1. INTRODUCTION

A. Purpose

B. Scope

C. Definitions, Acronyms, Abbreviations

D. Reference Documents

E. Document Structure

2. ARCHITECTURAL DESIGN

A. Overview: High level components and their interaction

C. Component view

D. Deployment view

E. Runtime view: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases

F. Component interfaces

G. Selected architectural styles and patterns: Please explain which styles/patterns you used, why, and how

H. Other design decisions

3. ALGORITHM DESIGN: Focus on the definition of the most relevant algorithmic part

4. USER INTERFACE DESIGN: Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have already done, possibly, providing here some extensions if applicable.

5. REQUIREMENTS TRACEABILITY: Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.

6. EFFORT SPENT: In this section you will include information about the number of hours each group member has worked towards the fulfillment of this deadline.

7. REFERENCES

# INTRODUCTION

## Purpose

The purpose of this document is to give more technical details about PowerEnJoy service than the RASD does.

This document is addressed to developers and aims to identify:

* the high level architecture;
* the design patterns;
* the main components and their interfaces provided one for another;
* the Runtime behavior.

## Scope

In particular, the main purposes of the system are:

1. to be reliable and efficient for the users of a car-sharing service;
2. guarantee an energetic and accessible car service.
3. The system allows clients to reserve a taxi via a mobile or web app, using his
4. GPS position our inserting his position manually to find a taxi in the same zone.

The system allows clients to register as a user by supplying the valid driving license information and credit card information.

After receiving an access the system and logging in it, users are able to search the available cars around the location they are or the specified position. System response the search request by enumerating the available cars around them in an available queue as well as the basic information of them. The basic information of a car includes the current location, the distance to the starting position, the rest of the charge and the seating.

Users are able to reserve no more than one car every time, they can cancel the reservation free within a certain time after the reservation request. The user who has not picked the reserved car within one hour from the reservation, he/she should pay a fee of 1 euro, and the reservation is withdrawn automatically as well.

The system removes the car from the available queue as soon as the car is reserved. It moves back if a user cancels the reservation or the car isn’t picked up in one hour.

If the user reaches the reserved car within the time limit, he/she should send request to pick up the car. The system unlocks the door and lets the user enter. The system starts charging for the car as soon as the user ignites the engine, and the charging information is notified through the screen in the car.

The user is able to choose the “money saving option” after picking up the car, and the system shows the optimal terminal station of the car parking.

Users are able to terminate their ride in the safe area. The charging stops and the car is locked as soon as all passengers leave the car, regardless of the fact the engine of the car works or not.

If the user parks the car in the unsafe area and leaves, it is necessary to lock the car for the reason of security; the system will consider that as a complete driving process with unsafe parking.

Users can recharge the car in the power gird station by plugging the car into power grid.

The system estimates the behavior of users in their ride according to the riding clauses that show what to do to operate the payment for the riding:

1. If the user took more than 2 passengers, apply 10% discount.
2. If the user parked the car in safe area with more than 50% dump energy, apply 20% discount.
3. If the user recharged the car after parking, apply 30% discount.
4. If the place user parked car is in a longer distance than 3KM to the nearest power grid station, apply 30% compensation.
5. If the user parked the car with less than 20% dump energy, apply 30% discount.
6. If the user parks the car in an unsafe area, apply 30% compensation.

## Definitions, Acronyms, Abbreviations

* API: application programming interface; it is a common way to communicate with another system.
* BCE: business controller entity
* DD: design document
* MVC: model view controller
* Push notification: it is a notification sent to a smartphone using the mobile application, so it must be installed.
* Push service: it is a service that allows to send push notifications with own API
* RASD: requirements analysis and specifications document
* SMS: short message service; it is a notification sent to a mobile phone, an SMS gateway is needed to use it.
* SMS gateway: it is a service, which allows to send SMS via standard API.
* URL: uniform resource locator
* UX: user experience design
* Safe area: it is an admissive parking area, pre-defined by the PowerEnJoy system; the power grid stations are the subset of safe area.
* Car: it is an electric car supplied for the car-sharing service in the PowerEnJoy system.
* Basic car information: it is a basic information that helps guests and users to make decisions and includes the dump energy, location information, distance to the setting location, the passenger capacity.
* Starting position: a current positon of user or a position, which a user inputted to start a ride.
* Car state: every car has four states:

1. Available: the car has dump energy more than 50% and be parked in the safe area.
2. Reserved: the car is reserved by a user, but still is in the safe area.
3. In use: the user has picked up the car.
4. Out of service: the car has battery more than 50% empty or it is not been parked in the safe area

* Available queue: it is a queue that maintains available cars
* Electronic devices: the GPS and power plug sensor, weight sensor, display screen, battery sensor, door state sensor, locks of door in the car, and the sensor on the power grid.
* Unsafe area: the area that is not included in the pre-defined safe area list.
* Incorrect parking: it is a parking in the unsafe area, a parking of the low battery car in power gird station and a parking of the low battery car in power grid station without charging the car with power grid.
* System: it is the whole system including the electronic devices and the PowerEnJoy system background.
* Complete driving process: in PowerEnJoy service, it is a process starts with igniting the engine and ends when all passengers leave the car.

## Reference Documents

* RASD produced before, version 1.1;
* Specification Document: Assignments AA 2016-2017.pdf;
* Sample Design Deliverable Discussed on Nov. 2.pdf.

## Document Structure

* **Introduction**: this section introduces the design document. It contains a justification of his utility and indications on which parts are covered in this document that are not covered by RASD.
* **Architectural design**: this section is divided into several parts:

1. Overview: this section explains the division in tiers of our application and gives a global view of the components of the application and their interactions.
2. Component view: this section gives a more detailed view of the components of the applications.
3. Deployment view: this section shows the components that must be deployed to have the application running correctly.
4. Runtime view: this section represents sequence diagrams to describe the way components interact to accomplish specific tasks typically related to use cases of our application.
5. Component interfaces: the interfaces between the components are presented in this section
6. Selected architectural styles and patterns: this section explains the choice of the styles and patterns taken for development of our application
7. Other design decisions

* **Algorithm design**: this section describes the most critical parts via some algorithms and focuses on the definition of the most relevant algorithmic part with using pseudo code.
* **User interface design**: this section contains an overview on how the user interfaces of our system will look like via UX and BCE diagrams.
* **Requirements traceability**: this section explains how the requirements defined in the RASD are linked to the design elements defined in this document.
* **Effort spent**: this section contains information about the number of hours each group member has worked towards the fulfillment of this deadline.

# Architectural design

## Overview



## Component view



## Deployment view



## Runtime view

## Component Interfaces

