POLITECNICO
MILANO 1863

Politecnico di Milano
A.A. 2015/2016
Software Engineering 2

Assignment 4: Integration Test Plan

Version 1.0

Alessandro Baldassari (mat. 841561)
Alberto Bendin (mat. 841734)
Francesco Giarola (mat. 840554)

January 11, 2016

Contents

1 Introduction

1.1 Revision History

This is the first version of the document. There are no previous versions.

Revision	Last Edited	Changes
1.0	xx/01/2016	Document redaction

1.2 Purpose and Scope

The Test Plan Document (ITPD) describes the plan to accomplish the integration test. This document is supposed to be written before the integration test really happens and takes the architectural description of the software system as a starting point, for this reason it is often redacted in parallel with the Design Document. It explain to the development team what to test, in which sequence, which tools are needed for testing (if any), which stubs/drivers/oracles need to be developed.
The purpose of integration testing is to verify functional, performance, and reliability requirements of individual software modules of the product when they are combined and tested as a group; i.e., units (or groups of units) are exercised through their interfaces. The aim is to test the modules interactions incrementally, with success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components interact correctly, for example across procedure calls or process activations.
This is done after testing individual modules, i.e., unit testing; the overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages up to the complete final system (the testing on the complete system is not part of this integration testing phase).

1.3 List of Definitions and Abbreviations

The following acronyms are used in this document:

- RASD: Requirements Analysis and Specification Document
- DD: Design Document

The following definitions are used in this document:

- Driver: are considered dummy modules which are always distinguished as "calling programs that are handled in bottom up integration testing; they are only used when main programs are under construction.
- Stub: in computer science, test stubs are programs that simulate the behaviors of software components (or modules) that a module undergoing tests depends on. Test stubs are mainly used in incremental testing's top-down approach.
- Oracle: in computing, software testers and software engineers can use an oracle as a mechanism for determining whether a test has passed or failed. The use of oracles involves comparing the output(s) of the system under test, for a given test-case input, to the output(s) that the oracle determines that product should have.

1.4 List of Reference Documents

- Specification document: myTaxiService project
- Requirements Analysis and Specification Document (RASD) for myTaxiService
- Design Document (DD) for myTaxiService

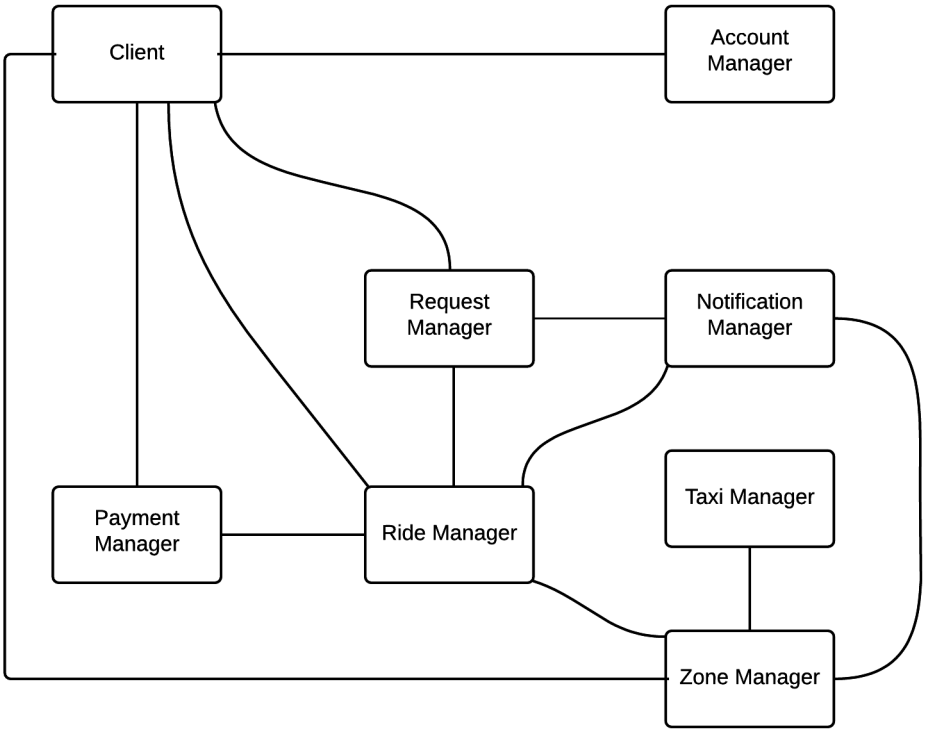
2 Integration Strategy

2.1 Entry Criteria

It is supposed that the unit testing phase has already been completed successfully.

2.2 Elements to be Integrated

The scheme below show the main components of the system.



ID	Integration Test	Paragraphs

2.3 Integration Testing Strategy

The bottom-up approach will be used to accomplish the analysis. This means that first the interactions inside the sub-systems of the components will be considered and consequently the integrations between the components themselves.

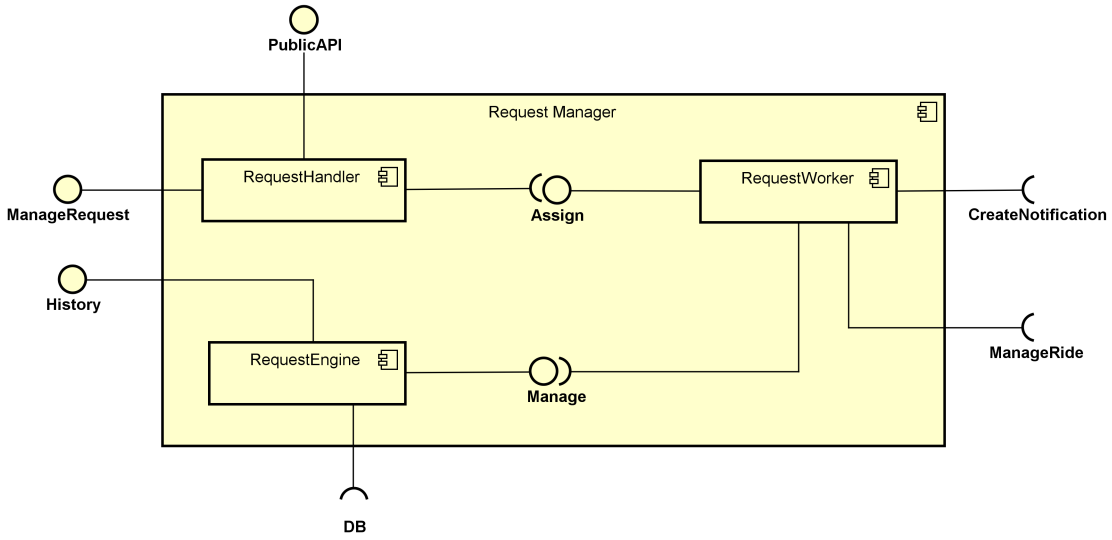
2.4 Sequence of Component/Function Integration

2.4.1 Software Integration Sequence

2.4.2 Subsystem Integration Sequence

3 Individual Steps and Test Description

3.1 Integration test X



Test case identifier	ID X
Test item(s)	Request Handler →Request Worker
Input specification	Request Handler assigns a task to a Request Worker
Output specification	Check if the correct functions are called in the Request Worker
Environmental needs	Client (Passenger) driver to simulate a typical communication input (creation or modification of request/reservation)

Test case identifier	ID X
Test item(s)	Request Engine →Request Worker
Input specification	Request Engine assigns a task to a Request Worker
Output specification	Check if the correct functions are called in the Request Worker
Environmental needs	Client (Passenger) driver to simulate a typical communication input (browsing history of rides)

4 Tools and Test Equipment Required

Mockito, Arquillian, JMeter

5 Program Stubs and Test Data Required

6 References

Material from Wikipedia

- Integration testing: https://en.wikipedia.org/wiki/Integration_testing
- Oracles: https://en.wikipedia.org/wiki/Oracle_(software_testing)

7 Appendix

7.1 Software and tools used

- TeXstudio 2.10.6 (http://www.texstudio.org/) to redact and format this document.
- Astah Professional 7.0 (http://astah.net/editions/professional)

7.2 Hours of work

The time spent to redact this document:

- Baldassari Alessandro: 20 hours.
- Bendin Alberto: 20 hours.
- Giarola Francesco: 20 hours.