

Design Document

Simone Mosciatti & Sara Zanzottera

November 29, 2016

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, Abbreviations	3
1.4	Reference Documents	5
1.5	Document Structure	5
2	Architectural Design	6
2.1	Overview	6
2.2	Component View	6
2.3	Deployment View	6
2.4	Runtime View	6
2.5	Component Interfaces	6
2.6	Architectural Styles and Patterns	6
2.7	Other Design Decisions	6
3	Algorithm Design	7
4	User Interface Design	8
5	Requirements Traceability	9
6	Conclusions	10
6.1	Tools used	10
6.2	Hours of work	10

1 Introduction

1.1 Purpose

This Design Document aims to provide to everyone involved in the actual development of the application specific insights about the structure of PowerEnJoy, its architecture's details, the design patterns we chose to implement, but also some details about its high level components, their interactions and general behavior.

1.2 Scope

PowerEnJoy is a digital management system for car sharing that exclusively employs electric cars to provide its service. The system provides all the functionalities normally provided by a car sharing service: registering to the service, find the location of nearby available cars, reserve cars up to a short amount of time, unlock the chosen car once found, ride it and then park it in a safe area, when it will be automatically locked and the fee paid.

In addition, the system gives bonuses and penalties in term of discounts or over-prices depending on the behavior of the user, in order to promote virtuous behaviors.

PowerEnJoy is therefore a inherently distributed system, based on a central server interactions with many distributed nodes. In detail the system can be divided into four main parts:

- a public app, used by customers to access the service
- a centralized backend that provides the service
- the cars' onboard system, that communicates only with the centralized backend
- a reserved frontend, used exclusively by the staff members to better organize their job

All these four components will be examined in more detail in the subsequent sections of the document.

1.3 Definitions, Acronyms, Abbreviations

RASD Requirements and Specification Document.

DD Design Document.

User A customer of PowerEnJoy using the service.

Staff Operator An employee of PowerEnJoy which takes care of the cars.

Ride The action of getting onboard of a PowerEnJoy car, start its engine, drive to destination and park.

Running Time The time an user spends using the PowerEnJoy service.

Issue Any problem a car may incur in, or a user may face while using the service.

Nearby Cars Cars located within a maximum distance to a specific position.

Nearby Issues Issues that are affecting cars close to a specific position.

Booking (Reservation) The act to reserve a car for a limited amount of time for future use by a user.

Reservation's maximum time The maximum amount of time a car can be reserved.

Driver Whoever is driving a regularly booked PowerEnjoy car.

Passenger Whoever is inside a PowerEnjoy car but is not the driver.

Driving License The state's issued driving license of the user.

Notification A form of communication where the user is actively notified of some event.

Issue Report An incoming notification that states a car incurred in an issue.

Fine A fine issued by the local law enforcing officers to a user while driving a PowerEnjoy car.

Pending Bills Bills that a user still needs to pay to PowerEnjoy .

Safe Area A parking area, predefined by the company, where it is possible to safely park the cars of the PowerEnjoy fleet.

Battery Charge The amount of charge that is kept inside the car's battery.

Charging Station Dedicated areas where it is possible to plug the PowerEnjoy cars to charge their batteries.

Car's Onboard System The control system of the car that is able to exchange data with the central system and to reevaluate operation parameters.

Customer's App An implementation of the system frontend tailored to the need of the customers.

Operator's App An implementation of the system frontend tailored to the need of the staff.

Central System The central system for PowerEnjoy . All the commands and all the data are streamed, analyzed and used here.

Credentials Pair {Username, Password} necessary to access the PowerEnjoy system.

GPS : Global Positioning System is a global navigation satellite system (GNSS) that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

System's Frontend The interface provided to the user of the PowerEnJoy system.

System's Backend The whole technical infrastructure necessary to PowerEnJoy .

1.4 Reference Documents

- *Assignments AA 2016-2017.pdf* (Assignments document given by the teacher)
- *Sample Design Deliverable Discussed on Nov. 2.pdf* (Sample document provided by the teacher)

1.5 Document Structure

1. Introduction

This sections aims to explain the purpose and the scope of the document, introducing the reader to subsequent sections of the document itself.

2. Architectural Design

;DESCRIPTION HERE;

3. Algorithm Design

In this section we focus on the most critical code section and we provide an in-depth analysis of how they should be structured, eventually providing pseudocode for them.

4. User Interface Design

In this section we carry on the UX design with the help of UX and BCE diagrams, eventually completing them with updated and extended application mockups.

5. Requirements Traceability

In this section we map the requirements stated in the RASD to the actual component or processes that fulfill these requirements.

6. Conclusions

In this section we enumerate the tools we used to redact this document, the hours of work spent by each group member and the (eventual) revision history of the document itself.

2 Architectural Design

2.1 Overview

[High level components and their inteactions]

2.2 Component View

2.3 Deployment View

2.4 Runtime View

[Includes sequence diagrams to show how components interact to accomplish specific use cases]

2.5 Component Interfaces

2.6 Architectural Styles and Patterns

[Explain patterns used above]

2.7 Other Design Decisions

3 Algorithm Design

[Definition of critical sections of code]

4 User Interface Design

[UX Diagrams and mockups, possibly extended]

5 Requirements Traceability

[Map requirements with components]

6 Conclusions

6.1 Tools used

During the development of this document we used the following tools:

- **Github** to version control the project
- **L^AT_EX** on TeXworks to redact this document
- **www.draw.io** to draw UML graphs
- **Gimp v.2.8** to mockup the application

6.2 Hours of work

- SZ: overall X hours
- SM: overall X hours