Politecnico di Milano

PowerEnjoy Service - Design Document

December 1, 2016

Version 1.1

Authors:

- Domenico FAVARO (Mat. 837995)
- Matheus FIM (Mat. 876069)
- Caio ZULIANI (Mat. 877266)

Prof. Elisabetta DI NITTO

Contents

1 Introduction **2**
- 1.1 Purpose . . . 2
- 1.2 Scope . . . 2
- 1.3 Glossary: Definitions, Acronyms, Abbreviations . . . 2
- 1.4 Reference Documents . . . 3
- 1.5 Document Structure . . . 3

2 Architectural Design **4**
- 2.1 Overview . . . 4
- 2.2 High level components and their interaction . . . 5
- 2.3 Component View . . . 6
- 2.4 Deployment View . . . 6
- 2.5 Runtime View . . . 6
- 2.6 Component Interfaces . . . 6
- 2.7 Selected Architectural Styles and Patterns . . . 6
- 2.8 Other Design Decisions . . . 6

3 Algorithm Design **6**

4 User Interface Design **6**

5 Requirements Traceability **11**

6 Effort Spent **11**

7 References **12**

8 Used Tools **13**

9 Changelog **14**

1 Introduction

1.1 Purpose

This Design Document serves the purpose to present to all parties interested the description of the structure for the PowerEnjoy Service. It provides documentation of Software, Architecture and other important aspects of Design to help the understanding and development of the System. Every component that is implemented for the System will be explained as well as the purpose they serve to contribute to the fullfilment of all the project Requirements. All strategies and design decisions will be documented as well. Software Design Description, decisions and key information to be used to communicate the general purpose of the structure to our stakeholders, and serve as detailed documentation for the components of the System for the developers.

1.2 Scope

This Document presents the PowerEnjoy Service System, an electric car sharing service. To better understand the broader scope of the service, it is presented in the RASD Document. This Document will not include detailed information on all the tools and protocols that will be used in the development of the System but rather their purpose and functionality inside the PowerEnjoy System. As example, general knowledge of the Client-Server structure is expected as it will not be rigorously explained but instead how such structure will be used to satisfy our System's Requirements.

1.3 Glossary: Definitions, Acronyms, Abbreviations

- **RASD:** Reqirements And Specifications Document.
- **DD:** Design Document.
- **Java EE:** Java Enterprise Edition. Software Development Platform.
- **EJB:** Enterprise Java Beans. Component in the Business Tier for the Application Logic.
- **DBMS:** DataBase Management System.
- **App:** Application. Refers to the deployed service as Web or Mobile Application.

For other concepts concerning the Service definition look in the **Glossary** section of the RASD.

1.4 Reference Documents

- Specification Document: Assignments AA 2016-2017.pdf
- PowerEnjoy Requirements And Specifications Document (RASD)
- IEEE Std 1016-2009 IEEE Standard for Information Technology-Systems Designs-Software Design Descriptions (SDD IEEE 1016-2009.pdf)
- Example Documents:
  - Sample Design Deliverable Discussed on Nov. 2.pdf
  - Software Design Document (SDD) Template (sdd_template.pdf)

1.5 Document Structure

Section 1 - Introduction: This section provides a general description of the purpose and structure of this Design Document.

Section 2 - Architectural Design: This section illustrates a broader to specific view of the components that form part of the System, presenting from the overview of the architecture of the system to a description of how each component will interface inside the structure.

Section 3 - Algorithm Design: Important functionalities of the System that require the development of algorithms will be described in this section.

Section 4 - User Interface Design: All details refering to how the User will interface with the System, from the Web Application to the screens inside the Cars will be shown in this section. As well as general mockups of the Graphical User Interface (GUI) screens.

Section 5 - Requirements Traceability: In this section is explained how the Design decisions and structure help fulfill the Requirements for the System that were defined in the RASD.

Section 6 - Effort Spent: Detailed record of the hours worked for each member of the team is documented in this section.

Section 7 - References: Any reference to additional external sources that can help the better understanding of this Document is documented in this section.

2 Architectural Design

2.1 Overview

As was show in the Proposed System part of the RASD, the PowerEnjoy structure will consist on a 3-tier Client-Server Architecture. The server part built on a JEE Platform with access to the Company's Database Server and the Client consisting of Users, CRM and Cars that will use a browser to access our Web Application and a Mobile Device and Screen inside the Cars will access the Mobile Application. As we need to interface our application with the Car functions (i.e. Lock, Unlock, Battery state) the application must have some Logic implemented client side. This distributed Logic structure will allow GUI to be created on the Client side as well. The Database Server will belong to the Company and will allow registration of User, Reservations, Rides and Payments. Cars, CRM Employees and Parking Locations are asumed to be added outside of this System.

- Client Tier (Web App, Mobile App, Car)
- Server Tier (Java EE Web and Application Server)
- Enterprise Information System (EIS) Tier (DataBase Server)

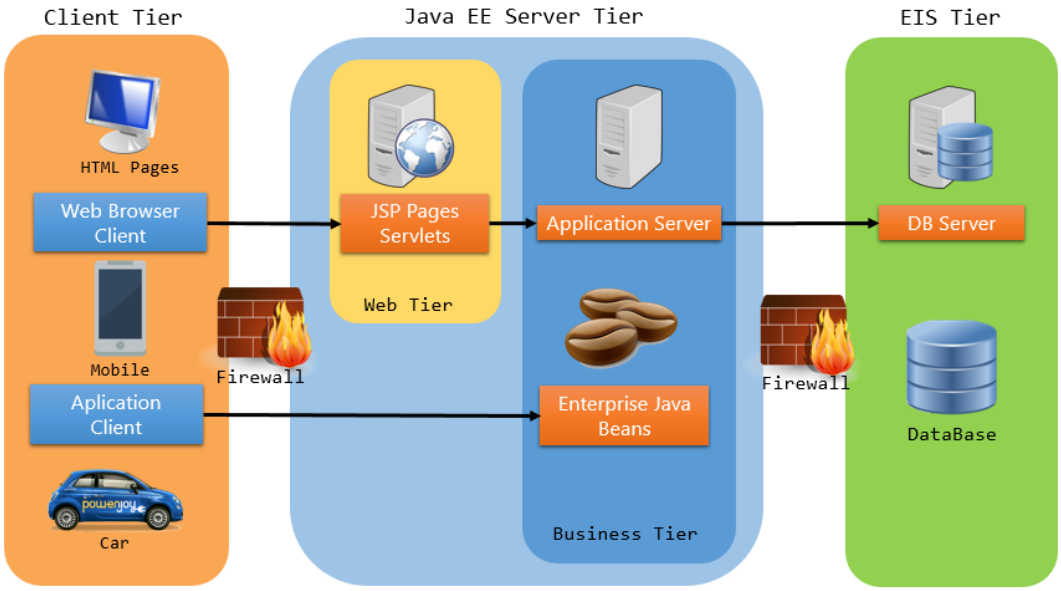

Figure 1: Proposed System Architecture

2.2 High level components and their interaction

Following our Architecture we organize the High level components on the 3 Tiers. This overview shows the interaction between the components and the entities they represent inside the System. Clients will have their User and CRM Components, distributed as Interfaces and Web Pages for the Apps. Inside the Server through Servlets and Session Beans a Session Manager will provide the services to the clients by communicating with the respective Controllers. Each Controller will manage and represent the System Entities previously explained in the RASD and will interface with a Persistent DataBase Connection to the DBMS. Helpers and Extra Feature Components are represented as well.

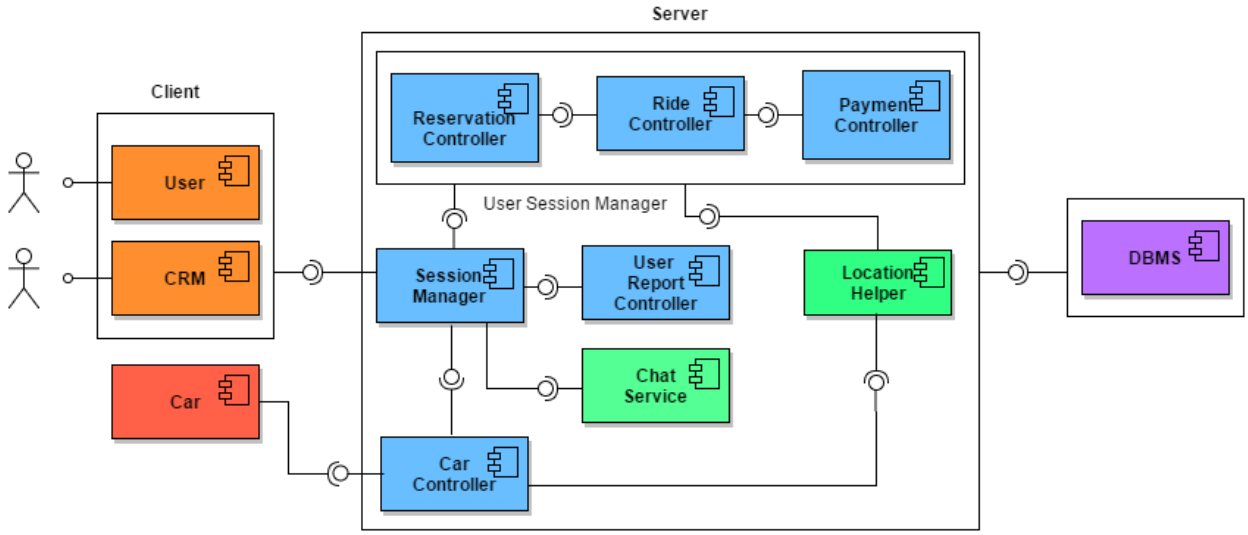

Figure 2: High Level Component Structure

- User and CRM Components: Implement the ServiceRequestors for each Actor (Web Pages and App) and will have to interface with the central Server Component.
- Session Manager: Will manage sessions with servlets and session beans and forward requests to the correspondant Controller.
- Reservation Controller: Will manage Reservations made by Users.
- Ride Controller: Manages Rides created by an activated Reservation.
- Payment Controller: Manages Payment at the end of a Ride.

- Location Helper: Will provide an interface for Location queries. Can interface to an external Service Provider API (Google Maps).
- User Report Controller: Will manage User Reports made by a CRM.
- Car Controller: Will manage Cars Status and interface to Cars to send them instructions.
- Car: Represents the Interface to control the Car and the screen inside it.
- Chat Service: Will provide the chat feature to provide communication between Users and available CRMs.
- DBMS: Each Controller will implement Persistent Entity Beans for each object that will be mapped on the DB.

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

3 Algorithm Design

4 User Interface Design

For our User Interface (UI) we'll offer a mobile App for Users and a desktop web App for Users and CRM. They will offer:

- **LogIn Page:** First screen of the app with the LogIn and SignIn options.

Figure 3: UI LogIn Page

- **Main Page:** Main page of the app where the user sees the map his location and selects the available cars. In Case of CRM, it can see all Cars.
- **Car Details Page:** Once a Car is selected this page allows to reserve a car and contais informations for the decision.

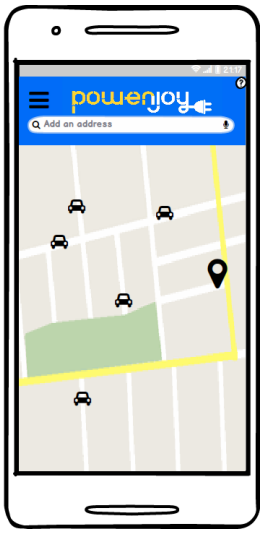

Figure 4: UI Main Page

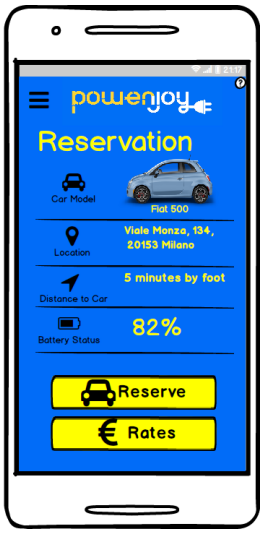

Figure 5: UI Car Details Page

- **Reservation Page:** Once the Reservation is made, the User can see the details of her/his Reservation and have the option to cancel it.
- **Ride Page:** Once a Ride started this page shows the details of the current ride, it's present also in the Car screen App.

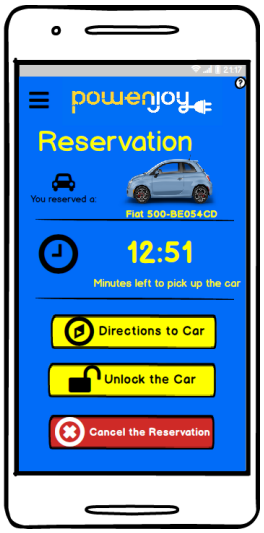

Figure 6: UI Reservation Page

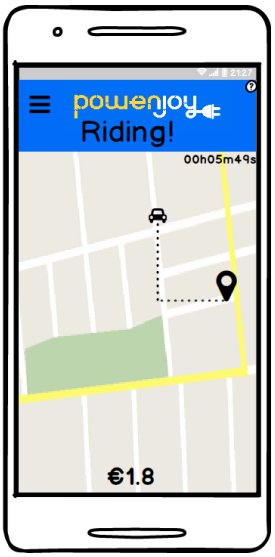

Figure 7: UI Mobile Ride Page

- **Email Page:** While not part of the App UI, the System should respond with an email to the User at the end of the Ride showing the final details for it.

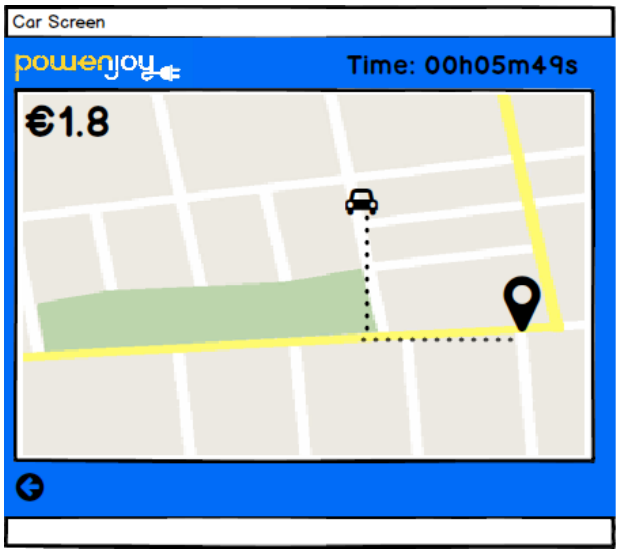

Figure 8: UI Car Ride Page.

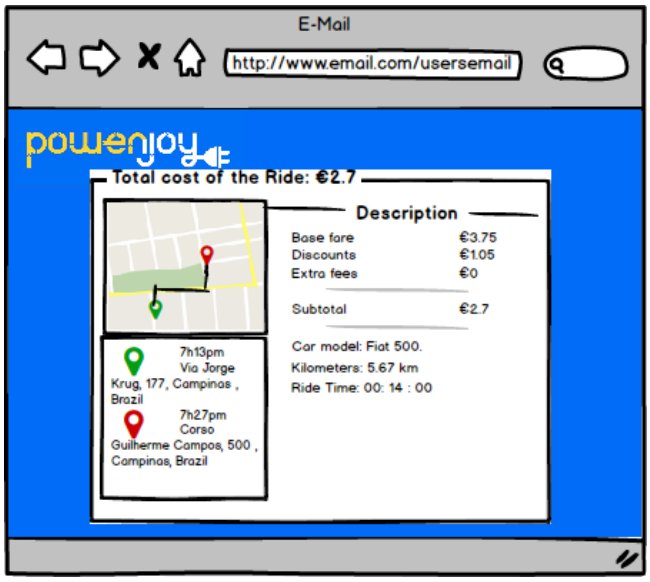

Figure 9: UI Email Page

- **Contact Page:** Page where the User can contact the CRM operator.

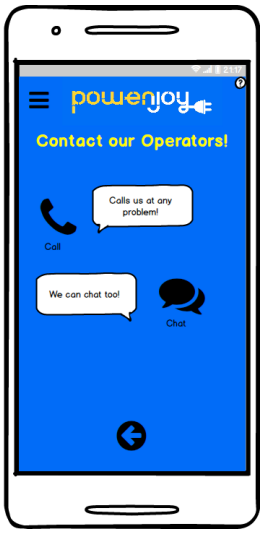

Figure 10: UI Contact Page

5 Requirements Traceability

6 Effort Spent

Date	Domenico	Caio	Matheus
27/11/16	1h	1h	1h
28/11/16	2h	-	-
29/11/16	2h	-	-
30/11/16	3h	-	-
31/11/16	1h	-	-
01/12/16	-	-	-
02/12/16	-	-	-
03/12/16	-	-	-
04/12/16	-	-	-
05/12/16	-	-	-
06/12/16	-	-	-
07/12/16	-	-	-
08/12/16	-	-	-
09/12/16	-	-	-
10/12/16	-	-	-
11/12/16	-	-	-

7 References

8 Used Tools

Keeping track of the Tools used during develop the DD document were:

- **GitHub:** for Version Control
- **Dia Diagram Editor:** for UML Diagrams
- **TeXworks:** for LaTex editing of this Document

9 Changelog

As the project and design decisions may change during the development this document is also prone to change. We'll document every version in this part.

- **Version 1.1:** 11/12/2016