



POLITECNICO DI MILANO
Computer Science and Engineering

Acceptance Testing Deliverable

eMall - e-Mobility for All

Software Engineering 2 Project
Academic year 2022 - 2023

09 February 2023
Version 1.0

Authors:
Balestrieri Niccolò
Bertogalli Andrea
Tombini Niccolò

Professor:
Matteo Camilli

Version history

Date	Revision	Notes
09/02/2023	v.1.0	First release.

Contents

Contents	1
1 Introduction	2
1.1 Purpose	2
1.2 Scope	2
2 Installation and setup	3
3 Acceptance test cases	4
3.1 Test cases	5
4 Additional notes	10
4.1 Source code documentation	10
4.2 General opinion about the whole project	10
5 Effort Spent	11
5.1 Team discussions	11
5.2 Balestrieri Niccolò	11
5.3 Bertogalli Andrea	11
5.4 Tombini Nicolò	11

Introduction

1.1 Purpose

This document serves as the Acceptance Test Deliverable (ATD) for the e-Mall project. Its main objective is to thoroughly evaluate and test the project that has been developed by another group, with the aim of highlighting any potential bugs or system issues, as well as any discrepancies between the requirements described in the RASD (Requirements and Analysis Specification Document) and DD (Design Document) documents and the actual implementation.

The ATD is an important step in the software development process, as it allows the project team to verify that the final product meets all the specified requirements and functions as intended before it is released to the customer. This in turn helps to ensure the overall quality and success of the project.

1.2 Scope

The project being analyzed was developed by Wu Tianjian Ren Yifan. The source code and all relevant documents, including the RASD, DD, ITD, and any others pertaining to the project, are included. Below the github repository of the analyzed project:

[Wu Tianjian Ren Yifan project repository](#)

Installation and setup

To install the project we followed the installation instructions provided in the ITD document. The platform includes a MySQL database, a Redis database, and Java code for the eMall application. The instructions cover downloading and installing Docker, pulling and running Docker containers for the MySQL and Redis databases, setting up the database connection in IDEA, executing SQL statements in the database, decompressing code for the eMall application, and setting up the Java environment in IDEA to run the eMall modules.

Acceptance test cases

Initially, we compared the documents provided along with the application code, The first thing we noticed is that the the introductory chapter is identical in both RASD and DD, so we think that there is a lack of attention on this aspect. Due to the fact explained above in the introduction chapter in the DD the document structure doesn't follow the guidelines provided during the course.

Some sections of the RASD are quite short, and perhaps some diagrams are missing, but overall the document is clear and well structured. The **Formal analysis using alloy** maybe is too short and lacks details such as assertions and predicates, also no world is reported.

Also for the DD, there are some flaws to report. The non-high-level component diagram components are missing and the descriptions are a bit offhand. In the deployment, a more detailed explanation of the tiers could have been useful for those reading the document. In the run-time view, it would have been useful to have a short but concise description of the sequence diagrams, however they are well done. The **Other Design Decision** chapter has not been developed but left within the document, the description of the database structure using the **er schema** and the **relational schema** would have been appropriate. The flow diagrams of the emsp and cpms are missing in the user interface design. The **Requirement Traceability** are well done. The **Implementation, Integration, and Test Plan** could be more detailed, but it is good.

The installation guide is clear for what is written, but some steps should be added that may come in handy to make the installation easier.

Before testing the application in actual usage conditions, we wanted to make sure that the basic building blocks and functionalities of the application were working as intended. This is why we tested the application following the tests provided in the ITD, which are designed to test the single functionalities of the application.

Then we have tested the application in realistic scenarios, to see how it performs when used by a realistic user.

Therefore, we tried various test cases to simulate different types of user interactions with the application, and monitored its performance in each scenario. These tests covered various aspects of the application, from functionality and usability, to security and performance.

The majority of the tests that were performed are listed in the following section for reference.

3.1 Test cases

Id	AT001_User_SignUp
Description	Creation of a new user.
Input	Data required in the form.
Expected output	New account creation.
Test results	New account created.
Outcome	Positive.
Notes	None.

Id	AT002_User_SignIn
Description	Login into user account.
Input	Username and password.
Expected output	Dashboard access.
Test results	Dashboard viewed correctly.
Outcome	Positive.
Notes	None.

Id	AT003_User_SignOut
Description	Logout from user profile.
Input	None.
Expected output	User correctly logs out.
Test results	The user logs out correctly.
Outcome	Positive.
Notes	None.

Id	AT004_User_UpdateAccount
Description	The user can edit the information of his profile.
Input	Requested form inputs.
Expected output	The information updates correctly.
Test results	The information updates correctly.
Outcome	Positive.
Notes	In the phone number field there aren't any format control.

Id	AT005_User_SearchStation
Description	The sistem shows to the user the nerby stations.
Input	None
Expected output	The nearby station are shown correctly.
Test results	The nearby stations are shown.
Outcome	Positive.
Notes	None.

Id	AT006_User_BookCharge
Description	The user can book a time-slot in a station.
Input	The socket type.
Expected output	The nearby station are shown correctly
Test results	The nearby stations are shown.
Outcome	Positive
Notes	It's not possible to book for a specific date, only at the current time.

Id	AT007_Cpo_SignUp
Description	Creation of a new cpo.
Input	Data required in the form.
Expected output	New cpo account creation.
Test results	New cpo account created.
Outcome	Positive.
Notes	The password can't be choosed, all the accounts have the same password

Id	AT008_Cpo_SignIn
Description	Login into cpo account.
Input	Email and password.
Expected output	Dashboard access.
Test results	Dashboard viewed correctly.
Outcome	Positive.
Notes	None.

Id	AT009_Cpo_SignOut
Description	Logout from cpo profile.
Input	None.
Expected output	Cpo correctly logs out.
Test results	The Cpo logs out correctly.
Outcome	Positive.
Notes	None.

Id	AT010_Cpo_UpdateAccount
Description	The cpo can edit the information of his profile.
Input	.
Expected output	The information are updated.
Test results	The information are updated correctly.
Outcome	Positive.
Notes	No format control in update form.

Id	AT011_Cpo_AddStation
Description	The cpo can add a station.
Input	Requested form inputs.
Expected output	The station is insterted.
Test results	The station is inserted correctly.
Outcome	Positive.
Notes	None.

Id	AT012_Cpo_EditStation
Description	The cpo can edit a station.
Input	Requested form inputs.
Expected output	The station is modified.
Test results	The station is modified correctly.
Outcome	Positive.
Notes	None.

Id	AT013_Cpo_DeleteStation
Description	The cpo can delete a station.
Input	None.
Expected output	The station is deleted.
Test results	The station is removed correctly.
Outcome	Positive.
Notes	None.

Id	AT014_Cpo_AddSocket
Description	The cpo can add a Chrging socket.
Input	Requested form inputs.
Expected output	The Charging socket is created.
Test results	The Charging socket is created correctly.
Outcome	Positive.
Notes	None.

Id	AT015_Cpo_DeleteSocket
Description	The cpo can delete a Chrging socket.
Input	None.
Expected output	The Charging socket is deleted.
Test results	The Charging socket is deleted correctly.
Outcome	Positive.
Notes	None.

Id	AT016_Cpo_ModifySocket
Description	The cpo can modify a Chrging socket.
Input	Requested form inputs.
Expected output	The Charging socket is updated.
Test results	The Charging socket is updated correctly.
Outcome	Positive.
Notes	None.

Id	AT017_Cpo_AddPriceStrat
Description	The cpo can add a pricing strategy.
Input	Requested form inputs.
Expected output	The pricing strategy is inserted.
Test results	The pricing strategy is inserted correctly.
Outcome	Positive.
Notes	None.

Id	AT018_Cpo_ModifyPriceStrat
Description	The cpo can modify a pricing strategy.
Input	Requested form inputs.
Expected output	The pricing strategy is modified.
Test results	The pricing strategy is modified correctly.
Outcome	Positive.
Notes	None.

Id	AT019_Cpo_DeletePriceStrat
Description	The cpo can delete a pricing strategy.
Input	None.
Expected output	The pricing strategy is deleted.
Test results	The pricing strategy is deleted correctly.
Outcome	Positive.
Notes	None.

Id	AT020_Dso_SignUp
Description	Creation of a new dso.
Input	Data required in the form.
Expected output	New dso account creation.
Test results	New dso account created.
Outcome	Positive.
Notes	The password can't be choosed, all the accounts have the same password.

Id	AT021_Dso_SignIn
Description	Login with a dso.
Input	Email and password.
Expected output	The dso logs into his account.
Test results	The dso logs into his account correctly.
Outcome	Positive.
Notes	None.

Id	AT022_DsoCpo_Link
Description	Allow to link a dso with a cpo.
Input	Required form inputs.
Expected output	Cpo and Dso linked together.
Test results	Cpo and Dso linked together correctly.
Outcome	Positive.
Notes	It would have been better to associate a CPO with a charging station.

Id	AT023_Dso_SignOut
Description	Logout from dso profile.
Input	None.
Expected output	Dso correctly logs out.
Test results	The Dso logs out correctly.
Outcome	Positive.
Notes	None.

Id	AT024_Dso_UpdateAccount
Description	The dso can edit the information of his profile.
Input	.
Expected output	The information are updated.
Test results	The information are updated correctly.
Outcome	Positive.
Notes	No format control in update form.

Id	AT025_Dso_SetPrice
Description	The dso set the price of energy.
Input	.
Expected output	The price is updated.
Test results	The price is updated correctly.
Outcome	Positive.
Notes	None.

Additional notes

4.1 Source code documentation

The code is well formatted but uncommented: commenting code helps to improve the readability of your code, which is important for other developers.

A good comment should describe what the code is doing, as well as any special considerations or limitations. This makes it easier for other people to understand your code and makes it easier to maintain or troubleshoot in the future.

4.2 General opinion about the whole project

In general, the application is very usable and human-friendly. There are only some minor problem understandable considering the time in which the project was developed.

In conclusion, it's a very good product with a nice and well-organised user interface.

Effort Spent

5.1 Team discussions

Topic	Hours
General discussion	2h

5.2 Balestrieri Niccolò

Topic	Hours
Application testing	3h

5.3 Bertogalli Andrea

Topic	Hours
Introduction	0.5h
Acceptance test cases (first part)	1.5h
Application testing	1h

5.4 Tombini Nicolò

Topic	Hours
Installation	1.5h
Document review	1.5h