Politecnico di Milano

A.A 2016-2017

Requirements Analysis and Specifications Document

Version 1.0

PowerEnjoy

Instructor : Prof. Di Nitto

Authors:
Amico Simone
Chianella Claudia Beatrice
Giovanakis Yannick

CONTENTS

1. INTRODUCTION

1.1 Purpose

The aim of this document is to give an overview of the requirements and specifications of system to be developed. The goal of the document is to describe in detail all functional and non-functional requirements of the system, analyzing the needs of the customer and explaining common use case scenarios. It will set a baseline for project planning and cost estimation, giving a detailed insight to all stakeholders which include the PowerEnjoy Board , Investors and finally engineers (present and future) involved in development, testing and maintenance.

1.2 System

The to be developed software system provides a complete support to the new PowerEnjoy car-sharing service. It will allow users to register ,log-in and use the car sharing service within the limits imposed by the below specifications.

1.3 Scope

The new PowerEnjoy software system will allow visitors to register and retrieve all necessary information of common domain (ToS , contact information...).
Once the visitor has successfully registered to the system and his/her driving license has been approved, he/she will be provided with a unique username-password combination. This enables an user to log into the system at any time , from any mobile device.
Once logged in, the user can search within a certain distance for vehicles based on his/her current location or based on an input address. Eventually the user chooses to reserve an available vehicle, which can be picked-up within a time span of one hour.
Should the user not pick-up the car within the one hour availability limit, then the system will tag the vehicle as available and charge the user a fee

of 1 Euro.

Should the user reach the car within the limit,he/she must be able to interact with the system in order to unlock the vehicle and grant the user access to the car. As soon as the engine ignites,the system starts charging the user for a given amount of money per minute. The current charge will be notified through a display inside the car.

The system stops charging the user as soon as the car is parked in a safe area and the user exits the car. At this point the system will automatically lock the car. Should the user stop in a non-safe area he/she won't be granted to end the transaction and will be charged until he/she parks in a safe area. The set of safe area parking spots is defined inside the system and must be available all the time to the user's knowledge through the on-board display.

1.3.1   Goals

The PowerEnjoy software system needs to be:

- **[G1] : Practical**
  It wants to give the users a better connectivity among places in the participant cities that common public transport often can't offer.

- **[G2] : Convenient**
  The service needs to be convenient: it's less expensive than taking a taxi. It's an eco-friendly solution : car-sharing reduces the number of vehicles circulating in the cities and doesn't pollute thanks to electric engines.

- **[G3] : User friendly**
  The system needs to be as user friendly as possible in order to be used easily by everyone through their own intuitive knowledge.

- **[G4] : Fast**
  The system needs to be fast and responsive even during high peaks of activity.

- **[G5] : Available**
  The system needs to be available at every time and on the largest possible number of devices (mobile and not) with an internet connection.

1.3.2 Applications

To achieve the above mentioned goals to following software tools need to be developed:

- **Mobile Application**: to be developed for all major operating systems (iOS, Android , Windows Phone). The mobile software must allow the user to perform all the tasks in the *Scope section (1.3)*.

- **Web Application**: must be compatible with all major browser applications. The user must be allowed to perform all tasks in the *Scope section (1.3)*.

- **On board car application**: this application needs to handle user interaction such as displaying general car information (charge, fee, tire pressure...), letting the user validate his/her identity through a pin code and report about the status of the vehicle. Furthermore it must control if the vehicle is parked in a safe area.

- **Back-End Application**: this application handles all search and reservation requests, payment transactions and car communications. It is also an essential tool for customer support operators.

1.4   Reference Documents

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

- Project description: Assignments AA 2016-2017.pdf

- Alloy Language Reference : `http://alloy.mit.edu/alloy/documentation/book-chapters/alloy-language-reference.pdf`

- UML Language Reference : `https://www.utdallas.edu/œchung/Fujitsu/UML_2.0/Rumbaugh--UML_2.0_Reference_CD.pdf`

1.5   Glossary

TODO

1.6 Document overview

1. **Introduction**: this section gives a general description of the software and its characteristics

2. **Overall description**: this section gives a detailed overview of the document focusing on domain assumptions, requirements (functional and not) and the characteristics of different types of interacting users.

3. **Specifications**: this section gives a deep insight about the system's main functionality, analyzing scenarios with their relative use-cases and includes an extensive alloy model. Event flows are model using sequence and state diagrams.

2. Overall Description

The overview section highlights all the main functionality and constraints of the PowerEnjoy software.

2.1 Software overview
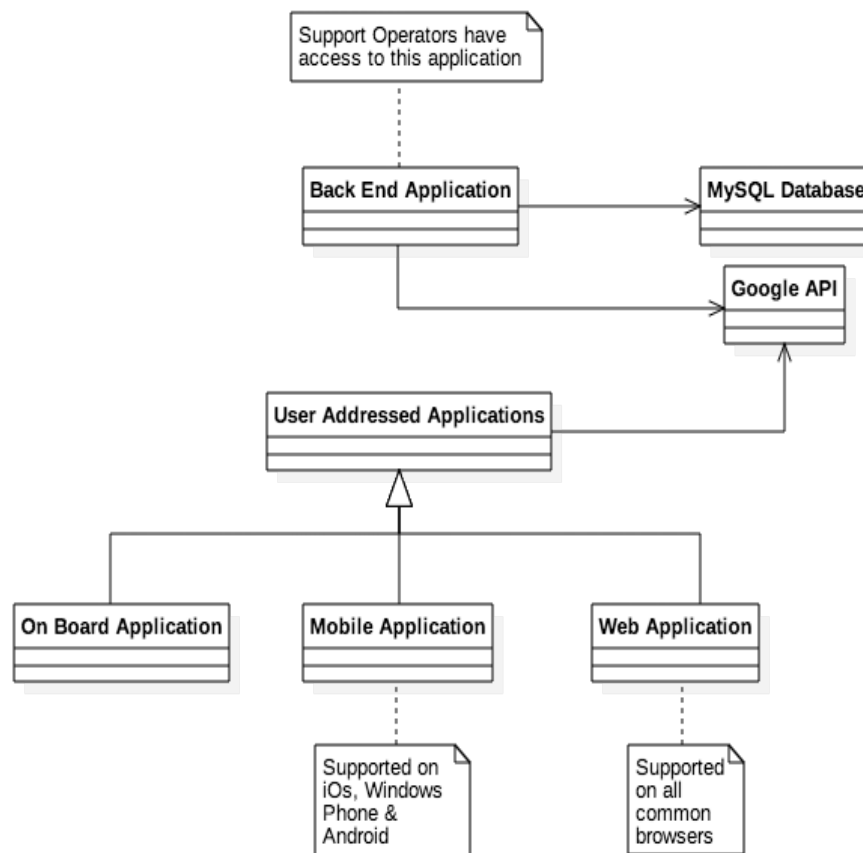
2.2 Actors

- *Visitor*: a visitor is defined as a person who is not logged into the system. A visitor can see only the log-in page with a registration form and some common information ,such as the company's customer office contact information and terms of service.

- *Registered User*: is defined as any person using the system who is also logged into the system. A registered user can perform any of the above mentioned actions.

- *Car*: is defined as the vehicle to be used by a registered user. A car must be able to communicate to with the system all the time. If a car is available (i.e currently not reserved by a registered user) then its current location must be stored in the system so to be shown on all users' devices. As soon as the engine ignition starts , the car must time the vehicle's usage in order to charge the user the right amount of money. Once the user has parked the car inside a safe area and has exited, the car informs the system about the fee to charge ,locks the doors automatically and finally informs the system that it is available again.

- *Support operator*: The support operator is an employee who is trained and has a full grasp of the different software application .His job is to offer costumer support concerning software issues and help out in case of emergency.
  Contrary to an user, he has access to the back-end software which enables him to overview the system status.

2.3 User characteristics

There are no constraints about visitors : every person can access the main site and try to register.
If a visitor wants to register there are some constraints to follow:

- the user's has to be **at least 18** years old

- the user must have a **valid** driving license (*aggiungere link al glossario *)

- the user must enter a valid payment method (*aggiungere link al glossario *)

2.4 Domain assumptions

2.4.1 General Domain Assumptions

We hold the assumption that the system will be deployed in large metropolitan cities with a maximum of 3M inhabitants.
We assume that the mean highest peak of activity is around dinner time on Fridays and Saturdays. We assume that during the peak the system will be used by around 50k people , 80% of which gain access to the system through the Mobile Application, while the remaining 20% through the Web Application.
We assume that the the fleet of vehicles will be proportional to the number of inhabitants, reaching a peek of 300 vehicles in the biggest area.

2.4.2 User Assumptions

We assume that all users are provided with an enabled internet connection equipped with GPS tracking which is the only way to search, reserve and unlock a car. We assume that all active users carry their driving license with them and drive following the traffic code[1].

2.4.3 Vehicle Assumptions

* chiedere se e' bisogna inserire assunsiozioni sulla carica delle macchine etc...*

2.5 Constraints

2.5.1 Software Limitations

The Mobile and Web Application can not exceed 50MB of RAM usage.
The On-Board application can not exceed 75MB of RAM usage.
The Back-End application cannot exceed 100GB of RAM usage.

---

[1]http://www.mit.gov.it/mit/site.php?p=normativa&o=vd&id=1&id_cat=&id_dett=0

2.5.2 Hardware limitations

The Mobile Application must be available for the following mobile operating systems:

- iOS version 7.0 or higher.

- Android version 4.0 or higher.

- Windows Phone 8 or higher.

The applications must interface with Google Maps API [2] and follow the design guidelines provided by each platform [3] [4].

The Web Application must be developed in order to be correctly used by the following web browsers [5]:

- Mozilla Firefox

- Google Chrome

- Safari

- Microsoft Edge

- Opera

The on-board software system must be equipped with a GPS Module and interface Google Maps API. The following safety regulations must be followed:

- **Night mode**: an automatic night mode must be implemented. This will avoid that drivers get blinded by excessive brightness of the display.

- **Safe - area parking recognition**: the system must recognize whether the vehicle is parked in a safe-area or not.

- **Additional passenger recognition**: the system must recognize passenger. This feature is crucial for bonus credit distribution to the driver.

---

[2]https://developers.google.com/maps/documentation/javascript/
[3]https://developer.android.com/design/index.html
[4]https://developer.apple.com/design/
[5]http://www.w3schools.com/browsers/default.asp

2.5.3 Concurrency

The back-end application must be able to handle correctly multiple user requests without creating conflicts : if two users desire two reserve the same car simultaneously ,only one can be granted a reservation.

2.5.4 Regulatory Policies

- **Privacy Policy**: Data should be collected and stored following the privacy policy guidelines provided by Mozilla Foundation `https://developer.mozilla.org/en-US/Marketplace/PublishingPolicies_and_Guidelines/Privacy_policies`

- Network connections must be encrypted.

- Data storage on PowerEnjoy databases must be encrypted

2.6 Functional Requirements

1.Mobile and Web Application

- **[R1.1]**: Allow a guest to register entering all relevant personal data and a valid driving license whose validity will be check on the spot

- **[R1.2]**: Allow a registered user to log-in

- **[R1.3]**: Allow a registered user to update his personal information

- **[R1.4]**: Allow a registered user to search for an available vehicle given his position within a selected distance

- **[R1.5]**: Allow a registered user to search for an available vehicle given an input position within a selected distance

- **[R1.6]**: Allow a registered user reserve a selected vehicle.

- **[R1.7]**: Allow a registered user to check his payment history

- **[R1.8]**: Allow a registered user to recover his account log-in information in a secure way

- **[R1.9]**: Allow a registered user to search for an available vehicle given an input position.

- **[R1.10]**: Allow a registered user to unlock *his/her* reserved car if the user is nearby.

- **[R1.11]**: **Chiedere per cancellazione prenotazioni**

2.On-Board Application

- **[R2.1]**: Allow a registered user to correctly identify himself/herself by entering a user specific pin-code

- **[R2.2]**: Enable a registered user to keep track of the distance covered , time spent using the car ,the current owed fee and the remaining battery charge.

3.Back-End Application

- **[R3.1]**: .

2.7 Non functional Requirements

Some non functional requirements have been to be necessary in the to be developed system in order to provider a correct operation flow and an optimized UX.

- *User friendliness*: the Mobile/Web as well as the On-Board Application must be easily handled by the average web user with basic web knowledge.

- *Performance*: to supply a suitable service the system must be responsive even at peak activity times. To achieve this client-server transaction must be reduced to a minimum.

- *Reliability*: the PowerEnjoy service must be active 24h/24h . Maintenance must be performed without compromising functionality. At all the time , a user must be able to be charged the owed fee.

- *Data integrity and availability*: Data have to be always available and accessible.Data availability can be achieved by duplicating data on back-up servers. For instance the use of RAID, mirroring and backup could present a valid solution.

- *Maintainability*: the software should be implemented so as to be as modular as possible to allow future expansions and easy modifications.
  Google API are required but can be easily updated from time to time on all systems.

- *Security*: user accounts are provided with two factor authentication with a code sent to the user's mobile phone to add a secure connection method to the system. Sensible user information are stored using a hashing mechanism .
  User data input should be filtered to avoid malicious modifications (i.e SQL Injections).
  All connections should be encrypted and run on a secure HTTPS protocol with valid certificates that uses the secure TSL protocol to avoid Man in the Middle Attacks.

2.8 Future implementation

The system must be as modular as possible: future implementations can require a deployment in very large metropolitan cities that exceed by far the maximum amount of planned users.
An important thing to keep in mind is that the nature of vehicles could change in the distant future : not only cars , but also electric powered motorcycles could be added to the PowerEnjoy vehicle fleet.This means that the on-board software should be as expandable as possible using for instance Hierarchy Design Patterns in the vehicle type definition.