Politecnico di Milano

Software Engineering 2 Project
a.y. 2015-16

# MyTaxiService
## Integration Test Plan Document
Version 1.0

Casati Fabrizio, 853195
Castelli Valerio, 853992

Referent professor: Di Nitto Elisabetta

January 15, 2016

# Contents

# Chapter 1

# Introduction

## 1.1 Revision History

| Version | Date | Author(s) | Summary |
|---|---|---|---|
| 1.0 | 21/01/16 | Valerio Castelli & Fabrizio Casati | Initial release |

## 1.2 Purpose and Scope

This document represents the Integration Testing Plan Document for myTaxiService. Integration testing is a key activity to guarantee that all the different subsystems composing myTaxiService interoperate consistently with the requirements they are supposed to fulfill and without exhibiting unexpected behaviors. The purpose of this document is to outline, in a clear and comprehensive way, the main aspects concerning the organization of the integration testing activity for all the components that make up the system. In the following sections we're going to provide:

- A list of the subsystems and their subcomponents involved in the integration activity that will have to be tested
- The criteria that must be met by the project status before integration testing of the outlined elements may begin
- A description of the integration testing approach and the rationale behind it
- The sequence in which components and subsystems will be integrated
- A description of the planned testing activities for each integration step, including their input data and the expected output

- A list of all the tools that will have to be employed during the testing activities, together with a description of the operational environment in which the tests will be executed

## 1.3 Definitions, Acronyms, Abbreviations

### 1.3.1 Definitions

- Subcomponent
- SubsystemTBD tbd

### 1.3.2 Acronyms

- SDD: Software Design Description.
- DD: Design Document. Used as a synonym of SDD.
- DBMS: Database Management System.
- API: Application Programming Interface.
- RASD: Requirement Analysis and Specification Document.
- SRS: Software Requirements Specifications. Synonym of RASD.
- ETA: Estimated Time of Arrival.
- UI: User Interface.
- GPS: Global Positioning System.

### 1.3.3 Abbreviations

- Req. as for Requirement.
- WebApp as for Web Application.

## 1.4 Reference Documents

- The project description document: Assignments 1 and 2 (RASD and DD).pdf
- Assignment document: Assignment 4 - integration test plan.pdf
- myTaxiService Requirement Analysis and Specification Document: RASD.pdf
- myTaxiService Design Document: DD.pdf

- The Integration Test Plan Example document: Integration Test Plan Example.pdf

# Chapter 2

# Integration Strategy

## 2.1 Entry Criteria

## 2.2 Elements to be Integrated

In the following paragraph we're going to provide a list of all the components that need to be integrated together.

As specified in myTaxiService's Design Document, the system is built upon the interactions of many high-level components, each one implementing a specific set of functionalities. For the sake of modularity, each subsystem is further obtained by the combination of several lower-level components. Because of this software architecture, the integration phase will involve the integration of components at two different levels of abstraction.

At the lowest level, we'll integrate together those components that depend strongly on one another to offer the higher level functionalities of myTaxiService. In our specific case, this involves the integration of the **Reservation Management**, **Request Management**, **Location Management** and **Taxi Management** subcomponents in order to obtain the **Taxi Management System** subsystem.

For what concerns the building of the **System Administration** and **Account Management** subsystems, the integration activity is actually quite limited; in fact, they simply represent a collection of functionalities belonging to the same area which however are not dependent on one another. As a result of this, their subcomponents don't really interact with each other, and the integration phase will be limited to the task of ensuring that the set of functionalities of each subcomponent is properly exposed by the subsystem. The components involved in this phase are:

- The **API Permissions Management**, **Zone Division Management**, **Taxi Driver Management**, **Service Statistics** and **Plugin Management** subcomponents in order to obtain the **System Administration** subsystem.

- The **Passenger Registration**, **Login**, **Password Retrieval** and **Settings Management** subcomponents in order to obtain the **Account Management** subsystem.

Some of these subcomponents also directly rely on higher level, atomic components: that is the case, for instance, of the dependency on the **Data Access Utilities** component. This dependency will be taken care of in the integration process.

Finally, we will proceed with the integration of the higher level subsystems. In particular, the integration activity will involve:

- A number of commercial, already existing components, used to achieve specific functionalities: these are the **DBMS**, **Mapping Service**, **Notification System** and **Remote Services Interface** components.
- Those components and subsystems specifically developed for myTaxiService, specifically:
  - On the server side: the **Taxi Management System**, **System Administration**, **Account Management** subsystems, together with the **Data Access Utilities** component.
  - On the client side: the **Administration Web Application**, **Passenger Web Application**, **Passenger Mobile Application** and **Taxi Driver Mobile Application** components.

## 2.3 Integration Testing Strategy

## 2.4 Sequence of Component/Function Integration

### 2.4.1 Software Integration Sequence

### 2.4.2 Subsystem Integration Sequence

# Chapter 3

# Individual Steps and Test Description

# Chapter 4

# Tools and Test Equipment Required

# Chapter 5

# Program Stubs and Test Data Required

# Appendix A

# Hours of work

To redact this document, we spent ?? hours per person.