

**PowerEnjoy**

**D**esign **D**ocument

Version 1.0

**Software Engineering 2 (A.A. 2016/2017)**

* Simone Boglio (mat. 772263)
* Lorenzo Croce (mat. 807833)

Summary

[1. Introduction 4](#_Toc468279516)

[1.1. Purpose 4](#_Toc468279517)

[1.2. Scope 4](#_Toc468279518)

[1.3. Definitions, Acronyms, Abbreviations 4](#_Toc468279519)

[1.4. Reference Documents 4](#_Toc468279520)

[1.5. Document Structure 5](#_Toc468279521)

[2. Architectural Design 6](#_Toc468279522)

[2.1. Overview 6](#_Toc468279523)

[2.2. Component View 6](#_Toc468279524)

[2.3. Deployment View 6](#_Toc468279525)

[2.4. Runtime View 6](#_Toc468279526)

[2.5. Component Interfaces 6](#_Toc468279527)

[2.6. Architectural Styles and Patterns 6](#_Toc468279528)

[2.7. Other Design Decisions 6](#_Toc468279529)

[3. Algorithm Design 6](#_Toc468279530)

[4. User Interface 6](#_Toc468279531)

[5. Requirements Traceability 6](#_Toc468279532)

[6. References 6](#_Toc468279533)

[7. Appendix 6](#_Toc468279534)

[7.1. Software and Tools Used 6](#_Toc468279535)

[7.2. Hours of Work 6](#_Toc468279536)

1. **Introduction**
   1. **Purpose**

This document presents the architecture on which PowerEnjoy will be developed, it describes the decisions taken during the design process and justifies them.

The whole process is reported including all the improvements and modifications to provide additional information in case of future changes of the architecture structure.

* 1. **Scope**

This document will focus on the non- functional requirements of PowerEnjoy. Since the system architecture defines constraints on the implementation, this document will be used to provide fundamental guidelines in the development phase of PowerEnjoy.

* 1. **Definitions, Acronyms, Abbreviations**

The following definitions are used in this document:

* Thin client: is a computer that depends heavily on another computer (its server) to perform its computational tasks.
* Server: is a type of server that provides most of the functionality to a client’s machine within a client/server computing architecture.
* Event-driven architecture: is a software architecture pattern promoting the production, detection, consumption of, and reaction to events, that is a change of state of an object.
* Availability: is the “status” of a PowerEnjoy car, it can be:
  + Available: the car is ready for a new reservation by the customer.
  + Unavailable: the car is not available in the system, for example when car need external maintenance, or it is moved from one area to another to balance the availability of the cars in the city.
  + In use: the car is used by a customer for a ride.
  + Reserved: the car is waiting for the customer who reserve its and cannot be reserved or use by another user.
* Area: the city of Milan is partitioned in different type of area.
  + Safe Area: the user can finish is ride leaving the car in one of this area.
  + Special Area: like safe area, also in this type of area the car can be left in charge in one of the power grid station.
  + Unsafe Area: the user cannot end is ride in this type of area.

The following acronyms are used in this document:

* JEE: Java Enterprise Edition
* RASD: Requirements Analysis and Specification Document
* UI: User Interface
* API: Application Programming Interface
  1. **Reference Documents**
* Specification document for Software Engineering 2: PowerEnjoy project.
* Template for the Design Document.
* IEEE Standard 1016-2009 - IEEE Standard on Design Descriptions.
* Requirements Analysis and Specification Document (RASD): PowerEnjoy.
  1. **Document Structure**

This document specifies the architecture of the system using different levels of detail, it also describes the architectural decisions and justifies them.

The design is developed in a top-down way, then the document reflects this approach.

The document is organized in the following sections:

1. Introduction

Provides a brief of the architectural descriptions.

1. Architectural design

The core of the design document, in this section are presented all the components of the system and the interaction between them, in increasing level of detail starting from a high-level overview.

1. Algorithm design

Pseudocode and flowchart of the fundamental algorithms of PowerEnjoy.

1. User interface design

Mock-ups of the UI to understand better how the functionalities will been implemented from a graphical viewpoint.

1. Requirements traceability

The mapping between the requirements and the components in the architectural design used to satisfy them.

1. References

List of sources used in this document: internet links, documents.

1. **Architectural Design**
   1. **Overview**
   2. **Component View**
   3. **Deployment View**
   4. **Runtime View**
   5. **Component Interfaces**
   6. **Architectural Styles and Patterns**
   7. **Other Design Decisions**
2. **Algorithm Design**
3. **User Interface**
4. **Requirements Traceability**
5. **References**

* Material from Wikipedia
  + Thin client: https://en.wikipedia.org/wiki/Thin client
  + Event-driven architecture: https://en.wikipedia.org/wiki/Event-driven architecture
* Documents
  + ---list of documents used---

1. **Appendix**
   1. **Software and Tools Used**

* Microsoft Office Word: to redact and format this document.
* Astah Professional 7.0 (http://astah.net/editions/professional): to create Use Cases Diagrams, Sequence Diagrams, Class Diagrams and State Machine Diagrams.
  1. **Hours of Work**
* Simone Boglio: 3 hours.
* Lorenzo Croce: x hours.

Guide Lines

6.2 Assignment2

The Design document (DD) must contain a functional description of the system, and any other view you find useful to provide. You should use all the UML diagrams you need to provide a full description of the system. Alloy may also be useful, although its use is not mandatory here. You will also include information on the number of hours each group member has worked towards the fulfilment of this deadline. As a reference structure for your document please refer to the following one:

1. INTRODUCTIONA. Purpose B. Scope C. Definitions, Acronyms, Abbreviations D. Reference Documents E. Document Structure2. ARCHITECTURAL DESIGN

A. Overview: High level components and their interaction

C. Component view

D. Deployment view

E. Runtime view: You can use sequence diagrams to describe the way components

interact to accomplish specific tasks typically related to your use cases

F. Component interfaces

G. Selected architectural styles and patterns: Please explain which styles/patterns you

used, why, and how H. Other design decisions

3. ALGORITHM DESIGN: Focus on the definition of the most relevant algorithmic part

4. USER INTERFACE DESIGN: Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have

already done, possibly, providing here some extensions if applicable.

5. REQUIREMENTS TRACEABILITY: Explain how the requirements you have defined in the RASD

map to the design elements that you have defined in this document.

6. EFFORT SPENT: In this section you will include information about the number of hours each

group member has worked towards the fulfilment of this deadline.

7. REFERENCES