PowerEnJoy

## Design Document

Cattaneo Davide El Hariry Matteo Frontino Francesco

Contents

1. ***Introduction***
   1. Purpose
   2. Scope
   3. Definitions, acronyms and abbreviations
   4. Reference documents
2. ***Architectural design***
   1. Overview
   2. High level components and interactions
   3. Component view
   4. Deploying view
   5. Runtime view
   6. Component interfaces
   7. Architectural styles and design patterns
      1. Overall architecture
      2. Protocols
      3. Design patterns
   8. Other design choices
3. ***Algorithm design***
   1. Algoritmo1
   2. Algoritmo2
   3. Algoritmo3
4. ***User interface design***
   1. Mockups
   2. UX diagrams
5. ***Requirements traceability***
6. ***Revision***
   1. Software and tools used
   2. Team work
7. ***Introduction***
   1. ***Purpose***

Purpose of this document is to define and design an architecture on which to rely when developing the software platform.

Addressed to developers, this document contains all the major guidelines to follow while developing the PowerEnJoy system. Both the application and server side architecture are treated in this document.

* 1. ***Scope***

This project aims at designing an electric-car sharing software system.

Car Sharing is a very cost-effective and useful service for anyone who needs a car occasionally. It allows people to use and pay for the car according to their personal use, without the hassle and costs of owning their own vehicle (parking, purchase costs, maintenance, insurance etc.).

The system we will develop is meant for cities which are provided with an efficient amount of parking lots and a wide distribution of electric car-charging platforms throughout the urban areas.

The application must allow the users which are registered to perform several easy and effective operations. Once logged in, the user can find available cars around him/her or in specified locations of the city, and chose the one to reserve.

Afterwards the user, who needs to reach the car before a given time slot expiration, will be able, by unlocking the car using the app, to easily enter the vehicle and drive to his/her destination.

* 1. ***Definitions, acronyms and abbreviations***

Here is a brief description of the most important actors and words used in our system:

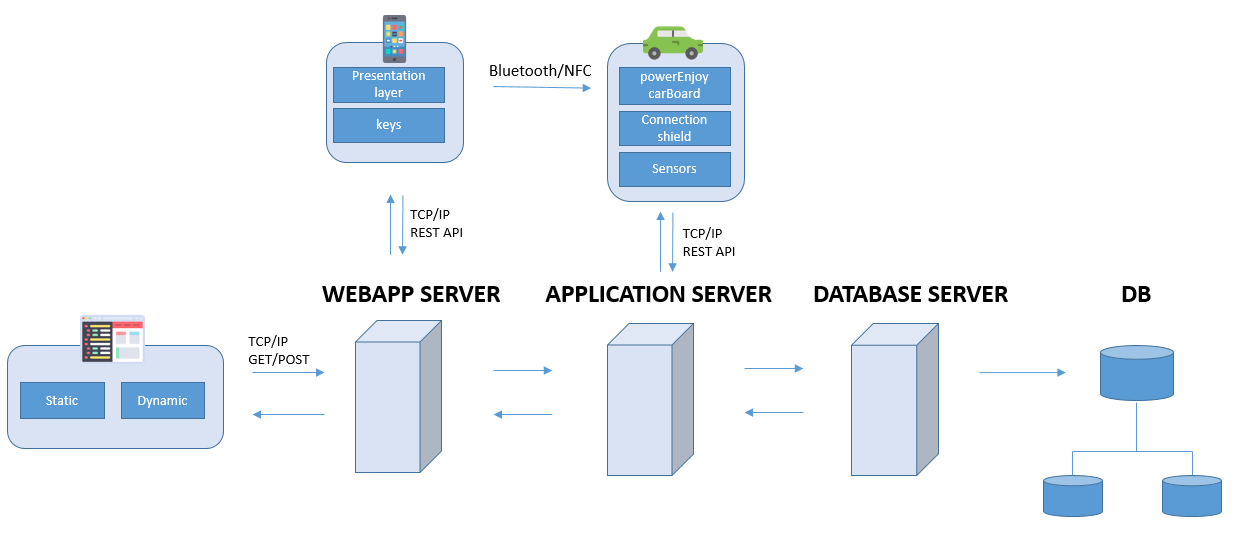
* **User:** by user is meant a person already registered in the system, so that has a profile, uses the features provided by the system and performs actions accordingly. (S)He can use all the functionalities described below (see Functional Requirements).
* **Guest:** a guest is a person that probably for the first time accesses the system or that hasn’t already signed up. Guest has less power in the system; his/her actions are limited to access an introduction view and register to the service.
* **System:** is the application core. The software system which will perform all the operations and monitor interactions and be a medium between users and cars.
* **Reservation:** the allocation of a car to a user, which starts when the booking request arrives and ends either when the expiration time ends or when the car is unlocked. In this last case it triggers the start of the first travel so it initiates a ride.
* **Car:** the vehicle used by the users, which contains different sensors and an embedded computer. It has seat sensors to detect passengers, sensor to know battery level and charging actions. The computer, of course, has as main functionality to provide navigation facilities through a GPS system and to send all the relevant data to the main system server.
* **Ride:** conceptually is the use of the car, and it can be identified by the time duration of the user’s journey, from unlocking the vehicle until the final parking (having user selecting “end ride” or “end ride & charge” on the car screen) with the car locked.
* **Travel:** is considered as the ride segment and is identified by a change of the status of the car. More travels can be part of a single ride.
* **Operator:** is a flexible actor in our system. He’s part of a set of people operating under the administrator directions. Their normal tasks are to bring to charging stations cars left with less than 15% battery level, interact with users which call for help during a ride, intervene when necessary (e.g. a wheel brakes during a ride). Their exceptional task can be the case in which they have to go and get back cars taken by the police or cars involved in incidents etc.
* **Administrator:** the administrator of the system is the person allowed to manage eventual unexpected cases (like incidents and damaging situations). He is the person notified every time a problem occurs, and once analyzed the situation (s)he’ll decide how to handle it (call for support, send operators, call the police etc.).
* **Safe Area:** is a part of a set of areas considered safe for parking cars after a ride is over. Temporary stops can be everywhere, but long term parks can only occur in safe areas. They must be very spread and every neighborhood should have at least one.
* **Normal rate:** the charging rate applied when the car engine is ON.
* **Halt rate:** the charging rate applied when the engine is OFF and either the user is inside or (s)he has parked the car in temporary stop mode.
  1. ***Reference documents***

Specification Document:

\* Assignments 1 and 2 (RASD and DD).pdf

\* Design Part I.pdf and Design Part 2.pdf from lecture slides

Examples documents:  
\* Sample Design Deliverable Discussed on Nov. 2

1. ***Architectural design***
   1. ***Overview***

PowerEnjoy will be available to our customers as a mobile application available either to android, iOS and windows mobile.

We have decided to provide also a web application accessible either from users and administration.

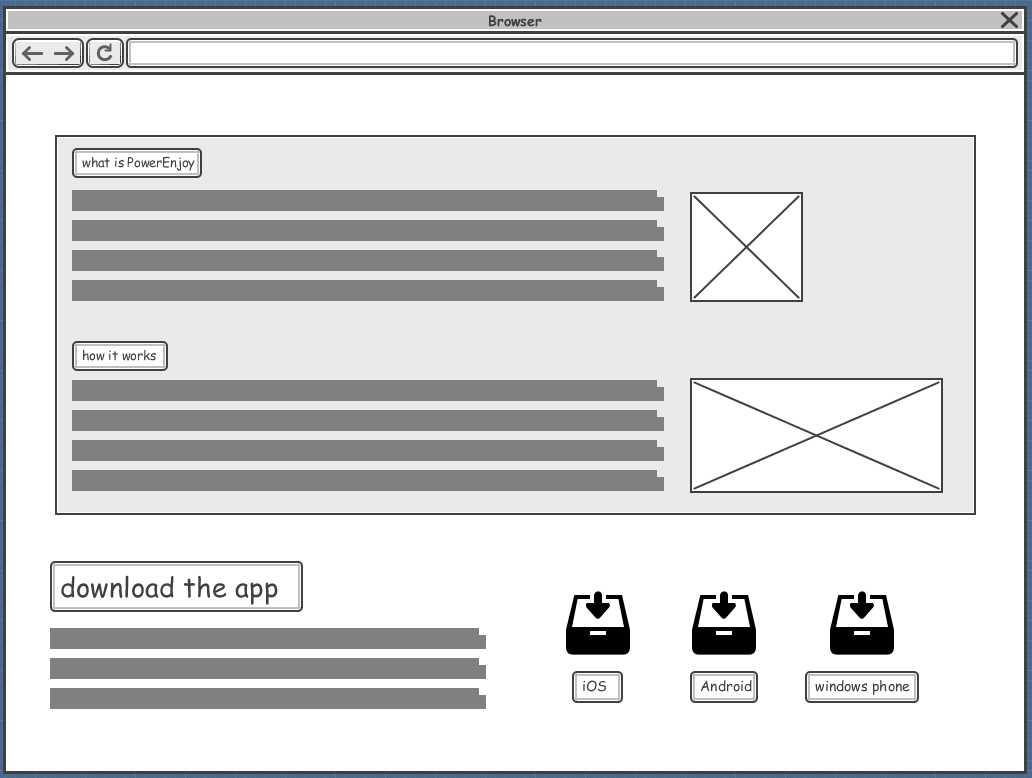
The maintenance side will be enabled only after a successful login. Administrator can manage and edit all data stored into the database, assign tasks to employee and manage car's remote functionality.

The choice of a distributed database was made to guarantee the reliability property. We have also expected a list of servers divided on all the users and cars. If one server go down, cars and users related to that server will be transferred to another one and the system will keep working with a low performance.

The main protocol used by our system is TCP/IP.

* 1. ***High level components and interactions***
  2. ***Component view***
  3. ***Deploying view***
  4. ***Runtime view***
  5. ***Component interfaces***
  6. ***Architectural styles and design patterns***
     1. ***Overall architecture***
     2. ***Protocols***
     3. ***Design patterns***
  7. ***Other design choices***

1. ***Algorithm design***
   1. ***Algoritmo 1***
   2. ***Algoritmo 2***
2. ***User interface design***
   1. ***Mockups***



* 1. ***UX diagrams***

1. ***Requirements traceability***

aaa

1. ***Revision***
   * 1. ***Software and tools used***

The following software have been used:

* Microsoft Word (document writing)
  + 1. ***Team work***

Here is reported a compact table showing how the work was brought on by all the members of the group.

|  |  |
| --- | --- |
| *Member* | *Hours of work* |
| Cattaneo Davide |  |
| El Hariry Matter |  |
| Frontino Francesco |  |