



**POLITECNICO
DI MILANO**

Computer Science and Engineering

A.A. 2016/2017

Software Engineering 2 Project:

Power&Joy

Requirements Analysis and Specification Document

November 11, 2016

Prof. Luca Mottola

Joshua Nicolay Ortiz Osorio Mat: 806568

Michelangelo Medori Matr: 878025

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed functional description of the PowerampersandJoy car sharing management system. It is aimed at developers, and contains all the information about the architectural and design choices made, which include design patterns, the identification of the main components and the interaction among them, and the system's Runtime behavior.

1.2 Scope

The management system for the PowerampersandJoy electric car sharing service has two different targets of people: users and operators. Both of them can register and log into the system via mobile or web app (the registration and login are different for users and operators, but they are accessible via the same web AND mobile app). The system allows users to view on a map the position of all the available cars placed inside the safe area, to search for cars near a specific area, reserve a car and ride it (a user can only ride a car he reserved, and only if he manages to unlock it within an hour by the moment the reservation was made). In case a car has some kind of issues users can communicate it to the system via an on-board computer placed inside every car. The system allows users to make a pit-stop of maximum 60 minutes. In case of accident, users can report it to an operator starting a call from the on-board computer, in order to receive assistance. The system allows operators to view a list of all the cars, providing their current position , battery charge, and status. The status of a car tells weather it is functional or not, and in the latter case, describes the kind of issues (accident/lowBattery/AssistanceNeeded) affecting that car. Operators' job is to resolve any kind of issue on those cars that need it and provide assistance to users involved in accidents. Operators can inform the system they are taking care of an issued car clicking on a button available on the row of the list associated to an issued car. The system manages to allow only a single operator to take care of a specific car, by hiding the corresponding button to all operators once an operator pressed it, and informing everyone about which cars are already being taken care of. An operator who took care of a car, can inform the system that he has finished his job clicking on a dedicated button on the same list. Cars with no issues are listed as well.

1.3 Definitiona, acronyms

1.4 Reference Documents

1.5 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.

Architecture Design This section id divided into several parts:

1. Overview : in this section is explained the architecture of our system in terms of layers and tiers
2. High level components and their interaction : this section gives a global view of the components of the system and how they ?interact with each other
3. Component view : this section gives a more detailed view of the ?components of the system
4. Deploying view : this section shows how the components must be ?deployed to have the application running correctly.
5. Runtime view : sequence diagrams are represented in this section to ?show the flow of events in terms of interaction among the components
6. Component interfaces :in this section are ?presented the interfaces among the components .
7. Selected architectural styles and patterns : this section explain the ?architectural choices taken during the creation of the application and the proposed design patterns
8. Other design decisions

Algorithms Design: this section describes the most critical parts via some algorithms. Pseudo code is used in order to hide unnecessary implementation details in order to focus on the most important parts.

User Interface Design: this section presents user experience explained via UX and BCE diagrams.

Requirements Traceability: this section aims to explain how the decisions taken in the RASD are linked to design elements.

2 Architecture Design

2.1 Overview

2.2 Component View

2.2.1 High Level Component View

The figure below shows the main high level components of the application and the interaction among them. The nexus is where the core of the application is allocated, which contains all the business logics. It needs to communicate with several other components as follows.

- The User can establish a connection with the nexus via the mobile app and web browser logging in and requesting the information about the cars he can reserve. This is a synchronous communication, since the user needs to wait the response from the nexus, before going on asking for a car reservation, etc.
- The operators can establish an asynchronous communication with the nexus in order to take care of cars: they can request to the nexus the list of cars and their information logging in, and, once the nexus provided it to them, they can choose a car to take care of and click on the "Operate" button associate to that car, so that the nexus can update the list, and the same when the issue is resolved.
- The nexus always needs to maintain the information on the lists provided to the operators updated in real time: in order to do that it needs an asynchronous communication with the operators.
- The on board computer of the cars has an asynchronous communication with the nexus, in order to provide updated information about the status of all the cars.
- The on-board computer also needs to establish a synchronous communication with the nexus when users want to unlock the cars they reserved, start and end their ride
- The nexus also needs to store and access all the information about users, operators, rides, cars positions and status on the database

2.2.2	Deployment View
2.2.3	Runtime View
2.2.4	Component Interfaces
2.2.5	Selected Architectural Styles and Patterns
2.2.6	Other Design Decisions
3	Algorithm Design
4	User Interface Design
5	Requirement Traceability
6	Effort Spent
7	References