Politecnico di Milano, A.A. 2016/2017

Software Engineering 2: PowerEnJoy **R**equirements **A**nalysis and **S**pecification **D**ocument

Binosi Lorenzo 876022

November 2, 2016

Contents

1 Introduction

1.1 Purpose

This is the Requirement Analysis and Specification Document for the PowerEnJoy car-sharing service. Its purpose is to completely describe the system, its components, functional and non-functional requirements, constraints, and relationships with the external world, and to provide typical use cases and scenarios for all the actors involved. It is also a strong baseline for project planning and cost estimization, for software evaluation and change control.

This document is written for project managers, developers, testers and systems analysts. Users are not generally interested in detailed software requirements, but they can still find it useful. It may be used in a contractual requirement.

1.2 Scope

PowerEnJoy is a digital management system for a car-sharing service that exclusively employs electric cars. This car rental system provides an alternative solution to public transport, thus being not only eco-friendly, but also simple and reliable.

The service is available to whomever has a valid driving license. Anybody who wants to register must provide a copy of it, along with his credentials and payment informations. Once registered, they will be allowed to search a car in a specific area, and reserve one for up to an hour before picking it up. The system will cancel the reservation in case of them failing to arrive to the car in time, will make that specific car available again and apply a fine to the driver. Upon arrival to the car, the driver will be able to get inside and drive it, at the cost of a per-minute fee of which he will be notified by the system through a screen on the car itself; this until the car will be left parked in a safe area, where the system will close the doors and make the vehicle available again.

1.3 Goals

The goals of the PowerEnJoy service are the following:

- [G1] Allow a driver to rent a car.
  - [G1.1] Allow a driver to search and reserve a car.
  - [G1.2] Allow a driver to use a reserved car.

  - [G1.3] Allow a driver to park a car in a safe area, suggested by the system.

- [G2] Notify the driver about every fee, discount, or sanction/fine that will be applied to him while using the service

- [G3] Allow a guest to signup to the system.

- [G4] Motivate a driver into car pooling.

- [G5] Motivate a driver to a better behavior.

1.4 Definitions, acronyms, and abbreviations

RASD: Requirements Analysis and Specification Document.

System: the whole software system to be developed, comprehensive of all its parts.

Client: the web application component or the mobile app.

Guest: anyone who's not registered to the system, who's going to, or is looking for informations about the service.

Driver: a registered user who using the car-sharing service. Only drivers can rent a car.

User: anyone who subscribes to the service, the driver.

Car pooling: the sharing of car journeys so that more than one person travels in a car.

1.5 References

This document refers to the project rules of the Software Engineering 2 project [1] and to the RASD assignment [2].

This document follows the IEEE Standard 830-1998 [3] for the format of Software Requirements specifications.

1.6 Overview

This document is structured in three parts:

section 1: Introduction. It provides an overall description of the system scope and purpose, together with some information on this document.

section 2: Overall description. Provides a broad perspective over the principal system features, constraints, and assumptions about the users and the environment.

section 3: Specific requirements. Goes into detail about functional and nonfunctional requirements. This chapter is arranged by feature.

2 Overall Description

3 Specific Requirements

4 Appendix

4.1 Used tools

To create the RASD we used the following tools:

- **MikTex**
  Distribution of the typesetting system LaTeX
  http://www.miktex.org/download

- **TexStudio**
  OpenSource cross-platform LaTeX editor we used to write the RASD.
  http://texstudio.sourceforge.net

- **StarUML**
  UML modelling tool we used to build the graphs
  http://staruml.io/download

- **Balsamiq**
  The mockup builder we used to design the mockups.
  https://balsamiq.com/download/

- **Alloy analyzer 4**
  used to build strong and substantial models
  http://alloy.mit.edu/alloy/download.html

- **GitHub desktop**
  Desktop application of the web-based Git repository hosting service. Used to collaborate in the team and to have a track of the changes.
  https://desktop.github.com/

4.2 Hours of work

This is the time spent redacting the RASD

- Lorenzo Binosi - 3 hours

References

[1] Software Engineering 2 Project, AA 2016/2017 - *Project goal, schedule and rules*

[2] Software Engineering 2 Project, AA 2016/2017 - *Assignments 1*

[3] IEEE Standard 830-1998: *IEEE Recommended Practice for Software Requirements Specifications*