

DD - SOFTWARE ENGINEERING 2



POLITECNICO
MILANO 1863

PowerEnjoy

Marini Alberto
862838

alberto2.marini@mail.polimi.it

Marrone Matteo
810840

matteo.marrone@mail.polimi.it

Sabatelli Antonella
875666

antonella.sabatelli@mail.polimi.it

December 11th, 2016

Politecnico di Milano

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, Abbreviations	3
1.3.1	Definitions	3
1.3.2	Acronyms and Abbreviations	3
1.4	Document Structure	4
2	Architectural Design	5
2.1	Definitions	5
2.2	High level components and their interaction	5
2.3	Component view	5
2.4	Deployment view	5
2.5	Runtime view	5
2.6	Component interfaces	5
2.7	Selected architectural styles and patterns	5
2.8	Other design decisions	5
3	Algorithm Design	6
4	User Interface Design	7
5	Requirements Traceability	8
6	References and Effort Spent	9

Chapter 1

1 Introduction

1.1 Purpose

The Design Document here presented is meant to provide relevant information regarding the architectural layers, components and interface of the Power Enjoy system to be developed as well as a description of the interaction occurring between the different architectural parts at different levels, directed to project managers, developers, testers and Quality Assurance staff; within the DD we will make use of graphical representations such as component views.

1.2 Scope

The PowerEnjoy system to be built will be able to offer a service of car sharing within the city of Milan involving electric cars only. Our description will not only include a description of the architectural tiers with a gradually increasing amount of detail, but will specify the relations between the tiers and the cars, characterized by a limited AI.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- Session Bean: is a component of the application logic used to model business functions.
- Stateless Session Bean: no state is maintained with the client.
- Stateful Session Bean: the state of an object consists in the values of its instance variables. They represent the state of a unique client/bean session. When the client terminates, the bean is no longer associated with the client.
- Singleton Session Bean: is instantiated once per application and exists for the whole application lifecycle. A single bean instance is shared across and concurrently accessed by clients.
- Java Server Faces: a component-based MVC framework built on top of the Servlet API.

1.3.2 Acronyms and Abbreviations

- RASD: Requirements Analysis and Specification Document

- Java EE: Java Enterprise Edition.
- JSF: Java Server Faces.
- REST: Representational State Transfer.
- XHTML: Extensible HyperText Markup Language.
- EJB: Enterprise Java Beans.
- UX Diagram: User Experience Diagram.
- RDBMS: Relational DataBase Management System.
- DB: the database layer, handled by a RDBMS.
- UI: User Interface.
- SOA: Service-oriented Architecture.
- MVC: Model-View-Controller.
- JDBC: Java DataBase Connectivity.
- JPA: Java Persistence API.

1.4 Document Structure

The document is divided in seven parts, as of requirements:

Chapter 1: Introduction. This section provides general information about the DD document and the system to be developed.

Chapter 2: Architectural Design. This section shows the main components of the systems with their sub-components and their relationships, along with their static and dynamic design. This section will also focus on design choices, styles, patterns and paradigms.

Chapter 3: Algorithm Design. This section will present and discuss in detail the algorithms designed for the system functionalities, independently from their concrete implementation.

Chapter 4: User Interface Design. This section shows how the user interface will look like and behave, by means of concept graphics and UX modeling.

Chapter 5: Requirements Traceability. This section shows how the requirements in the RASD are satisfied by the design choices of the DD.

Chapter 2

2 Architectural Design

2.1 Definitions

2.2 High level components and their interaction

2.3 Component view

2.4 Deployment view

2.5 Runtime view

2.6 Component interfaces

2.7 Selected architectural styles and patterns

2.8 Other design decisions

Chapter 3

3 Algorithm Design

Chapter 4

4 User Interface Design

Chapter 5

5 Requirements Traceability

Chapter 6

6 References and Effort Spent