PowerEnJoy
Software Engineering II

Requirements Analysis and Specification Document

Giovanni Scotti, Marco Trabucchi

October 24, 2016

Contents

Section 1

Introduction

1.1 Purpose

The Requirement Analysis and Specification Document for the *PowerEnJoy* digital management system is intended to describe the system itself, the functional and non-functional requirements, its components as well as its constraints and the relationship with the real world and the users by providing several use cases and scenarios. Furthermore part of the documentation makes use of Alloy, a language for describing structures and a tool for exploring them, and gives a formal specification of some features of the system to be.

This document establishes some baselines for the project planning, its estimation and evaluation and it may be legally binding; it is mainly addressed to developers and programmers, who have to implement the requirements, testers, who have to determine whether the requirements have been met, project managers, who control the development process and - last but not least - users, who validate the system goals.

1.2 Scope

The product is a digital management system to support a car-sharing service that exclusively employs electric cars.

The system consists of a back-end server application that manages rental requests remotely and two front-end applications, a web-based one to provide the final user with a friendly interface to take advantage of the services of *PowerEnJoy* and the one that runs on embedded on-board computers of each vehicle, used to interact with the car itself, unlock it and access a personalized GPS/sat-nav service. Moreover, the system will include a mobile application

that allows the user to easily access the service anywhere he/she needs to.

The system is intended for only one type of user: drivers, who should be allowed to register and access the system via username and password, in order to make the renting and payment processes easier and quicker to carry out. Moreover, the system aids the users by locating nearby available vehicles and keeps track of the distance driven, all while notifying them about the amount of money they are being charged. Predefined safe parking areas are signaled by an on-board computer.

Lastly, the system aims to motivate drivers to maintain a virtuous behavior providing discounts when it detects signs of responsible and ecologic actions.

In further detail, the goals of *PowerEnJoy* are the following:

1. Let the user register to the service and login via the provided credentials;

2. Let the driver find the location of nearby available cars;

3. Let him/her reserve a chosen car up to an hour before picking it up;

4. Improve the efficiency of the service by assuring that no car stays reserved if not actually in use;

5. Allow the user to easily access the cars by unlocking them once the driver is in proximity;

6. Actively keep track of the driver's current charges, continuously notifying it;

7. Let the user know which and where the pre-defined parking spots are, showing the nearest ones;

8. Incentivize responsible behaviors, providing discounts for the worthiest users.

1.3   Definitions, Acronyms and Abbreviations

Back-end application: any computer program that remains in the background and offers application logic and communication interfaces to work with the front-end counterpart. It does not involve any graphical user interface, but it can provide a data access layer.

CAN bus: the Controller Area Network is a vehicle bus standard designed to allow microcontrollers and other devices to communicate with each other without a host computer.

DBMS: Database Management System.

Driver: See **User**.

Front-end application: any application the users interact with directly. It provides the so called presentation layer.

GPS: Global Positioning System.

m: meter, *Interational System of Units (SI)* unit of measure.

Mobile broadband: it is the marketing term for wireless internet access delivered through mobile phone towers to any digital device using a portable modem.

OS: Operating System.

PC: Personal Computer.

PIN: Personal Identification Number.

RASD: Requirements Analysis and Specification Document.

System: The software system-to-be, in all of its entirety.

UMTS: Universal Mobile Telecommunications System.

User: Any person subscribed to the service who rents a car using *PowerEnJoy*.

1.4   References

This document follows the guidelines provided by ISO/IEC/IEEE 29148:2011 [1] and IEEE 830-1998 [2] respectively related to the requirements engineering for systems and software products and the recommended practice for software requirements specifications.

Moreover it is strictly based on the specifications concerning the RASD assignment [3] for the Software Engineering II project, part of the course held by professors Luca Mottola and Elisabetta Di Nitto at the Politecnico di Milano, A.Y. 2016/17.

1.5   Overview

This document consists of three sections:

Section 1: Introduction. A general introduction and overview of the system-to-be purpose, scope and goals, along with some important information about this document.

Section 2: Overall description. It describes the general factors that affects the product and its requirements. The section provides a background for those requirements which are defined in detail in Section 3 and makes them easier to figure out.

Section 3: Specific Requirements. All the software requirements are specified to a level of detail which is sufficient to let the designers satisfy them. Both functional and non-functional requirements are mentioned.

At the end of the document are an **Appendix** and a **Bibliography**, providing additional information about the sections listed above.

Section 2

Overall Description

2.1 Product Perspective

2.1.1 User interfaces

The users have several ways to access the system: a web application can be executed on any personal computer while a mobile application provides flexibility, portability and can be used literally everywhere. Despite the fact that the hardware interfaces running the application are rather different, a unified and common user interface is provided. It should be user friendly and very intuitive to allow everyone to easily use it without any specific knowledge.

Moreover the users have to interact with the on-board computer installed on each electric vehicle, therefore it should offer an interface as straightforward as the one implemented by the web and mobile applications.

2.1.2 Hardware interfaces

The web application can be executed on any general purpose computer that complies with the minimum system requirements specified in subsection 2.4.2.

The mobile application has to exchange data with the GPS module located on any recent smart-phone. Moreover it has to access the mobile broadband in order to communicate with the main system server.

An on-board computer is set up in each electric car and it talks to the vehicle control unit through the CAN bus and to the system server via the mobile broadband. Also, the cars are equipped with a thermographic camera, used to detect the number of passengers, drivers included.

2.1.3   Software interfaces

The web based application must support the main browsers such as IE, Google Chrome and Mozilla Firefox.

The mobile application has to be compatible with iOS and Android. The server side of the application, that is the system back-end, stores data in a relational DBMS and runs on any web server supporting Java.

The on-board computers run a Linux-based OS that provides advanced features such as GPS/sat-nav, car monitoring and user identification and access.

The back-end has mainly to deal with the rental service and data management, essentially aimed towards keeping track of transactions and customers information.

2.2   Product Functions

The system allows users to reserve available cars, to manage their requests and charge them for the rental.

The users can:

- Register to the service;

- Login with their personal account;

- Edit and manage personal information;

- Delete their existing account;

- Locate nearby available-to-rent vehicles;

- Reserve one of said cars for a fixed amount of time;

- Unlock the reserved vehicle based on their mobile device GPS position;

- Alternatively, unlock the reserved car by inserting a vehicle-specific code (found on the windshield) into the application;

- Start the engine by inserting their PIN;

- Perform payments for the used service via credit card or $PayPal^{TM}$.

2.3 User Characteristics

Target of the service are *drivers* that wish to rent a car.

The base assumption is that every registered user is in possess of a driving license which is valid in the country where he/she makes use of the service.

Moreover, the user must have the means of accessing the service from anywhere: this includes having the possibility to use a stable internet connection both from a mobile device and a stationary one such as a laptop or desktop PC.

Lastly, the user should provide the system a valid payment method upon the act of registration to the service.

2.4 Constraints

2.4.1 Regulatory policies

The system must be allowed by the user to collect, process and store personal data. Furthermore, the system is capable to delete all personal data upon user request and to keep track of each payment.

The user has the responsibility to use the system properly in order to comply with the local laws and policies.

2.4.2 Hardware limitations

The system should comply to these following minimum hardware requirements:

- Mobile application:

  > 3G UMTS connection at its maximum speed of 2 Mb/s

  > 50 MB of available space

  > 1 GB of RAM

  > GPS module

- Web application:

  > Internet connection at 7 Mb/s

  > 800x600 screen resolution

2.4.3 Parallel operation

The system must support parallel operations from different users and the DBMS relies heavily on concurrent transactions.

2.4.4 Reliability requirements

The system reliability, that is the probability to operate without a failure for a specific period of time, must be 99%.

2.4.5 Criticality of the application

Life-critical applications do not concern the system to be developed.

2.4.6 Safety and security consideration

The user must have a valid driving license in order to take advantage of the car sharing service. The license number is asked and stored by the system for security reason. The locations and the travels performed by the users must be tracked but kept private.

2.5 Assumptions and Dependencies

For the rest of the RASD, it is given for granted that:

- The GPS location of users and cars is accurate, with an uncertainty of ± 1 m;

- All users can access a reliable internet connection;

- The users' mobile devices feature a working GPS;

- All users pay accordingly to their charges after the ride, and the payment is successfully carried out;

- No user can reserve more than one car at a time;

- All users only ever park in the predetermined safe areas, indicated by the system;

- The on-board computers always notify the correct charge to the driver during the ride;

- All cars unlock properly upon insertion of the code by the user who reserved them or in case he/she is detected to be in proximity;

- The thermographic cameras on cars always give a correct reading of the number of passengers, driver included, for the current ride;

- The number of passenger never changes for the duration of the ride;

2.6   Future Extensions

The system implementation will open the possibility of including new features. For example:

- The possibility of including more categories of vehicles, such as motorcycles and bycicles;

- The possibility of receiving active feedback from the users, mainly concerning car conditions;

- The possibility for everyone, registered and unregistered people, of notifying *PowerEnJoy* of incorrect or unsafe situations involving their vehicles.

Appendix A

Appendix

A.1 Software and tools used

- LaTeX used as typesetting system to build this document.

Bibliography

[1] ISO/IEC/IEEE 29148:2011 *Systems and software engineering - Life cycle processes - Requirements engineering*

[2] IEEE 830:1998 *Recommended Practice for Software Requirements Specifications*

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