

Requirements Analysis and Specifications Document (RASD)

Simone Mosciatti & Sara Zanzottera

November 3, 2016

Contents

1 Introduction

In this section we are providing an overview of the whole PowerEnJoy Project, we will start highlight what the product will do, what functions it will provide, what constraints it will have and what assumption we made.

1.1 Purpose

The purpose of this document is to analyze the requirements for the project and provide detailed specifications for PowerEnJoy, a digital management system for a car sharing service that features only electric cars.

1.2 Overall Description of the given problem

PowerEnJoy is a digital management system for car sharing that exclusively employs electric cars to provide its service. The system provides all the functionalities normally provided by a car sharing service, like registering to the service, find the location of nearby available cars, reserve cars up to a short amount of time (namely one hour), unlock the chosen car once found, ride it and then park it in a safe area, when it will be automatically locked and the fee paid.

In addition, the system gives bonuses and penalties in term of discounts or extra fees depending on the behavior of the user, in order to incentivize a virtuous behavior. Some examples are:

- a discount of 10% for users who brings other passengers with him during the ride
- a discount of 20% if the car is left with at least 50% of the battery still full, or a 30% if the car is left near a charging station and plugged.
- if the car is left more than 3km from the nearest charging station or with less than 20% of charge left, the user is charged 30% more

PowerEnJoy is divided in two main part: a frontend, used by the customers, and a backend that provides the service.

Product Perspective

PowerEnJoy provide users the ability to rent cars for a limited amount of time in a bounded area, for example a city. The frontend part provides the user the ability to look for available cars, reserve cars for up to one hour and unlock the cars once they are near the reserved vehicle.

The backend part takes care of providing all the required data to the frontend. It also tracks the position of every single car, the duration of each ride and charges them accordingly.

Product Functions

PowerEnJoy is provided to the user via a mobile application. A registered user can log in into its account, while a new user is prompted to register and then to log in.

After the user logged in they will be able to see the position of each car in the city. Once the user has decided which car he wants to use, they will be able to book the car for one hour. After the car is been booked, the user can unlock the car only when in proximity of the car itself.

The system starts charging the user when the engine starts running, and will stop once the car is been parked and the user exit from it. Several forms of discount or overfees will be applied depending on the behaviour of the user.

Parallel Operation

The system is capable to operate and serve users concurrently, however an, HA, single point of coordination is required in order to avoid conflicts on the booking system.

1.3 Text Assumptionss

Assumptions on the car system

1. Each car is provide with internet connectivity and capable to send and receive data to the main server
2. Each car is provided of a GPS of reasonable accuracy.
3. Each car can capture all these events correctly:
 - the ignition of the engine,
 - the users getting into the car
 - the users exiting the car
 - the number of passengers.
 - the user plugging the battery to the power grid.
4. Each car is able to monitor the residual charge of its battery.

Assumptions on the final user

1. The final user has a smartphone with the PowerEnJoy app installed
2. The final user has a smartphone with a GPS
3. The final user has a smartphone with internet connectivity

External Services

The system depends on external services that are served by third-parties. These services are:

- Charging the fee to the user
- Determinate if the car is been parked in a safe position
- Determinate the distance from the closest power grid station
- Determinate the distance between two different GPS signals

The decision to rely on external partners to run these functions has been made since those services are not the core business for PowerEnJoy, and other external companies has already found very efficient and cost-effective solution to them.

1.4 Constraints

Constraints on the user

Users who wish to use PowerEnJoy are required to fulfill some specific requirements:

1. The user must be registered to the service.
2. The user must have a valid driving license.

Platform Contraints

Also, the system has some platform constraints coming from the device that is running the app:

1. The app should be small enough to fit the memory of ideally all smartphones.
2. The app should be thin enough to fit the computational capabilities of ideally all smartphones.
3. The app should send and receive only small quantity of data through the smartphone's internet connection, in order to prevent the user to be charged a heavy fee from telecom companies.
4. The app should be able to get information from the smartphones' GPS.

Privacy regulations

The system should complain to the most recent privacy regulations in managing the data that the user are generating, thus ensuring:

1. The user must be aware that its data is being recorded by the system, including its GPS position.

2. The user must be able to see which data the system collected about him and, in case, to delete it.
3. No third parties should be able to gain access on users' data.
4. Users' data must be protected from third-parties attacks.
5. Users' data should not be possible to obtain by reverse-engineering the behavior of the system.

1.5 Proposed System

The proposed system features a client-server architecture, so it is divided into two parts: a frontend app for smartphones, which allows the users to use the service, and a backend system which deals with all the operations and coordinates them. The backend also interacts with the cars, that can be seen as a third part of the system.

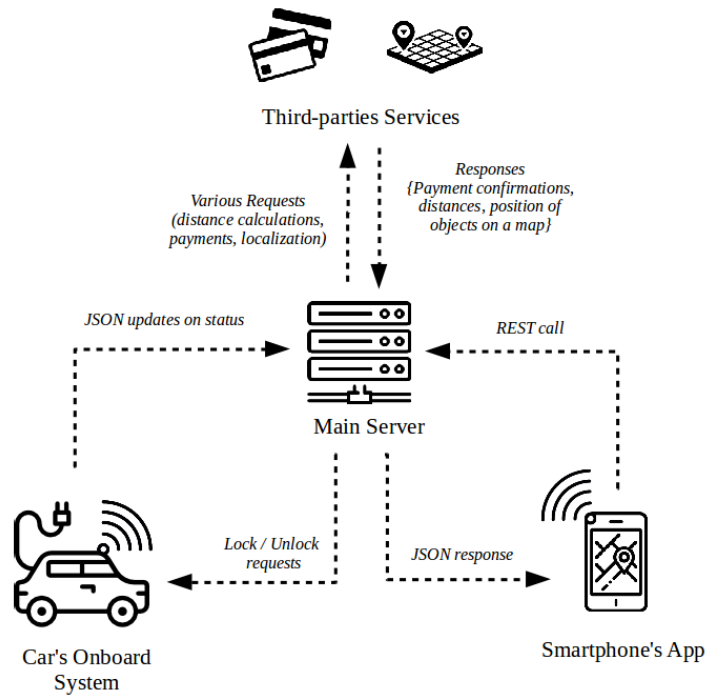


Figure 1: Description of the proposed system.

App Frontend

The frontend is a thin app that relies on the smartphone's internet connections in order to work. Almost no operations can be performed with the app alone: all transactions are sent to the main server first, then processed and the result sent back to the app.

The app can be classified as a thin client.

Centralized Backend

The backend is the core of the system. Being able to process a lot of parallel operations, it can deal with all the requests coming from the clients in a reasonable amount of time (see Non Functional Requirements). The backend is based on an MVC architecture and a REST API.

Car's Onboard system

The cars are equipped with an onboard system that monitors the status of the car, its location, and can send all the necessary informations to the main server. There won't be direct interactions between the car and the user's app.

1.6 Actors and stakeholders

Actors

The actors involved in our system are mainly **final users**, that is, people who are renting cars. Indeed the system needs support from a **technical team** that takes care of the digital infrastructure, and a **staff** that takes care of the cars in some special situations:

- Cars left far from the power grid with an empty battery, that needs to be charged in-place
- Cars left far from a safe parking area, that needs to be brought back to one of them
- Cars not plugged to the power grid, that needs to be plugged.

Stakeholders

The main stakeholder for our system are **users** themselves, who requires a easy-to-use service of electric cars sharing. Other stakeholders are the **city government**, who supports our service as a way to improve sustainable mobility inside the city and takes care of the road infrastructures, and the **energy companies**, which provide power for the cars.

1.7 Reference Documents

- *Assignments AA 2016-2017.pdf* (Assignments document given by the teacher)
- *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*
- Other Sample documents:
 - *RASD sample from Oct. 20 lecture.pdf*
 - *Libra: An Economy-Driven Cluster Scheduler Software Requirements Specification*

2 Specific Requirements

In this section we are going to illustrate the specific requirement of PowerEnJoy .

We analyze the goals that the application should fullfill, then moving on functional and not functional requirements.

2.1 Goals

2.2 Domain Properties

2.3 Definitions, acronyms, abbreviations

GPS : Global Positioning System is a global navigation satellite system (GNSS) that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

Frontend

Backend

3 Scenarios and Use Cases

3.1 Scenarios

3.2 Use Cases

4 Alloy Model

5 Hours