Politecnico di Milano  
Industrial and Information Engineering  
Computer Science and Engineering

Software Engineering II Assignment

PowerEnJoy - car sharing

Academic Year: 2016/2017

Prof. Elisabetta Di Nitto

Alice Segato matr. 875045

Mark Edward Ferrer matr. 876650

Davide Bonacina matr. 876199

PowerEnJoy



Car Sharing App

# TABLE OF CONTENT

[TABLE OF CONTENT 3](#_Toc471329780)

[1 INTRODUCTION 4](#_Toc471329781)

[1.1 Revision History 4](#_Toc471329782)

[1.2 Purpose and Scope 4](#_Toc471329783)

[1.3 List of Definitions and Abbreviations 4](#_Toc471329784)

[1.4 List of Reference Documents 5](#_Toc471329785)

[2 INTEGRATION STRATEGY 5](#_Toc471329786)

[2.1 Entry Criteria 5](#_Toc471329787)

[2.2 Elements to be Integrated 5](#_Toc471329788)

[2.3 Integration Testing Strategy 6](#_Toc471329789)

[2.4 Sequence of Component/Function Integration 6](#_Toc471329790)

[2.4.1 Software Integration Sequence 6](#_Toc471329791)

[2.4.2 Subsystem Integration Sequence 6](#_Toc471329792)

[3 INDIVIDUAL STEPS AND TEST DESCRIPTION 6](#_Toc471329793)

[4 TOOLS AND TEST EQUIPMENT REQUIRED 6](#_Toc471329794)

[5 PROGRAM STUBS AND TEST DATA REQUIRED 6](#_Toc471329795)

[6 EFFORT SPENT 6](#_Toc471329796)

# 1 INTRODUCTION

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Authors | Summary |
| 1.0 |  | Mark Edward Ferrer, Alice Segato, Davide Bonacina | Initial release |

## Purpose and Scope

This document describes the Integration Testing Plan for PowerEnjoy system.

This testing is necessary in order to avoid unexpected behavior of the system and guarantee its ability to fulfill all the requirements.

The ITPD outlines the testing activities organization for the subsystems that make up the system.

This document is composed by five parts:

* Integration Strategy: explains the selection of subsystems and their subcomponents for the testing and outlines, for each one, the project status that has to be met in order to start the testing.
* Individual Steps and Test Description: describes the integration testing approach, the sequence in which components and subsystems will be integrated and the planned testing activities for each integration step, including their input data and the expected output.
* Performance Analysis: performance measures on the components to check in order to verify the requirements fulfillment.
* Tools and Test Equipment Required: list of tools that will be employed during the testing activities and description of the environment for the test execution.
* Required Program Stubs and Test Data: list of program stubs and drivers to perform the necessary method invocations on the components to be tested.

## 1.3 List of Definitions and Abbreviations

* DD: Design Document.
* RASD: Requirements analysis and Specification Document.
* DB: the database layer, handled by a DBMS.
* UI: User Interface.
* GUI:graphical user interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators;
* Application server: the layer, which provides the application logic and interacts with the DB and with the front-ends.
* Back-end: term used to identify the Application server.
* Front-end: the components, which use the application server services, namely the web front-end and the mobile applications.
* Web server: the component that implements the web-based front-end. It interacts with the application server and with the users’ browsers.
* Acknowledge: is a [signal](https://en.wikipedia.org/wiki/Message) passed between communicating [processes](https://en.wikipedia.org/wiki/Process_%28computing%29) or [computers](https://en.wikipedia.org/wiki/Computer) to signify acknowledgement, or receipt of response, as part of a [communications protocols](https://en.wikipedia.org/wiki/Communications_protocol).

## 1.4 List of Reference Documents

* RASD;
* Assignment AA 2016-2017.pdf;
* …
* …

# 2 INTEGRATION STRATEGY

## 2.1 Entry Criteria

In order to start testing the integration of all the components previously described in the Design Document, there are some conditions that have to be fulfilled before the integration:

* The components must have been already designed in the Design Document in order to figure out their role in the system;
* The components must have been unit tested and they must be correct.

## 2.2 Elements to be Integrated

The components that we are going to test are all the internal components of the Main System, the Car Computer and the Client App (as described in the section 2.3.1 of the Design Document). The components of the Third Party System and the relative clients (both web and mobile applications) have to be already tested and fully working, according to our assumptions described in the section 1.6 of the Requirements Analysis and Specification Document.

[imagine dei component che devono essere testati]

Realistically speaking, handling a full system testing is very difficult, for this reason we clamp together the components that have strong dependencies into subsystems to be tested, and then we integrate the subsystem gradually until we obtain the complete system.

*Main System*

* The Application Controller depends on User Manager, Reservation Manager, Utility Manager subsystems;
* The API Controller depends on the Algorithm Controller;
* The API Controller depends also on the Application Controller.

*Car Computer*

* The Mobile application controller relies on the Sensor Manager.

*User App*

* The Mobile application controller depends on the GPS Manager;

These three components generate five subsystems and the components themselves must be integrated in this order:

1. Main System with Car Computer;
2. User App with Main System;
3. All these three components with the Third Party System and the Database.

## 2.3 Integration Testing Strategy

To approach the integration test phase we decided to adopt the bottom-up strategy to test first the lower level components until we obtain greater subsystems. At this point we follow the critical-module-first approach to integrate together the subsystems found in the previous step. We will need only one stub of the Main System component to be used during User App subsystem testing since some of the functionalities of the User App involve strongly some subcomponents of the Main System.

## 2.4 Sequence of Component/Function Integration

### 2.4.1 Software Integration Sequence

### 2.4.2 Subsystem Integration Sequence

# 3 INDIVIDUAL STEPS AND TEST DESCRIPTION

# 4 TOOLS AND TEST EQUIPMENT REQUIRED

# 5 PROGRAM STUBS AND TEST DATA REQUIRED

# 6 EFFORT SPENT