POLITECNICO
MILANO 1863

Politecnico di Milano
A.A. 2016/2017
Software Engineering 2: "*PowerEnJoy*" v. 1.0

Integration **T**est **P**lan **D**ocument

Matteo Bresich (mat. 774366)

January 15, 2017

Table of Content

1 Introduction

1.1 Revision History

Revision	Changes
1.0	First version of the document.

1.2 Purpose and Scope

The Integrated Test Planning Document describes the plan to accomplish the integration test. This document should be used as a reference of the testing methodologies used by the developer team. 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.

1.3 List of Definitions and Abbreviations

1.3.1 Definitions

- ITn: it is the identifier of the integration test.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- DD: Design Document.
- MOM: Message Oriented Middleware.

1.4 List of Reference Documents

Revision	Document Name	File Name
-	*PowerEnJoy* Project Description	Assignments AA 2016-2017.pdf
1.1	RASD	RASD.pdf
1.0	DD	DD.pdf

2 Integration Strategy

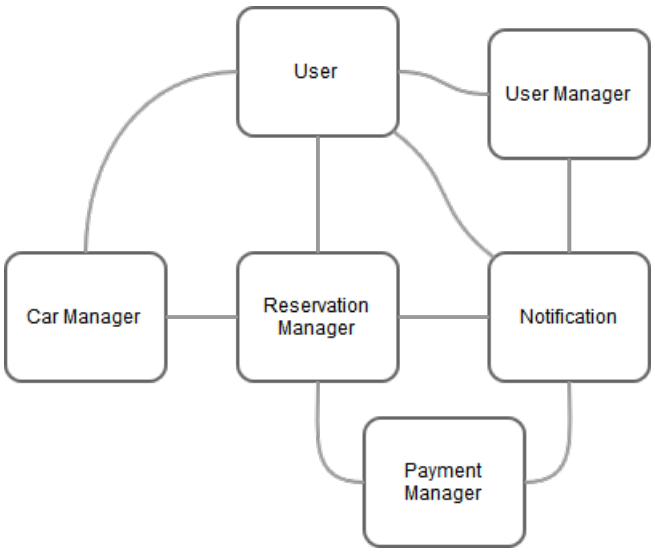
2.1 Entry Criteria

This section describes the prerequisites that need to be met before integration testing can be started. It is supposed to proceed after a successfully testing phase with a minimum coverage of 70% of the lines of code. In particular the following components are considered already tested and thus trusted as working:

- Notification
- MOM
- Google Maps APIs
- GPS APIs

2.2 Elements to be Integrated

Follows a scheme that shows the main high level components of the system.



Test ID	Integration Test	Paragraphs		
IT1	Login →User Factory	2.4.1	3.1.1	3.2.1
IT2	Profile Manager →Login	2.4.1	3.1.1	3.2.1
IT3	Reservation Handler →Reservation Factory	2.4.2	3.1.2	3.2.2
IT4	Reservation Handler →Reservation	2.4.2	3.1.2	3.2.2
IT5	User →User Manager	2.4.3	3.1.3	
IT6	User →Car Manager	2.4.3	3.1.4	
IT7	User →Reservation Manager	2.4.3	3.1.5	
IT8	User →Notification	2.4.3	3.1.6	
IT9	User Manager →Notification	2.4.4	3.1.7	
IT10	Payment Manager →Notification	2.4.5	3.1.8	
IT11	Reservation Manager →Payment Manager	2.4.6	3.1.9	
IT12	Reservation Manager →Car Manager	2.4.6	3.1.10	
IT13	Reservation Manager →Notification	2.4.6	3.1.11	

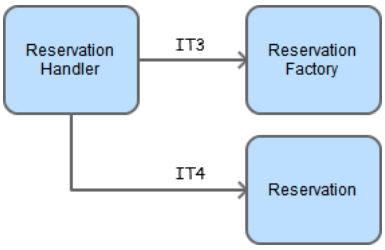
2.3 Integration Testing Strategy

To accomplish the analysis will be used an hybrid approach. It consists in an integration of sub-systems of the high level components followed by the integration between the high level components themselves. The choice is motivated by the fact that the development and testing in this way follows the functionality to be realized. The order of the elements to be developed is dictated by the importance of the functionality they offer.

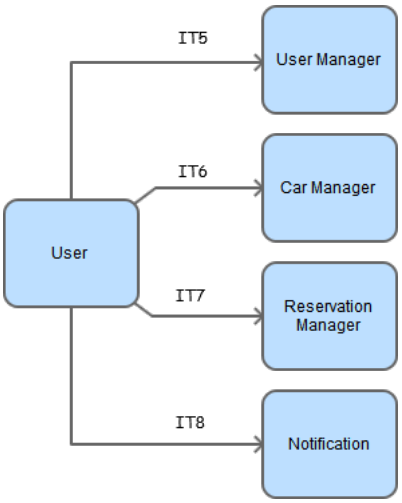
2.4 Sequence of Component/Function Integration

2.4.1 Sequence of integration for User Manager sub-system

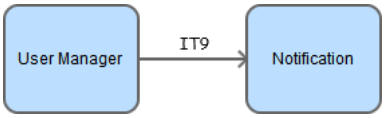
2.4.2 Sequence of integration for Reservation Manager sub-system



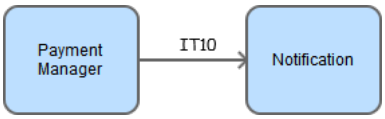
2.4.3 Sequence of integration for User



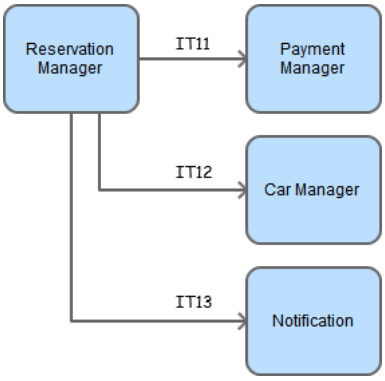
2.4.4 Sequence of integration for User Manager



2.4.5 Sequence of integration for Payment Manager



2.4.6 Sequence of integration for Reservation Manager



3 Individual Steps and Test Description

3.1 Integration Test Case Specifications

3.1.1 Integration Test for User Manager sub-system

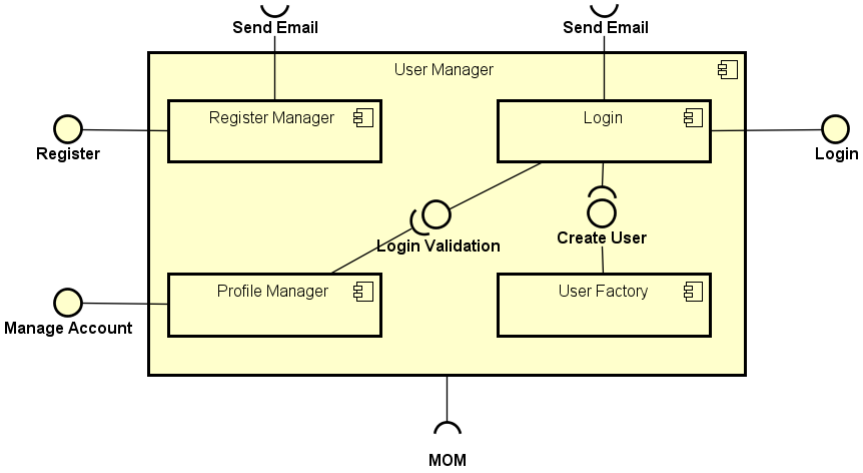


Image taken from the Design Document (paragraph 2.3.1)

Test case identifier	IT1
Test items	Login →User Factory
Input specification	Methods call from Login to User Factory with typical input
Output specification	Check the correctness of User Factory's methods
Environmental needs	User Driver, MOM and Notification Stub

Test case identifier	IT2
Test items	Profile Manager →Login
Input specification	Methods call from Profile Manager to Login with typical input
Output specification	Check the correctness of Login's methods
Environmental needs	User Driver, MOM and Notification Stub

3.1.2 Integration Test for Reservation Manager sub-system

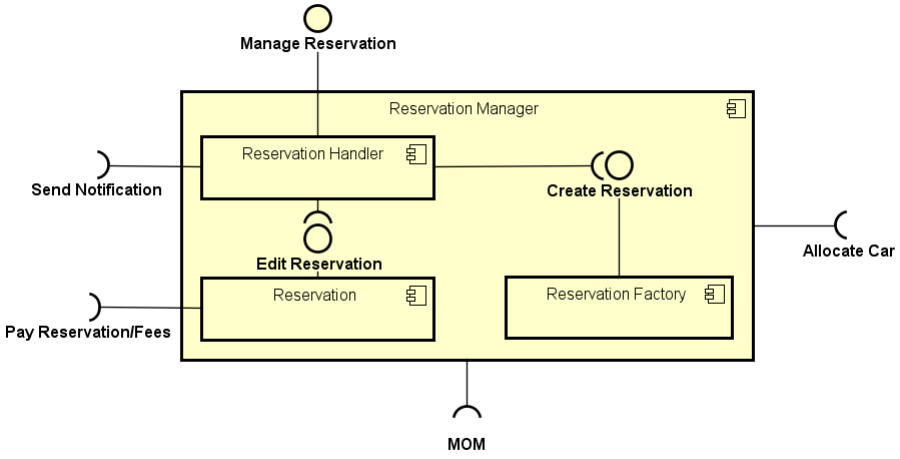
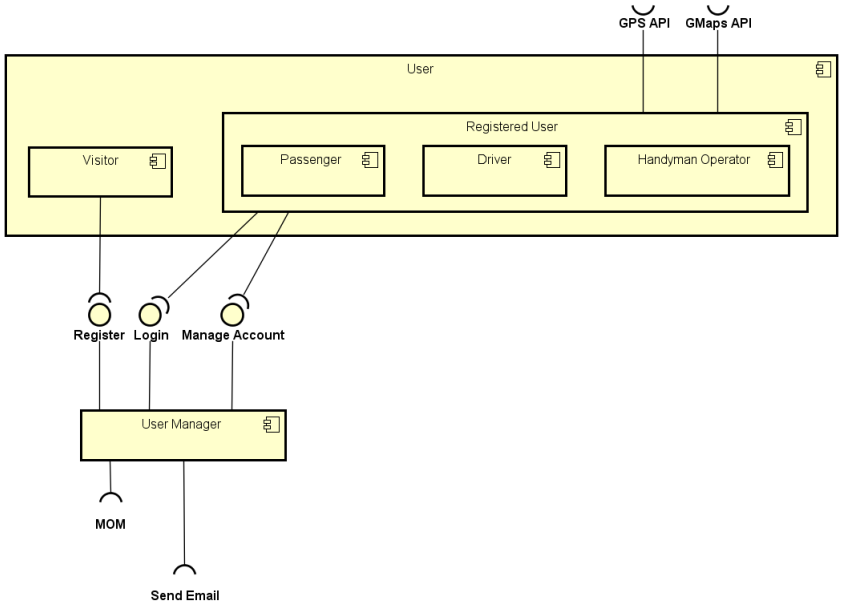


Image taken from the Design Document (paragraph 2.3.3)

Test case identifier	IT3
Test items	Reservation Handler →Reservation Factory
Input specification	Methods call from Reservation Handler to Reservation Factory with typical input
Output specification	Check the correctness of Reservation Factory's methods
Environmental needs	User Driver, MOM and Car Manager Stub

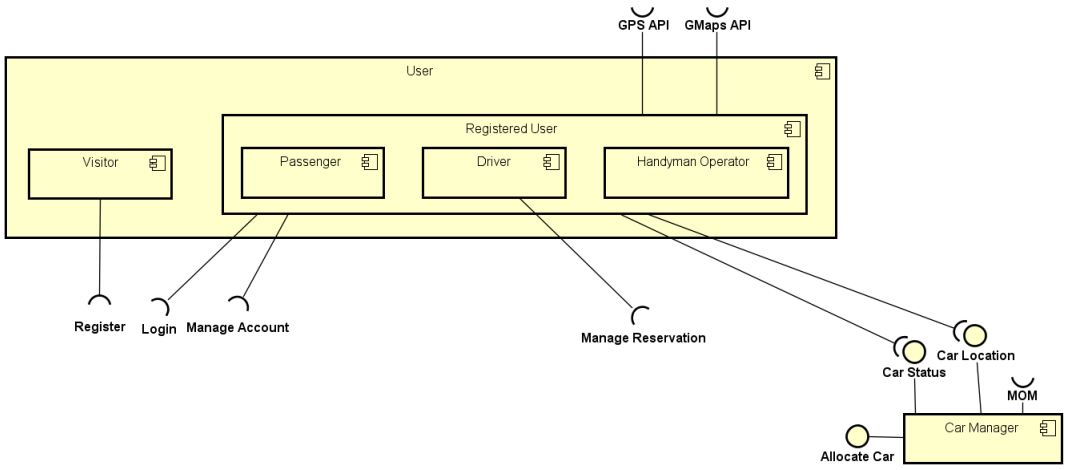
Test case identifier	IT4
Test items	Reservation Handler →Reservation
Input specification	Methods call from Reservation Handler to Reservation with typical input
Output specification	Check the correctness of Reservation's methods
Environmental needs	User Driver, MOM and Car Manager Stub

3.1.3 Integration Test for User and User Manager



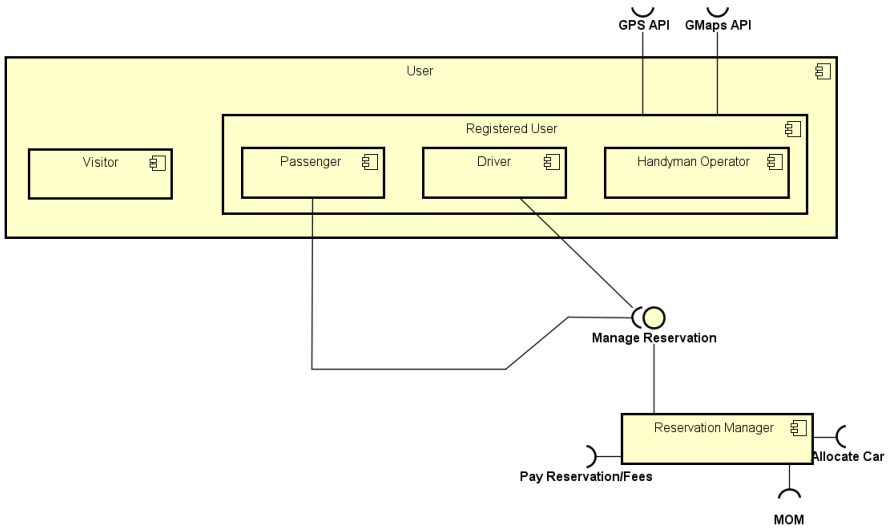
Test case identifier	IT5
Test items	User →User Manager
Input specification	Methods call from User to User Manager with typical input
Output specification	Check the correctness of User Manager's methods
Environmental needs	IT1 and IT2 passed, User Driver and MOM

3.1.4 Integration Test for User and Car Manager



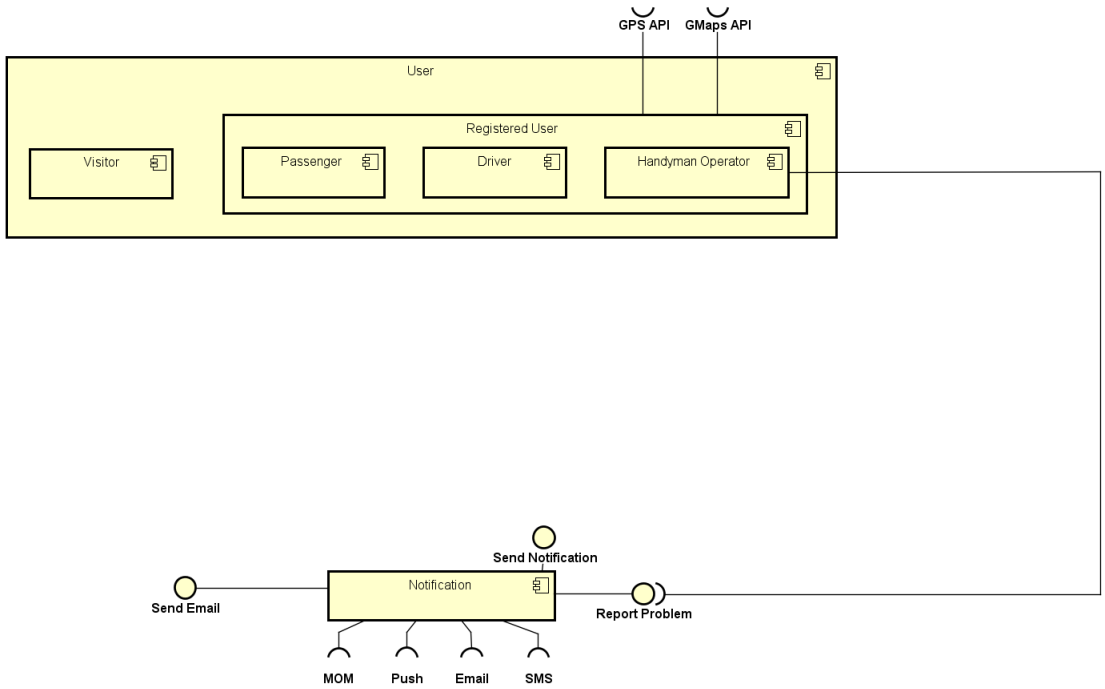
Test case identifier	IT6
Test items	User →Car Manager
Input specification	Methods call from User to Car Manager with typical input
Output specification	Check the correctness of Car Manager's methods
Environmental needs	User Driver and MOM

3.1.5 Integration Test for User and Reservation Manager



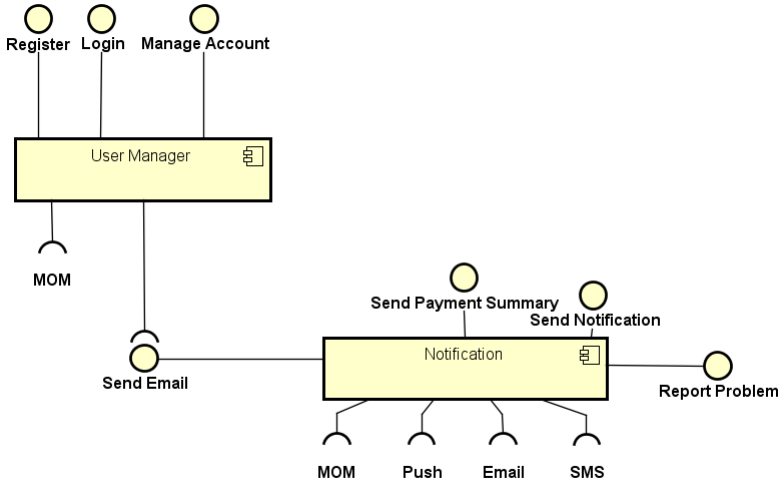
Test case identifier	IT7
Test items	User →Reservation Manager
Input specification	Methods call from User to Reservation Manager with typical input
Output specification	Check the correctness of Reservation Manager's methods
Environmental needs	IT3 and IT4 passed, User Driver, Payment Manager Stub and MOM

3.1.6 Integration Test for User and Notification



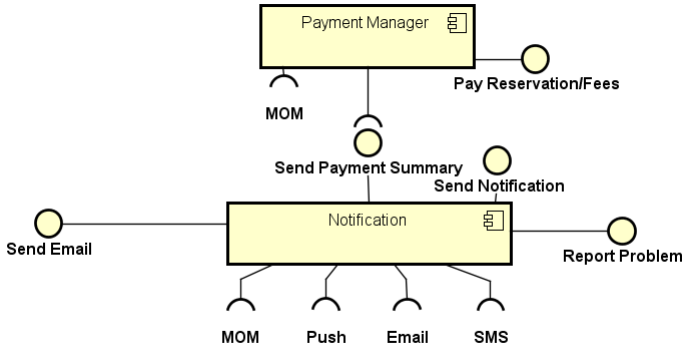
Test case identifier	IT8
Test items	User →Notification
Input specification	Methods call from User to Notification with typical input
Output specification	Check the correctness of Notification's methods
Environmental needs	User Driver and MOM, Email, SMS and Push notification stubs

3.1.7 Integration Test for User Manager and Notification



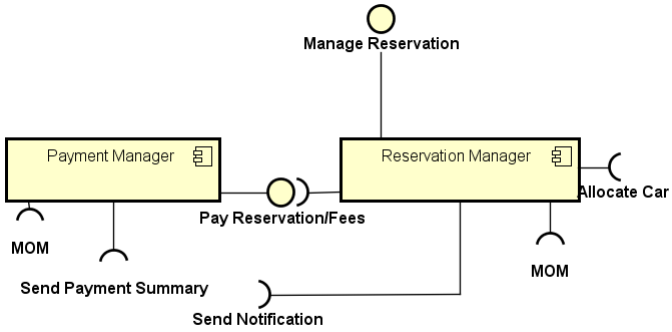
Test case identifier	IT9
Test items	User Manager →Notification
Input specification	Methods call from User Manager to Notification with typical input
Output specification	Check the correctness of Notification's methods
Environmental needs	IT1 and IT2 passed, User Driver and MOM, Email, SMS and Push notification stubs

3.1.8 Integration Test for Payment Manager and Notification



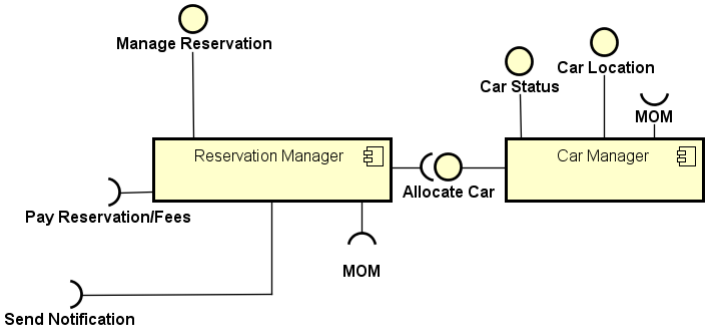
Test case identifier	IT10
Test items	Payment Manager →Notification
Input specification	Methods call from Payment Manager to Notification with typical input
Output specification	Check the correctness of Notification's methods
Environmental needs	Reservation Manager Driver, MOM, Email, SMS and Push notification stubs

3.1.9 Integration Test for Reservation Manager and Payment Manager



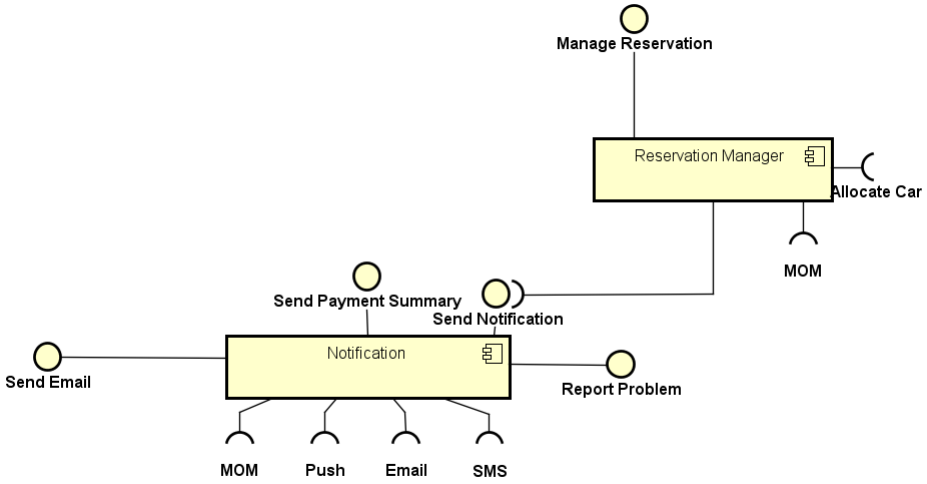
Test case identifier	IT11
Test items	Reservation Manager →Payment Manager
Input specification	Methods call from Reservation Manager to Payment Manager with typical input
Output specification	Check the correctness of Payment Manager's methods
Environmental needs	IT3 and IT4 passed, MOM, Car Manager and Notification Stubs

3.1.10 Integration Test for Reservation Manager and Car Manager



Test case identifier	IT12
Test items	Reservation Manager →Car Manager
Input specification	Methods call from Reservation Manager to Car Manager with typical input
Output specification	Check the correctness of Car Manager's methods
Environmental needs	IT3 and IT4 passed, MOM, Payment Manager and Notification Stubs

3.1.11 Integration Test for Reservation Manager and Notification



Test case identifier	IT13
Test items	Reservation Manager →Notification
Input specification	Methods call from Reservation Manager to Notification with typical input
Output specification	Check the correctness of Notification's methods
Environmental needs	User Driver, MOM, Email, SMS and Push notification stubs

3.2 Test procedures

3.2.1 Test procedure for User Manager sub-system

Test procedure identifier	TP1
Purpose	This test verifies that the User Manager component: • Can handle user input • Can output the requested information to a user • Can modify informations about a user • Can retrieve the correct information about a user
Procedure steps	Execute in order IT1 and IT2

3.2.2 Test procedure for Reservation Manager sub-system

Test procedure identifier	TP2
Purpose	This test verifies that the Reservation Manager component: • Can handle the creation, modification and cancellation of rides • Can handle the creation, modification and cancellation of seats reservations • Can handle payment process • Can retrieve information about the rides • Can send notifications to users
Procedure steps	Execute in order IT3 and IT4

3.2.3 Test procedure for components involved in user login

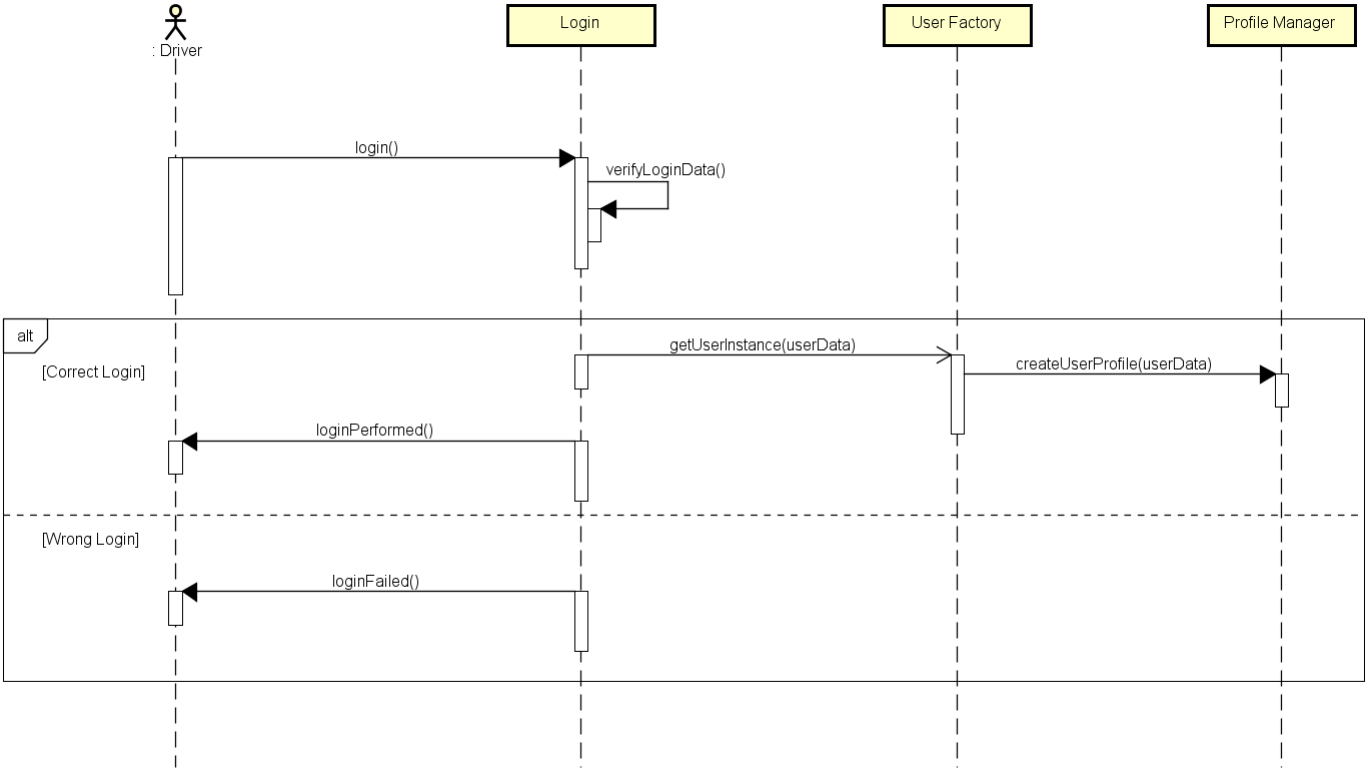


Image taken from the Design Document (paragraph 2.5.2.1)

Test procedure identifier	TP3
Purpose	This test verifies the login procedure. The purpose of the procedure is to check the correct integration of all the involved components: Login, User Factory and Profile Manager.
Procedure steps	IT1 followed by IT2

3.2.4 Test procedure for components involved in car/seat reservation

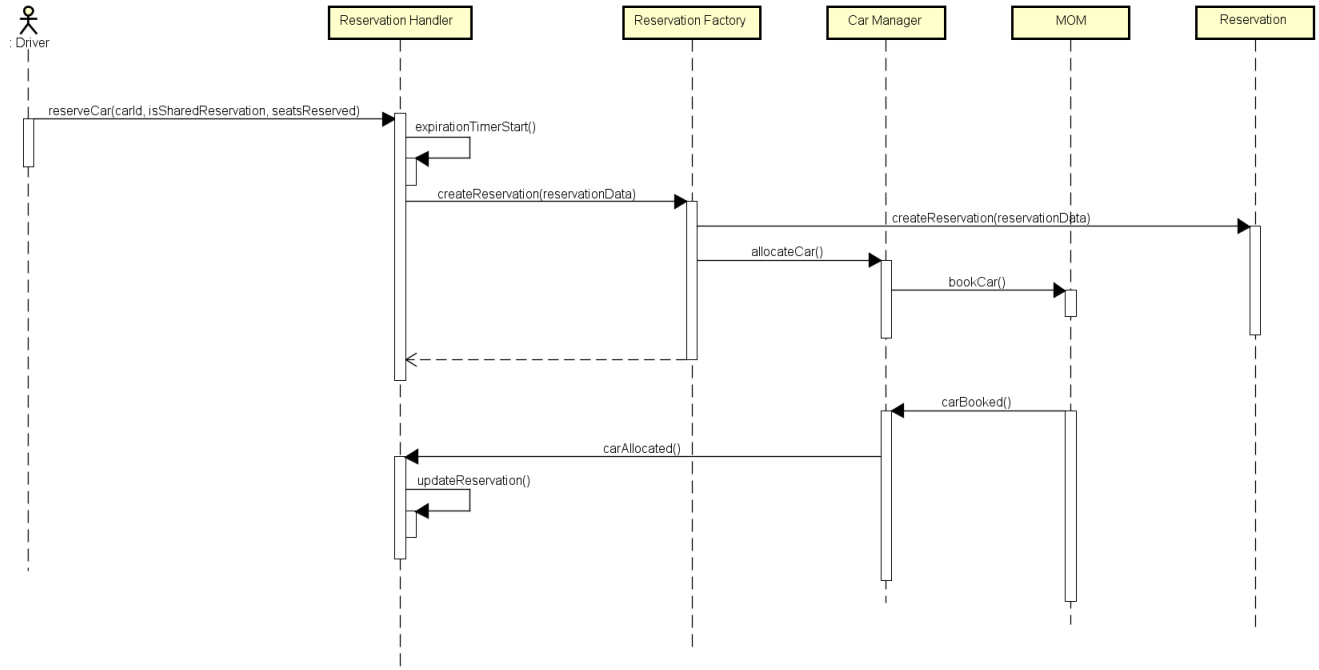


Image taken from the Design Document (paragraph 2.5.2.2)

Test procedure identifier	TP4
Purpose	This test verifies the car reservation procedure. The purpose of the procedure is to check the correct integration of all the involved components: Reservation Handler, Reservation Factory, Car Manager, MOM and Reservation.
Procedure steps	Execute in order IT3, IT4, IT12

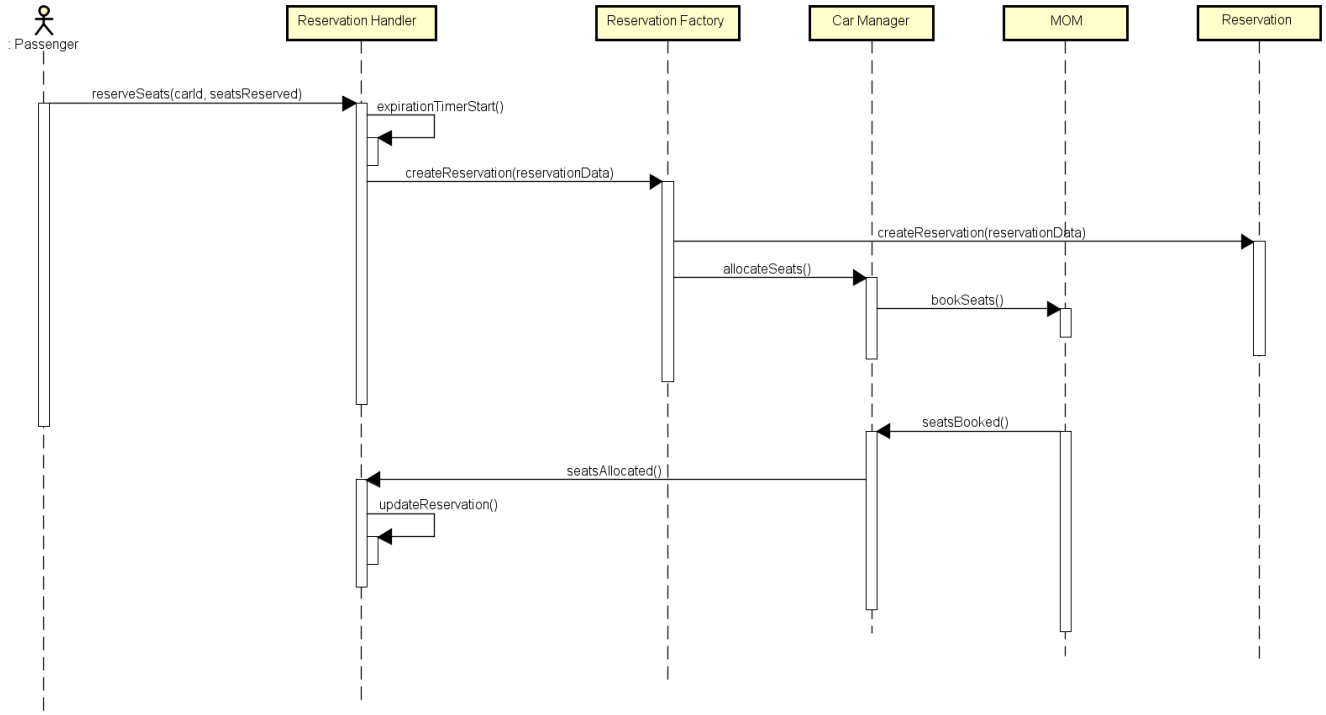


Image taken from the Design Document (paragraph 2.5.2.3)

Test procedure identifier	TP5
Purpose	This test verifies the seat reservation procedure. The purpose of the procedure is to check the correct integration of all the involved components: Reservation Handler, Reservation Factory, Car Manager, MOM and Reservation.
Procedure steps	Execute in order IT3, IT4, IT12

3.2.5 Test procedure for components involved in delete a car reservation

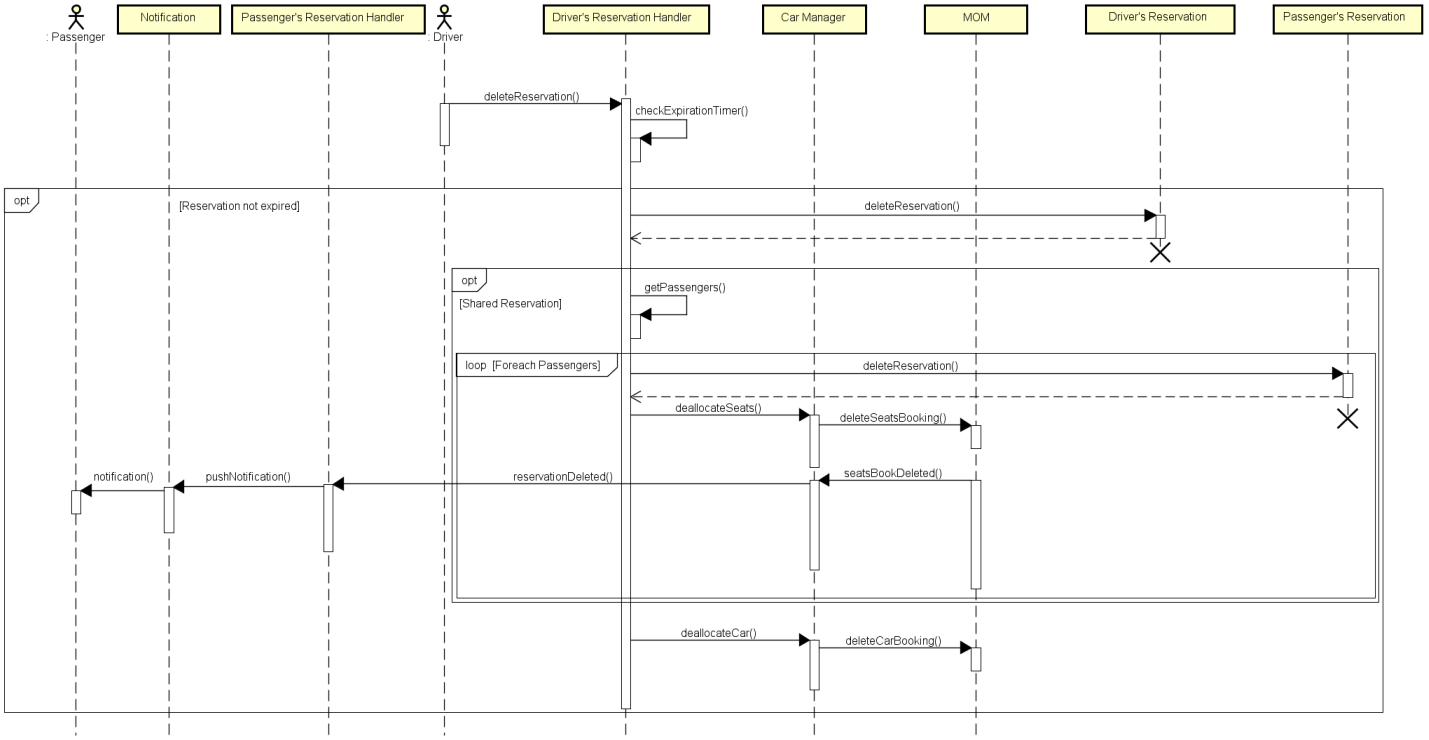


Image taken from the Design Document (paragraph 2.5.2.4)

Test procedure identifier	TP6
Purpose	This test verifies the delete a car reservation procedure. The purpose of the procedure is to check the correct integration of all the involved components: Reservation Handler, Car Manager, MOM, Reservation and Notification.
Procedure steps	Execute in order IT3, IT4, IT12 and IT13

4 Tools and Test Equipment Required

To perform integration tests will be used the following software tools:

- Junit: for unit tests of the single components.
- Arquillian: for tests containers and their integration with JavaBeans.
- Mockito: for generate mock objects, stubs and drivers.

5 Program Stubs and Test Data Required

In order to fulfill integration tests during the development, it is necessary to use stubs and drivers that will temporarily replace software components that do not yet exist.

5.1 Stubs

- Payment Manager
- Notification
- Car Manager
- Payment service
- SMS service
- Email service
- Push notification service
- MOM

5.2 Drivers

- User (Passenger, Driver, Handyman Operator)
- Reservation Manager
- MOM

5.3 Data required

To perform a correct integration testing it is necessary to implement a dummy DB that will contains a reduced set of instances and simulate MOM messages.

6 Effort Spent

6.1 Hours of work

The time spent to redact this document:

- Bresich Matteo: 36 hours.

Days	Hours of work
02/01/17	6h
03/01/17	7h
04/01/17	7h
10/01/17	4h
11/01/17	4h
12/01/17	3h
13/01/17	2h
14/01/17	2h
15/01/17	1h

7 References

- TeXstudio v2.11.2 (http://www.texstudio.org/) to produce this document.
- Evolus Pencil v2.0.5 (http://pencil.evolus.vn/) to generate diagrams.
- Astah Professional 7.1.0 (http://astah.net/) to create Use Cases Diagrams, Sequence Diagrams, Class Diagrams and State Machine Diagrams.