myTaxiService

-

Test Plan Document

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

January 15, 2016

Contents

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 reservation 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 consist in testing the low level components for first and then the level just above until you reach the top level components. This strategy is suitable for our system because evary 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 need to be tested to unlock the testing of Component B.

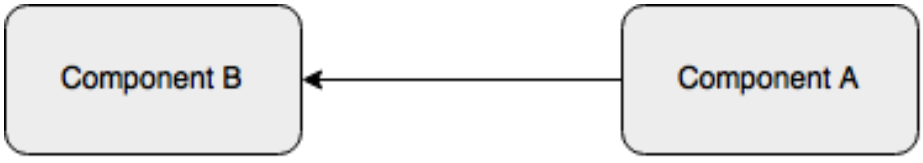

Figure 2.1: Priority Example

2.4.1 Software Integration Sequence

Here we present the integration sequences divided by 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 it's illustrated the integration sequence for the Profile Manager package:

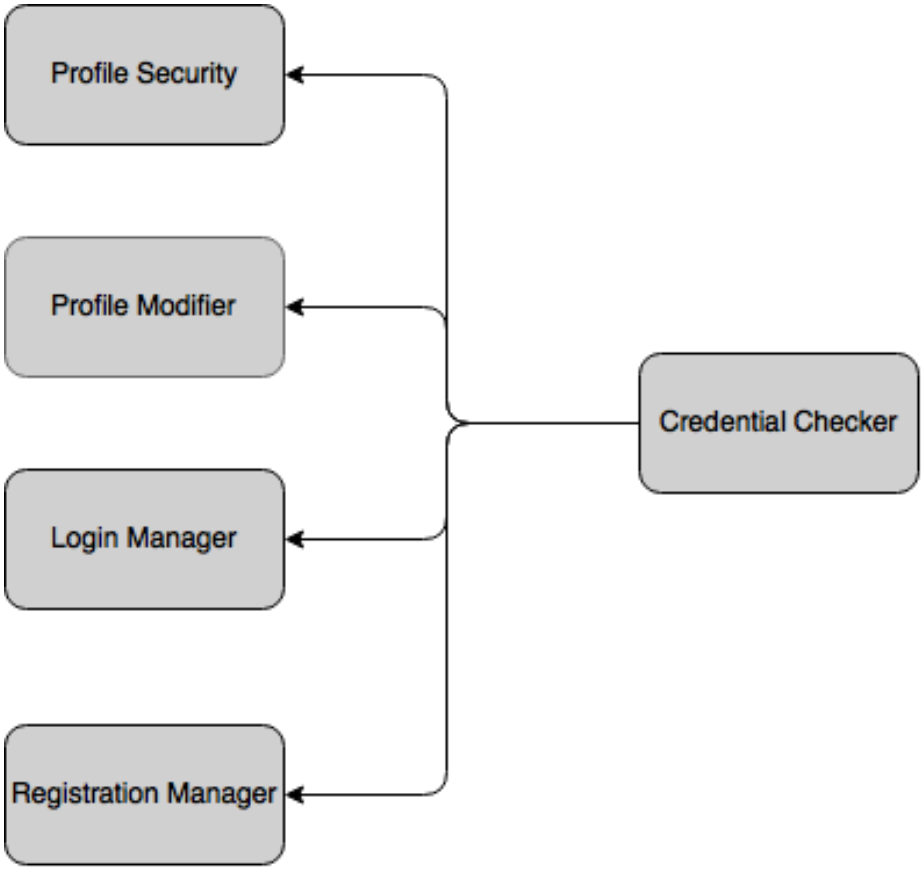

Figure 2.2: Profile Manager Integration Sequence

  - Here it's illustrated the integration sequence for the Zone Manager package:

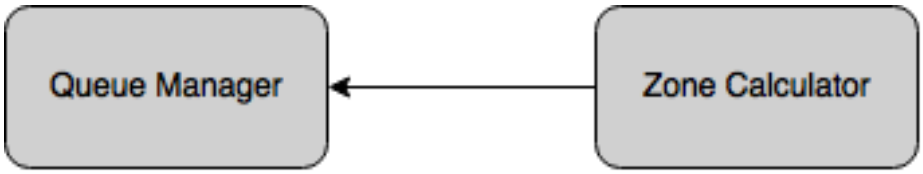

Figure 2.3: Zone Manager Integration Sequence

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

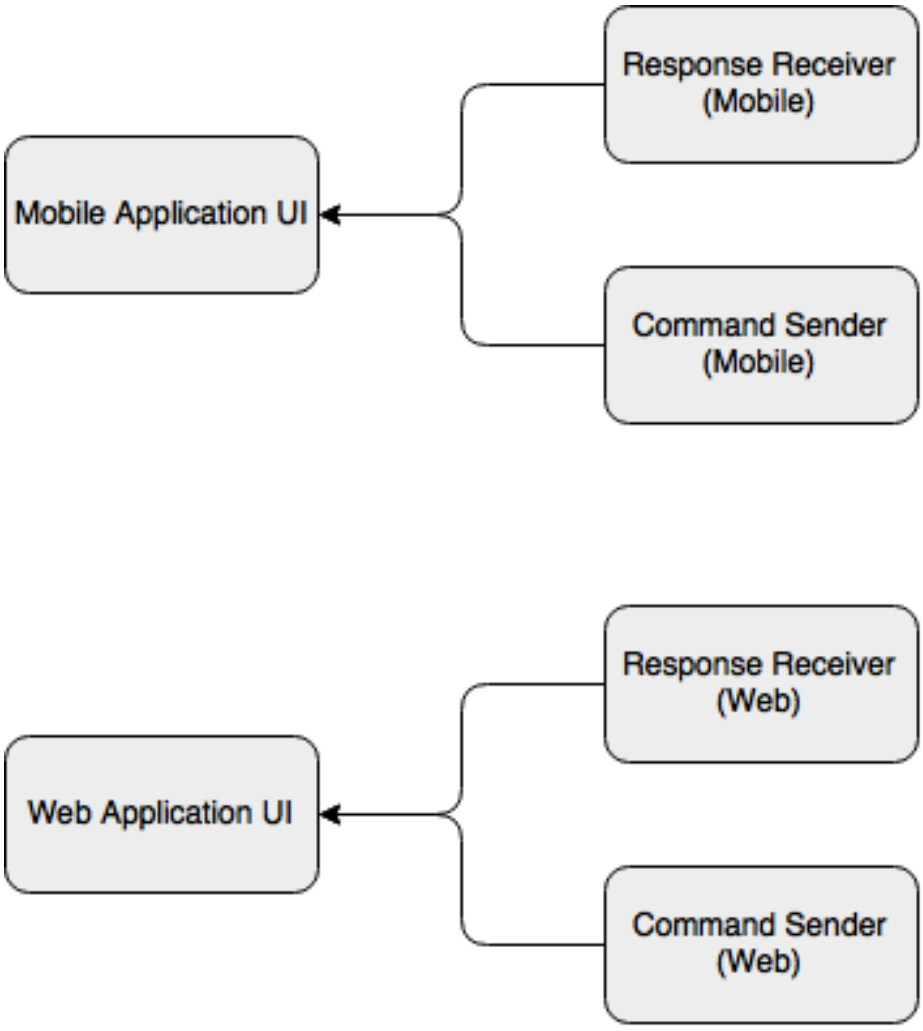

Figure 2.4: Mobile and Web Application UI Integration Sequence

2.4.2 Subsystem Integration Strategy

Here it's illustrated how the various components of the subsystems are integrated toghether:

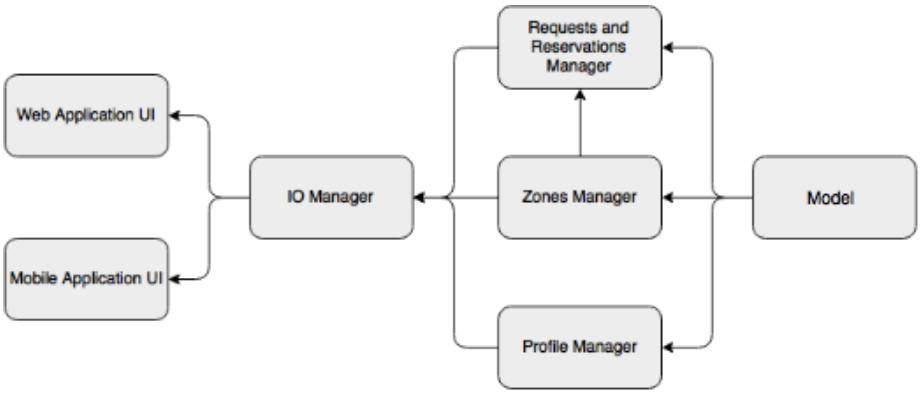

Figure 2.5: Subsystems Integration Sequence

Chapter 3

Individual Steps and Test Description

Test Case Identifier
Test Item(s)
Input Specification
Output Specification
Environmental Needs
Purpose of Test

Table 3.1: Integration Test IT.number

Chapter 4

Tools and Test Equipment Required

Chapter 5

Program Stubs and Test Data Required

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