



POLITECNICO DI MILANO
MSC COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING 2
ACADEMIC YEAR 2016-2017

Requirements Analysis and Specification Document

PowerEnJoy

Author:

Melloni Giulio 876279

Renzi Marco 878269

Testa Filippo 875456

Reference Professor:

MOTTOLA Luca

Release Date: October 25th, 2016
Version 0.1

Table of Contents

1	Introduction	2
1.1	Description of the problem	2
1.2	Scope	2
1.3	Definitions, acronyms and abbreviations	2
1.4	Reference Documents	3
1.5	Overview	3
2	Overall Description	4
2.1	Product perspective	4
2.2	Product functions	4
2.3	User characteristics	4
2.4	Constraints	4
2.5	Assumptions	4
2.5.1	Text assumptions	4
2.5.2	Domain assumptions	4
3	Specific Requirements	5
3.1	External Interface Requirements	5
3.1.1	User Interfaces	5
3.1.2	Hardware Interfaces	5
3.1.3	Software Interfaces	5
3.1.4	Communication Interfaces	5
3.2	Functional Requirements	5
3.3	Non Functional Requirements	5
4	Scenarios	6
5	Modeling the System	7
5.1	UML Models	7
5.1.1	Use Cases	7
5.1.2	Class Diagram	7
5.1.3	Sequence Diagram	7
5.1.4	Statechart Diagram	7
5.1.5	Activity Diagram	7
5.2	Alloy	7
5.2.1	Source Code	7
6	RASD Preparation	8
6.1	Tools	8
6.2	Timing	8

1 | Introduction

1.1 Description of the problem

This project is modeling a car sharing service. The cars can be reserved or picked up if free thanks to a web-application. This can be used by a mobile phone or thanks to a web browser. The application can be used by two different type of people:

- Users
- Workers

User account granted different functionalities: he can, for instance, search for a car using GPS or specifying location, reserve or pick a car up. Worker account is used by PowerEnjoy employees: they can find exhausted battery of cars and charge them thanks to GPS location. We are modelling only the user system and we are not considering the employees. All these functionalities can be access by users if and only if they are registered into the system: this means that the client has to put his personal data, for instance the name, surname and other info, a valid driving licence, checked by the system, and an existing external payment account. When the registration is complete, the user must confirm using the email sent by the system. After he is registered, he can log into the system and use it in order to take a car.

1.2 Scope

The project has a specific purpose: the user can register himself on the website and access to the main functionalities of this service. So, the user can search for a car giving GPS position, specifying the location or choosing manually it from the map; after he has searched for it, the user can choose the one which prefers and then select if pick it up or reserve for most an hour, removing this car from the available ones.

1.3 Definitions, acronyms and abbreviations

Guest: the client who isn't already registered. So he can't access to the system functionalities because he isn't an user yet.

Client: the client is the user who interacts with the application. He can use the system functionalities, designed to be used by people who are registered with valid data.

System: the system is the server side of the application. This is meant to be the side of the process which answers to user's question and sends back to him data.

Registration: this let a guest becomes a valid user recognized by the system. The registration has three steps:

- **Personal Data** Here the user must insert his personal data, which means, for instance, the name and surname, the date of birth, the residence address and the billing address.

- **Driving Licence** In order to take the car and drive it, the user must have a driving licence which is recognized by the system thanks to its data.
- **External Payment** An user can perform these actions if he has a positive balance on the external payment account that is bind to the PowerEnjoy account when registering.

If an user puts wrong data, or close the registration process not completing all these steps, the registration isn't done. *Search*: this functionality lets the user searching for a car in three different ways:

- **MAP** - The user scrolls the map on the start screen of the application and he can select cars displayed by green indicators.
- **GPS** - The user can select to use it location and then automatically display all the available cars on the map.
- **POSITION** - The user gives the position where he wants to find the car.

In the last two cases, the user has to specify also a range where he wants to find the cars in.

Reservation: after searching, the user can reserve the car for most an hour. That means that the user can take the car in one hour, and the car is displayed as unavailable. If the car is not took in one hour, the car becomes again available and the user has to pay a fee of 1?.

Manage reservation: if the user has reserved a car, he can delete the reservation. If the user delete the reservation, the car become available and the user can reserve another car (maybe nearer for instance).

Pick Up:

Stop driving:

1.4 Reference Documents

1.5 Overview

2 | Overall Description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions

2.5.1 Text assumptions

2.5.2 Domain assumptions

3 | Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 Functional Requirements

3.3 Non Functional Requirements

4 | Scenarios

5 | Modeling the System

5.1 UML Models

5.1.1 Use Cases

5.1.2 Class Diagram

5.1.3 Sequence Diagram

5.1.4 Statechart Diagram

5.1.5 Activity Diagram

5.2 Alloy

5.2.1 Source Code

6 | RASD Preparation

6.1 Tools

6.2 Timing