Requirement Analysis and Specification Document for PowerEnJoy

Enrico Migliorini, Alessandro Paglialonga, Simone Perriello

October 27, 2016

Contents

1 Introduction

1.1 Purpose

This document presents the requirements of the *PowerEnJoy* system, aimed at powering a car-sharing service. Said requirements will be presented using both natural language and diagrams.

1.2 Intended Audience

This document is addressed to all the stakeholders in the *PowerEnJoy* project. This includes, but is not limited to, the CEO, the end users, the development committee, product designers and engineers, quality assurance and marketing.

1.3 Product Scope

The *PowerEnJoy* is a partially automated electric car-sharing service. The system keeps track of users and vehicles, addresses users to available cars, locks cars when not in use, and charges the users for use and abuse of the vehicles. The system also needs to keep track of the battery level of the vehicles and dispatch personnel to connect low-on-battery vehiclesto the power grid.

It is important to notice that this document only describes the requirements for the software dealing with the cars, not the hardware on which the software will run or the management system.

1.4 Definitions, Acronyms and Abbreviations

1.4.1 Business terms glossary

1.4.1.1 Car-sharing A *Car-sharing* service allows *Users* to rent *Vehicles* for a limited amount of time, being charged a *Fee* according to time and eventually applying a *Discount* or an *Increase*.

1.4.1.2 Database A structure that holds informations linked logically according to relationships. For instance, a *Database* could hold records of every registered *User*, every available *Vehicle* and every time a *User* rented a *Vehicle*.

1.4.1.3 Discount A reduction in the *Fee* to be paid because of good behaviour on the part of the *User*, e.g. leaving the *Vehicle* plugged or bringing it back with a mostly-full battery.

1.4.1.4 Fee The amount of money that the *User* will be charged for his usage of the *Car-sharing* service.

1.4.1.5 Increase An increase in the *Fee* to be paid because of improper behavious on the part of the *User*, e.g. bringing the *Vehicle* back with a mostly-empty battery.

1.4.1.6 Management System An external system that allows administrative access to the internal *Database*.

1.4.1.7 System The automated software structure this document is about. It tracks *Users* and *Vehicles* and deals with all the details needed for *Car-sharing*, from GPS mapping to charging *Users* with *Fees*.

1.4.1.8 User A person registered on the *System*, who will use the *Vehicles* for a *Fee*.

1.4.1.9 Vehicle An electric car owned by the *Car-sharing* service, rented to the *User* and tracked by the *System*.

1.4.2 Document specific terms

1.4.2.1 Alloy A descriptive language that allows to describe a set of structures through constraints.

1.4.2.2 DBMS Data Base Management System. A software interface allowing to interact with the *Database*.

1.4.2.3 RASD Requirements Analysis and Specification Document. This document, describing the *System* to be developed.

1.4.2.4 UC Use Case. A description of interaction between *Users* and *System*.

1.4.2.5 UML Unified Modeling Language. A language for modeling Object-Oriented sfotware systems.

2 General Description

A description of the software, with the how and the why.

2.1 General Description

2.2 Product Perspective

2.3 Product Functions

2.4 User Classes and Characteristics

2.5 Operating Environment

2.6 Design and Implementation Constraints

2.7 Assumptions and Dependencies

3 External Interface Requirements

The interfaces the system should interact with.

3.1 User Interface

3.2 Hardware Interface

3.3 Software Interface

3.4 Communication Interface

4 Functional Requirements

The functionality for the various users.

4.1 Use cases specification

4.2 Use Case Diagram

4.3 Alloy representation of requirements

5 Non-functional Requirements

Additional requirements that may be added to improve on the program.

5.1 Performance Requirements

5.2 Safety and Security Requirements

5.3 Software Quality Attributes

5.3.1 Availability

5.3.2 Reliability

5.4 Business Rules