



Politecnico di Milano

# PowerEnjoy Service - Integration Test Plan Document

January 3, 2017

## **Version 1.1**

Authors:

- Domenico FAVARO (Mat. 837995)
- Matheus FIM (Mat. 876069)
- Caio ZULIANI (Mat. 877266)

Prof. Elisabetta DI NITTO

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Revision History . . . . .	2
1.2	Purpose and Scope . . . . .	2
1.3	Definitions and Abbreviations . . . . .	2
1.4	Reference Documents . . . . .	3
<b>2</b>	<b>Integration Strategy</b>	<b>4</b>
2.1	Entry Criteria . . . . .	4
2.2	Elements to be Integrated . . . . .	4
2.3	Integration Testing Strategy . . . . .	5
2.4	Sequence of Component/Function Integration . . . . .	5
2.4.1	Software Integration Sequence . . . . .	5
2.4.2	Subsystem Integration Sequence . . . . .	5
<b>3</b>	<b>Individual Steps and Test Description</b>	<b>6</b>
<b>4</b>	<b>Performance Analysis</b>	<b>6</b>
<b>5</b>	<b>Required Tools and Test Equipment</b>	<b>6</b>
5.1	Tools . . . . .	6
5.2	Test Equipment . . . . .	6
<b>6</b>	<b>Required Program Stubs and Test Data</b>	<b>6</b>
6.1	Program Stubs . . . . .	6
6.2	Test Data . . . . .	6
<b>7</b>	<b>Effort Spent</b>	<b>7</b>
<b>8</b>	<b>Changelog</b>	<b>8</b>

# 1 Introduction

## 1.1 Revision History

This section records all revisions to the Document.

Version	Date	Authors	Summary
1.1	15/01/16	Domenico Favaro, Caio Zuliani, Matheus Fim	Initial Release

## 1.2 Purpose and Scope

The Integration Test Plan Document (ITPD) serves to present the integration sequence and testing for all Subsystems and Components that conform PowerEnjoy Car Sharing Service. This is a key part to guarantee the functioning and quality of the software. The Document will present the division of the System in Subsystems and Components that will endure individual testing as independent modules and then be subject to integration on the whole System.

## 1.3 Definitions and Abbreviations

- **RASD:** Requirements And Specifications Document.
- **DD:** Design Document.
- **ITPD:** Integration Test Plan Document.
- **SDK:** Software Development Kit
- **App:** Application, referring to Web or Mobile App.
- **Subsystem:** Part of the system the generally encapsulates one or more features.
- **Component:** Self sustained part of the System that provides with functionalities and is part of one or more subsystems.
- **Bottom-up:** Referring to Bottom-up testing. Each component at lower hierarchy is tested individually and then the components that rely upon these components are tested.
- **Top-down:** Top-down integration testing is an integration testing technique used in order to mock or simulate the behaviour of the lower-level modules that are not yet integrated.
- **Mock:**Simulation that mimic the behavior of certain objects and fuctions in controlled ways, done to test the behavior of some other object.

For other concepts concerning the Service definition look in the **Glossary** section of the RASD and DD.

## 1.4 Reference Documents

- Specification Document: Assignments AA 2016-2017.pdf
- PowerEnjoy Requirements And Specifications Document (RASD)
- PowerEnjoy Design Document (DD)
- Example Document - Integration testing example document.pdf
- Testing Tools Documents:
  - Mockito
  - JMeter

## 2 Integration Strategy

### 2.1 Entry Criteria

We define the criteria that must be met before integration testing of the system components. We consider Integration a part of the production development. In order for production to start all documentation must first be written and up to date, including RASD and DD, to have a clear and full scope of the system components functionalities and importance. Once in production, the integration of a single component can be done when the following criteria is met:

- The Component feature must be 100% complete, that is all classes and functions must have been implemented.
- No tickets must be opened for the Component, no bugs or considered missing features must be present.
- Individual component testing must have been performed, using JUnit to test its classes and functions.
- All the interfaces the Component has to communicate to have to be present or at least mocked to be able to test its coupling.

### 2.2 Elements to be Integrated

As stated in the Design Document, our system is composed by several High level Components presented in 3 tiers. Specifically these components are:

- Client Tier:
  - User Client Component
  - CRM Client Component
  - Car Component
- Server Tier:
  - User Controller
  - CRM Controller
  - Car Controller
  - Reservation Controller
  - Ride Controller
  - Payment Controller
  - User Report Controller

- Email Helper
- Location Helper
- Chat Service
- EIS Tier:
  - Database

## 2.3 Integration Testing Strategy

Our approach following the 3-tiered structure will follow a **Bottom-up** strategy, working on components that do not depend on others to function first.

This implies following the next tier order: EIS -> Server -> Client for development and testing.

Inside each Tier Bottom-up strategy will be used again to integrate independent modules first and then those that depend on others. This strategy will help in contrast to Top-down to minimize the mock-up testing to be done, testing on top of already deployed modules. The order in which the 'Bottom' modules will be picked will follow a Critical-Module-First Integration Strategy, giving priority to those that will have dependencies of other modules, this will help not only to spot any error on critical modules first but also to unblock the integration of dependant modules earlier on.

## 2.4 Sequence of Component/Function Integration

### 2.4.1 Software Integration Sequence

For each subsystem, identify the sequence in which the software components will be integrated within the subsystem; relate this sequence to any product features that are being build up.

### 2.4.2 Subsystem Integration Sequence

Identify the order in which subsystems will be integrated.

- 3 Individual Steps and Test Description
- 4 Performance Analysis
- 5 Required Tools and Test Equipment
  - 5.1 Tools
  - 5.2 Test Equipment
- 6 Required Program Stubs and Test Data
  - 6.1 Program Stubs
  - 6.2 Test Data

## 7 Effort Spent

Date	Domenico	Caio	Matheus
27/12/16	2h	2h	2h
28/12/16	-	-	-
29/12/16	1h	-	-
30/12/16	2h	-	-
31/12/16	-	-	-
01/01/17	-	-	-
02/01/17	2h	-	-
03/01/17	-	-	-
04/01/17	-	-	-
05/01/17	-	-	-
06/01/17	-	-	-
07/01/17	-	-	-
08/01/17	-	-	-
09/01/17	-	-	-
10/01/17	-	-	-
11/01/17	-	-	-
12/01/17	-	-	-
13/01/17	-	-	-
14/01/17	-	-	-



## 8 Changelog

As the project and design decisions may change during the development this document is also prone to change. We'll document every version in this part.

- **Version 1.1:** 15/01/2017