Power EnJoy

Design Document

Version 1.0.0

Redaelli Marco 877622 Zanoli Francesco 877471

16/11/2016

Contents

Figure Contents

Chapter 1

Introduction

This document contains the complete design description of *PowerEnJoy*. This includes the architectural features of the system down through details of what operations each code module will perform and the database layout. It also shows how the use cases detailed in the RASD will be implemented in the system using this design. The primary audiences of this document are the software developers but the level of the description is high enough for all the stakeholders to capture the information they need in order to decide whether the system meets their requirements or in order to begin the development work.

1.1 Scope

PowerEnJoy is a car sharing service developed to fit all the reality, as small or large city. The main goals of the system are:

- simplify the access of driver to the service
- guarantee a fair management of reservation and use of electric car

The system architecture will be a three-tier architecture: client, server application and server database. It will be created by using the MVC architectural pattern. The system will be divided into components with respect to the principles leading to good design:

- Each individual component will be smaller in order to be easier to understand
- Coupling will be reduce where possible
- Reusability and flexibility will be increase in order to make easier future implementation

The system will have efficient algorithm in order to increase its performance; in the document will be given special attention to the sharing algorithm.

1.2 Definitions, acronyms and abbreviations

In the document are often used some technical terms whose definitions are here reported:

- **Layer:** A software level in a software system.
- **Tier:** An hardware level in a software system.
- **Relational Database:** A digital database whose organisation is based on the relational model of data, as proposed by E.F. Codd in 1970.
- **Cocoa MVC:** A strict application of MVC principles.
- See the correspondent section in the **RASD** for more definitions.

For sake of brevity, some acronyms and abbreviations are used:

- **DD:** Design Document.
- **GPS:** Global Positioning System.
- **GUI:** Graphic User Interface.
- **API:** Application Programming Interface.
- **MVC:** Model View Controller.
- **ETA:** Estimated Time of Arrival.
- See the correspondent section in the **RASD** for more acronyms and abbreviations.

1.3 References

- Software Engineering 2 Project AA 2016/2017: Assignments AA 2016-2017
- ***PowerEnJoy* RASD v1.0:** Requirements Analysis and Specification Document for *PowerEnJoy*
- **IEEE Std 1016-2009:** IEEE Standard for Information Technology - Systems Design - Software Design Descriptions
- **ISO/IEC/IEEE 42010:** International Standard for Systems and software engineering - Architecture description

1.4 Document Structure

This document is essentially divided in seven main sections:

- **Introduction:** it gives a description of the document and some information about the system design and architecture.
- **Architectural Design:** This is the core of the document. It gives general information about the architectural design. It also describes how the system will be divided into components and how the components communicate. It also has a description of the design pattern and architectural styles that will be used.
- **Algorithm Design:** it gives a description of the main algorithm that will be implemented. Without using a specific language it describe the different steps the algorithm will do.
- **User Interface Design:** it gives a description of the user interfaces of the system and a detailed list of mockup
- **Requirements Traceability:** this section documents the life of a requirement and provides bi-directional traceability between various associated requirements.
- **Appendix:** it provides informations that are not considered part of the actual DD. It includes: software and tools used, project group organisation.

Chapter 2

Architectural Design

2.1 Overview

2.2 High level components and their interaction

2.3 Component View

2.4 Deployment view

2.5 Runtime view

2.6 Component interfaces

2.7 Selected architectural styles and patterns

2.8 Other design decisions

Chapter 3

Algorithm Design

Chapter 4

User Interface Design

Chapter 5

Requirements Traceability

Appendix A

Appendix

A.1 Tools

- **TeXstudio:** LaTeX editor used to write the document.
- **Alloy Analyzer 4.2:** Used to build an Alloy Model and to check its consistency.
- **StarUML:** Used to build UML Class Diagram and UML Use Case Diagrams.
- **draw.io WebSite:** Used to design the UML Sequence Diagram and the Mockup (wep portal and smartphone app).

A.2 Hours of work

In the following are listed the hours of work that each member of the group did:

1. Marco Redaelli: 41 *hours*
2. Francesco Zanoli: 41 *hours*

A.3 Version History

In the following are listed the differences between versions:

1. First version