Software Engineering 2 - Prof. Di Nitto Elisabetta
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

eMall - e-Mobility for All

ATD
Acceptance Test Deliverable

February 5, 2023

Giovanni De Lucia (10700658)
Lorenzo Battiston (10618906)
Matteo Salvatore Currò (10940719)

Repository: `https://github.com/gio-del/BattistonDeLuciaCurro-swe2`

Contents

1 Introduction

Project Analyzed: `https://github.com/bighands2304/ManoniSgaravattiFerretti`

- Lorenzo Ferretti
- Lorenzo Manoni
- Carlo Sgaravatti

1.1 Acronyms

- eMSP: e-Mobility Service Provider
- CPMS: Charging Point Management System
- DSO: Distribution System Operator
- OCPP: Open Charge Point Protocol
- OCPI: Open Charge Point Interface
- OSCP: Open Smart Charging Protocol
- API: Application Programming Interface

2 Installation Setup

We followed the instructions in the ITD.pdf files to install the project.
We used the following commands to start the eMSP client:

```
yarn
yarn start
```

We used the following commands to start the CPMS client:

```
yarn
yarn start
```

Overall the instructions were clear and easy to follow, but we encounter some inconsistencies between the instructions in the ITD and the one in the README of the cpmsClient/cpms folder.
Also, in order to test the CPMS a charging point OCPP endpoint is needed. They provided a simulator built with Python, but we cannot found any instructions on how to start it, so we had to do it ourselves by using those commands:

```
pip install -r requirements
python ocppDriver.py
```

3 Acceptance

In this section we refer to the product functionalities described in the ITD.pdf file, in the same order

3.1 eMSP

1. It is possible to find a charging point in a specific area given the address. Although we cannot find information about the price (Tariff) and the Type.
2. It is possible to reserve a charging point in maximum the next 20 minutes.
3. It is possible to start a charging session, but we cannot see the updates of the battery level and the price at the end of the session.

3.2 CPMS

1. It is possible to insert new Charging Points, but it is not possible to remove them. Furthermore, it is possible to make some sockets not available. It is possible to add tariffs to socket in the phase of creation of the charging point. But we faced errors (NOT SERIALIZABLE) when we tried to

add a tariff to a socket that was already created. Also, it seems that it is not possible to remove them or modify them.

2. It is possible to manage the charging sessions and monitoring it.
3. It is possible to monitor the status of the batteries and charging profiles.
4. It is possible to see the current price of energy from the DSOs.
5. If the manual mode is selected, the add button to select a DSO does not call any API.
6. It is possible to see the battery status inserted manually or to select the automatic mode.

4 Additional points

1. There are some leftover files like the ”MealsToGo” project that we think should be removed from the repository. And some reference to a ”Restaurant Card” component that it’s not present in the eMSP Client folder. Also, the package names are not consistent with the project name, for example the eMSP client has the package name ”mealstogoproject” and the CPMS client has the package name ”jobster”.
2. We think that the backend is well-structured and easy to understand, and we appreciate the choice of an opinionated framework like Spring.
3. The use of standards protocols like OCPP and OCPI is definitely a plus for the project.