



POLITECNICO DI MILANO
SOFTWARE ENGINEERING II PROJECT:
POWERENJOY

Design Document

Gregori Giacomo and Ruaro Nicola

January 5, 2017
Version 0.1

Contents

Contents	I
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
2 Integration strategy	2
2.1 Entry criteria	2
2.2 Elements to be integrated	2
2.3 Integration testing strategy	2
2.4 Sequence of Component/Function Integration	2
2.4.1 Software Integration Sequence	2
2.4.2 Subsystem Integration Sequence	2
3 Individual steps and test description	4
3.1 Sample Integration test case I1	4
3.2 Sample Integration test procedure TP1	5
4 Tools and test equipment required	6
5 Program stubs and test data required	7
A Appendix A: Used Tools	I
A.1 L ^A T _E X	I
A.2 git	I
A.3 draw.io	I
B Appendix B: Hours of work	II
C Appendix C: Revisions	III
Bibliography	IV

Abstract

This document provides a detailed description of the Integration Test's planning for the PowerEnJoy system. It is based on the RASD and DD documents presented in the previous deliveries and must explain to the development team how to test the system.

Introduction

1.1 Purpose

The purpose of this document is to give a guideline for the development team in order to effectively test the component's integration. The tests are described individually and the required equipment and test-data are listed in the following sections.

1.2 Scope

PowerEnJoy is a car-sharing service based on mobile and web applications which should allow users to reserve vehicles and use them. The application logic must be designed and allocated into components that should improve software maintainability and ease future extensions.

Integration strategy

2.1 Entry criteria

Specify the criteria that must be met before integration testing of specific elements may begin (e.g., functions must have been unit tested).

2.2 Elements to be integrated

Identify the components to be integrated, refer to your design document to identify such components in a way that is consistent with your design.

2.3 Integration testing strategy

Describe the integration testing approach (top down, bottom up, functional groupings, etc.) and the rationale for the choosing that approach.

2.4 Sequence of Component/Function Integration

NOTE: The structure of this section may vary depending on the integration strategy you select in Section 2.3. Use the structure proposed below as a non mandatory guide.

2.4.1 Software Integration Sequence

Identify the sequence in which the software components will be integrated within the subsystem. Relate this sequence to any product features/functions that are being built up.

2.4.2 Subsystem Integration Sequence

Identify the order in which subsystems will be integrated.

If you have a single subsystem, 2.4.1 and 2.4.2 are to be merged in a single section. You can refer to Section 2.2 of the test plan example [1] as an example of what we expect.

Individual steps and test description

For each step of the integration process identified above, describe the type of tests that will be used to verify that the elements integrated in this step perform as expected. Describe in general the expected results of the test set. You may refer to Chapter 3 and Chapter 4 of the test plan example [1] as an example of what we expect. (NOTE: This is not a detailed description of test protocols. Think of this as the test design phase. Specific protocols will be written to fulfill the goals of the tests identified in this section.)

3.1 Sample Integration test case I1

Test Case Identifier	I1T1
Test Item(s)	Client Communicator → Client Translator
Input Specification	Create typical Client Communicator input
Output Specification	Check if the correct functions are called in the Client Translator
Environmental Needs	Client driver

3.2 Sample Integration test procedure TP1

Test Procedure Identifier	TP1
Purpose	<p>This test procedure verifies whether the dispatcher software:</p> <ul style="list-style-type: none">• can handle command-line input• can handle client input• can handle agent input• can output requested information to a client• can output requested information to an agent
Procedure Steps	Execute I5-I6 after I1-I4

Tools and test equipment required

Identify all tools and test equipment needed to accomplish the integration. Refer to the tools presented during the lectures. Explain why and how you are going to use them. Note that you may also use manual testing for some part. Consider manual testing as one of the possible tools you have available.

Program stubs and test data required

Based on the testing strategy and test design, identify any program stubs or special test data required for each integration step.

Appendix A: Used Tools

A.1 \LaTeX

Used to format and redact this document

A.2 *git*

Used as version control system in order to lead development

A.3 *draw.io*

Used to draw mockups and diagrams

Appendix B: Hours of work

These are the hours of work spent by each group member in order to redact this document:

- Ruaro Nicola: 0 hours
- Gregori Giacomo: 0 hours
- Total worktime: 0 hours

Appendix C: Revisions

These sections will be eventually redacted during future post-release updates in order to approach the ITPD modifiability providing a comfortable and highly effective way to trace changes:

Bibliography

- [1] Luca Mottola and Elisabetta Di Nitto, *Software Engineering 2: Project goal, schedule and rules*, 2016
- [2] Nicola Ruaro and Giacomo Gregori, *RASD: Requirements Analysis and Specification Document*, 2016
- [3] Nicola Ruaro and Giacomo Gregori, *DD: Design Document*, 2016