



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

ATD: Acceptance Testing Document

Author(s): **Lorenzo Ferretti**
Lorenzo Manoni
Carlo Sgaravatti

Version: 1.0

Date: 05/02/2023

URL: <https://github.com/bighands2304/ManoniSgaravattiFerretti>

Academic Year: 2022-2023

Contents

Contents	i
1 Tested Project	1
2 Installation	3
3 Test cases	5
3.1 CPMS Test Scenarios	5
3.1.1 Register	5
3.1.2 Login	5
3.1.3 Insert Charging Station	6
3.1.4 Update DSO Contract	6
3.1.5 Update Charging Mode	7
3.1.6 Monitor Charging Process	8
3.2 eMSP Test Scenarios	8
3.2.1 Login	8
3.2.2 Register	8
3.2.3 Book a Charge	9
3.2.4 Delete a Booking	10
3.2.5 Start Charging Process	10
3.2.6 Stop Charging Process	11
3.3 Additional points	11
3.3.1 Client applications	11
3.3.2 Application Server	12
3.3.3 RASD	12
4 Effort Spent	13
4.1 Lorenzo Ferretti	13
4.2 Lorenzo Manoni	13

4.3 Carlo Sgaravatti 13

1 | Tested Project

- **Authors**

- Andrea Bertogalli
- Nicolò Tombolini
- Niccolò Balestrieri

- **GitHub repository:**

- <https://github.com/andberto/BalestrieriBertogalliTombini>

- **Documents Considered:**

- RASD:
<https://github.com/andberto/BalestrieriBertogalliTombini/blob/main/DeliveryFolder/RASD3.pdf>
- DD:
<https://github.com/andberto/BalestrieriBertogalliTombini/blob/main/DeliveryFolder/DD2.pdf>
- ITD:
<https://github.com/andberto/BalestrieriBertogalliTombini/blob/main/DeliveryFolder/ITD1.pdf>

2 | Installation

For the Installation setup, we have followed the instructions in the *Installation Instructions* section of the ITD. The instructions to install the application server, the CPO app, and the end user app were very clear and detailed, and we didn't face any problems in making them run using the provided commands. However, we encountered some issues with the ChargingSocket application; indeed, the provided documentation didn't specify how to set up the connection with the server in order to test the QR-code scanning process and we had to request some further clarifications to make this work.

We already had MySQL and Node.js installed, so no further installations were needed in order to run the application server and the CPO application. In order to run the end-user application, the Flutter SDK was installed using the provided instructions.

Both the application server and the CPO application were run on a Windows 11 laptop. The CPO application was executed in Google Chrome. For the end-user application and the ChargingSocket application, we used some Android emulators from Android Studio; in particular:

- The end-user application was tested in a Google Pixel 3A emulated on the Android Virtual Device (AVD) environment, with SDK version 33 and Android 13.0
- The ChargingSocket application was tested in a Google Pixel 4 emulated on the AVD environment, with SDK version 30 and Android 11.0

3 | Test cases

3.1. CPMS Test Scenarios

3.1.1. Register

- **Goals:**
 - Register a new CPO.
- **Steps:**
 - Run the application
 - Push the 'Register Now' link
 - Fill out all the fields required
 - Press the Sign-Up button.
- **Test Cases:**
 - Providing all fields correctly
 - Providing a password in the 'repeat password' field different from the first one
 - Register with credential used yet.
- **Test Result:**
 - All cases were successfully executed.

3.1.2. Login

- **Goals:**
 - Log in a CPO.
- **Steps:**
 - Run the application

- Fill out all the fields required for the login.
- Press the 'Login' button.

- **Test Cases:**

- Providing all fields correctly
- Providing an incorrect password

- **Test Result:**

- If the credentials provided are correct, the system correctly logs the user into the application otherwise prevent the action.

3.1.3. Insert Charging Station

- **Goals:**

- Add a new Charging Station to the system.

- **Steps:**

- Login to the application.
- Navigate to the Charging Station page.
- Press the 'Add a new station' button.
- Fill out all required fields.
- Insert at least one socket
- Press Confirm Button.

- **Test Cases:**

- Providing all fields correctly.
- Providing all fields correctly without inserting any socket.

- **Test Result:**

- All cases were successfully executed. The CPO can't insert a new Charging Point without providing any socket.

3.1.4. Update DSO Contract

- **Goals:**

- Change DSO provider for a certain Charging Station
- **Steps:**
 - Login to the application.
 - Navigate to the DSO Provider station page.
 - Press the 'Buy' button for a certain CP.
 - Press Confirm Button.
- **Test Cases:**
 - Change DSO Provider.
- **Test Result:**
 - The DSO provider is changed and the database system is updated.

3.1.5. Update Charging Mode

- **Goals:**
 - Change the DSO-mode, DSO-mode, Auto-mode
- **Steps:**
 - Login to the application.
 - Navigate to the DSO Provider station page.
 - Choose between: Change the DSO-mode, Battery-mode, and Auto-mode with the toggle buttons
- **Test Cases:**
 - DSO-mode
 - Battery-mode
 - Auto-mode
- **Test Result:**
 - In all cases the information is updated in the DBMS. But there is no real implementation in the server in order to manage the automatic mode of both DSO and Battery cases.

3.1.6. Monitor Charging Process

- **Goals:**
 - Monitor the data relative to a Charging process happening.
- **Steps:**
 - Login to the application.
 - Navigate to the Battery Cluster status page
- **Test Result:**
 - The data displayed are correctly updated and reflect the behavior of the Charging Process.

3.2. eMSP Test Scenarios

3.2.1. Login

- **Goals:**
 - Login to the application
- **Steps:**
 - Run the Application.
 - Fill out all required fields.
 - Press the login button.
- **Test Cases:**
 - Providing correct credentials
 - Providing incorrect credentials
- **Test Result:**
 - If the credentials are correct the system logs in the user correctly, otherwise an error message is displayed.

3.2.2. Register

- **Goals:**

- Register in the application
- **Steps:**
 - Run the Application.
 - Insert the IP address of the server.
 - Fill out all the fields required for the registration
 - Press the register button
- **Test Cases:**
 - Providing correct credentials
 - Providing incorrect credentials
 - Providing correct IP address
 - Providing incorrect IP address
- **Test Results:**
 - If the Ip-address provided is wrong the application doesn't allow the user to insert a new one and it can't access the application.
 - If the Ip-address provided is correct and the credentials are incorrect an error message is displayed.
 - If the Ip-address and the credential are correct, the system registers the user successfully.

3.2.3. Book a Charge

- **Goals:**
 - Booking a Charging Session
- **Steps:**
 - Login to the application
 - Go to the Map View
 - Select a Charging Station
 - Select a Time slot
 - Press the reserve Button.

- **Test Cases:**

- Select a Time slot Free
- Select a Time slot that is reserved.

- **Test Result:**

- If the time slot selected is reserved the reservation correctly fails and an error message is displayed.
- If the time slot selected is free the reservation is correctly made.

3.2.4. Delete a Booking

- **Goals:**

- Delete a Booking previously made.

- **Steps:**

- Login to the application
- Press My Booking button.
- Press on a Booking item.
- Press Delete Booking.

- **Test Result:**

- The booking is correctly eliminated from the system.

3.2.5. Start Charging Process

- **Goals:**

- Start a charging Session.

- **Steps:**

- Login to the application
- Navigate to 'My Bookings'
- Select a Booking
- Scan the QR code on the socket via Charging Socket Application.

- Navigate to the Charge Status page.
 - Press 'Start Charging'.
- **Test Cases:**
 - Select a valid Booking (with a time range consistent, not expired...).
 - Select an invalid Booking (with a time range inconsistent)
- **Test Result:**
 - If the Booking is valid is possible to start the Charge, otherwise, the system prevents doing it.

3.2.6. Stop Charging Process

- **Goals:**
 - Stop a Charging session started.
- **Steps:**
 - Login to the application
 - Navigate to the Charge Status Page
 - Press the 'Stop Charging' button.
- **Test Result:**
 - The Charging is stopped correctly.

3.3. Additional points

3.3.1. Client applications

- There was a problem in the End User application with the WebSocket connection to the server in order to start a charging session after the QR-code scan. Indeed, with the provided configuration, the client wasn't able to establish a connection to the server due to a wrong endpoint. Figure 3.1 shows both the problem and the solution that we adopted to test the charging sessions.



Figure 3.1: The WebSocket problem in the lib/view/set_ip_address.dart file (on the left), with the adopted solution (on the right).

- We appreciated the design of both the End User application and CPO application that allow easy and comfortable interaction with the system.

3.3.2. Application Server

- There is no access control for any operation provided to the server. For example, anyone can access data (e.g. from postman) and trigger any operation via API without any authentication.
- The code structure of the application server doesn't reflect the one provided in the component diagram of the DD.

3.3.3. RASD

- There is no management of any Optimizer (DSO energy acquisition and Battery Management).
- The system can't interact with other systems since the communication standards are not respected.
- There is no control over the number of reservations that a user can make. Potentially a user can reserve all the sockets of all CPs in the application.
- A very good point in our opinion is the choice to insert a QR code verification to start a charging process since it prevents many possible issues that can happen in different otherwise.

4 | Effort Spent

4.1. Lorenzo Ferretti

Task	Hours Spent
Testing	8

4.2. Lorenzo Manoni

Task	Hours Spent
Testing	8

4.3. Carlo Sgaravatti

Task	Hours Spent
Testing	8
Environment installation	2

