



Politecnico di Milano

PowerEnjoy Service - Design Document

November 16, 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	2
1.5	Document Structure	2
2	Architectural Design	3
2.1	Overview	3
2.2	Component View	5
2.3	Deployment View	5
2.4	Runtime View	5
2.5	Component Interfaces	5
2.6	Selected Architectural Styles and Patterns	5
2.7	Other Design Decisions	5
3	Algorithm Design	5
4	User Interface Design	5
5	Requirements Traceability	10
6	Effort Spent	10
7	References	11
8	Used Tools	12
9	Changelog	13

1 Introduction

1.1 Purpose

1.2 Scope

1.3 Glossary: Definitions, Acronyms, Abbreviations

- **First-Time User:**

1.4 Reference Documents

- Specification Document: Assignments AA 2016-2017.pdf
- IEEE Std 1016-2009 IEEE Standard for Information Technology-Systems Designs-Software Design Descriptions (SDD IEEE 1016-2009.pdf)
- Example Documents:

-

1.5 Document Structure

Section 1 - Introduction:

Section 2 - Architectural Design:

Section 3 - Algorithm Design:

Section 4 - User Interface Design:

Section 5 - Requirements Traceability:

Section 6 - EffortSpent:

Section 7 - References:

2 Architectural Design

2.1 Overview

Our Proposed System consist in a 3-tier Client-Server Architecture, with web clients for Users and CRM, mobile App clients for Users and the Car interface that will be implemented as a Client as it's external to our Server System. The Server side will consist on a Java EE Server that will implement a Web Tier for Web Clients and a Business Tier that'll communicate directly to the App Clients. The Server will then connect to the Company's Database that'll be located in our EIS Tier. Project will be developed using a JEE Platform based in a 3-Tier Client/Server Architecture. First idea consists in a Distributed Logic structured layers where the Client Apps will interface with the Server that will store the logic and will connect with the Data Layer of the System. The architecture as explained briefly in the **Proposed System** section will consist of:

- Client Tier:
 - Web Client on a browser with abilitated JavaScript for the Web App use.
 - Application Client for the Mobile App (Android and iOS), with GPS functionalities. We will consider the screen present in the Car as a Client App of the System that'll implement some logic to interface to the Car Software to Lock, Unlock, read the Battery level and other functions.
- Server Tier:
 - Web Server that'll answer connections for Web Clients. It will interface with the Application Server.
 - JavaEE Application Server, that'll implement the logic of the System. Application Clients (Users and Cars) will connect directly to it. Will implement JavaBeans to handle request from multiple Clients at a time.
- Enterprise Information System (EIS) Tier:
 - DataBase Server, accessed by the Application Server to store all the data refering to the System.

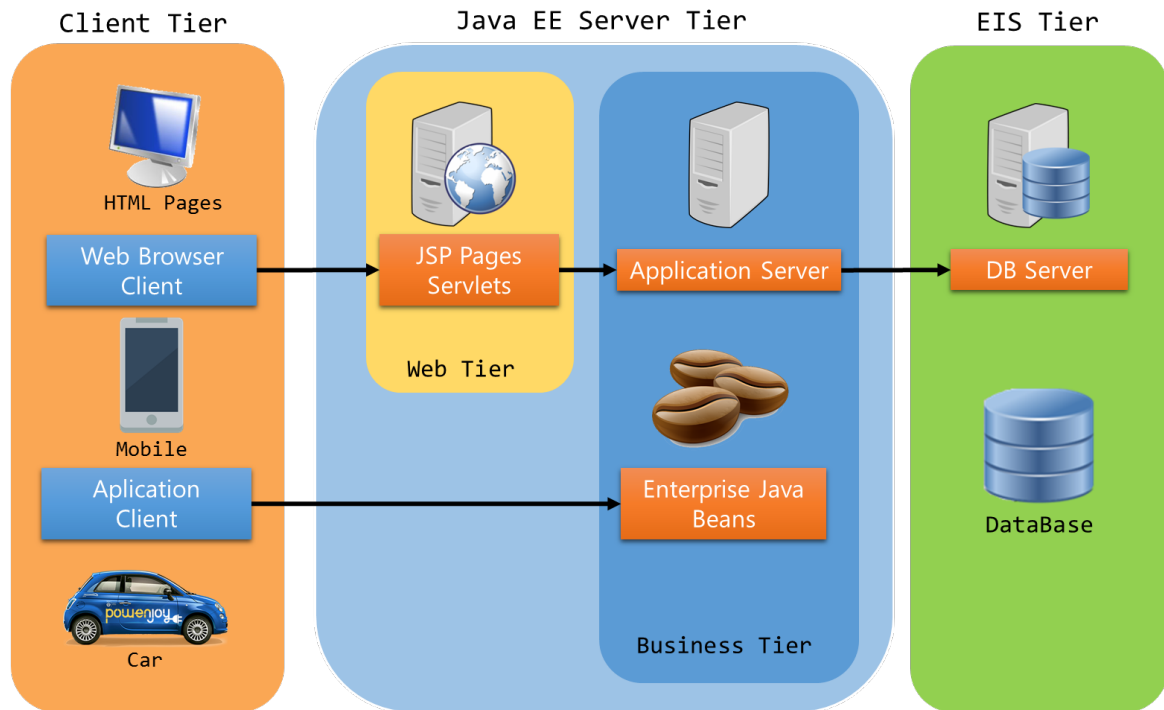


Figure 1: Proposed System Architecture

2.2 Component View

2.3 Deployment View

2.4 Runtime View

2.5 Component Interfaces

2.6 Selected Architectural Styles and Patterns

2.7 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 2: 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.

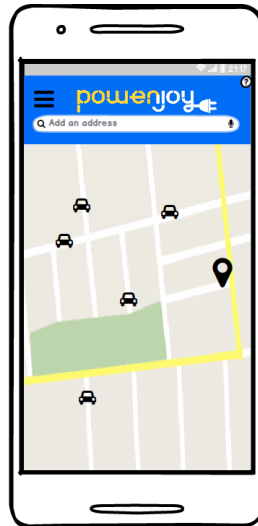


Figure 3: UI Main Page

- **Car Details Page:** Once a Car is selected this page allows to reserve a car and contains informations for the decision.

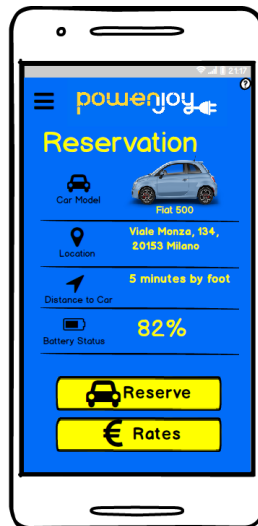


Figure 4: UI Car Details Page

- **Reservation Page:** Once the Reservation is made, the User can see the details of her/his Reservation and have the option to cancel it.

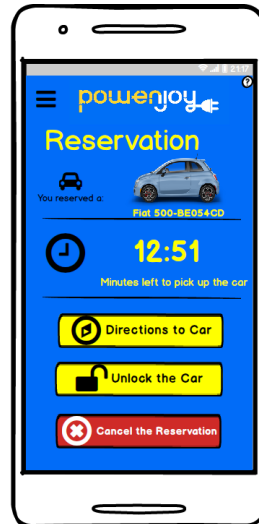


Figure 5: UI Reservation Page

- **Ride Page:** Once a Ride started this page shows the details of the current ride, it's present also in the Car screen App.

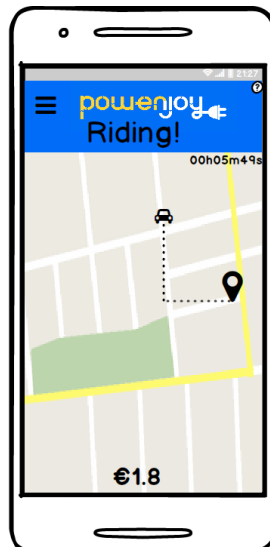


Figure 6: UI Mobile Ride Page

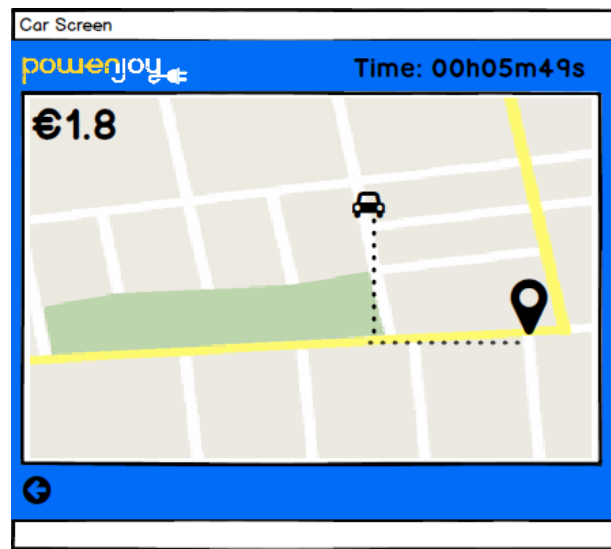


Figure 7: UI Car Ride Page.

- **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.

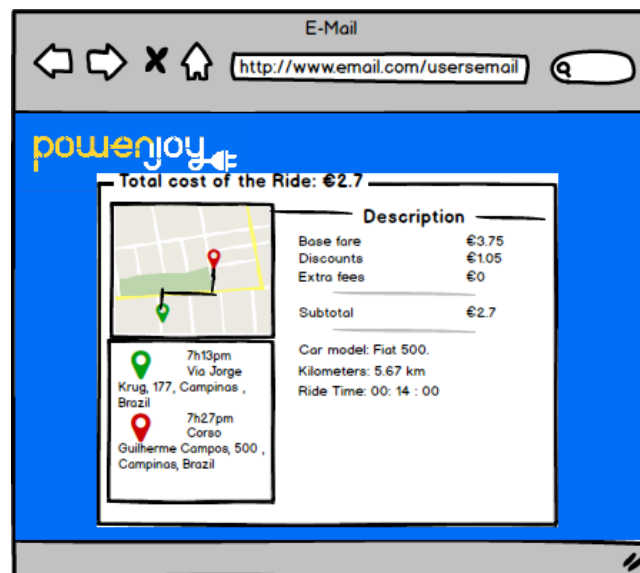


Figure 8: UI Email Page

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

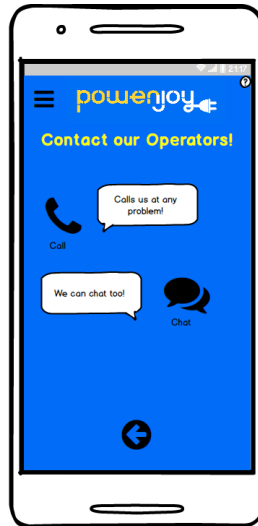


Figure 9: UI Contact Page

5 Requirements Traceability

6 Effort Spent

Date	Domenico	Caio	Matheus
24/11/16	-	-	-
25/11/16	-	-	-
26/11/16	-	-	-
27/11/16	-	-	-
28/11/16	-	-	-
29/11/16	-	-	-
30/11/16	-	-	-
31/11/16	-	-	-
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