­



Politecnico di Milano

A.A. 2015-2016

Software Engineering 2: “MyTaxi”

Code inspection

Manzi Giuseppe (mat. 854470) &

Nicolini Alessandro (mat. 858858)

CONTENTS

[1. Introduction 4](#_Toc440293362)

[1.1 Revision History 4](#_Toc440293363)

[1.2 Purpose and Scope 4](#_Toc440293364)

[1.3 List of definitions and Abbreviations 4](#_Toc440293365)

[1.4 List of Reference Document 4](#_Toc440293366)

[2. Integration Strategy 4](#_Toc440293367)

[2.1 Entry Criteria 4](#_Toc440293368)

[2.2 Element to be Integrated 4](#_Toc440293369)

[2.3 Integration Testing Strategy 4](#_Toc440293370)

[2.4 Sequence of Component 4](#_Toc440293371)

[2.4.1 Software Integration Sequence 4](#_Toc440293372)

[2.4.2 Subsystem Integration Sequence 4](#_Toc440293373)

[3. Individual Steps and Test Description 4](#_Toc440293374)

[4. Tools and Test Equipment Required 4](#_Toc440293375)

[5. Program Stubs and Test Data Required 4](#_Toc440293376)

# 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 definitions and Abbreviations

## 1.4 List of Reference Document

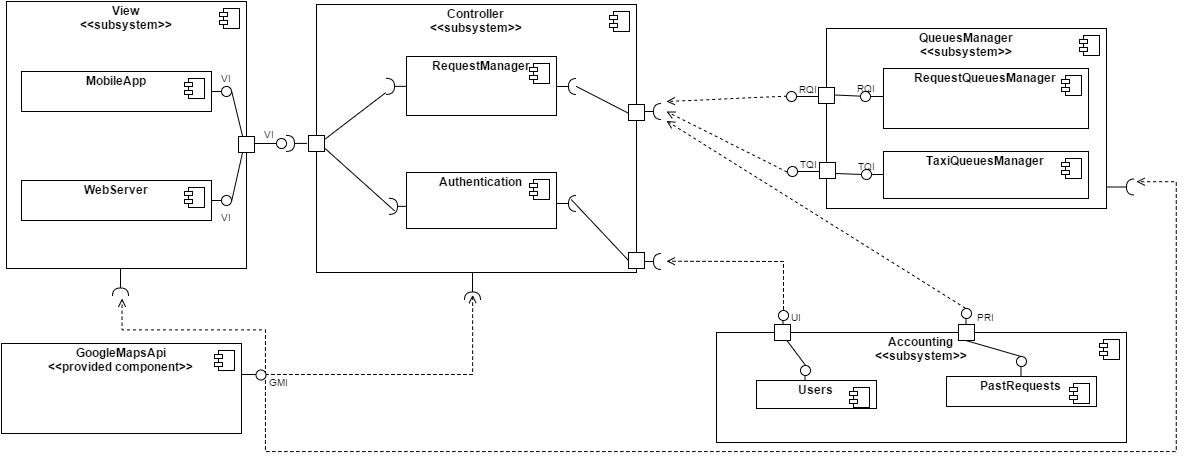
# 2. Integration Strategy

## 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 think that a **BOTTOM UP** strategy is the best in our case.

This means that integration testing starts at the bottom level.

All low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing.

This method helps to determine the levels of software developed and makes it easier to report testing progress in time.

## 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

# 3. Individual Steps and Test Description

# 4. Tools and Test Equipment Required

# 5. Program Stubs and Test Data Required