POLITECNICO DI MILANO

SOFTWARE ENGINEERING 2 (2015 - 2016)

Integration Test Plan Document

myTaxiService V1

Authors:

Sun Chao

Daniel Naveda

Bakti Ariani Melinda Pertiwi

January 20th 2016

Contents

1. Introduction

1.1 Revision History

Version	Date	Author(s)	Summary
0.1.0	17-01-2016	SunChao	Initial revision. Document creation
0.2.0	20-01-2016	Daniel	Draft
0.3.0	20-01-2016	Bakti Ariani	Internally accepted

1.2 Purpose and Scope

1.2.1 Purpose

The purpose of the integration test plan is to describe the necessary tests to verify that all of the components of myTaxiService are properly assembled. Integration testing ensures that the unit-tested modules interact correctly.

1.2.2 Scope

MyTaxiService allows registered user to use a Mobile Device to order a taxi service in a convenient way. Also, it is more useful to manage the behaviors of the registered drivers. The basic architecture is a two-tier Server-Client model paradigm. The server will be our management system in charge of assigning the taxis, and the clients will be either a regular user or a taxi driver.

1.3 List of Definitions and Abbreviations

1.3.1 Definitions

- Mobile App : Mobile Application
- Taxi : Vehicle used to transport passengers as a service provided

- Registered Passenger : Passenger that have registered
- Registered Driver : Taxi drivers that have registered and have been authenticated by the system manager
- Authentication : To make sure the driver is really existed and belong to the one taxi company
- Android application : An application created on the Android operating system
- Google map : An electronic map developed by Google company

1.3.2 Abbreviations

- RASD : Requirements analysis and specification document
- DD : Design Document
- GPS : Global Positioning System
- UI : User Interface
- DBMS : Database management system
- API : Application Programming Interface
- JVM : Java Virtual Machine
- JEE : Java Enterprise Edition
- SDK : Software Development Kit
- ADT : Android Development Tools
- App : Software Application
- ITPD :Integration Test Plan Document

1.4 List of Reference Documents

- Specification document: “Assignments 1.pdf” assigned by the Professor of the course “Software Engineering 2”.

- Specification document: “Assignments 2.pdf” assigned by the Professor of the course “Software Engineering 2”.
- Specification document: “Assignments 4.pdf” assigned by the Professor of the course “Software Engineering 2”.
- Delivery for the design document: “myTaxiService_DD.pdf” completed by our group.
- Delivery for the Requirement document “RASD_TaxiServ_FinalV1.pdf” completed by our group
- IEEE Std 1016-2009 (Revision of IEEE Std 1016-1998) IEEE Standard for Information Technology—Systems Design— Software Design Descriptions

2. Integration Strategy

2.1. Entry Criteria

Before Integration test, it should be ensure that each component of myTaxiService is interacting with other components as expected. Also, we assume that each module (function) has been unit-tested successfully and all the features described on DD have been achieved on the coding section. Moreover, the RASD and DD for this project have been provided.

2.2. Elements to be Integrated

In the component view diagram from the "Design Document" (Assignment 2), we can view the elements which require integration to other elements. In the following diagram, we will illustrate these elements, with emphasis on the interconnection among them.

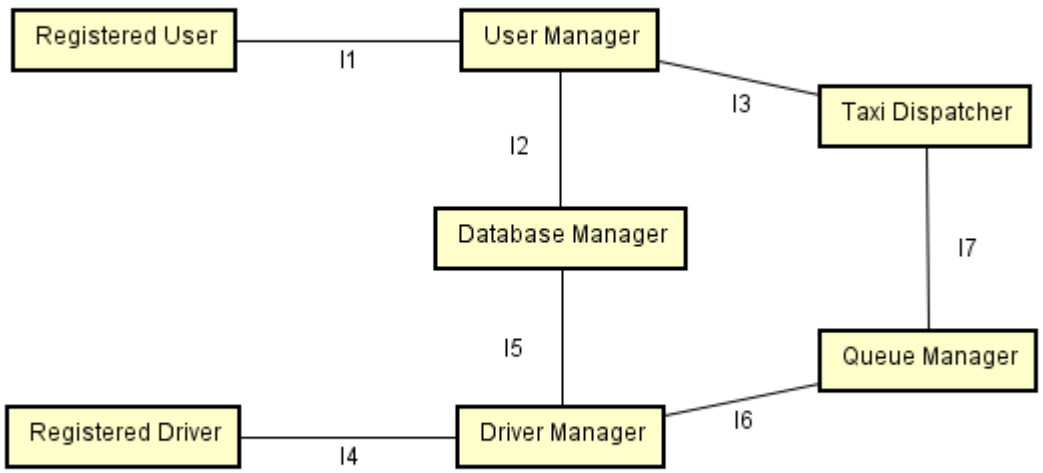

Figure 1 – Elements to Integrate

As we can see from the Figure shown above, we have 7 interfaces, named: I1, I2, … , I7. These will be the identifiers used to describe their testing plan.

2.3. Integration Testing Strategy

The integration testing strategy adopted was “Bottom Up approach”. In this case, we will be developing “Drivers” in order to check that the functionality is being performed as expected.

2.4. Sequence of Component

2.4.1. Software Integration Sequence

Integration tests of the myTaxiService System:

ID	Integration Test	Paragraphs
I1	Registered User -> User Manager	3.1 3.8 5.1
I2	User Manager -> Database Manager	3.2 3.8 5.2
I3	User Manager -> Taxi Dispatcher	3.3 3.8 5.3
I4	Registered Driver -> Driver Manager	3.4 3.8 5.4
I5	Driver Manager -> Database Manager	3.5 3.8 5.5
I6	Driver Manager -> Queue Manager	3.6 3.8 5.6
I7	Taxi Dispatcher -> Queue Manager	3.7 3.8 5.7

3. Individual Steps and Test Description

3.1. Integration test case I1

Test Case Identifier	**I1**
Test Item(s)	Registered User -> User Manager
Input Specification	Create typical User's Communication input
Output Specification	Check if the correct functions are called in the User Manager
Environmental Needs	User Driver

3.2. Integration test case I2

Test Case Identifier	**I2**
Test Item(s)	User Manager -> Database Manager
Input Specification	Create typical User Manager input
Output Specification	Check if the correct methods are called in the Database Manager
Environmental Needs	I1 succeeded

3.3. Integration test case I3

Test Case Identifier	**I3**
Test Item(s)	User Manager -> Taxi Dispatcher
Input Specification	Create typical User Manager input
Output Specification	Check if the correct functions are called in the Taxi Dispatcher
Environmental Needs	I1 succeeded

3.4. Integration test case I4

Test Case Identifier	**I4**
Test Item(s)	Registered Driver -> Driver Manager
Input Specification	Create typical Driver's Communication input
Output Specification	Check if the correct functions are called in the Driver Manager
Environmental Needs	Registered-Driver Driver

3.5. Integration test case I5

Test Case Identifier	**I5**
Test Item(s)	Driver Manager -> Database Manager
Input Specification	Create typical Driver Manager input
Output Specification	Check if the correct methods are called in the Database Manager
Environmental Needs	I4 succeeded

3.6. Integration test case I6

Test Case Identifier	**I6**
Test Item(s)	Driver Manager -> Queue Manager
Input Specification	Create typical Driver Manager input
Output Specification	Check if the correct methods are called in the Queue Manager
Environmental Needs	I4 and I5 succeeded

3.7. Integration test case I7

Test Case Identifier	**I7**
Test Item(s)	Taxi Dispatcher -> Queue Manager
Input Specification	Create typical Taxi Dispatcher input
Output Specification	Check if the correct methods are called in the Queue Manager
Environmental Needs	I1,I2 and I3 succeeded

3.8. Test Integration Description

Integration test procedure of the myTaxiService system:

Test Procedure Identifier	**TP 1**
Purpose	This test procedure verifies whether the myTaxiService Server: • Can handle Registered Drivers

	• Can handle Registered Users
Procedure Steps	Execute I1-I3 and I7 after I4-I6

Test Procedure Identifier	**TP 2**
Purpose	This test procedure verifies whether the user software: • Can handle request from the client • Can output information to the client
Procedure Steps	Execute I1-I3

Test Procedure Identifier	**TP 3**
Purpose	This test procedure verifies whether the driver software: • Can handle client request • Can output information to the client
Procedure Steps	Execute I4-I6

4. Tools and Test Equipment Required

4.1. Tools and test in requirement management

Tools and test supporting requirement management enable

- To store requirements and specification
- To check consistency and ambiguities in requirements
- To prioritize requirements and specification
- To ensure traceability of each test.

4.2. Other test supporting tools

There are many tools supporting software and system test :

- Website analysis tools, able to identify missing pages
- Spellers and syntax checkers
- Tools to test network safety or vulnerability
- Specific tools like
    - Junit
    - Mockito

5. Program Stubs and Test Data Required

5.1. Test case I1

Test Case Identifier	I1
Test Item(s)	Registered User -> User Manager
Program Stubs Required	Not required
Test Data Required	The Registered User should have valid credentials in order to communicate with the Server

5.2. Test case I2

Test Case Identifier	I2
Test Item(s)	User Manager -> Database Manager
Program Stubs Required	Not required
Test Data Required	The User Manager should have the right credentials to access the Database Manager

5.3. Test case I3

Test Case Identifier	I3
Test Item(s)	User Manager -> Taxi Dispatcher
Program Stubs Required	Not required
Test Data Required	Not required

5.4. Test case I4

Test Case Identifier	I4
Test Item(s)	Registered Driver -> Driver Manager
Program Stubs Required	Not required
Test Data Required	The Registered Driver should have valid credentials in order to communicate with the Server

5.5. Test case I5

Test Case Identifier	I5
Test Item(s)	Driver Manager -> Database Manager
Program Stubs Required	Not required
Test Data Required	The Driver Manager should have the right credentials to access the Database Manager

5.6. Test case I6

Test Case Identifier	I6
Test Item(s)	Driver Manager -> Queue Manager
Program Stubs Required	Not required
Test Data Required	Not required

5.7. Test case I7

Test Case Identifier	I7

Test Item(s)	Taxi Dispatcher -> Queue Manager
Program Stubs Required	Not required
Test Data Required	The queue manager should have a well-established queue with valid taxis