

POLITECNICO DI MILANO

2016-2017

Software Engineering 2: PowerEnJoy

Design Document

Version 1.0

Peverelli Francesco

Reppucci Federico

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| 1. INTRODUCTION  1.1. PURPOSE  1.2. SCOPE  1.3. GLOSSARY  1.4. REFERENCES  1.5. SUMMARY  2. BODY  2.1. OVERVIEW  2.1.1 REQUIREMENTS MAPPING APPROACH  2.1.2. DESIGN DECISIONS METHODOLOGY  2.2. COMPONENT VIEW  2.2.1. EXTERNAL COMPONENTS AND INTERFACES  2.2.2 SOFTWARE COMPONENTS VIEW  2.3. HIGH-LEVEL SYSTEM ARCHITECTURE  2.3.1. SOFTWARE ARCHITECTURAL PATTERN  2.3.2. SYSTEM ARCHITECTURE  2.4. COMPONENTS INTERACTION  2.4.1. COMPONENTS INTERFACES  2.4.2. RUNTIME SEQUENCE DIAGRAMS  2.5. COMPONENTS ARCHITECTURE AND PATTERNS  2.5.1. COMPONENTS ARCHITECTURAL VIEW (ALSO DB: ER DIAGRAM)  2.5.2. OBJECT DIAGRAMS  2.6. USER INTERFACE DESIGN  2.7. SELECTED TOOLS  2.8. DEPLOYMENT VIEW  2.8.1. RUNTIME DEPENDENCIES VIEW  2.8.2. SOFTWARE COMPONENTS MAPPING  2.8.3 TECHNICAL ENVIRONMENT REQUIREMENTS  2.9. ALGORITHM DESIGN  2.10. REQUIREMENTS TRACEABILITY  3. EFFORT SPENT |  |

***1. INTRODUCTION***

***1.1. PURPOSE***

The aim of this document is to provide a complete specification of the PowerEnJoy system's architecture which fulfills the requirements identified during the requirements specification phase. More specifically, this document is meant to serve both as a way to clarify to the stakeholders how the specified system will fulfill the requirements and how the requirements have motivated the design decisions presented, as well as a reference for the developers of the system that will guide the implementation phase.

***1.2. SCOPE***

The PowerEnJoy project aims to develop a car-sharing service run exclusively employing electric cars. The system will provide a mobile application by means of which the users, once registered, will be able to use the car sharing services. The main goals of the service are to provide a sustainable and environmentally-friendly car sharing service as well as to promote virtuous behaviors from its users.

***1.3. GLOSSARY***

***1.4. REFERENCES***

***1.5. SUMMARY***

*1. OVERVIEW:* this section explains how the document is structured, and the reasoning behind the choice of this particular approach.

*2. COMPONENT VIEW:* this section shows the external software services that interact with the system and the structure of the system's main software components.

*3. HIGH-LEVEL SYSTEM ARCHITECTURE:* this section describes the Software Architectural Pattern of choice as well as the physical architecture of the system.

*4. COMPONENTS INTERACTION:* this section describes more precisely how the software components are connected, which interfaces they provide and how they interact with each other.

5. COMPONENTS ARCHITECTURE AND PATTERNS: this section specifies for each software component its architecture, as well as the main design patterns used to construct it and their specific purpose.

*6. USER INTERFACE DESIGN:* this section describes in detail the functionalities of the user interfaces.

*7. SELECTED TOOLS:* this section specifies all the main frameworks, languages, libraries and tools to use during the development of the system.

*8. DEPLOYMENT VIEW:* this section describes how the software components are mapped onto the system's hardware and how they interact with each other at runtime. Moreover, a set of requirements regarding the execution environment of the software is identified.

*9. ALGORITHM DESIGN:* this section describes the most critical algorithms to implement, providing a pseudo-code implementation.

*10. REQUIREMENTS TRACEABILITY:* this section explains how the requirements previously identified are met by the system described in the document.

***2. BODY***

***2.1. OVERVIEW***

The structure of this document is meant to reflect the actual stages of the development of the architecture and design of the system, in order to illustrate not only the final result of the design phase, but also the intermediate steps of this process, and the rationale behind every major design decision. In order to achieve this, the document does not follow a strictly top-down structure, but starts from the design decisions that follow most directly from the requirements, namely the design of the system's main functional components. It is important to note that despite this difference in the rationale behind the structure of the document, for the most part the approach is still top-down, since such an approach is the most natural way of obtaining a cohesive design.

***2.1.1 REQUIREMENTS MAPPING APPROACH***

***2.1.2. DESIGN DECISIONS METHODOLOGY***

***2.2. COMPONENT VIEW***

***2.2.1. EXTERNAL COMPONENTS AND INTERFACES***

***2.2.2 SOFTWARE COMPONENTS VIEW***

***2.3. HIGH-LEVEL SYSTEM ARCHITECTURE***

***2.3.1. SOFTWARE ARCHITECTURAL PATTERN***

***2.3.2. SYSTEM ARCHITECTURE***

***2.4. COMPONENTS INTERACTION***

***2.4.1. COMPONENTS INTERFACES***

***2.4.2. RUNTIME SEQUENCE DIAGRAMS***

***2.5. COMPONENTS ARCHITECTURE AND PATTERNS***

***2.5.1. COMPONENTS ARCHITECTURAL VIEW (ALSO DB: ER DIAGRAM)***

***2.5.2. OBJECT DIAGRAMS***

***2.6. USER INTERFACE DESIGN***

***2.7. SELECTED TOOLS***

***2.8. DEPLOYMENT VIEW***

***2.8.1. RUNTIME DEPENDENCIES VIEW***

***2.8.2. SOFTWARE COMPONENTS MAPPING***

***2.8.3 TECHNICAL ENVIRONMENT REQUIREMENTS***

***2.9. ALGORITHM DESIGN***

***2.10. REQUIREMENTS TRACEABILITY***

***3. EFFORT SPENT***