

# Politecnico di Milano

# A.A. 2015-2016

# Software Engineering 2

# **Integration Test Plan**

# Alessandro Macchi

# Caterina Finetti

# Simone Manzoli

#### Abstract

This document describes the Integration Test Plan (ITP) for the Taxy App project.  
This document contains the description of the integration tests for the project.   
This project is one of five assignments for the course SOFTWARE ENGINEERING 2 at Politecnico di Milano.

#### Document Title: Integration Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Summary** |
| 0.1 | 22/01/2016 | A. Macchi,  C. Finetti,  S. Manzoli | Document creation |

[1 Introduction 5](#_Toc441244948)

[1.1 Purpose 5](#_Toc441244949)

[1.2 Scope 5](#_Toc441244950)

[1.3 Definitions, acronyms and abbreviations 6](#_Toc441244951)

[1.4 Documents 7](#_Toc441244952)

[1.5 Overview 8](#_Toc441244953)

1 Introduction

# 1.1 Purpose

This document describes the plans for testing the integration of the created components. The  
purpose of this document is to test the interfaces between the components as described in  
test report. Every team member who cooperates in the integration tests should read  
this document

# 1.2 Scope

The program allows users to access an app to make a reservation for a taxi and the taxi drivers to stay in the queue for the zone they belong and receive calls from users. The program's database contains user's and taxi driver's personal information, this information is used by the program to provide the service.

# 1.3 Definitions, acronyms and abbreviations

**User**: is a person who is registered in the database of the application. He has access to all the functions of the program that involves the requiring of a taxi, shared or not. He also has the possibility to save a list of preferred locations, that he can automatically choose when the System require from him an address as starting position or destination.

**Guest**: is a person who is using the application but is not registered in the database. He has access only to the registration functions.

**User Information**: all the information that concern a user; most of them have to be inserted during the registration (Name, Surname, tel. Number, e- mail, password), some of them can be inserted at any time after the registration (such as the personals locations) and others are assigned by the system (for example the number of blank-calls or the feedback).

**Feedback**: the feedback measures the reliability of a user. Is a simple relation between the total number of calls and the number of blank calls that a user have made (so a feedback equals to 1 means that he never missed a call).

**Basic User Information**: The information that a taxi driver visualizes when he receive a call. They are: Name, Surname, Feedback, Telephone Number of the user.

**Blank Call**: we define Blank-call a call for a taxi where the client is not at the starting location when the taxi arrives with a maximum late of X minutes, or a call that the user cancel before X minutes.

**Missed Call**: we define Missed-call a call for a taxi where the client is not at the starting location when the taxi arrives (X+1) or more minutes late.

**Partner**: someone who share a run with a user

**Pick-up place**: the Address where a user asks a taxi to come

**Taxi Driver**: a registered Taxi Driver

**BCE diagram**: boundary-controller-entity diagram

**UX diagram**: User experience diagram

# 1.4 Documents

# 1.5 Overview

In the second chapter the items to be tested are mentioned. A specification for each test case

is given in the third chapter. The fourth chapter specifies the procedures for these test cases.

In the fifth chapter the reports for all test cases are presented.

2. Integration Strategy

# 2.1 Entry Criteria

Before the integration test every function must be tested individually. The code needs to be complete and functional in its entirely. Every known exception needs to be treated and the developer must have a copy of the hardware that will be used (one driver's device witch gps and taximeter and one Smartphone).

# 2.2 Elements to be Integrated

The sub-system that will be integrated and tested are:

Call sub-system

Shared-call cost sub-system

User sub-system

Manage location sub-system

Manage account sub-system

Manage settings sub-system

Log in sub-system

Sign up sub-system

# 2.3 Integration Testing Strategy

The items to be tested consist of the integration of the code modules developed, for the MyTaxiDrive project. For testing we choose the bottom-up approach. This means that integration

testing starts at the bottom level. This way the project will be built up from

the bottom level.

# 2.4 Sequence of Component Integration

## 2.4.1 Software Integration Sequence