# Integration Test Plan Document

Filippo Agalbato
850481

Andrea Cannizzo
790469

January 17, 2016

# Contents

# Chapter 1

# Introduction

## 1.1 Purpose and scope

This is the Integration Test Plan Document for the project software My Taxi Service. The goal of this document is to describe the system's required integration testing framework and infrastructure that ought to prove its compliance with specifications, starting from these and proceeding up to the design phase of the different testing procedures.

This document is targeted to the project testers, to correctly build their infrastructure so that the integration testing phase can proceed according to plan.

General information on the My Taxi Service project itself may be found in the reference documents, as specified below.

## 1.2 Definitions, acronyms and abbreviations

## 1.3 References

- Specification Document: My Taxi Service project specification for the Academic Year 2015-16 – *Assignments 1 and 2 (RASD and DD).pdf*;
- Specification Document: Testing assignment specification for the Academic Year 2015-16 – *Assignment 4 – Integration test plan.pdf*;
- Agalbato, Cannizzo, *Requirement Analysis and Specification Document*;
- Agalbato, Cannizzo, *Design Document.*

# Chapter 2

# Integration strategy

## 2.1 Entry criteria

Before the integration testing phase proper may begin, it is important that the code has been inspected, manually or otherwise, and that unit tests have been run, so that any bug or issue found in this phase may be safely flagged as being a problem of integration and not something that works as it shouldn't at a lower level.

## 2.2 Elements to be integrated

As detailed in the *Design Document*, the architecture is organized in several higher level components themselves divided in lower level modules. Refer to the *Design Document* itself for a detailed listing of the elements.

## 2.3 Integration testing strategy

Since, as stated, the high level components are divided in low level modules, it is at first necessary to test the integration of the modules that make a single component before that same component can be tested against the others at the higher level.

A top down approach will be followed for the integration of the components. The process is very much simplified by the almost complete independence of most of the different components. As for the modules, since their number is rather small, a top down or functional grouping approach can be used indifferently without much impact overall.

## 2.4 Sequence of integration

### 2.4.1 Components

Since a top down (from the interface) approach will be used, the integration order will be as follow:

1. Client App GUI;

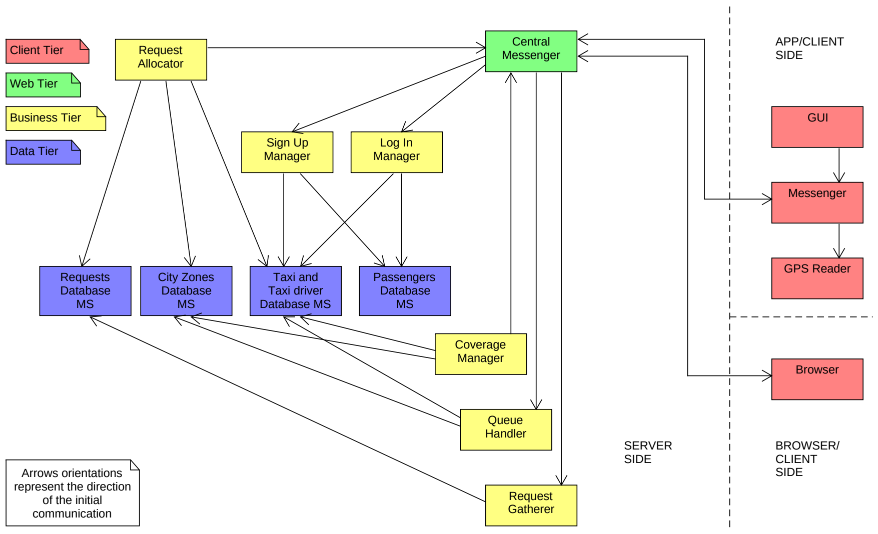

Figure 2.1: The different architectural components and their interaction – see the *Design Document*

2. Client App Messenger;
3. Client App GPS Reader;
4. Client Webapp (on par with the three previous components);
5. Central messenger;
6. Log in manager;
7. Sign up magager;
8. Request allocator;
9. Coverage manager;
10. Queue handler;
11. Request gatherer;
12. All Database Management Systems.

### 2.4.2 Modules

# Chapter 3

# Individual steps and test description

# Chapter 4

# Tools and test equipment required

# Chapter 5

# Program stubs and testing data required

# Appendix A

# Document and work information

## A.1 Revisions

This is the first version of this document. There are currently no revisions.

## A.2 Tools used

**TeXworks editor** With PDFLaTeX, for composing and editing this document.

## A.3 Overall time spent

The authors spent about 20 hours of their time, equally divided among them, working on this document.