

***School of Industrial and Information Engineering
Department of Computer Science and Engineering***



***Software Engineering II PowerEnjoy -
Integration Test Plan Document***

Presented by:

Bachar Senno

Gianluigi Iobizzi

Onell Saleeby

Under the supervision of:

Prof. Luca Mottola

Prof. Elisabetta Di Nitto

V1.0, 15-01-2017

Table of Contents

Table of Contents	2
Introduction	3
Purpose:	3
Scope:	3
Definitions, acronyms, abbreviations:	3
Reference documents:	3
Test Plan	4
Integration strategy	4
Elements to be integrated	4
Entry criteria	5
Test Case Specifications	6
Integration Test Case T1	6
Integration Test Case T2	6
Integration Test Case T3	6
Integration Test Case T4	6
Integration Test Case T5	7
Integration Test Case T6	7
Integration Test Case T7	7
Integration Test Case T8	7
Integration Test Case T9	8
Integration Test Case T10	8
Integration Test Case T11	8
Test Procedures	9
Integration Test Procedure TP1	9
Integration Test Procedure TP2	9
Integration Test Procedure TP3	9
Integration Test Procedure TP4	9
Revision History	10
Effort Spent	10

Introduction

Purpose:

This document describes the plans for testing the interfaces between the components of the “PowerEnjoy” system.

This document shows the hierarchy of the components, the integration strategy of these components, as well as a description of testing steps.

Scope:

In this document, we will be discussing our proposed approach for testing the different interfaces of the components which make up our system. As a first step, we will give a broad description of the testing method we’ll be using along with an explanation as to the reasons behind our choice, and then we will be going more into the details of each test that is to be performed. Finally, we will wrap it up by integrating the different tests we designed among each other in order to have more complete and accurate results.

Definitions, acronyms, abbreviations:

- RASD: Requirements Analysis and Specifications Document
- DD: Design Document
- API: Application Programming Interface; it is a common way to communicate with another system
- DBMS: Data-Base Management System

Other system-specific definitions can be found in the previously released RASD and DD.

Reference documents:

- RASD & Design documents
- Verification and Validation – part I & II.pdf
- Sample Integration Test Plan Document.pdf

Test Plan

Integration strategy

In order to perform the necessary testing of our system, we decided to follow a bottom-up approach when designing the different tests to perform.

Since we have a limited number of low-level components, writing drivers for these components won't be an issue.

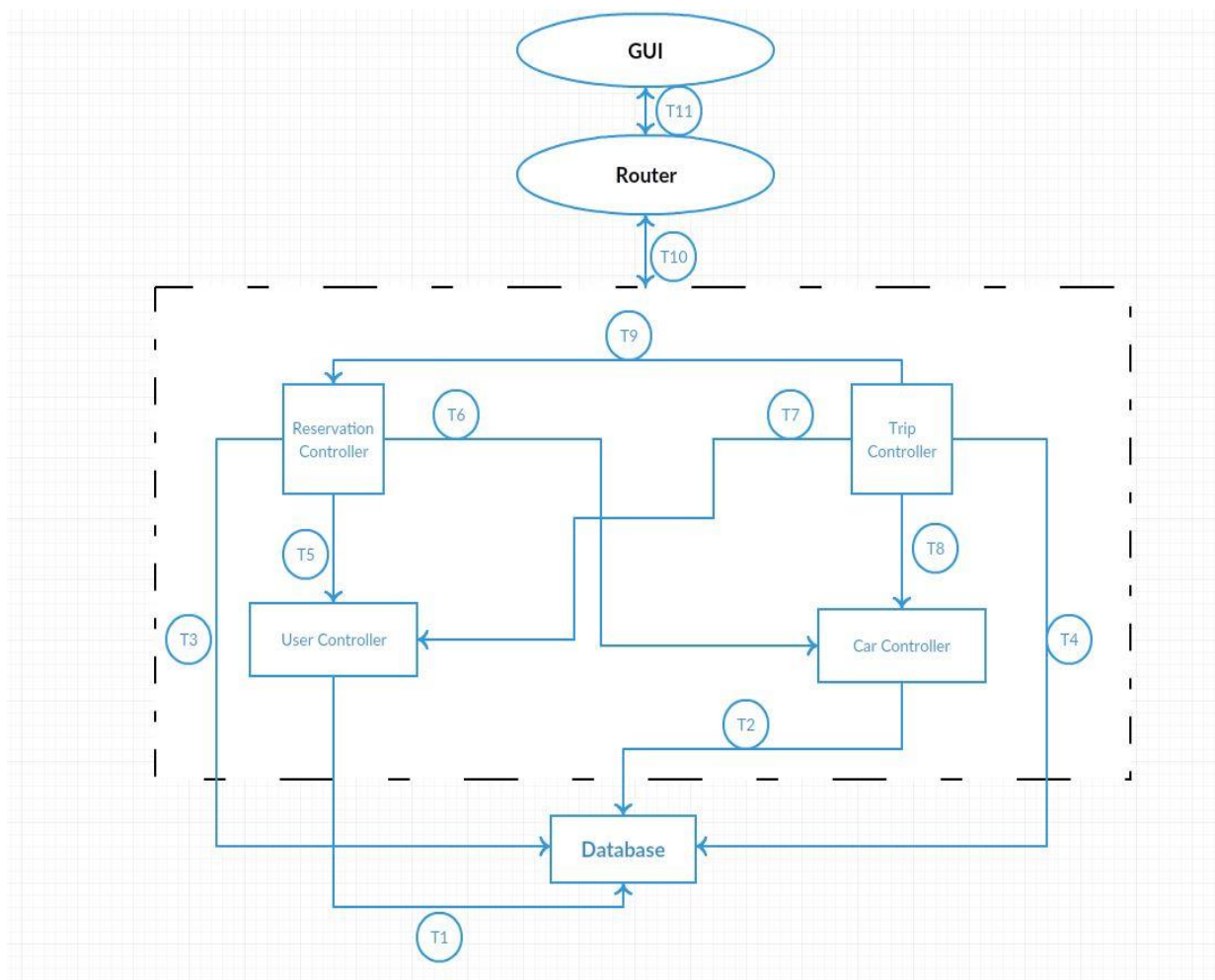
Furthermore, adopting the bottom-up strategy will allow us to granularly identify issues in any components as soon as they appear and promptly fix them, guaranteeing solid performance from our system.

The design of the integration test procedures is done in order to obtain a good confidence about the functioning of the system's core features related to the interaction between different components, that are: correct data generation, upgrading and storage, correct management of car's registration requests and trip processes and correct integration of the user's mobile app with the system's main server.

Elements to be integrated

The following list and schema show in detail the software components to be integrated and tested together (this is mainly based on the design proposed in the DD) and identify which of them are involved in a single integration step. The components following the arrows are integrated with the previous ones.

- T1** User Controller -> Database
- T2** Car Controller -> Database
- T3** Reservation Controller -> Database
- T4** Trip Controller -> Database
- T5** Reservation Controller -> User Controller
- T6** Reservation Controller -> Car Controller
- T7** Trip Controller -> User Controller
- T8** Trip Controller -> Car Controller
- T9** Trip Controller -> Reservation Controller
- T10** Router -> Lower Level Components
- T11** GUI -> Router



Entry criteria

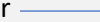
The following items must be delivered before integration testing can begin:

- Chapters 1 and 2 of the Design Document [DD]
- Chapters 1, 2, 3 of this document

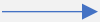
Furthermore, each component has to be unit tested and the required testing drivers must be developed (as defined for each integration test in the Test Case Specifications section).

Test Case Specifications


Integration Test Case T1

Test Case Identifier	T1
Test Item	User Controller  Database
Input Specification	Dummy user info
Output Specification	Correct interaction with the database
Environment Needs	N/A

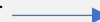
Integration Test Case T2

Test Case Identifier	T2
Test Item	Car Controller  Database
Input Specification	Dummy car info
Output Specification	Correct interaction with the database
Environment Needs	N/A

Integration Test Case T3

Test Case Identifier	T3
Test Item	Reservation Controller  Database
Input Specification	Dummy reservation info
Output Specification	Correct interaction with the database
Environment Needs	User and Car Driver

Integration Test Case T4

Test Case Identifier	T4
Test Item	Trip Controller  Database
Input Specification	Dummy trip info
Output Specification	Correct interaction with the database
Environment Needs	User, Car, and Reservation Driver

Integration Test Case T5

Test Case Identifier	T5
Test Item	Reservation Controller —————> User Controller
Input Specification	Reservation Info
Output Specification	Correct reservation output (ex. car reserved to user)
Environment Needs	T1, T2, T3 successful

Integration Test Case T6

Test Case Identifier	T6
Test Item	Reservation Controller —————> Car Controller
Input Specification	Reservation Info
Output Specification	Correct reservation output (ex. car reserved to user)
Environment Needs	T1, T2, T3 successful

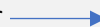
Integration Test Case T7

Test Case Identifier	T7
Test Item	Trip Controller —————> User Controller
Input Specification	Trip Info
Output Specification	Correct trip execution output (ex. Reservation deleted, trip start and finish procedures are correct)
Environment Needs	T1, T2, T3, T6 successful


Integration Test Case T8

Test Case Identifier	T8
Test Item	Trip Controller —————> Car Controller
Input Specification	Trip Info
Output Specification	Correct trip execution output (ex. Reservation deleted, trip start and finish procedures are correct)
Environment Needs	T1, T2, T3, T6 successful

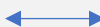
Integration Test Case T9

Test Case Identifier	T9
Test Item	Trip Controller  Reservation Controller
Input Specification	Trip Info
Output Specification	Correct trip execution output (ex. Reservation deleted, trip start and finish procedures are correct)
Environment Needs	T1, T2, T3, T6 successful

Integration Test Case T10

Test Case Identifier	T10
Test Item	Router  Lower-level Components
Input Specification	Different operations (reservation, trips etc...)
Output Specification	Make sure that the router is correctly routing the calls between the user-side(GUI) and the application server
Environment Needs	T6, T7, T8, T9 successful

Integration Test Case T11

Test Case Identifier	T11
Test Item	Gui  Router
Input Specification	Different operations (reservation, trips etc...)
Output Specification	Make sure that the GUI is correctly relaying the user inputs to the Router, and correctly displaying the information received from the router
Environment Needs	T10 successful

Test Procedures

Integration Test Procedure TP1

Test Procedure Identifier	TP1
Procedure	This test procedure verifies whether all the core components related with data production and updating correctly interact with the DBMS component
Procedure Steps	Execute T1, T2, T3, T4

Integration Test Procedure TP2

Test Procedure Identifier	TP2
Procedure	This test procedure verifies whether the software application logic can correctly manage an incoming reservation request
Procedure Steps	Execute T5, T6

Integration Test Procedure TP3

Test Procedure Identifier	TP3
Procedure	This test procedure verifies whether the software application logic can correctly manage a trip process in all its phases
Procedure Steps	Execute T7, T8, T9

Integration Test Procedure TP4

Test Procedure Identifier	TP4
Procedure	This test procedure verifies whether the user's mobile application and the in-car system can correctly interact with the application server's components
Procedure Steps	Execute T10, T11

Revision History

V 1.0 – Initial version

Effort Spent

Bachar Senno:

- 10/1: 2 hours
- 12/1: 1 hour
- 13/1: 1 hour

Onell Saliby

- 10/1: 2 hours
- 12/1: 1 hour
- 13/1: 1 hour

Gianluigi Iobizzi

- 10/01: 2 hours
- 13/01: 1 hour
- 14/1: 1 hour