**Politecnico di Milano**

**Software Engineering 2**

**Integration test plan**

**PowerEnjoy**

**Authors:**

**Simone Bruzzechesse**

**Luca Franceschetti**

**Gian Giacomo Gatti**

# Index

[Index 2](#_Toc470798320)

[1 Introduction 3](#_Toc470798321)

[1.1 Revision History 3](#_Toc470798322)

[1.2 Purpose 3](#_Toc470798323)

[1.3 Scope 3](#_Toc470798324)

[1.4 Definitions, Acronyms, Abbreviations 3](#_Toc470798325)

[1.5 Reference Documents 4](#_Toc470798326)

[2 Integration Strategy 5](#_Toc470798327)

[2.1 Entry Criteria 5](#_Toc470798328)

[2.2 Elements to be integrated 5](#_Toc470798329)

[2.3 Integration Testing Strategy 6](#_Toc470798330)

[2.4 Sequence of Component/Function Integration 7](#_Toc470798331)

[2.4.1 Software Integration Sequence 7](#_Toc470798332)

[2.4.2 Subsystem Integration Sequence 7](#_Toc470798333)

[3 Individual Steps and Test Description 8](#_Toc470798334)

[4 Performance analysis 9](#_Toc470798335)

[5 Tools and Test Equipment Required 10](#_Toc470798336)

[6 Required Program Stubs and Drivers 11](#_Toc470798337)

[7 Appendix 12](#_Toc470798338)

[7.1 Used tools 12](#_Toc470798339)

[7.2 Effort spent 12](#_Toc470798340)

# Introduction

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Description |
| 1.0 | 29-12-2016 | Simone Bruzzechesse,  Luca Franceschetti,  Gian Giacomo Gatti | Document created |
| 1.1 |  | Simone Bruzzechesse,  Luca Franceschetti,  Gian Giacomo Gatti | Document completed |

## Purpose

This document represents the Integration Test Plan Document (ITPD) for Power Enjoy project, which describes the plans for testing the integration of Power Enjoy project’s components. The purpose of this document is to highlight the main aspects regarding the organization of the integration testing activity for all components of our system.

## Scope

The Integration Test Plan Document describes the plan for the integration testing, which takes as input software components (described in DD) that have been unit tested, groups them in larger aggregates, tests their interfaces, and delivers as its output the integrated system ready for system testing.

## Definitions, Acronyms, Abbreviations

* RASD: Requirements Analysis and Specification Document
* DD: Design document
* DBMS: Database Management System
* API: Application Programming Interface
* UI: User interface
* GPS: Global Positioning System
* ETA: Estimated Time of Arrival
* OS: Operating System

## Reference Documents

* Our RASD document
* Our DD document
* Specification Document: Assignments AA 2016-2017.pdf
* Sample integration test plan document

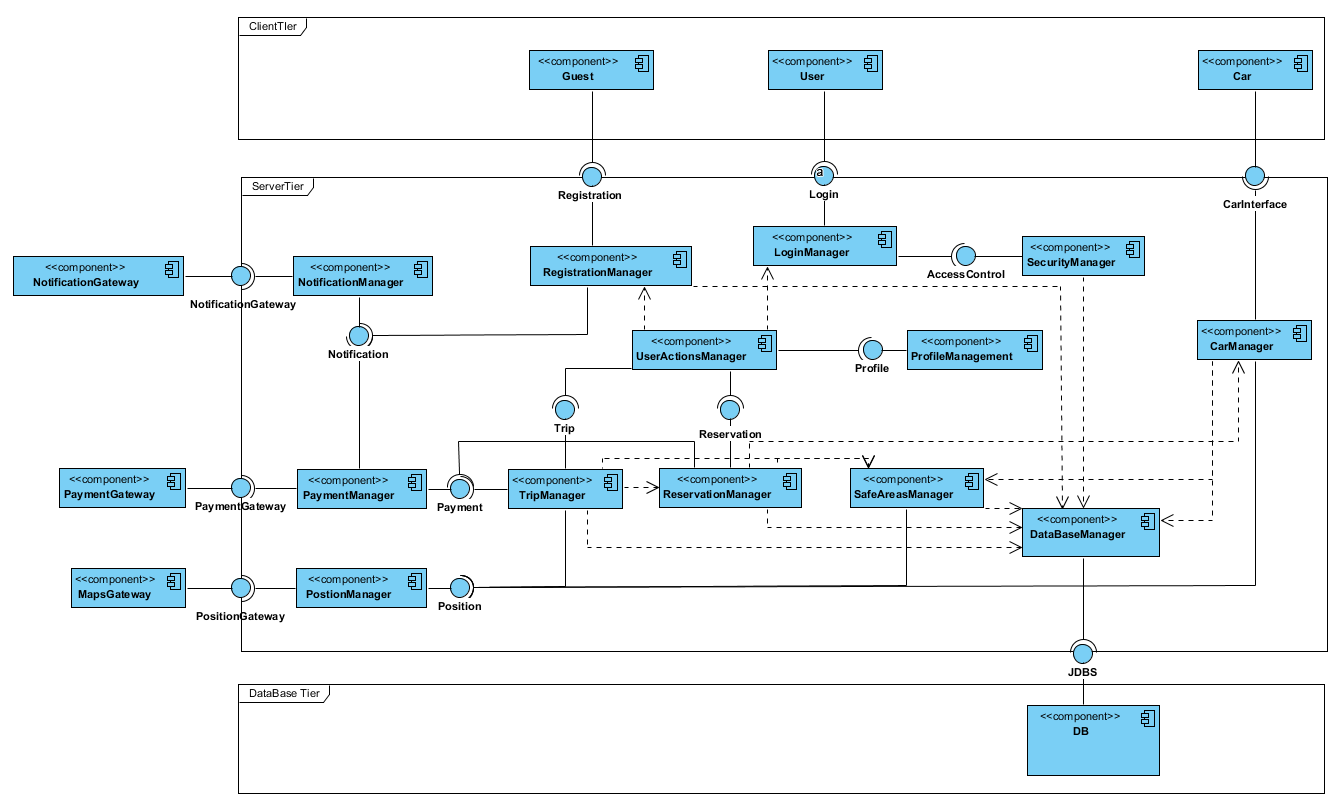
# Integration Strategy

## Entry Criteria

Before the integration test can begin, the RASD document and the DD document must be completed and successfully delivered. Then, all software components must have been unit tested: this is important because in case of failure we know the problem is in the implementation of interfaces and not in how modules have been developed.

## Elements to be integrated

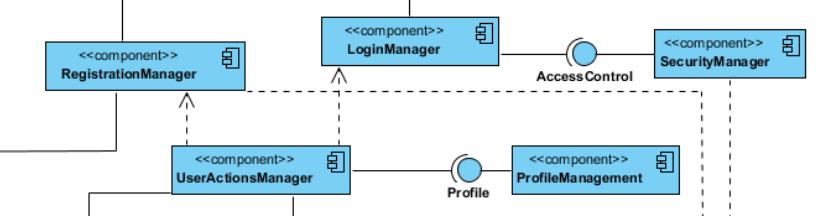
In this paragraph, we are going to list all components that must be integrated. We report our component diagram (taken from Design Document) for a clearer comprehension of interfaces and main components.

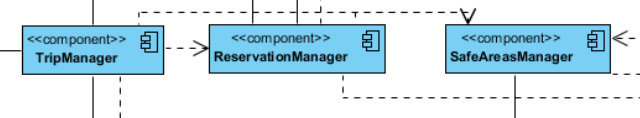


## Integration Testing Strategy

As integration testing strategy, we decided to use a mixture of bottom-up and functional grouping integration strategies. We chose bottom-up approach since we already know the architecture of the software and all components have been implemented and unit-tested, so we group components which do not rely on other components. Then, we decided to adopt the functional grouping where we group components with similar functionalities, so we try to avoid malfunctioning while managing to integrate larger number of components. After grouping these components, we integrate them with other components which are interfaces with external system. Observe that there is no need to test DMBS modules since they are commercial components and they have already been tested from their software house, as well as other external system such as payment system, notification system and maps system.

We identify two main groups:

* User basic functionalities, which includes:
  + RegistrationManager
  + LoginManager
  + SecurityManager
  + UserActionManager
  + ProfileManagement
* Trip management functionalities, which includes:
  + TripManager
  + ReservationManager
  + SafeAreasManager



## Sequence of Component/Function Integration

### Software Integration Sequence

The IDs represents the order in which the integration testing should proceed.

* User basic functionalities



|  |  |  |
| --- | --- | --- |
| ID | Integration Test | Paragraph |
| I1 | RegistrationManager 🡪 UserActionManager |  |
| I2 | LoginManager 🡪 SecurityManager |  |
| I3 | LoginManager 🡪 UserActionManager |  |
| I4 | UserAction 🡪 ProfileManagement |  |

* Trip management functionalities



|  |  |  |
| --- | --- | --- |
| ID | Integration Test | Paragraph |
| I5 | TripManager 🡪 ReservationManager |  |
| I6 | TripManager 🡪 SafeAreasManager |  |

Then we integrate these two main groups with other components which work as interfaces.

So, first User basic functionalities: we call the result User subsystem.



|  |  |  |
| --- | --- | --- |
| ID | Integration Test | Paragraph |
| I7 | UserBasicFunctionalities 🡪 ReservationManager |  |
| I8 | UserBasicFunctionalities 🡪 SafeAreasManager |  |

Then trip management functionalities: we call the result Trip subsystem.



|  |  |  |
| --- | --- | --- |
| ID | Integration Test | Paragraph |
| I9 | TripManagementFunctionalities🡪 PositionManager |  |
| I10 | TripManagementFunctionalities🡪 PaymentManager |  |
| I11 | TripManagementFunctionalities🡪 CarManager |  |

### Subsystem Integration Sequence

Now we first integrate both User subsystem and Trip subsystem with DBMS, then we integrate them together.

# Individual Steps and Test Description

# Performance analysis

# Tools and Test Equipment Required

# Required Program Stubs and Drivers

# Appendix

## Used tools

## Effort spent