Integration Test Plan Document

Version 1.0

Giorgio Pea(Mat. 853872), Andrea Sessa(Mat. 850082)

21/1/2016

Contents

1 Introduction

1.1 Revision History

Date	Description	Authors
21/01/2016	Delivers of version 1.0	Pea, Sessa

1.2 Purpose

The purpose of the integration test plan is to describe the necessary tests to verify that all the components(see **DD**, reference section) of MyTaxiServive are properly assembled. Integration testing ensures that the unit-tested components interact correctly.

1.3 Scope

The aim of this project is to develop MyTaxiService, a web/mobile application that makes easier and quicker taking taxies within the city's borders. Thanks to MyTaxiService, anyone can request or book a taxi and get realtime information about how long it will take to be picked up or about the taxi's current position and identification code. In addition to that, MyTaxiService provides an efficient way to allocate taxies by dividing the city in zones and using a queue based allocation system, in order to reduce the average waiting time and city's traffic.

1.4 Terms Definition

1.4.1 Glossary

- **Mtaxi:** A taxi that joined MyTaxiService
- **User:** Refers to either a logged registered user or a generic user(see RASD)
- **MyTaxiService(B):** see RASD
- **Administrator:** see RASD
- **Mtaxi bad beahvior**: see RASD

1.4.2 Acronyms

- **DD:** Design Document
- **MVC:** Model View Controller
- **FIFO:** First In First Out

- **API:** Application Programming Interface
- **GUI:** Graphic User Interface
- **GPS:** Global Positioning System
- **AWT:** Approximate waiting time to be picked up

1.5 Reference Documents

- [**RASD**]Requirements and analisys specification document(RASD)
- [**DD**] Design Document(DD)

2 Integration Strategy

2.1 Overview

In the first part of this section the components to be tested are mentioned, moreover the testing strategy is chosen and the integration testing order defined.
A more detailed specification for each test case is given in the third chapter.

2.2 Entry Criteria

The main entry conditions for this phase is that each of the system components low level functions has been previously subjected to a unit test process.

2.3 Elements to be integrated

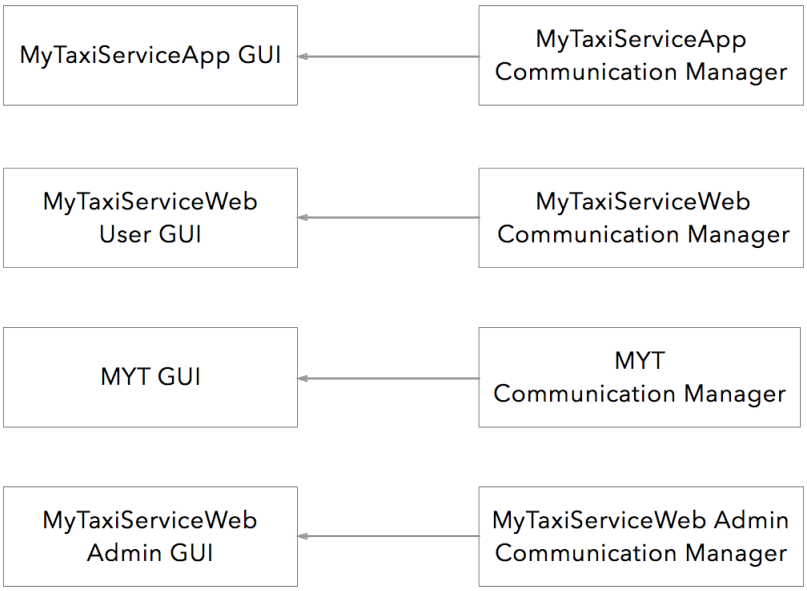
For a detailed description of each components function and interaction refer to the **DD**
Two main subsystem can be individuated in the general architecture of MyTaxiService:

1. **Client side Component**
    - MyTaxiServiceAppUser GUI
    - MyTaxiServiceMYT GUI
    - MyTaxiServiceWebUser GUI
    - MyTaxiServiceWebAdmin GUI
    - MyTaxiServiceAppUser Communication Manager
    - MyTaxiServiceMYT Communication Manager
    - MyTaxiServiceWebUser Communication Manager
    - MyTaxiServiceWebAdmin Communication Manager
2. **Server side Component**
    - Dispatcher
    - Request Manager
    - External Services Manager
    - Location Manager
    - Queue Manager
    - Request Receiver
    - Data Manager

2.4 Integration Testing Strategy

The strategy chosen for the integration of the components of the application is a grouped Bottom-Up integration: for each subsystem S of MyTaxiService, its components are orchestrated together following a bottom-up approach, when this procedure is finished a general integration of the subsystems of the application with each other is performed. This integration testing strategy has been selected because it mimics very well the structure and the modularity of the software system. The bottom-up approach consists in integrating first those components that are at the low level of the component's dependency tree (see DD's component diagram).

2.5 Sequence of Component/Function Integration



3 Individual Step and Test Description

4 Tools and Test Equipment Required

5 Program Stub and Test Data Required