myTaxiService
-
Test Plan Document

Davide Cremona (matr. 852365), Simone Deola (matr. 788181)

January 19, 2016

Contents

1 Introduction **2**

1.1 Revision History . . . 2

1.1.1 Document Data . . . 2

1.1.2 Document Releases . . . 2

1.2 Purpose and Scope . . . 3

1.2.1 Purpose . . . 3

1.2.2 Scope . . . 3

1.3 List of Definitions and Abbreviations . . . 3

1.3.1 Definitions . . . 3

1.3.2 Abbreviations . . . 3

1.4 List of Reference Documents . . . 3

2 Integration Strategy **4**

2.1 Entry Criteria . . . 4

2.2 Elements to be Integrated . . . 4

2.3 Integration Testing Strategy . . . 5

2.4 Sequence of Component/Function Integration . . . 5

2.4.1 Software Integration Sequence . . . 6

2.4.2 Subsystem Integration Strategy . . . 8

3 Individual Steps and Test Description **10**

3.1 Integration test case IT1 . . . 10

3.2 Integration test case IT2 . . . 10

3.3 Integration test case IT3 . . . 11

3.4 Integration test case IT4 . . . 11

3.5 Integration test case IT5 . . . 11

3.6 Integration test case IT6 . . . 12

3.7 Integration test case IT7 . . . 12

3.8 Integration test case IT8 . . . 12

3.9 Integration test case IT9 . . . 13

3.10 Integration test case IT10 . . . 13

3.11 Integration test case IT11 . . . 13

3.12 Integration test case IT12 . . . 14

3.13 Integration test case IT13 . . . 14

3.14 Integration test case IT14 14
3.15 Integration test case IT15 15
3.16 Integration test case IT16 15
3.17 Integration test case IT18 15
3.18 Integration test case IT17 16

4 Tools and Test Equipment Required 17

5 Program Stubs and Test Data Required 18

6 Appendix 19
6.1 Used Software 19
6.2 Hours of Work 19

Chapter 1

Introduction

1.1 Revision History

1.1.1 Document Data

- **Title:** Test Plan Document.
- **ID:** myTaxiService - TPD.
- **Authors:** Deola Simone, Cremona Davide.
- **Last Version:** 0.1.
- **Status:** In Writing.

1.1.2 Document Releases

- **Version 0.1**
  - **Date:** 13/01/2016
  - **Authors:** Deola Simone, Cremona Davide.
  - **Changes:** Creation of the document.

1.2 Purpose and Scope

1.2.1 Purpose

The purpose of this document is to provide a description of how the Integration Testing is planned. The purpose of the integration testing is to test the right interaction between the various components described in the Design Document (DD,2.3).

1.2.2 Scope

The scope of the developed software is to provide an interface between the customers and the taxi drivers. The application is in charge to simplify the requests and the reservations of the rides and the management of the city queues.

1.3 List of Definitions and Abbreviations

1.3.1 Definitions

- **Taxi Drivers**: Taxi Drivers are the workers that drive a taxi. They are also users of myTaxiService.
- **Customers**: Customers are the end-users of the application.

1.3.2 Abbreviations

- **DD**: State for Design Document
- **DD-x.y**: Indicates the section y of the chapter x of the Design Document
- **RASD**: States for Requirement Analysis and Specification Document.

1.4 List of Reference Documents

- **Application Description**: downloadable from here: `https://github.com/SimoneDeola/CremonaDeola/blob/master/source/testPlan/documents/Assignments%201%20and%202%20(RASD%20and%20DD).pdf`
- **RASD**: downloadable from here: `https://github.com/SimoneDeola/CremonaDeola/raw/master/Deliveries/%5B1%5Drasd.pdf`
- **DD**: downloadable from here: `https://github.com/SimoneDeola/CremonaDeola/raw/master/Deliveries/%5B2%5Ddd.pdf`
- **Documentation of Tools Used:**

Chapter 2

Integration Strategy

2.1 Entry Criteria

To start with the Integration test, all the components must be tested with an unit tool (at least 90% lines of code). Also, the documentation about test results and public methods must be provided for each component.

2.2 Elements to be Integrated

The components that must be integrated are divided in three subsystems (Logic, Data and Presentation layers as specified in the DD). Each subsystem is composed by some components that can be seen as small sub-subsystems. Here are reported all the public components and sub-components that need to be integrated.:

- **Data**:
  - Model.
    - Database Manager.
- **Logic**:
  - IOManager.
    - Message Handler.
  - Requests and Reservations Manager.
    - Taxi Requests Adder.
    - Taxi Reservations Adder.
    - Ride Ender.
    - Taxi Allocator.
  - Profile Manager.
    - Registration Manager.

        - Login Manager.
        - Profile Modifier.
        - Profile Security.
        - Profile Viewer.
        - Credential Checker.
    - Zones Manager.
        - GPS Updater.
        - Zone Calculator.
        - Queue Manager.

- **Presentation**:
    - Web Application UI.
    - Mobile Application UI.
    - For each of these two components, these sub-components needs to be tested:
        - Response Receiver.
        - Command Sender.

2.3 Integration Testing Strategy

The strategy that we want to follow is the Bottom-Up Strategy. This kind of approach consists in testing the low level components first and then the level just above, until you reach the top level components. This strategy is suitable for our system because every component is divided in low level components.

2.4 Sequence of Component/Function Integration

Here we present the integration sequence for each component of the system. The arrows represents the priorities of the components' integration.
Example:
Component A needs to be tested to unlock the testing of Component B.

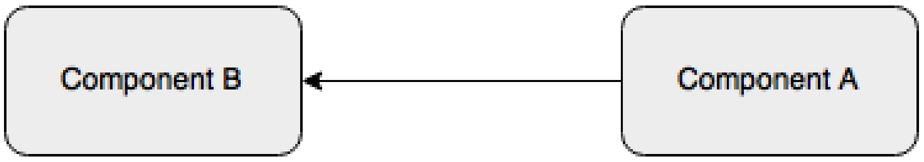

Figure 2.1: Priority Example

2.4.1 Software Integration Sequence

Here we present the integration sequences divided in subsystems, here are listed only the components that have internal interactions between sub-components, for the interaction between components you can refer to the next section:

- **Logic**:
  - Here is illustrated the integration sequence for the Profile Manager package:

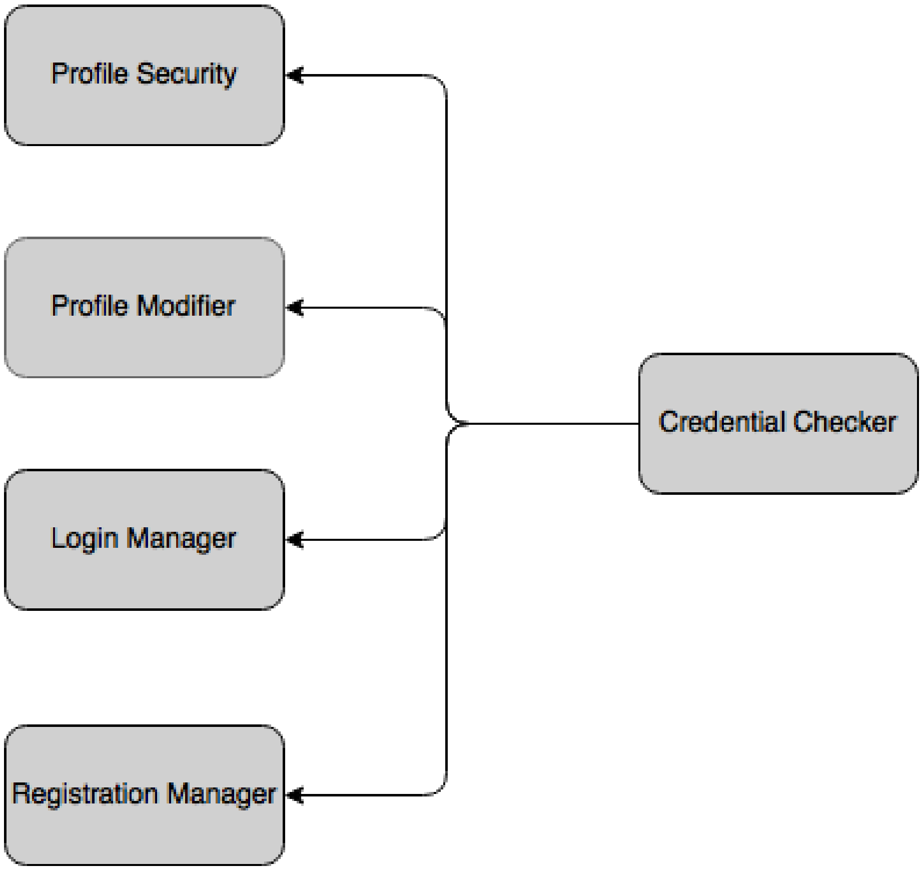

Figure 2.2: Profile Manager Integration Sequence

  - Here is illustrated the integration sequence for the Zone Manager package:

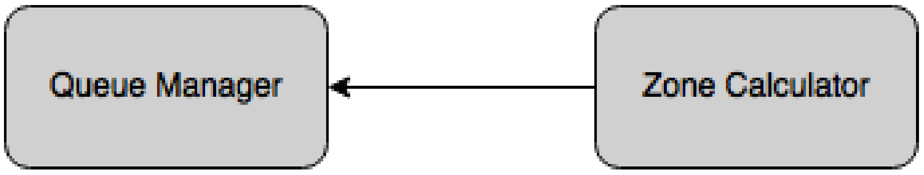

Figure 2.3: Zone Manager Integration Sequence

- **Presentation**:
  - Here is illustrated the integration sequence for the Web Application UI and Mobile Application UI packages:

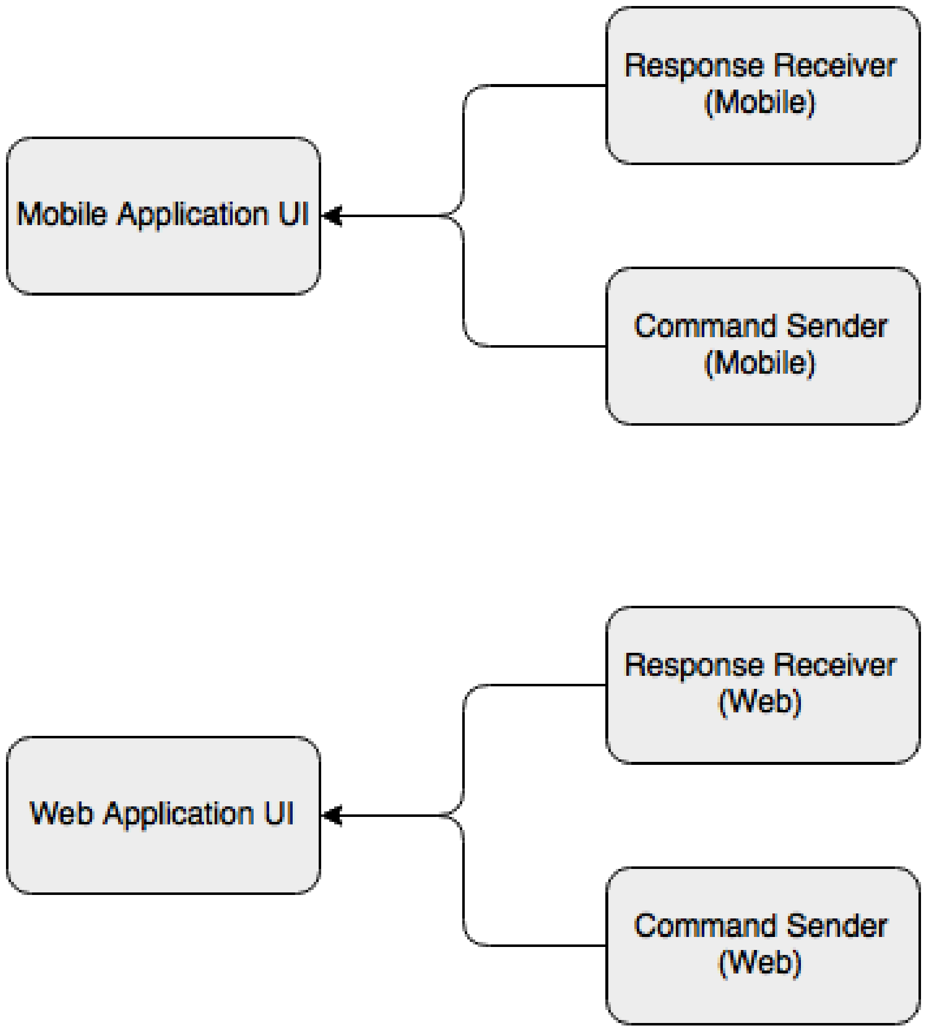

Figure 2.4: Mobile and Web Application UI Integration Sequence

2.4.2 Subsystem Integration Strategy

Here is illustrated how the various components of the subsystems are integrated toghether:

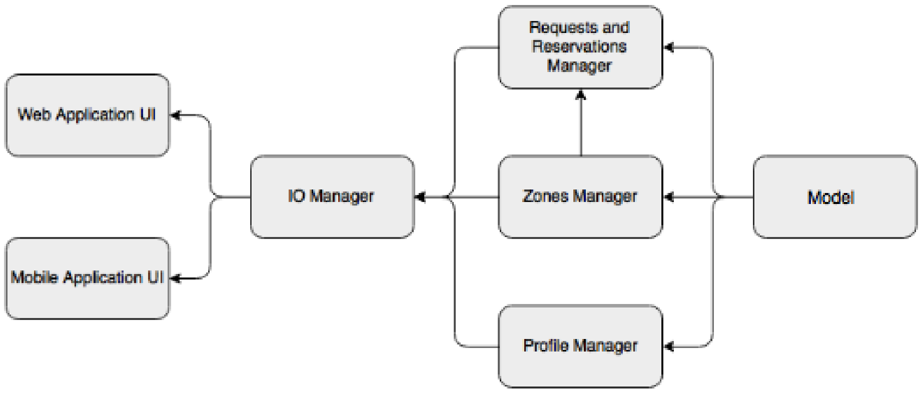

Figure 2.5: Subsystems Integration Sequence

Chapter 3

Individual Steps and Test Description

3.1 Integration test case IT1

Test Case Identifier	IT1
Test Item(s)	CredentialChecker $\rightarrow$ ProfileSecurity
Input Specification	Create typical CredentialChecker input
Output Specification	Check if the correct functions are called in the ProfileSecurity
Environmental Needs	N/A
Purpose of Test	Verify that the mail is checked properly

Table 3.1: Integration Test IT1

3.2 Integration test case IT2

Test Case Identifier	IT2
Test Item(s)	CredentialChecker $\rightarrow$ ProfileModifier
Input Specification	Create typical CredentialChecker input
Output Specification	Check if the correct functions are called in the ProfileModifier
Environmental Needs	N/A
Purpose of Test	Verify that the user credential checking process works properly

Table 3.2: Integration Test IT2

3.3 Integration test case IT3

Test Case Identifier	IT3
Test Item(s)	CredentialChecker $\rightarrow$ LoginManager
Input Specification	Create typical CredentialChecker input
Output Specification	Check if the correct functions are called in the LoginManager
Environmental Needs	N/A
Purpose of Test	Verify that the user credential checking process works properly

Table 3.3: Integration Test IT3

3.4 Integration test case IT4

Test Case Identifier	IT4
Test Item(s)	CredentialChecker $\rightarrow$ RegistrationManager
Input Specification	Create typical CredentialChecker input
Output Specification	Check if the correct functions are called in the RegistrationManager
Environmental Needs	N/A
Purpose of Test	Verify that the user credential checking process works properly

Table 3.4: Integration Test IT4

3.5 Integration test case IT5

Test Case Identifier	IT5
Test Item(s)	ZoneCalculator $\rightarrow$ QueueManager
Input Specification	Create typical ZoneCalculator input
Output Specification	Check if the correct functions are called in the QueueManager
Environmental Needs	N/A
Purpose of Test	Verify that the queue assignement process works properly

Table 3.5: Integration Test IT5

3.6 Integration test case IT6

Test Case Identifier	IT6
Test Item(s)	ResponseReceiver $\rightarrow$ MobileApplicationUI
Input Specification	Create typical ResponseReceiver input
Output Specification	Check if the correct functions are called in the MobileApplicationUI
Environmental Needs	N/A
Purpose of Test	Verify that the notifications from the server are correctly showed

Table 3.6: Integration Test IT6

3.7 Integration test case IT7

Test Case Identifier	IT7
Test Item(s)	CommandSender $\rightarrow$ MobileApplicationUI
Input Specification	Create typical CommandSender input
Output Specification	Check if the correct functions are called in the MobileApplicationUI
Environmental Needs	N/A
Purpose of Test	Verify that the user gestures are correctly translate in message for the server

Table 3.7: Integration Test IT7

3.8 Integration test case IT8

Test Case Identifier	IT8
Test Item(s)	ResponseReceiver $\rightarrow$ WebApplicationUI
Input Specification	Create typical ResponseReceiver input
Output Specification	Check if the correct functions are called in the WebApplicationUI
Environmental Needs	N/A
Purpose of Test	Verify that the notifications from the server are correctly showed

Table 3.8: Integration Test IT8

3.9 Integration test case IT9

Test Case Identifier	IT9
Test Item(s)	CommandSender $\rightarrow$ WebApplicationUI
Input Specification	Create typical CommandSender input
Output Specification	Check if the correct functions are called in the WebApplicationUI
Environmental Needs	N/A
Purpose of Test	Verify that the user gestures are correctly translated in messages for the server

Table 3.9: Integration Test IT9

3.10 Integration test case IT10

Test Case Identifier	IT10
Test Item(s)	Model $\rightarrow$ ProfileManager
Input Specification	Create typical Model input
Output Specification	Check if the correct functions are called in the ProfileManager
Environmental Needs	N/A
Purpose of Test	Verify that the ProfileManager can access correctly to the user information on the database

Table 3.10: Integration Test IT10

3.11 Integration test case IT11

Test Case Identifier	IT11
Test Item(s)	Model $\rightarrow$ ZoneManager
Input Specification	Create typical Model input
Output Specification	Check if the correct functions are called in the ZoneManager
Environmental Needs	N/A
Purpose of Test	Verify that the ZoneManager can access correctly to the positions information on the database

Table 3.11: Integration Test IT11

3.12 Integration test case IT12

Test Case Identifier	IT12
Test Item(s)	Model $\rightarrow$ RequestAndReservationManager
Input Specification	Create typical Model input
Output Specification	Check if the correct functions are called in the RequestAndReservationManager
Environmental Needs	N/A
Purpose of Test	Verify that the RequestAndReservationManager can access correctly to the ride information on the database

Table 3.12: Integration Test IT12

3.13 Integration test case IT13

Test Case Identifier	IT13
Test Item(s)	ZoneManager $\rightarrow$ RequestAndReservationManager
Input Specification	Create typical ZoneManager input
Output Specification	Check if the correct functions are called in the RequestAndReservationManager
Environmental Needs	N/A
Purpose of Test	Verify that the ride requests are assigned correctly to the most appropriate taxiDriver

Table 3.13: Integration Test IT13

3.14 Integration test case IT14

Test Case Identifier	IT14
Test Item(s)	RequestAndReservationManager $\rightarrow$ IOManager
Input Specification	Create typical RequestAndReservationManager input
Output Specification	Check if the correct functions are called in the IOManager
Environmental Needs	Integration test cases IT13 and IT12 needed.
Purpose of Test	Verify that the request from users relative to the rides are right handled

Table 3.14: Integration Test IT14

3.15 Integration test case IT15

Test Case Identifier	IT15
Test Item(s)	ZoneManager $\rightarrow$ IOManager
Input Specification	Create typical ZoneManager input
Output Specification	Check if the correct functions are called in the IOManager
Environmental Needs	Integration test cases IT5 and IT11 needed.
Purpose of Test	Verify that the information about the users positions are right stored

Table 3.15: Integration Test IT15

3.16 Integration test case IT16

Test Case Identifier	IT16
Test Item(s)	ProfileManager $\rightarrow$ IOManager
Input Specification	Create typical ProfileManager input
Output Specification	Check if the correct functions are called in the IOManager
Environmental Needs	Integration test cases IT1, IT2, IT3, IT4 and IT10 needed.
Purpose of Test	Verify that the actions between the user and his information on the server (Login, Registration, Modify, ...) are correctly handled

Table 3.16: Integration Test IT16

3.17 Integration test case IT18

Test Case Identifier	IT18
Test Item(s)	IOManager $\rightarrow$ WebApplicationUI
Input Specification	Create typical IOManager input
Output Specification	Check if the correct functions are called in the WebApplicationUI
Environmental Needs	Integration test cases IT16, IT15 and IT14 needed.
Purpose of Test	Verify that the action performed by the user on the web interface makes the right changes on the rest of the system

Table 3.17: Integration Test IT18

3.18 Integration test case IT17

Test Case Identifier	IT17
Test Item(s)	IOManager $\rightarrow$ MobileApplicationUI
Input Specification	Create typical IOManager input
Output Specification	Check if the correct functions are called in the MobileApplicationUI
Environmental Needs	Integration test cases IT16, IT15 and IT14 needed.
Purpose of Test	Verify that the action performed by the user on the mobile application makes the right changes on the rest of the system

Table 3.18: Integration Test IT17

Chapter 4

Tools and Test Equipment Required

In general, the tools used for integration testing depend on the programming language used for the development of the system. We assume that the system is currently developed in Java EE (Java Enterprise Edition). With respect to this assumption, required tools for the integration test are:

- JUnit: to make unit testing. (reference: `http://junit.org`)
- Arquillian: to make drivers for the bottom-up testing strategy (reference: `http://arquillian.org`)
- Manual testing can be considered a testing tool and can be used if necessary.

Chapter 5

Program Stubs and Test Data Required

For the integration testing, we have to populate the Database with some demo users. It will be required to check credentials, update positions etc... To emulate the GPS (that is external to the system) the integration testing needs some classes, that simulate a real GPS. To perform integration testing on the login and registration with Facebook it will be required to emulate the responses from the Facebook Login API. In general, if the components of the system are integrated during the development of the system itself, it will be required to create some drivers that emulate the desired behavior.

Chapter 6

Appendix

6.1 Used Software

To create this document we have used some common softwares:

- For the latex we have used two different softwares:
  - Simone Deola: TexShop, provided with the MacTex package (link)
  - Davide Cremona: Sublime Text editor (link) with LaTeXTools (link) and the Basic Package of MacTex (link)
- For the diagrams we have used Draw.io service (`http://www.draw.io`)

6.2 Hours of Work

- Cremona Davide (852365): 8 Hours
- Deola Simone (788181): 8 Hours