



**POLITECNICO**  
**MILANO 1863**

15 January 2017

# **PowerEnJoy**

## Integration Test Plan Document

Blanco	Federica	875487
Casasopra	Fabiola	864412

*Software Engineering 2 Project*  
2016/2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Revision History . . . . .	1
1.2	Purpose and Scope . . . . .	1
1.3	List of Definitions and Abbreviations . . . . .	1
1.4	List of Reference Documents . . . . .	1
1.5	Document overview . . . . .	2
<b>2</b>	<b>Integration Strategy</b>	<b>3</b>
2.1	Entry Criteria . . . . .	3
2.2	Elements to be Integrated . . . . .	3
2.3	Integration Testing Strategy . . . . .	4
2.4	Sequence of Component/Function Integration . . . . .	4
<b>3</b>	<b>Individual Steps and Test Description</b>	<b>9</b>
<b>4</b>	<b>Tools and Test Equipment Required</b>	<b>10</b>
<b>5</b>	<b>Program Stubs and Test Data Required</b>	<b>11</b>
<b>6</b>	<b>Appendix</b>	<b>12</b>
6.1	Used Tools . . . . .	12
6.2	Working Hours . . . . .	12

## List of Figures

1	Managed Beans and corresponding Managers connections . . .	4
2	Business-tier subcomponents connections . . . . .	5
3	Business-tier subcomponents and Java Persistence connections	6
4	Business-tier subcomponents connections . . . . .	7
5	Managed Beans and corresponding Managers connections . . .	8

## List of Tables

1	Managed Beans and corresponding Managers connections . . .	4
2	Business-tier subcomponents connections . . . . .	5
3	Business-tier subcomponents and Java Persistence connections	6
4	Business-tier subcomponents connections . . . . .	7
5	Managed Beans and corresponding Managers connections . . .	8

# 1 Introduction

The Integration Test Plan Document aims at describing the planning in order to accomplish the integration test for our application PowerEnJoy. This document is useful for the development team, which is responsible for the creation of the integration test scripts in accordance to what is described in the next sections. Moreover, a developer will be chosen and he will be responsible for execution of the test scripts and certifying that the integration testing is complete. Furthermore, integration testing includes interactions between all layers of an application, including interfaces to other applications, as a complete end-to-end test of the functionality.

## 1.1 Revision History

Version 1.0, on 15 January 2017.

## 1.2 Purpose and Scope

The aim of the project PowerEnJoy is to provide a car-sharing service that involves *only* electric cars. In this documents, what to test, in which sequence, which tools are needed for testing and which stubs, drivers or oracles need to be developed is explained. If you wish to have more details about the scope of our project, you may refer to the *Section 1* of the Requirements Analysis and Specifications Document.

## 1.3 List of Definitions and Abbreviations

Here there is the acronims and abbreviations list:

**DD** Design Document

**ITPD** Integration Test Plan Document

**IT** Integration Test

**RASD** Requirements Analysis and Specifications Document

**TP** Test Procedure

## 1.4 List of Reference Documents

- Specification document: Assignments AA 2016-2017.pdf
- IEEE Std 1016tm-2009 Standard for Information Technology - System Design - Software Design Descriptions.

- Requirements Analysis and Specifications Document: RASD.pdf  
(<https://github.com/fabiola-casasopra/sw-eng-2-project/tree/master/RASD/RASD.pdf>)
- Design Document: DD.pdf  
(<https://github.com/fabiola-casasopra/sw-eng-2-project/blob/master/DD/DD.pdf>)

## 1.5 Document overview

Here we show the structure of the document, with a brief overview of each section.

- Section 1** There is an introduction with this document's purpose and other general information about it.
- Section 2** There is the definition of all the items to be tested and the explanation of the integration testing approach.
- Section 3** Here, for each step of the integration process above, there is a description of the type of tests that will be used to verify that the elements integrated in this step perform as expected. Moreover, there is a general description of the expected results of the test set.
- Section 4** Here, we are going to identify all tools and test equipment needed to accomplish the integration and there will be an explanation on why and how we are going to use the specific tool.
- Section 5** Here, we are going to identify any program stubs or special test data required for each integration step, referring to the testing strategy and test design described in the previous section.
- Section 6** Here there are given additional information that may be useful to the reader, such as the tools used and the time spent to redact this document.

## 2 Integration Strategy

### 2.1 Entry Criteria

In this part of the document, we are going to specify the criteria that must be met before integration testing of specific elements may begin:

- The Requirements Analysis and Specifications Document and the Design Document must be already completed, in order to know the interaction of the various components and their expected behaviour;
- Each component of our software must have successfully passed the Unit Testing;
- So, the correct version of our application is moved into the integration testing environment;
- All the code of our project must be already written and so the major functionality must be present;
- Our project should satisfy the memory requirements specified in the RASD;
- The database should be ready and its tables should already be populated with the initial data.

### 2.2 Elements to be Integrated

As we have shown in the Design Document related to our project PowerEnJoy, the system relies on many high-level components, each one implementing a specific set of functionalities, that interacts between them. Since we have decided to follow a modular approach, each component is the result of the combination of various subcomponents. However, since we haven't fully defined all low level components needed for our system, we think it is better to focus our integration testing only on the Business Logic and its components (for further information, see *Section 2* of the Design Document). By doing this choice, we have to consider that, in the following evolution of our project, the needed subcomponents will be created and further Integration Test must be carried out.

So, for what we said above, the elements to be integrated are the following:

- Web Component and Business Logic Component, testing the direct connections between Managed Beans and their corresponding Managers;
- the subcomponents of the Business Logic Component, integrating them, each one with the needed others.

## 2.3 Integration Testing Strategy

As we explained above, in this stage of the development we haven't fully defined the hierarchy of all subcomponents and subsystem. For this reason, we will have an Integration Test strategy for a single abstract layer and we have to keep in mind that other lower level subcomponents will be implemented. Although it is not possible to define the final integration test strategy, we think that, as far as we know at this stage, the better strategy we can apply is the top-down approach. Moreover, the chose of this strategy allow us to test the new subcomponent folowing the downward development.

## 2.4 Sequence of Component/Function Integration

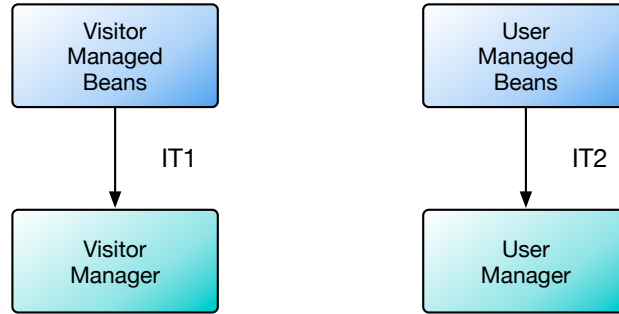


Figure 1: Managed Beans and corresponding Managers connections

ID	Components	IT	TP
IT1	Visitor Managed Beans → Visitor Manager	ref	ref
IT2	User Managed Beans → User Manager	ref	ref

Table 1: Managed Beans and corresponding Managers connections



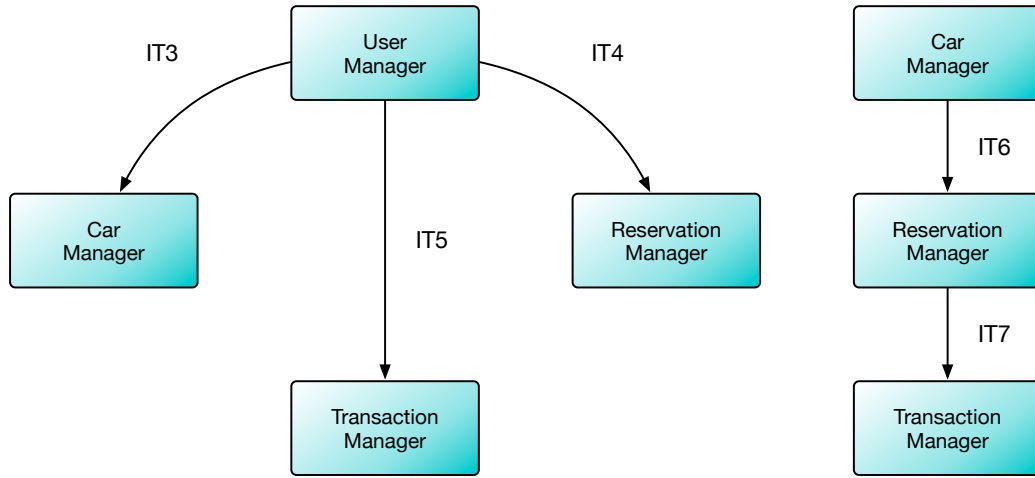


Figure 2: Business-tier subcomponents connections

ID	Components	IT	TP
IT3	User Manager → Car Manager	ref	ref
IT4	User Manager → Reservation Manager	ref	ref
IT5	User Manager → Transaction Manager	ref	ref
IT6	Car Manager → Reservation Manager	ref	ref
IT7	Reservation Manager → Transaction Manager	ref	ref

Table 2: Business-tier subcomponents connections

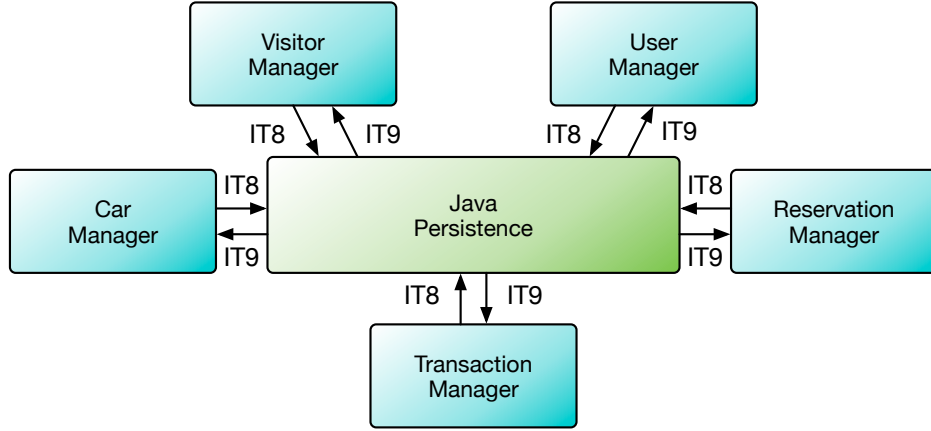


Figure 3: Business-tier subcomponents and Java Persistence connections

ID	Components	IT	TP
IT8	User Manager → Java Persistence	ref	ref
IT8	Visitor Manager → Java Persistence	ref	ref
IT8	Car Manager → Java Persistence	ref	ref
IT8	Reservation Manager → Java Persistence	ref	ref
IT8	Transaction Manager → Java Persistence	ref	ref
IT9	Java Persistence → User Manager	ref	ref
IT9	Java Persistence → Visitor Manager	ref	ref
IT9	Java Persistence → Car Manager	ref	ref
IT9	Java Persistence → Reservation Manager	ref	ref
IT9	Java Persistence → Transaction Manager	ref	ref

Table 3: Business-tier subcomponents and Java Persistence connections

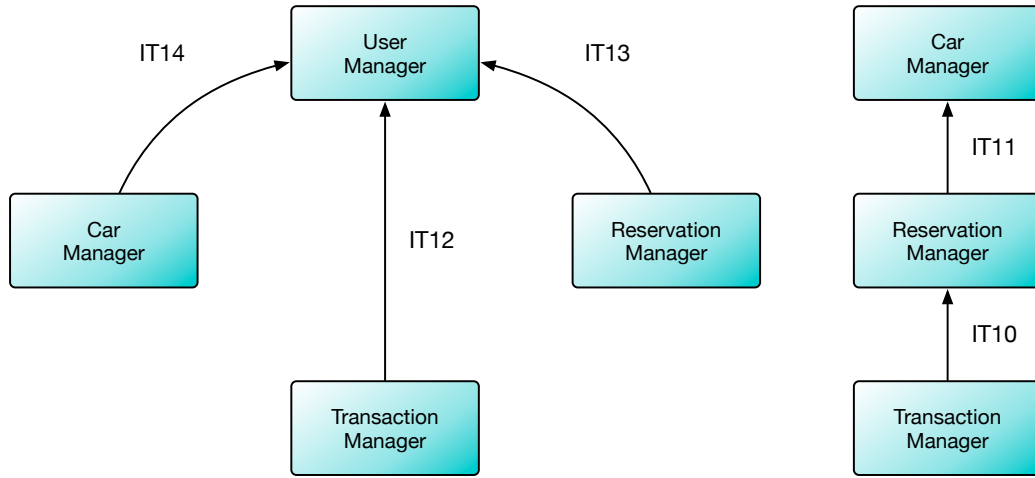


Figure 4: Business-tier subcomponents connections

ID	Components	IT	TP
IT10	Transaction Manager → Reservation Manager	ref	ref
IT11	Reservation Manager → Car Manager	ref	ref
IT12	Transaction Manager → User Manager	ref	ref
IT13	Reservation Manager → User Manager	ref	ref
IT14	Car Manager → User Manager	ref	ref

Table 4: Business-tier subcomponents connections

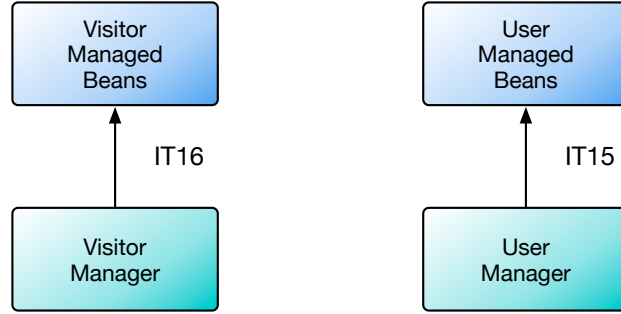


Figure 5: Managed Beans and corresponding Managers connections

ID	Components	IT	TP
IT15	User Manager→ User Managed Beans	ref	ref
IT16	Visitor Manager → Visitor Managed Beans	ref	ref

Table 5: Managed Beans and corresponding Managers connections

### 3 Individual Steps and Test Description

## 4 Tools and Test Equipment Required

## 5 Program Stubs and Test Data Required

## 6 Appendix

In this section, we will give the information about the used tools, the hours of work done by the members of the group.

### 6.1 Used Tools

In this phase of the project, the following tools have been used:

- $\text{\LaTeX}$  and TeXMaker editor: to redact and to format this document

### 6.2 Working Hours

Last Name	First Name	Total Hours
Blanco	Federica	? h
Casasopra	Fabiola	? h