Politecnico di Milano
A.A. 2015-2016
Software Engineering 2: “MyTaxi”
Inspection and Test Plan Document

Manzi Giuseppe (mat. 854470) &
Nicolini Alessandro (mat. 858858)

CONTENTS

1. INTRODUCTION 4
1.1 Revision History 4
1.2 Purpose and Scope 4
1.3 List of Reference Document 4

2. INTEGRATION STRATEGY 4
2.1 Entry Criteria 4
2.2 Element to be Integrated 4
2.3 Integration Testing Strategy 5
2.4 Sequence of Component 5
2.4.1 Software Integration Sequence...... 6

3. INDIVIDUAL STEPS AND TEST DESCRIPTION 7
3.1 Integration test case I1 7
3.2 Integration test case I2 7
3.3 Integration test case I3 7
3.4 Integration test case I4 8
3.5 Integration test case I5 8
3.6 Integration test case I6 8
3.7 Integration test case I7 9
3.8 Integration test case I8 9

4. TOOLS AND TEST EQUIPMENT REQUIRED 10

5. PROGRAM STUBS AND TEST DATA REQUIRED 10

1. Introduction

1.1 Revision History

1.2 Purpose and Scope

The Test Plan Document of MyTaxiService describe which test the development team have to do, in which sequence, which tools are used for testing (if any), which stubs/ drivers need to be developed.

1.3 List of Reference Document

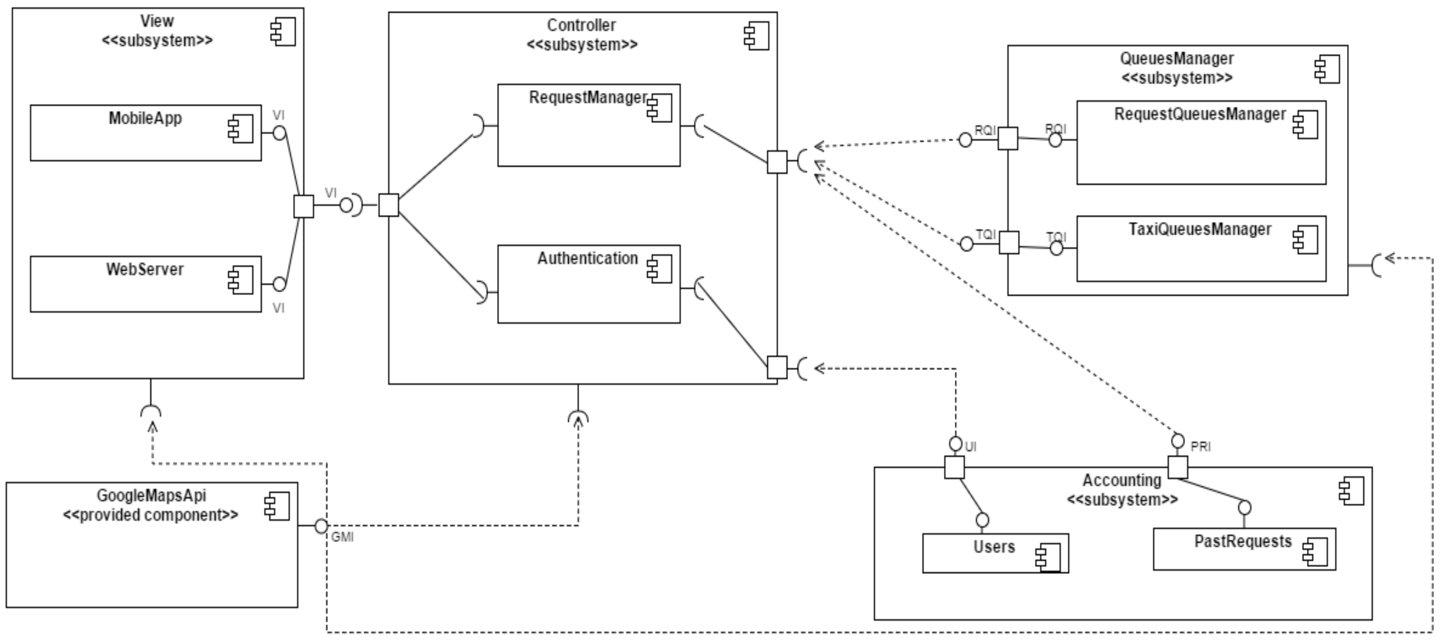
- **"My taxi" RASD** (authors: *Giuseppe Manzi, Alessandro Nicolini*);
- **"My taxi" DD** (authors: *Giuseppe Manzi, Alessandro Nicolini*);
- **Integration test plan assignment** (authors: *Raffaela Mirandola*).

2. Integration Strategy

2.1 Entry Criteria

The first assumption to take is that, at the end of the document, all the Element that we describe in the Design Document must be integrated and tested.
To reach this goal we assume that all the document such as the Design Document and RASD are complete and all the classic Integration Fault are take in consideration before the start of the integration document.

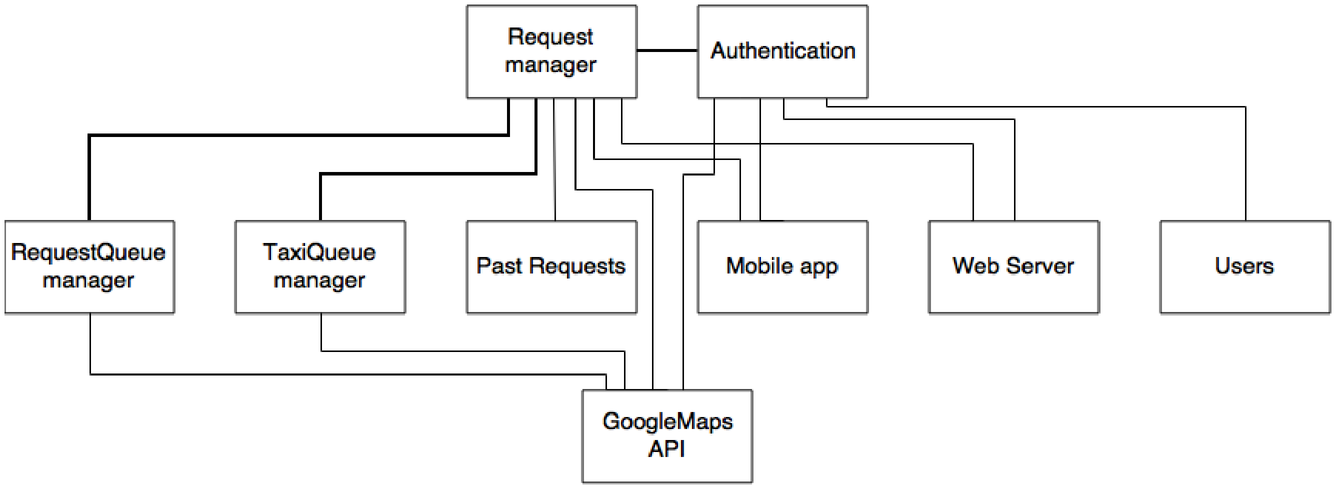
2.2 Element to be Integrated



See the Design Document
The element to be integrated are the View, with the sub component MobileApp and the Web Server, the Controller, with the sub-component Request Manager and the Authentication, the Queue Manager(Taxi and request Queue Manager) and the Accounting of User and PastRequest.

2.3 Integration Testing Strategy

We decided to use a **TOP-DOWN** strategy giving priority to risky modules. We applied this pattern, because we think that it is better to test problematic interactions first, in order to find big problems as soon as possible. In our system the hardest interactions are the ones between components of the controller subsystem and the ones between the Controller and the Queue Manager. TOP-DOWN integration starts from the components that has the highest number of "use" or "include" relation, so in our case from the Controller's ones. So applying TOP-DOWN strategy the riskiest interactions will be tested first.
The following diagram shows the levels of interactions. The hardest ones are highlighted using thicker lines:



2.4 Sequence of Component

Related to the section 2.3 where we had choose the bottom up strategy for the integration of the component.

2.4.1 Software Integration Sequence

Diagram:

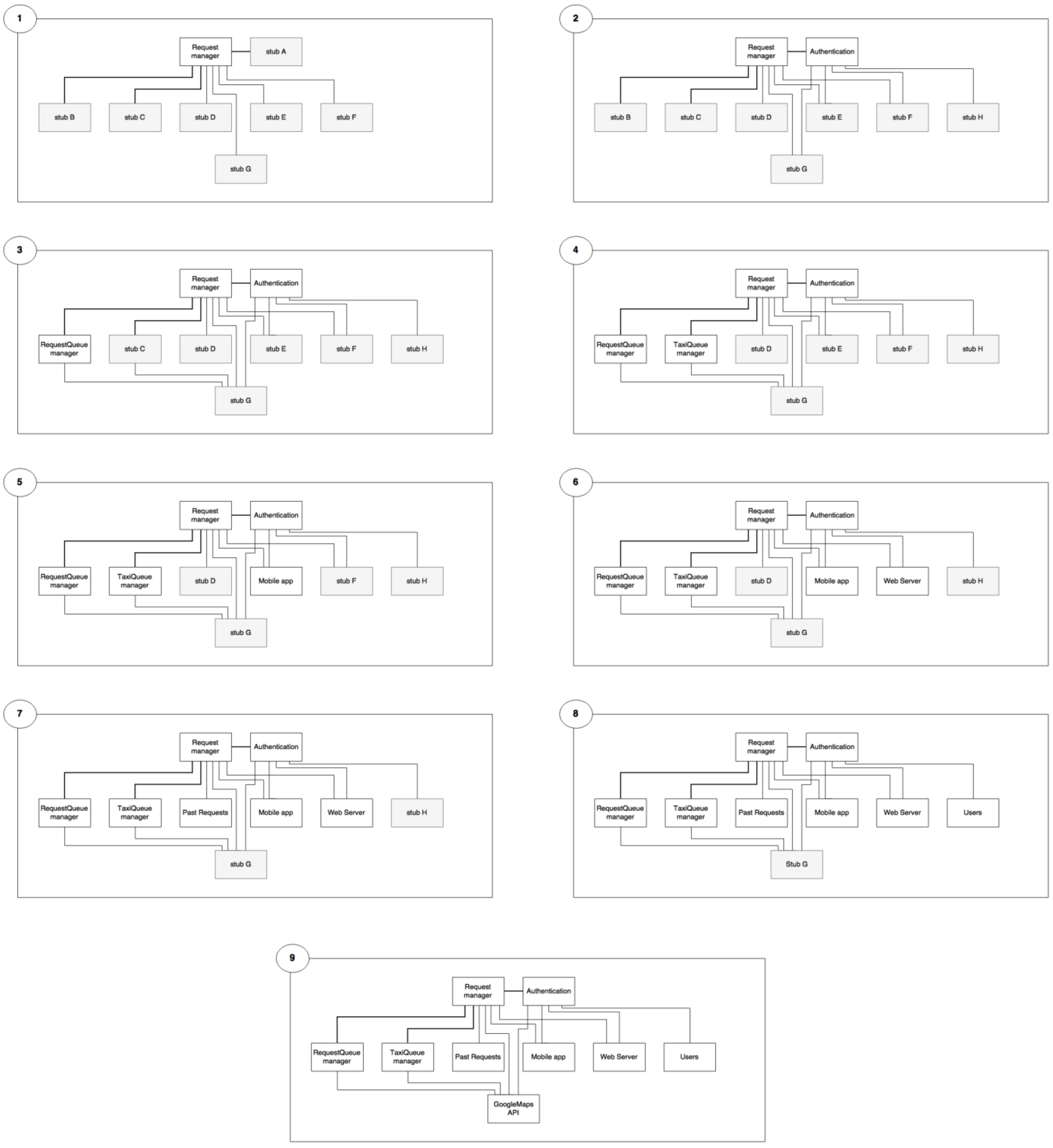


Table:

ID	Integration test	Paragraphs
I1	RequestManager → Authentication	3.1
I2	RequestManager → RequestQueueManager	3.2
I3	RequestManager → TaxiQueueManager	3.3
I4	RequestManager, Authentication → MobileApp	3.4
I5	RequestManager, Authentication → WebServer	3.5
I6	RequestManager → PastRequest	3.6
I7	Authentication → Users	3.7
I8	RequestManager, Authentication, RequestQueueManager, TaxiQueueManager → GoogleMapsAPI	3.8

3. Individual Steps and Test Description

3.1 Integration test case I1

Test Case Identifier	I1T1
Test Item(s)	RequestManager → Authentication
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in the Authentication
Environmental Needs	Mobile app, WebServer, GoogleMapsApi, Users, PastRequest, RequestQueueManager, TaxiQueuesManager Stubs

3.2 Integration test case I2

Test Case Identifier	I2T1
Test Item(s)	RequestManager → RequestQueueManager
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in the RequestQueueManager
Environmental Needs	I1 Succeeded and Mobile app, WebServer, GoogleMapsApi, Users, PastRequest, TaxiQueuesManager Stubs

3.3 Integration test case I3

Test Case Identifier	I3T1
Test Item(s)	RequestManager → TaxiQueueManager
Input Specification	Create typical RequestManager input

Output Specification	Check if the correct methods are called in the TaxiQueueManager
Environmental Needs	I2 Succeeded and Mobile app, WebServer, GoogleMapsApi, Users, PastRequest Stubs

3.4 Integration test case I4

Test Case Identifier	I4T1
Test Item(s)	RequestManager → MobileApp
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in the MobileApp
Environmental Needs	I3 Succeeded and WebServer, GoogleMapsApi, Users, PastRequest Stubs

Test Case Identifier	I4T2
Test Item(s)	Authentication → MobileApp
Input Specification	Create typical Authentication input
Output Specification	Check if the correct methods are called in the MobileApp
Environmental Needs	I1 Succeeded and WebServer, GoogleMapsApi, Users, PastRequest Stubs

3.5 Integration test case I5

Test Case Identifier	I5T1
Test Item(s)	RequestManager → WebServer
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in the WebServer
Environmental Needs	I3 Succeeded and WebServer, GoogleMapsApi, Users, PastRequest Stubs

Test Case Identifier	I5T2
Test Item(s)	Authentication → WebServer
Input Specification	Create typical Authentication input
Output Specification	Check if the correct methods are called in the WebServer
Environmental Needs	I1 Succeeded and GoogleMapsApi, Users, PastRequest Stubs

3.6 Integration test case I6

Test Case Identifier	I6T1
Test Item(s)	RequestManager → PastRequest
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in PastRequests
Environmental Needs	Users and GoogleMapsAPI stubs and I5 succeeded

3.7 Integration test case I7

Test Case Identifier	I7T1
Test Item(s)	Authentication → Users
Input Specification	Create typical Authentication input
Output Specification	Check if the correct methods are called in the Users
Environmental Needs	GoogleMapsAPI stub and I6 succeeded

3.8 Integration test case I8

Test Case Identifier	I8T1
Test Item(s)	RequestManager → GoogleMapsAPI
Input Specification	Create typical RequestManager input
Output Specification	Check if the correct methods are called in the GoogleMapsAPI
Environmental Needs	I7 succeeded

Test Case Identifier	I8T2
Test Item(s)	Authentication → GoogleMapsAPI
Input Specification	Create typical Authentication input
Output Specification	Check if the correct methods are called by the GoogleMapsAPI
Environmental Needs	I7 succeeded

Test Case Identifier	I8T3
Test Item(s)	RequestQueueManager → GoogleMapsAPI
Input Specification	Create typical RequestQueueManager input
Output Specification	Check if the correct methods are called by the GoogleMapsAPI
Environmental Needs	I7 succeeded

Test Case Identifier	I8T1
Test Item(s)	TaxiQueueManager → GoogleMapsAPI
Input Specification	Create typical TaxiQueueManager input
Output Specification	Check if the correct methods are called by the GoogleMapsAPI
Environmental Needs	I7 succeeded

4. Tools and Test Equipment Required

arquillian

A tool used to execute test cases against the container, good for application.
We can use this tool, for example, to take the test for the interact with the Database:

RequestManager -> PastRequest
Authentication-> Users

Apache JMeter

A load testing tool for analysing and measuring performance
Perfect for both WebApplication than App, It can be used to simulate a heavy load on a server, network or object to test its strength or to analyse overall performance under different load types.
For our Project, JMeter can set up test plans that simulate logging into our web site, filling out forms, clicking buttons and links.

5. Program Stubs and Test Data Required

- **Stub A:** it simulates Authentication behaviour.
- **Stub B:** it simulates RequestQueueManager behaviour.
- **Stub C:** it simulates TaxiQueueManager behaviour.
- **Stub D:** it simulates PastRequests behaviour.
- **Stub E:** it simulates Mobile app behaviour.
- **Stub F:** it simulates Web server behaviour.
- **Stub G:** it simulates Users behaviour.
- **Stub H:** it simulates GoogleMapsAPI behaviour.