Politecnico di Milano
Scuola di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Computer Science and Engineering
A.Y. 2015-2016

Software Engineering 2 Project
"myTaxiService"
Integration **T**est **P**lan

January 16, 2016

Principal Adviser: Prof. **Di Nitto**

Authors:
Davide Fisicaro 854043
Gianmarco Giummarra 852667
Salvatore Ferrigno 850130

Contents

1 Introduction

1.1 Revision History

Record all revisions to the document.

1.2 Purpose and Scope

The aim of this document is to provide an integration test plan.

The integration test detects the bugs not discovered during unit tests, focusing the attention on a group of components to be tested.

Two or more components are integrated and analyzed, and when bugs are detected more components can be added to fix the errors.

In more details, interfaces and interactions between all the components of the myTaxiService system will be tested following the approaches that will be described below.

The scope can be found in the RASD, section 1.3.

1.3 List of Definitions, Acronyms and Abbreviations

- **RASD**: Requirement Analysis and Specification Document;

1.4 List of Reference Documents

List all reference documents, for instance:

- The RASD
- The Design document
- ******The documentation of any tool you plan to use for testing******

2 Integration Strategy

2.1 Entry Criteria

Some criteria that have to be met before starting with the integration testing are listed below:

- All of the atomic functions have to be unit-tested
- Every integration between components have to be defined
- Every needed interface have to be defined

2.2 Elements to be Integrated

The elements to be integrated are, according to the strategy explained in the section 2.3 of this document, the single components of the system, as described in the Design Document.

To give a best description of these “atomic” elements of the system, here is a list:

- Authentication manager
- Customer profile manager
- Immediate call manager
- Reservation call manager
- Taxi driver profile manager
- Incoming call manager
- Assistance call manager
- Data component
- Payment manager
- Notification manager
- Guest component
- Customer component
- Taxi driver component

2.3 Integration Testing Strategy

The chosen integration testing approach for this Integration Plan is the bottom-up approach. The reason of this choice is that by following this strategy it's possible to begin testing the system from the little components that compose every single part and continue the integration at a higher level for each step. This justifies also the presence of the first entry criterium in the point 2.1 of this document; in the rare case that some functions or some part of code at the moment of the integration testing is not developed yet, it should be possible to build a Stub in order to go on with the testing of the whole single functions at the low level. For the same reason, in the integration testing for different components, should be possible to build Drivers that allow to testing the integration when an interface is not completely developed yet.

2.4 Sequence of Component/Function Integration

NOTE: The structure of this section may vary depending on the integration strategy you select in Section 2.3. Use the structure proposed below as a non mandatory guide.

2.4.1 Software Integration Sequence

The sub-systems below described are composed by the components described in the section 2.2 of this document.

Guest Component sub-system
This is the sub-system which represents the Guest Component environment.

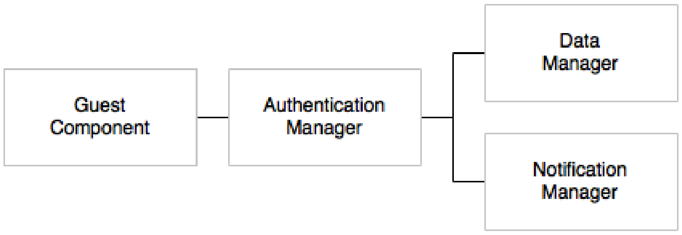

Figure 2.1: Integration sequence for the Guest Component environment

Customer Component sub-system

This is the sub-system which represents the Customer Component environment.

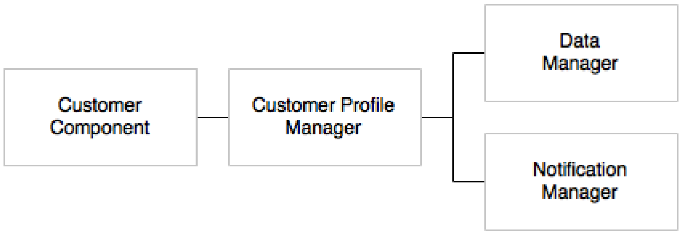

Figure 2.2: Integration sequence for the Customer Profile management

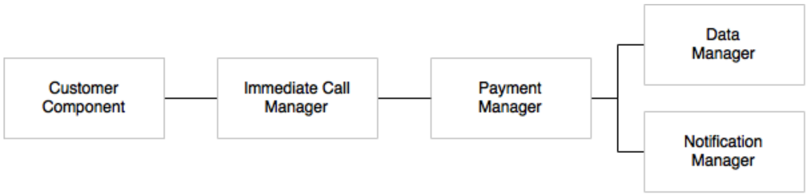

Figure 2.3: Integration sequence for the Immediate Call by a Customer

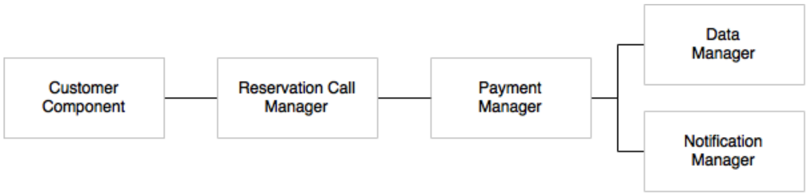

Figure 2.4: Integration sequence for the Reservation Call by a Customer

Taxi Driver Component sub-system

This is the sub-system which represents the Taxi Driver Component environment.

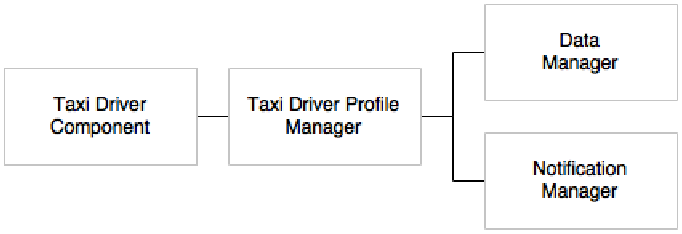

Figure 2.5: Integration sequence for the Profile management

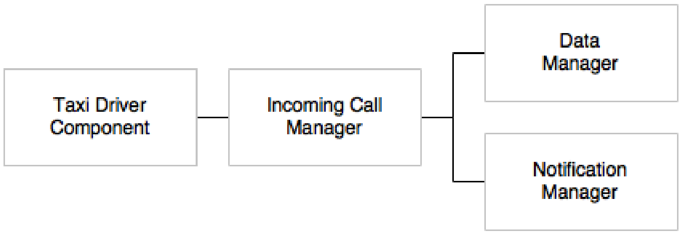

Figure 2.6: Integration sequence for the Guest Component environment

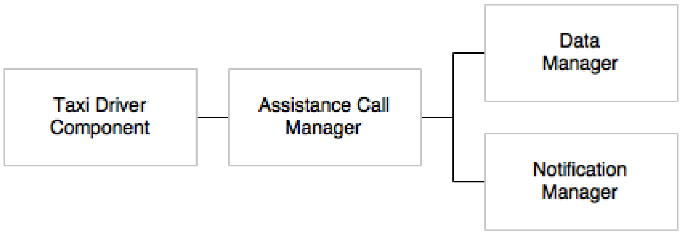

Figure 2.7: Integration sequence for the Guest Component environment

2.4.2 Subsystem Integration Sequence

Identify the order in which subsystems will be integrated. If you have a single subsystem, 2.4.1 and 2.4.2 are to be merged in a single section. You can refer to Section 2.2 of the test plan example [1] as an example of what we expect.

3 Individual Steps and Test Description

For each step of the integration process identified above, describe the type of tests that will be used to verify that the elements integrated in this step perform as expected. Describe in general the expected results of the test set. You may refer to Chapter 3 and Chapter 4 of the test plan example [1] as an example of what we expect. (NOTE: This is not a detailed description of test protocols. Think of this as the test design phase. Specific protocols will be written to fulfill the goals of the tests identified in this section.)

4 Tools and Test Equipments Required

Identify all tools and test equipment needed to accomplish the integration. Refer to the tools presented during the lectures. Explain why and how you are going to use them. Note that you may also use manual testing for some part. Consider manual testing as one of the possible tools you have available.

5 Program Stubs and Test Data Required

Based on the testing strategy and test design, identify any program stubs or special test data required for each integration step.