

Luca Scannapieco - 877145

Andrea Pasquali - 808733

Emanuele Torelli - 876210

11/12/2016

PowerEnJoy

Design Document

Table of contents

[1. Introduction 3](#_Toc468215116)

[1.1. Purpose 3](#_Toc468215117)

[1.2. Scope 3](#_Toc468215118)

[1.3. Definitions, acronyms, abbreviations 3](#_Toc468215119)

[1.4. Reference documents 3](#_Toc468215120)

[1.5. Document structure 3](#_Toc468215121)

[2. Architectural design 3](#_Toc468215122)

[2.1. Overview 3](#_Toc468215123)

[2.2. Component view 3](#_Toc468215124)

[2.3. Deployment view 3](#_Toc468215125)

[2.4. Runtime view 3](#_Toc468215126)

[2.5. Component interfaces 3](#_Toc468215127)

[2.6. Selected architectural styles and patterns 3](#_Toc468215128)

[2.7. Other design decisions 3](#_Toc468215129)

[3. Algorithm design 4](#_Toc468215130)

[4. User interface design 4](#_Toc468215131)

[5. Requirements traceability 4](#_Toc468215132)

[6. Effort spent 4](#_Toc468215133)

[7. References 4](#_Toc468215134)

# 1. Introduction

## 1.1. Purpose

## 1.2. Scope

## 1.3. Definitions, acronyms, abbreviations

## 1.4. Reference documents

## 1.5. Document structure

# 2. Architectural design

## 2.1. Overview

The PowerEnJoy service is implemented as a common client-server application, in which the offered services are essentially three:

* User interface
* Application logic
* Database

Each of these logic services is placed in the corresponding physic layer, the result is that we adopted a three-tier architecture.

We provide a mockup to better understand the structure of the PowerEnJoy service.

MOCKUP

The user interface has two different implementations, one is constituted by a web app that can be executed on a modern browser, and the other one is the PowerEnJoy mobile application. Furthermore, during a ride, there is another component with which the user interacts: the screen of the car. On the other hand, the assistance coordinator has an interface built ad hoc to performs his work, in fact the tasks that the coordinator must do are forbidden to the common users.

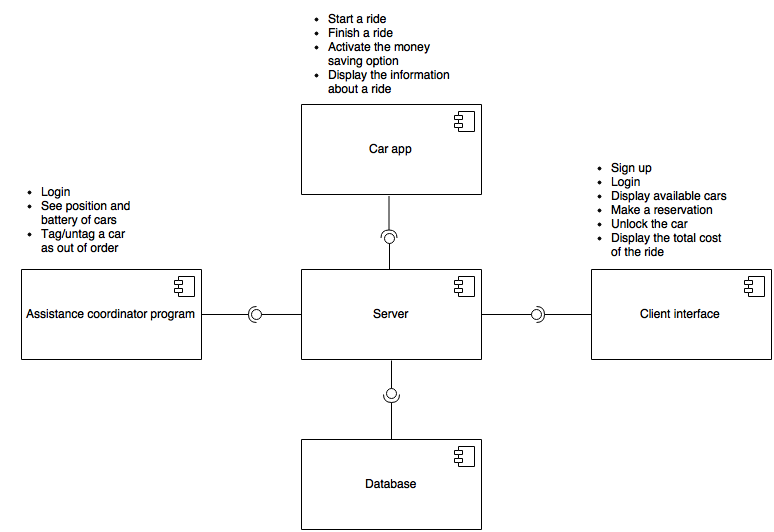
The core of our architecture is the main central server, where is provided the logic necessary to the system to run properly. In addition to the central server we need an external web server to run the PowerEnJoy web app properly.

Separating the layer for the application logic and the database we can ensure a high modularity of the system, and for example the company can decide to move one of the layers (or both) to a cloud service, for example to amazon AWS where it would have dedicated cloud servers with load balance for database and other for application logic on demand.

## 2.2. Component view

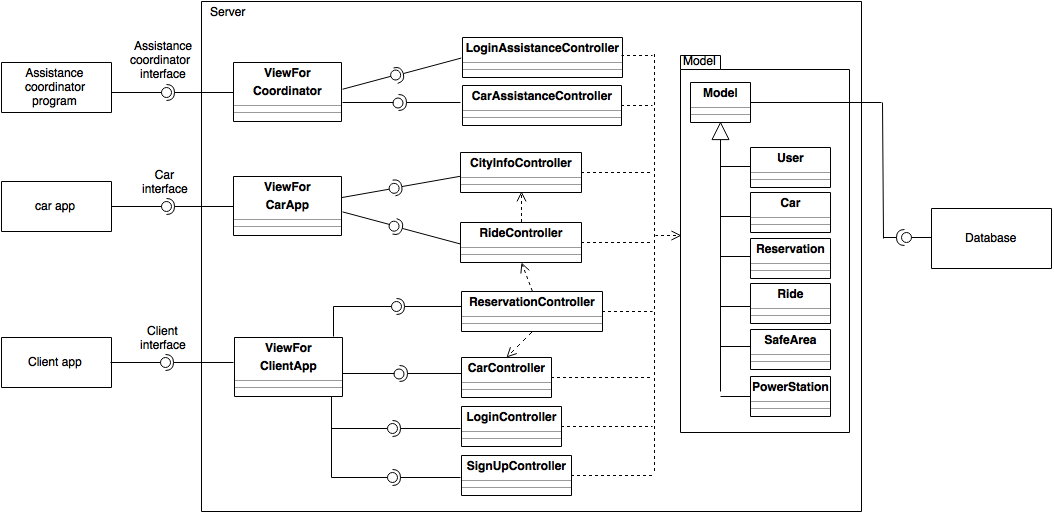
In the following diagram we show the main components of the PowerEnJoy system, seen at the highest level as possible.

Note that for each of the three components of the Tier 1 (the client app, the car app and the coordinator app) we describe the functionalities that the system must ensure; we derived those functionalities from the use cases diagram presented in the RASD.



Esplosioine del component view

* Spiega a grandi linee la struttura del server
* Spiega le varie interfacce
* Spiega perché abbiamo le view
* Spiega che c’è un model che riflette la struttura del db



* Spiega il database (class diagram)

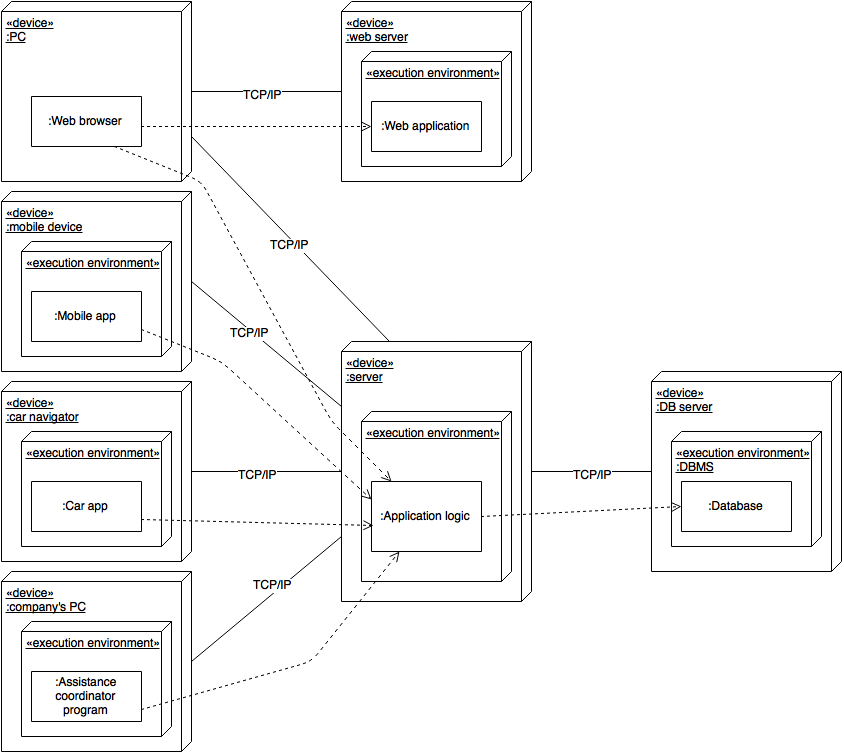
CLASS DIAGRAM DB

## 2.3. Deployment view

-Spiega a cosa serve

-Spiega perché TCP IP

-spiega del wer server



## 2.4. Runtime view

See available cars

Reservation

Unlock and start a ride

End ride

## 2.5. Component interfaces

Due parole introduttive

COMPONENT VIEW

## 2.6. Selected architectural styles and patterns

-mvc server

## 2.7. Other design decisions

inventa

# 3. Algorithm design

# 4. User interface design

# 5. Requirements traceability

# 6. Effort spent

# 7. References