATD strategy
12-2	3	Further take on trying to test function 7
13-2	3	Final revision of the documentation and fixes.

Lorenzo Cocchia:

11-2	6	Installation of the project to test and setting of the ATD strategy
12-2	4	Acceptance testing and documentation
13-2	4	Acceptance testing and documentation