POLITECNICO
MILANO 1863

Politecnico di Milano

COMPUTER SCIENCE AND ENGINEERING

Software Engineering 2

Integration Test Plan Document

PowerEnJoy

Authors:
Francesco Fabiani
Jagadesh Manivannan
Niccolò Pozzolini

Professors:
Elisabetta Di Nitto
Luca Mottola

Academic Year 2016/2017

Contents

Chapter 1

Introduction

1.1 Revision history

Version	Date	Authors	Summary
1.0	15/01/2017	Fabiani, Manivannan, Pozzolini	Initial release

Table 1.1: Changelog of this document

1.2 Purpose and scope

The Integration Test Plan Document (ITPD) is intended to indicate the necessary tests in order to verify that all the components of the previously described system are properly integrated. This process is critically important to ensure that the unit-tested modules interact correctly.

The product described is PowerEnJoy, a car-sharing service which offers to its users exclusively electric cars. It includes the common functionalities of its category: permitting to registered users to obtain the position of all the available cars, reserving one within a certain amount of time and continuously displaying the up-to-the-minute cost of the ride are just few of them. Moreover, PowerEnJoy stimulates users to behave virtuously towards the ecosystem by applying various types of discounts under specific conditions.

1.3 Definitions, acronyms, abbreviations

- *API*: Application Programming Interface
- *BCE*: Business Controller Entity

- *Car*: electric vehicle provided by the service
- *DB*: Database
- *DBMS*: Database Management System
- *DD*: Design Document
- *ER*: Entity-Relationship
- *GPS*: Global Positioning System
- *Guest* or *Guest user*: person not registered to the service
- *ITPD*: Integration Test Plan Document
- *MVC*: Model View Controller
- *OS*: Operating System, related both to desktop and mobile platforms
- *PIN*: Personal Identification Number
- *RASD*: Requirements Analysis and Specification Document
- *Registered user*: see *User*
- *REST*: REpresentational State Transfer
- *RESTful*: that follows the REST principles
- *Safe area*: set of parking spots where a user can leave a car without penalization
- *User*: person with a valid driving license registered to the service
- *UX*: User eXperience
- *W3C*: World Wide Web Consortium

1.4 Reference documents

The Integration Test Plan Document has been composed following the indications reported in the previous documents delivered for this project: the Requirements Analysis and Specification Document, describing fundamental aspects of PowerEnJoy such as domain assumptions, goals, functional and non-functional requirements, and the Design Document, which shows more accurately all the functionalities provided by focusing on the software design of the system.

With regards to the course named Software Engineering 2 and held by professors Luca Mottola and Elisabetta Di Nitto (Politecnico di Milano, a. y. 2016/17), the document conforms to the guidelines provided during the lectures and within the material of the course.

Chapter 2

Integration Strategy

2.1 Entry Criteria

This section shows the conditions that must be met before starting the integration in order to obtain significant results.

First, it is crucial that the RASD and DD documents are completely composed, so that a whole vision of the components of the system and their functionalities is available.

As regards the individual components, the development must go forward along with their unit testing, so that the new modules implemented do not interfere with the solidity if the system. For this reason, every component should have at least 90% of its functionalities completed before the integration with other components is tested.

Moreover, the integration process should start when the following percentages of development are achieved:

-
-
-

The decision of requiring different amounts of functionalities according to the component is based on the order the integration will take place and on the time needed to accomplish the integration testing phase of each one.

2.2 Elements to be Integrated

2.3 Integration Testing Strategy

The items to be tested consist of the integration of the code modules developed, for the Power Enjoy project. For testing we choose the bottom-up approach. This means that integration testing starts at the bottom level. We chose this because the top level component when built has to be tested from the bottom level components in your project, i.e. we can simply say that it has few dependencies on the bottom level components for testing.

Even though, it can be developed by creating stubs, we do not prefer that. We want to test using the real values and functionalities. The integration tests described in this documents are at the component level. The integration tests of lower level code modules are described in the corresponding components unit test. Unit tests are described in the [UTP].

2.4 Sequence of Component/Function Integration