



POLITECNICO
MILANO 1863

Politecnico di Milano

COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING 2

Design document

PowerEnJoy

Authors:

Francesco Fabiani
Jagadesh Manivannan
Niccolò Pozzolini

Professors:

Elisabetta Di Nitto
Luca Mottola

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, acronyms, abbreviations	3
1.4	Reference documents	4
1.5	Document structure	4

Chapter 1

Introduction

1.1 Purpose

The Design Document aims to describe more accurately all the functionalities provided by PowerEnJoy focusing on the software design of the system, such as the architectural components and the interfaces that allow them to interact.

The main objective of this DD is to illustrate to a team of developers how to scrupulously implement our service, with the help of UML diagrams explaining the purpose of every element.

1.2 Scope

The product described is PowerEnJoy, a car-sharing service which offers to its users exclusively electric cars. It includes the common functionalities of its category: permitting to registered users to obtain the position of all the available cars, reserving one within a certain amount of time and continuously displaying the up-to-the-minute cost of the ride are just few of them. Moreover, PowerEnJoy stimulates users to behave virtuously towards the ecosystem by applying various types of discounts under specific conditions.

There are four software components that constitute the PowerEnJoy project. First, a back-end server provides APIs in order to simplify the communication related to the interactions of a user with the cars. Then two applications are available for a user to allow him/her the employment of every functionality: a web-based application (intended for visualization from desktop) and a mobile one. Lastly, every vehicle will be equipped with an on-board computer, used by the driver to manage the ride with the available options and see real-time information related to it, such as the time spent,

the distance traveled and the total amount.

1.3 Definitions, acronyms, abbreviations

- *API*: Application Programming Interface
- *Car*: electric vehicle provided by the service
- *DB*: Database
- *DBMS*: Database Management System
- *DD*: Design Document
- *ER*: Entity-Relationship
- *GPS*: Global Positioning System
- *Guest* or *Guest user*: person not registered to the service
- *MVC*: Model View Controller
- *OS*: Operating System, related both to desktop and mobile platforms
- *PIN*: Personal Identification Number
- *RASD*: Requirements Analysis and Specification Document
- *Registered user*: see *User*
- *REST*: REpresentational State Transfer
- *RESTful*: that follows the REST principles
- *Safe area*: set of parking spots where a user can leave a car without penalization
- *User*: person with a valid driving license registered to the service
- *W3C*: World Wide Web Consortium

1.4 Reference documents

The Design Document has been composed following the indications and examples reported in the document ISO/IEC/IEEE 1016-2009, released by W3C, containing provisions for the processes and products related to the engineering of requirements for systems and software products and services throughout the life cycle.

The previous document delivered for this project, the Requirements Analysis and Specification Document, describes fundamental aspects of PowerEnjoy such as domain assumptions, goals, functional and non-functional requirements.

With regards to the course named Software Engineering 2 and held by professors Luca Mottola and Elisabetta Di Nitto (Politecnico di Milano, a. y. 2016/17), the document conforms to the guidelines provided during the lectures and within the material of the course.

1.5 Document structure

This Design Document is composed of five chapters, each of them describing a specific aspect of our project.

- *Introduction*: provides a general overview of this document.
- *Architectural design*: explains the main components, focusing on their interactions, architectural choices and patterns.
- *Algorithm design*: describes the most relevant algorithms utilized in order to optimize the system.
- *User interface design*: shows what is likely to be displayed to the user when using the client-side implementations.
- *Requirements traceability*: indicates how the requirements reported in the RASD map to the design elements described in this document.