myTaxiService
Integration **T**est **P**lan **D**ocument

A.Y. 2015/2016

Politecnico di Milano

Version 1.0

Cattaneo Michela Gaia, matr. 791685
Barlocco Mattia, matr. 792735

January 22, 2016

Contents

1 Introduction

1.1 Revision History

1.2 Purpose and Scope

This document represents the Integration Test Plan Document. It aims at specifying the plan for the integration testing of myTaxiService application, ensuring that all its modules interacts properly.
This document is intended for the development team, that needs to know what it is needed to test, in which sequence it should occur and with which tools.
myTaxiService is an application that simplifies the access of the customers to the taxi service, managing also the taxi driver's distribution over the city and the queues of the areas. The customer can use the mobile application or the web app to request or reserve a taxi and a notification is forwarded to the designated taxi driver, who is going to answer. This system should always be responsive and reliable, in order to keep up with all the requests of the passenger and the notifications forwarding.
This is the reason why the components that comprise the myTaxiService system are meant to be well integrated and tested.

1.3 List of Definitions and Abbreviations

- **Definitions**

- **User**: a person who requests a service from the system. It can be a visitor or a passenger.
- **Visitor**: a person who is not registered in the application.
- **Passenger**: a person who is registered in the application.
- **Taxi driver**: a taxi driver who access the application with a specific ID.
- **Request**: the request of a taxi in a certain area and position in the city made by a user.
- **Reservation**: the reservation of a taxi in a certain area, place and time that can be made only by passengers.

- **Acronyms and abbreviations**

- **RASD**: Requirement Analysis and Specification Document
- **DBMS**: Database Management System
- **JEE**: Java Enterprise Edition
- **API**: Application Programming Interface
- **UML**: Unified Modeling Language
- **RMI**: Remote Method Invocation
- **HTTP**: HyperText Transfer Protocol
- **UX**: User eXperience

1.4 List of Reference Documents

- Project Description and Rules (`https://github.com/MichelaCattaneo/myTaxiService/blob/master/Project%20Description%20And%20Rules.pdf`)
- Requirements Analysis and Specification Document (`https://github.com/MichelaCattaneo/myTaxiService/blob/master/Deliveries/RASD_1.1.pdf`)
- Design Document (`https://github.com/MichelaCattaneo/myTaxiService/blob/master/Deliveries/DD.pdf`)
- Integration Test Plan Example (`https://beep.metid.polimi.it/documents/3343933/5b3768d0-d949-4369-87e1-7a31b6943726`)

2 Integration Strategy

2.1 Entry Criteria

These are the criteria that must be respected before the integration testing phase may begin:

- Requirement Analysis and Specification Document is complete and revised
- Design Document is complete and revised
- Code Inspection Document is complete and revised
- The code is complete and bugs free
- The product satisfies the requirements and the assumptions specified in the RASD
- The product satisfies the architecture and the design specified in the DD
- Test environment, test cases and test data are ready

2.2 Elements to be Integrated

These components refer to the ones specified in the Component View in chapter 2.3 of the DD. This diagram shows how these components have to be integrated and the order of integration, according to the strategy adopted.

2.3 Integration Testing Strategy

The main advantage of the Bottom-Up approach is that bugs are more easily found. With Top-Down, it is easier to find a missing branch link. da database a moduli client La parte pi critica quella piu bassa!(che contiene la logica del sistema) meglio bottom up. cosi evitiamo di fare gli stub ma facciamo i driver che sono pi semplici. basic functionalities are tested at the beginning

2.4 Sequence of Component/Function Integration

2.4.1 Software Integration Sequence

2.4.2 Subsystem Integration Sequence

3 Individual Steps and Test Description

4 Tools and Test Equipment Required

5 Program Stubs and Test Data Required