Contract No. H2020 – 636078

INFORMATION TECHNOLOGIES FOR SHIFT TO RAIL

D2.4 – Travel Shopping Core Integration Report

Due date of deliverable: 31/08/2016
Actual submission date: 12/09/2016

Leader of this Deliverable: Leyre Merle Carrera (Indra)
Reviewed: Amadeus, Hacon, Leonardo

Document status		
Revision	Date	Description
0	19/05/2016	Template creation
1	18/07/2016	Indra sends draft version including test cases
2	27/07/2016	Included Amadeus´ and Leonardo´s contributions in Test Case descriptions
3	25/08/2016	Included section 2, updated sections 3 & 4
4	31/09/2016	Included Hacon´s contributions
5	05/09/2016	Generate final version including results for 2.1.5, 2.3.2, 2.5.x
6	09/09/2016	Include Executive Summary. Update 2.1.6, 2.4.2, review of all partners
7	12/09/2016	Final version after TMC approval

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/05/2015 Duration: 30 months

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Florian Briy	Amadeus	Describe Amadeus´ configuration fo section 3 Description of Test Cases and results in section 4 (leader of 2.2.1, 2.2.2, 2.3.1, 2.5.X)
Peter von Grumbkow	Hacon	Describe Hacon´s configuration of section 3 Description of Test Cases and results in section 4 (leader of 2.2.3, 2.3.2, 2.4.X)
Guido Mariotta	Leonardo	Describe Leonardo´s configuration of section 3 Description of Test Cases and results in section 4 (leader of 2.1.X)
Leyre Merle	Indra	Request and integrate contributions from partners. Describe Indra´s configuration of section 3. Include section 1, 2 and 5. Description of Test Cases and results in section 4 (leader of 2.3.3) Include executive summary and issue

EXECUTIVE SUMMARY

This document details the Test Categories and Test Cases identified by the partners for the Core Release of WP2. For each of the test cases identified, a description is included detailing the objectives, expected results and how to perform the testing.

The objectives of this test campaign in particular are to perform unit testing of each of the core modules of WP2, as well as to test the interfaces with other WP2 modules and other WPs. The results obtained for each of the test identified and described here are also included in this document, identifying weather the test has been performed satisfactorily and is considered “PASSED”, or not and therefore is considered “NOT PASSED”, in the cases when it has not been possible to carry the test for technical problems, or further developments are needed and will we performed for next IT2RAIl release. In some cases input data is simulated in other to perform unit testing not having still the completed connection or interfaces among modules. This way it is possible to evaluate the performance of the component and to assess if the output is the expected.

TABLE OF CONTENTS

LIST OF FIGURES

There are not referenced Figures in the document

LIST OF TABLES

There are not referenced Tables in the document

1. INTRODUCTION

This document details the Test Categories and Test Cases identified by the partners for the Core Release of WP2. For each of the test cases identified, a description is included detailing the objectives, expected results and how to perform the testing.

The objectives of this test campaign in particular are to perform unit testing of each of the core modules of WP2, as well as to test the interfaces with other WP2 modules and other WPs. The results obtained for each of the test identified and described here are also included in this document, identifying weather the test has been performed satisfactorily and is considered "PASSED", or not and therefore is considered "NOT PASSED", in the cases when it has not been possible to carry the test for technical problems, or further developments are needed and will we performed for next IT2RAIl release. In some cases input data is simulated in other to perform unit testing not having still the completed connection or interfaces among modules. This way it is possible to evaluate the performance of the component and to assess if the output is the expected.

The document includes an explanation of the text campaign and a global view of the test categories and test cases (Section 2), followed by an explanation of the infrastructure used by the partners for the testing (Section 3). Section 4 includes information of each of the Test Cases: description, objectives: expected results, how to perform the testing and results obtained. In Section 5 the results of the tests included in section 4 are summarized.

1.1 APPLICABLE DOCUMENTS

D 2.1 Travel Shopping Ontology document

D 2.2 Travel Shopping specifications document

1.2 NORMATIVE DOCUMENTS

Not Applicable.

2. CAMPAIGN STRATEGY

The objectives of the test campaign are to perform unit testing of each of the core modules of WP2, as well as to test the interfaces with other WP2 modules and other WPs.

In order to organize the testing, a number of Test Categories and Test Cases have been identified, for which this document collects the configuration and the results of each of the cases. It is worth mentioning that this document collects only the tests planned for the developments done for the **Core Release**. In some cases, inputs or outputs are simulated to test only specific functions, not being available yet the final version of the components and functions.

For identifying Test Categories and Test Cases, it has been taken into account the designs and analysis performed previously within WP2 and collected in Travel Shopping specifications document. In this document a number of Use Cases were identified, that will be linked to the Test Categories for testing purposes, and also a number of modules/components with related Functions and Function Exchanges, which will be used as a reference to identify the Test Cases.

Below is included the information used as a reference:

- **Use cases:**
  - Manage and decode the mobility request
  - Identify smartest routes corresponding to the mobility request
  - Build itinerary offers
  - Provide itinerary details
  - Alternative calculation

- **Modules/components:**
  - The Mobility Request Manager
  - The Shopping Orchestrator
  - The Meta-Route Explorer
  - The Offer Builder
  - The Travel Expert Journey Planner
  - The Travel Expert Offer Builder

Linked to the previous, a number of **Functions** such as *ManageMobilityRequest* and *PrepareTravelExpertList,* and also **Functional Exchanges** such as *Send Mobility Request with Traveller Preferences* and *Get Stop Places List* were also identified in previous stages of

WP2. For more details about the design and expected operation of each module, *D 2.2 Travel Shopping specifications document* can be consulted.

Taking all of this into consideration, the following list of Test Categories and Test Cases was elaborated for this particular C-Rel test campaign:

- **[TEST CATEGORY 2. 1] Manage mobility request and return offers**
  - [TEST CASE 2.1.1] Prepare mobility request
  - [TEST CASE 2.1.2] Send Mobility RequesT
  - [TEST CASE 2.1.3] Send Mobility Request with Traveller Preferences
  - [TEST CASE 2.1.4] Provide itinerary offers for BA computations
  - [TEST CASE 2.1.5] Get Traveller Preferences from tc
  - [TEST CASE 2.1.6] Itinerary offers provided to mobility request manager by the shopping orchestrator
  - [TEST CASE 2.1.7] Itinerary offers provided to travel companion by the mobility request manager

- **[test category 2.2] Identify smartest routes corresponding to the mobility request**
  - [TEST CASE 2.2.1] Select Smartest Routes
  - [TEST CASE 2.2.2] Select Smartest Routes with search options
  - [TEST CASE 2.2.3] Get mobility request from the shopping orchestrator and provide
  - metaroutes to the shopping orchestrator

- **[test category 2.3] Build itinerary offers**
  - [TEST CASE 2.3.1] Aggregate Itinerary Offers
  - [TEST CASE 2.3.2] Get request from the shopping orchestrator and provide itinerary
  - offers to the shopping orchestrator
  - [TEST CASE 2.3.3] GENERATion of offers by travel expert

- **[test category 2.4] Provide itinerary details**
    - [TEST CASE 2.4.1] Get Stop Places List
    - [TEST CASE 2.4.2] Orchestration of all shopping modules
    - [TEST CASE 2.4.3] Prepare Travel Expert List

- **[test category 2.5] Build network reference resource**
- [TEST CASE 2.5.1] Receive and decode air statistic file
- [TEST CASE 2.5.2] Receive and decode Rail statistic file
- [TEST CASE 2.5.3] Receive and decode coach statistic file
- [TEST CASE 2.5.4] Receive and decode urban transport statistic file
- [TEST CASE 2.5.5] Build network reference resource with air, coach, rail and urban transport

- **[test category 2.6] Complete test WP2**
    - [TEST CASE 2.4.1] Test the whole WP2 flow

Section *4. Test Descriptions* includes information of each of the Test Cases: description, objectives: expected results, how to perform the testing and results obtained. In *section 5* the results of the tests included in section 4 are summarized.

3. TEST MATERIALS DESCRIPTION

This chapter must list all the assets required to perform the test campaign.

3.1 Configuration 2.1

Amadeus WP2 Test Configuration

3.1.1 Infrastructure and Hardware

Computer & Internet connection

3.1.2 Setup & configuration

Each client should have:

- A LSS account (Amadeus security account)

- A Dedicated SAP
- Access to Amadeus Webservice Portal for technical references, user guide and implementation guides

3.1.3 Tested system

Involved modules:

- Metaroute Explorer
- Build Offer

3.1.4 System DATA PARAMETERS

Metaroute Explorer:

- Statistical Data provided by TrenItalia & SNCF.
- Airline schedule network stored on Amadeus side.

3.1.5 Simulators

Build Offer:

- Emulated response

3.1.6 Personel

Amadeus Personnel – henceforth referred as: Test Case Tester: [2.1]

3.2 CONFIGURATION 2.2

Hacon WP2 Test Configuration

3.2.1 Infrastructure and Hardware

State-of-the-art multi-core computer (x86) with a state-of-the-art Linux.

Internet connection

3.2.2 Setup & configuration

The Shopping Orchestrator is installed and deployed on Tomcat 8 at the following endpoint: http://demo.hafas.de/it2rail/shopping-orchestrator

It needs itself properly configured access credentials to the Offer Builder and the Meta Route Explorer.

Additionally it needs a VPN tunnel to access the Location Resolver and the Travel Expert Resolver.

3.2.3 Tested system

Tests concern the Shopping Orchestrator (C-REL version) itself and its communication with Location Resolver, Meta Route Explorer, Travel Expert Resolver and Offer Builder.

3.2.4 System DATA PARAMETERS

The Shopping Orchestrator does not use further data.

3.2.5 Simulators

No simulators used for any component of the Shopping orchestrator.

3.2.6 Personnel

One senior JAVA developer (Hacon Personnel). Henceforth referred as: Test Case Tester: [2.2]

3.3 Configuration 2.3

This section describes the configuration for the Mobility Request Manager module envisaged for the IT2Rail Core Release.

3.3.1 Infrastructure and Hardware

The Mobility Request Manager has to be deployed on an application server such as Tomcat 8.

The machine has to be exposed to the public Internet in order to receive mobility requests.

Software components will be running on a machine with these technical features:

- CPU & Core Xeon 1.9
- Centos 7 64 bit as operating system
- 8 GB RAM.

3.3.2 Setup & configuration

This section contains the setup and the configuration for performing the test campaign envisaged for the Core Release.

The Mobility Request Manager module is installed and deployed on Tomcat 8 at the following endpoint:

http://185.54.152.72:99/it2rail-wp2/mobilityrequest

In order to accomplish the test phase, the personnel in charge of testing the Mobility Request Manager module needs to have an internet connection perfectly working.

In addition, the following software applications have to be installed on the laptop where the tests will be carried out:

- A web browser (Google Chrome preferably).
- POSTMAN as Google Chrome plugin

3.3.3 Tested system

The Mobility Request Manager is deployed on Tomcat application server (v 8.0).

Java Runtime Environment (JRE) or Java Development Kit (JDK) version 8 has to be installed in order to run Tomcat and the Mobility Request Manager.

The Operating System is CentOS 7 64 bit.

3.3.4 System DATA PARAMETERS

This section describes parameters used as input for the Mobility Request Manager. This information will be sent in an HTTP Request by using POST method and enabling the following headers:

- Accept: application/json
- Content-Type: application/json

The following mobility request is formatted in JSON and will be used for the test compaing.

```
{

        "userId": "name.lastname@mail.com",

        "oneTimePreferences": {

                "directTravel": "true"

        },
        "metaJourneys": [{
                "origin": {
                    "name": "Grenoble",
                    "latitude": "2.456789",
                    "longitude": "44.56789"
                },
                "destination": {
                    "name": "Barcelone",
                    "latitude": "5.3456",
                    "longitude": "41.6543"
                },
                "wishedDepartureDate": "2016-05-03T03:55:37Z",
                "wishedArrivalDate": ""
        }, {
                "origin": {
                    "name": "Barcelone",
                    "latitude": "5.3456",
                    "longitude": "41.6543"
                },
                "destination": {
                    "name": "Madrid",
                    "latitude": "3.8196207",
                    "longitude": "40.4378698"
                },
                "wishedDepartureDate": "",
                "wishedArrivalDate": "2016-05-03T03:55:37Z"
        }]
}
```

3.3.5 Simulators

An example of itinerary offer has been provided by AMADEUS and will be used in order to return results as response of the Mobility Request Manager.

3.3.6 Personnel

The personnel required to run this test campaign has to be high-qualified people with a professional background on Information Technology and Computer Science. Leonardo Personnel – henceforth referred as: Test Case Tester: [2.3]

3.4 CONFIGURATION 2.4

Indra WP2 Test Configuration

3.4.1 Infrastructure and Hardware

Currently the software is deployed in a virtual machine with the following characteristics:

- Intel Xeon CPU E5-2630 x2
- 8 GB RAM
- Windows Server 2008 R2 Datacentre 64 bits

The services are deployed and or need the following software:

- SQL Server 2012
- Tomcat 9

3.4.2 Setup & configuration

The software is deployed in a Tomcat 9 accessible in an intranet at the URL: http://192.168.27.50:9090. In the future it is expected to be made accessible through a public address.

In order to use the application a modern web browser is needed (Firefox recommended).

3.4.3 Tested system

The current tested system has the following software versions:

- Java Runtime Environment 1.8.0_91
- Tomcat Server 9.0.0.M6
- Windows Server 2008 R2 Datacentre 64 bits

3.4.4 System DATA PARAMETERS

Data uploaded in the service: TMB transport data in GTFS format

Simulated fares

Simulated traveller information: origin, destination, hour of travelling

3.4.5 Simulators

Request and responses have been simulated and xml and JSON static files sent offline to other partners

3.4.6 Personnel

1 Senior Developer (Indra Personnel). Henceforth referred as: Test Case Tester: [2.4]

4. TEST DESCRIPTIONS

This chapter contains the test cases that were planned to be executed for the Core Release, as well as the results and observations collected.

4.1 [TEST CATEGORY 2. 1] MANAGE MOBILITY REQUEST AND RETURN OFFERS

Corresponds to Use Case *Manage and decode the mobility request*, and test the unitary performance of the *Mobility Request Manager component* and its interfaces with other WP2 modules and other WPs modules.

4.1.1 [TEST CASE 2.1.1] Prepare mobility request

2.1.1	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Component able to process incoming information and prepare mobility request including preferences
Description	Test the performance of the Mobility Request Manager to prepare mobility request processing information from TC and preferences Data Managed: Mobility request and preferences Comments: Input is simulated at this stage.
Status	OK
% passed	OK 80% completed.

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions:				
1	• Internet connection is available • Mobility Request Manager has to be working and running. • Travel Companion Cloud has to be reachable and able to retrieve user preferences.	Prepare the mobility request with the user preferences	The Mobility Request Manager is able to prepare the mobility request without preferences.	**80% Passed**	NA

4.1.2 [TEST CASE 2.1.2] Send Mobility RequesT

2.1.2	
Method Of Test	Demonstration
Type of test	Manual
Objectives	All information sent from WP5 is received by the Mobility Request Manager component

2.1.2	
Description	Test interfaces and communication among TC and Mobility Request Manager component Dependency with WP5: the mobile app is able to send a mobility request. Dependency with WP2: the Mobility Request Manager is able to receive requests from WP5.
Status	OK
% passed	NA

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions:				
1	• Internet connection available • Travel companion mobile app has to be working and running • Travel Companion mobile app has to be connected to the public Internet • Mobility Request Manager has to be working and running	The Mobility Request Manager is able to receive the mobility request sent by the Travel Companion mobile app.	The Mobility Request Manager is able to receive the mobility request sent by the Travel Companion mobile app.	**Passed**	NA

4.1.3 [TEST CASE 2.1.3] Send Mobility Request with Traveller Preferences

2.1.3	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Shopping orchestrator is able to receive the enriched mobility request.
Description	Test Shopping Orchestrator interface that will receive the Mobility Request enriched with User Preferences Dependency with Shopping Orchestrator interface: required to send the Mobility Request enhanced with user's preferences
Status	NOK
% passed	The test will be performed in the next IT2Rail release

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
Preconditions:					

Id	Step description	Expected result	Observed result	State	Associated defect
1	• Internet connection available • Mobility Request Manager has to be working and running • Travel Shopping Orchestrator has to be working and running	The Travel Shopping Orchestrator is able to receive mobility requests enriched with users' preferences. Result must be expressed under the form of a metric	Not available	**NA**	NA

4.1.4 [TEST CASE 2.1.4] Provide itinerary offers for BA computations

2.1.4	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Mobility Request Manages is able to send itinerary offers to WP6. The latter will store itinerary offers in its repositories for later computations.
Description	Test WP6 interface responsible for storing itinerary offers
	Dependency with WP6 interface: required for sending itinerary offers to WP6.
Status	OK
% passed	NA

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions				
1	• Internet connection is available • Mobility Request Manager has to be working and running • Itinerary offers have to be computed by the Travel Shopping Orchestrator • The Business Analytics module has to be working and running.	The IT2Rail Business Analytics is able to receive itinerary offers and to store them.	The Mobility Request Manager sends itinerary offers to WP6. The Business Analytics is able to receive itinerary offers and to store them.	**Passed**	NA

4.1.5 [TEST CASE 2.1.5] Get Traveller Preferences from tc

2.1.5	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Mobility Request Manager component is able to retrieve preferences information from WP5-Cloud for a particular user
Description	Test interfaces and communication among TC - Cloud and Mobility Request Manager component Dependency with WP5 Interface. Required for retrieving user's preferences.
Status	OK
% passed	

Configuration to apply:2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
Preconditions:					

Id	Step description	Expected result	Observed result	State	Associated defect
1	• Internet connection is available • Mobility Request Manager has to be working and running • Travel Companion Cloud has to be reachable in order to retrieve user preferences.	The Mobility Request Manager is able to retrieve user preferences from Travel Companion Cloud.	The Mobility Request Manager is able to retrieve user preferences from Travel Companion Cloud	**Passed**	NA

4.1.6 [TEST CASE 2.1.6] Itinerary offers provided to mobility request manager by the shopping orchestrator

2.1.6	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Mobility Request Manager component successfully retrieves itinerary offers and can handle them.
Description	The interface between Mobility Request Manager and Shopping Orchestrator is working and returns valid itinerary offers.
Status	OK
% passed	

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]/[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions:				
1	• Internet connection is available • Mobility Request Manager has to be working and running • Itinerary offers have to be computed by the Travel Shopping Orchestrator	The Mobility Request Manager is able to receive itinerary offers from the Travel Shopping Orchestrator.	The Mobility Request Manager is able to receive itinerary offers from the Travel Shopping Orchestrator.	**Passed**	NA

4.1.7 [TEST CASE 2.1.7] Itinerary offers provided to travel companion by the mobility request manager

2.1.7	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Mobility Request Manager successfully return itinerary offers to the TC
Description	Test that the computed itinerary offers are returned to the Travel Companion mobile app.

2.1.7	
Status	OK
% passed	NA

Configuration to apply: 2.3	
Regression	No
Test Case Tester	[2.3]

Id	Step description	Expected result	Observed result	State	Associated defect
Preconditions:					

Id	Step description	Expected result	Observed result	State	Associated defect
1	• Internet connection is available • Mobility Request Manager has to be working and running • Itinerary offers have to be computed by the Travel Shopping Orchestrator Travel Companion Mobile App has to be connected to the public Internet	The Travel Companion mobile app receives itinerary offers from the Mobility Request Manager	The Travel Companion mobile app receives itinerary offers from the Mobility Request Manager	**Passed**	NA

4.2 [TEST CATEGORY 2.2] IDENTIFY SMARTEST ROUTES CORRESPONDING TO THE MOBILITY REQUEST

Corresponds to Use Case *Identify smartest routes corresponding to the mobility request*, and test the unitary permformance of the *Meta Route Explorer component* and its interfaces with other WP2 modules and other WPs modules.

4.2.1 [TEST CASE 2.2.1] Select Smartest Routes

2.2.1	
Method Of Test	Analysis
Type of test	Manual
Objectives	Obtain most relevant metaroutes joining origin and destination

2.2.1	
Description	The user specifies an origin, destination, wished departure date… (Only mandatory data). Dependency: Testable only if networks in the metaroute explorer are completed
Status	OK
% passed	NA

Configuration to apply: 2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- Amadeus security account**				
1	Send a basic "air oriented" query with no options : 1 origin & associated airport ; 1 destination & associated airport	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 Air episode No irrelevant metaroute is returned	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 Air episode	**Passed**	

Id	Step description	Expected result	Observed result	State	Associated defect
2	Send a basic query with no options : 1 origin & associated stop places (1 airport, 1 train station) ; 1 destination & associated stop places (1 airport, 1 train station)	1 Metaroute is returned with rail & train metatravel expert episodes 1 Metaroute is returned with air metatravel expert episodes No irrelevant metaroute is returned	1 Metaroute is returned with rail & train metatravel expert episodes 1 Metaroute is returned with air metatravel expert episodes	**Passed**	
3	Send a basic query for multiple MetaJourneys, no options : For each: 1 origin & associated stop places (1 airport, 1 train station) ; 1 destination & associated stop places (1 airport, 1 train station)	For each metaJourney: 1 Metaroute is returned with rail & train metatravel expert episodes 1 Metaroute is returned with air metatravel expert episodes No irrelevant metaroute is returned	For each metaJourney: 1 Metaroute is returned with rail & train metatravel expert episodes 1 Metaroute is returned with air metatravel expert episodes	**Passed**	
4	Error case: No stop places indicated in query sent	An error is returned, no route is computed	An error is returned, no route is computed	**Passed**	
5	Error case: Past date indicated in query sent	An error is returned, no route is computed	An error is returned, no route is computed	**Passed**	

4.2.2 [TEST CASE 2.2.2] Select Smartest Routes with search options

2.2.2	
Method Of Test	Analysis
Type of test	Manual
Objectives	Obtain most relevant metaroutes joining origin and destination metaroutes are consistent with request
Description	The user specifies an origin, destination, wished departure date… and includes or excludes one or several modes Dependency: Testable only if networks in the metaroute explorer are completed
Status	OK
% passed	NA

Configuration to apply: 2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- Amadeus security account**				
1	Send a basic query with rail transport mode excluded : 1 origin & associated airport + train station; 1 destination & associated airport + train station	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 Air episode No metaroute including rail metatravel episodes are returned	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 Air episode	**Passed**	
2	Send a basic query with rail transport mode included : 1 origin & associated airport + train station; 1 destination & associated airport + train station	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 train episode No irrelevant metaroute is returned	1 Metaroute is returned with 3 Metatravelexpert Episode: 2 Urban Transport & 1 Train episode	**Passed**	
3	Send a basic query with rail + air transport mode excluded : 1 origin & associated airport + train station; 1 destination & associated airport + train station	No MetaRoute is returned: An error is returned (not made explicit for the core release)	No MetaRoute is returned: An error is returned	**Passed**	

4.2.3 [TEST CASE 2.2.3] Get mobility request from the shopping orchestrator and provide meta routes to the shopping orchestrator

2.2.3	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Meta route explorer successfully decodes mobility request; shopping orchestrator successfully decodes Meta route explorer replies
Description	The interface between the shopping orchestrator and the meta route explorer is working
Status	OK
% passed	NA

Configurayion to apply: 2.2	
Regression	No
Test Case Tester	[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
Preconditions: **- Internet connection** **- Amadeus security account**					

Id	Step description	Expected result	Observed result	State	Associated defect
1	Send grammatically correct requests from Shopping Orchestrator to Meta Route Explorer.	Receive a grammatically correct response and decodes it.	Received a grammatically correct response and decoded it.	**Passed**	The quality of the response content cannot be checked.
2	Send grammatically incorrect requests from Shopping Orchestrator to Meta Route Explorer.	Receive an error message response.	Received an error message response.	**Passed**	

4.3 [TEST CATEGORY 2.3] BUILD ITINERARY OFFERS

Corresponds to Use Case *Build itinerary offers*, and test the unitary performance of the *Offer Builder component* and its interfaces with other WP2 modules.

4.3.1 [TEST CASE 2.3.1] Aggregate Itinerary Offers

2.3.1	
Method Of Test	Analysis
Type of test	Manual
Objectives	Itinerary offer is built
Description	Aggregation using simulators
Status	OK
% passed	NA

Configuration to apply: 2.1

Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- internet connection** **- Amadeus security account**				
1	Send a grammatically valid query.	Receive the emulated Build Itinerary Offers response	Emulated Build Itinerary Offers response received	**Passed**	
2	Grammar checks – elements order: launch a query in which elements are not correctly ordered. The element Destination is inserted before the element Origin.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “Invalid content was found starting with element 'att:Origin'. One of '{”http://xml.amadeus.com/2010/06/DoorToDoorTypes_v1”:Destination}' is expected.”	**Passed**	

Id	Step description	Expected result	Observed result	State	Associated defect
3	Grammar checks - value constraint: launch a query in which one value is not compliant with the expected format. One of the ID does not start with a letter.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “The value '1000' of attribute 'id' on element 'att:Traveller' is not valid with respect to its type, 'ID'.”	**Passed**	
4	Grammar checks – ID unicity: launch a query in which two IDs have the same value.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “There are multiple occurrences of ID value 'ID_1000'.”	**Passed**	
5	Grammar checks – ID reference: launch a query in which an IDREF does not refer to an existing ID in the message.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “There is no ID/IDREF binding for IDREF 'ID_500'.”	**Passed**	

Id	Step description	Expected result	Observed result	State	Associated defect
6	Grammar checks – mandatory element: launch a query in which one mandatory element is missing. One of the MetaJourney does not contain any Origin element.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “Invalid content was found starting with element 'att:Destination. One of '{”http://xml.amadeus.com/2010/06/DoorToDoorTypes_v1”:Origin}' is expected.”	**Passed**	
7	Grammar checks – mandatory attribute: launch a query in which one mandatory attribute is missing. One of the MetaRoute does not have any id.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “Attribute 'id' must appear on element 'att:MetaRoute'.”	**Passed**	

Id	Step description	Expected result	Observed result	State	Associated defect
8	Grammar checks – element cardinality: launch a query in which one of the element occurrences exceeds the maximum number authorized. In one of the MetaJourney, the element Origin appears twice.	Receive an error code 100043, with indication on the grammar error.	Error code 100043 returned, with the message: “Invalid content was found starting with element 'att:Origin'. One of '{”http://xml.amadeus.com/2010/06/DoorToDoorTypes_v1”:Destination}' is expected.”	**Passed**	

4.3.2 [TEST CASE 2.3.2] Get request from the shopping orchestrator and provide itinerary offers to the shopping orchestrator

2.3.2	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Offer builder successfully decodes the request; shopping orchestrator successfully decodes offer builder replies
Description	Test that the interface between the shopping orchestrator and the offer builder is working
Status	OK

2.3.2
% passed

Configuration to apply: 2.2	
Regression	No
Test Case Tester	[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- Amadeus security account**				
1	Send grammatically correct requests from Shopping Orchestrator to Offer Builder.	Receive a grammatically correct response and decodes it.	Received a grammatically correct response and decoded it.	**Passed**	The quality of the response content cannot be checked.
2	Send grammatically incorrect requests from Shopping Orchestrator to Offer Builder.	Receive an error message response.	Received an error message response.	**Passed**	

4.3.3 [TEST CASE 2.3.3] GENERATion of offers by travel expert

2.3.3	
Method Of Test	Demonstration
Type of test	Manual
Objectives	Travel Expert Generation of Travel Offers by Travel Experts
Description	Indra/TMB Travel Expert Generates a number of offers to travel in the city of Barcelona. Although this information should be sent from the Travel Expert to the Shopping Offer Builder through the WP1 IF, for the C-REL a specific static file has been generated and sent to Amadeus in xml and JSON format, for a specific case.
Status	OK
% passed	NA

Configuration to apply: 2.4	
Regression	No
Test Case Tester	[2.4]

Id	Step description	Expected result	Observed result	State	Associated defect
Preconditions:					

Id	Step description	Expected result	Observed result	State	Associated defect
1	-Enable software based on Open Trip Planner for Route Planning -Upload TMB transport information in GTFS format (bus, Tram and metro) -Internet connection available -Only a limited number of functionalities available for route planning. Not possible to include preferences -10 different scenarios simulated for initial testing	GTFS information upload correctly	Successful read and upload of GTFS files. Software capable of using the information for route calculation.	**Passed**	NA
2	Origin/Destination and travelling times manually introduced for 2 specific scenarios (2 users arriving by different modes at specific times)	Obtain different routes and offers	Generated xml and JSON for request and response. 3 possible routes obtained for each of the 2 scenarios	**Passed**	NA

4.4 [TEST CATEGORY 2.4] PROVIDE ITINERARY DETAILS

Corresponds to Use Case *Provide itinerary details*, and test the unitary performance of the *Shopping Orchestrator component* and its interfaces with other WP2 modules and other WPs components.

4.4.1 [TEST CASE 2.4.1] Get Stop Places List

2.4.1	
Method Of Test	Demonstration
Type of test	Manual
Objectives	The list of nearest stop places is identified and integrated in the mobility request
Description	The location resolver identifies nearest stop places
Status	OK
% passed	

Configuration to apply: 2.2	
Regression	No
Test Case Tester	[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- VPN tunnel to Location Resolver**				
1	Send grammatically correct requests from Shopping Orchestrator to Location Resolver.	Receive a grammatically correct response and decodes it.	Received a grammatically correct response and decoded it.	**Passed**	The quality of the response content cannot be checked.

Id	Step description	Expected result	Observed result	State	Associated defect
2	Send grammatically incorrect requests from Shopping Orchestrator to Location Resolver.	Receive an error message response.	Received an error message response.	**Passed**	

4.4.2 [TEST CASE 2.4.2] Orchestration of all shopping modules

2.4.2	
Method Of Test	Demonstration
Type of test	Manual
Objectives	The Shopping Orchestrator successfully fulfils the four consecutive steps of shopping.
Description	The Shopping Orchestrator has to fulfil four consecutive steps of shopping. Data has to be transferred correctly from one step to the next.
Status	OK
% passed	

2.2	
Regression	No
Test Case Tester	[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- VPN tunnel to Location Resolver** **- Amadeus security account**				
1	Initiate a valid mobility request with the shopping orchestrator. Input from Travel Expert Resolver is simulated	The Location Resolver, Meta Route Explorer, Travel Expert Resolver, and Offer Builder are queried consecutively and their responses are used as for the following system, respectively.	The Location Resolver, Meta Route Explorer, Travel Expert Resolver (interface simulated for testing), and Offer Builder were queried consecutively and their responses were used as for the following system, respectively.	**Passed**	.

4.4.3 [TEST CASE 2.4.3] Prepare Travel Expert List

2.4.3	
Method Of Test	Demonstration

2.4.3

Type of test	Manual
Objectives	The Shopping Orchestrator successfully calls the Travel Expert Resolver and retrieves meta travel expert episodes enriched by appropriate travel experts.
Description	The Travel Expert Resolver has to supply meta travel expert episodes with appropriate travel experts.
Status	NOK
% passed	0%

Configuration to apply: 2.2

Regression	No
Test Case Tester	[2.2]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Internet connection** **- VPN tunnel to Location Resolver**				
1	Send grammatically correct requests from Shopping Orchestrator to Travel Expert Resolver.	Receive a grammatically correct response and decodes it.	Received a grammatically correct response and decoded it.	**Failed**	The quality of the response content cannot be checked.

Id	Step description	Expected result	Observed result	State	Associated defect
2	Send grammatically incorrect requests from Shopping Orchestrator to Travel Expert Resolver.	Receive an error message response.	Received an error message response.	Failed	

4.5 [TEST CATEGORY 2.5] BUILD NETWORK REFERENCE RESOURCE

Corresponds to Use Case *Identify smartest routes corresponding to the mobility request*, and test the unitary performance of the *Meta Route Explorer component* and its interfaces with other WP2 modules and other WPs components.

4.5.1 [TEST CASE 2.5.1] Receive and decode air statistic file

2.5.1	
Method Of Test	Analysis
Type of test	Manual
Objectives	Decode file with air content
Description	Air travel experts provide at least a statistic file with air content
Status	OK
% passed	

2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Statistical file with air content**				
1	Integrate air content in meta-route	The graph includes air content	The graph includes air content	**Passed**	

4.5.2 [TEST CASE 2.5.2] Receive and decode Rail statistic file

2.5.2	
Method Of Test	Analysis
Type of test	Manual
Objectives	Decode file with rail content
Description	Rail travel experts provide at least a statistic file with air content
Status	OK
% passed	

2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Statistical file with rail content**				
1	Integrate rail content in meta-route	The graph includes rail content	The graph includes rail content	**Passed**	

4.5.3 [TEST CASE 2.5.3] Receive and decode coach statistic file

2.5.3	
Method Of Test	Analysis
Type of test	Manual
Objectives	Decode file with coach content
Description	Coach travel experts provide at least a statistic file with air content
Status	NOK (Not Applicable)

2.5.3
% passed

2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Statistical file with coach content**				
NA	NA	NA	NA	NA	NA

4.5.4 [TEST CASE 2.5.4] Receive and decode urban transportstatistic file

2.5.4	
Method Of Test	Analysis
Type of test	Manual
Objectives	Decode file with urban transport content

2.5.4	
Description	Urban transport travel experts provide at least a statistic file with air content
Status	NOK (Not Applicable)
% passed	

2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- Statistical file with urban transport content**				
NA	NA	NA	NA	NA	NA

4.5.5 [TEST CASE 2.5.5] Build network reference resource with air, coach, rail and urban transport

2.5.5	
Method Of Test	Analysis

2.5.5	
Type of test	Manual
Objectives	Build Meta Networks
Description	MetaRoute Explorer builds networks with statistic data
Status	OK
% passed	

2.1	
Regression	No
Test Case Tester	[2.1]

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **- [List all pre-condition to the test case. EX: internet connection…]**				
1	Integrate all content resource received, mixing Air and Rail content.	The graph is fully built	The graph is fully built	**Passed**	NA

4.6 [Test category 2.6] Complete test WP2

Corresponds to the testing of the whole WP2 flow (integration among WP2 modules) and the integration within IT2Rail

4.6.1 [TEST CASE 2.4.1] Test the whole WP2 flow

2.4.1	
Method Of Test	Demonstration
Type of test	Manual
Objectives	A mobility request is sent from the TC and the appropriate list of itinerary offers is returned
Description	This test aims at testing the whole WP2 flow and the integration within IT2Rail. Needs that all the previous key Tests are Passed
Status	NOK
% passed	A few critical tests has not been passed for this campaign, making unable to test the complete flow from beginning to end

2.1-2.2-2.3	
Regression	NA
Test Case Tester	NA

Id	Step description	Expected result	Observed result	State	Associated defect
	Preconditions: **-All list pre-condition of the previous test cases apply** **-Previous Test Cases must be passed to perform the full test**				
NA	NA	NA	NA	NA	NA

5. TEST EXECUTION

The following table gives an overview of the results achieved during the first campaign described in the document. Previous section details the results obtained for the test executed.

Test Case Form - Summarized results			
Test Category	**Test Case ID**	**Results of Test Run (passed/not passed)** **More details of results in section 4**	**Comments**
2. 1: Manage mobility request and return offers	**Test Case 2.1.1**	Passed	80% completed: Able to prepare a complete mobility query but not to send it to the Travel Shopping Orchestrator
	Test Case 2.1.2	Passed	
	Test Case 2.1.3	Not passed	Moved to next IT2Rail release
	Test Case 2.1.4	Passed	
	Test Case 2.1.5	Passed	
	Test Case 2.1.6	Passed	
	Test Case 2.1.7	Passed	
2.2 : Identify smartest	**Test Case 2.2.1**	Passed	

routes corresponding to the mobility request	**Test Case 2.2.2**	Passed	
	Test Case 2.2.3	Passed	
2.3: Build itinerary offers	**Test Case 2.3.1**	Passed	
	Test Case 2.3.2	Passed	
	Test Case 2.3.3	Passed	
2.4: Provide itinerary details	**Test Case 2.4.1**	Passed	
	Test Case 2.4.2	Passed	Moved to next IT2Rail release
	Test Case 2.4.3	Not passed	Moved to next IT2Rail release
2.5 Build network reference resource	**Test Case 2.5.1**	Passed	
	Test Case 2.5.2	Passed	
	Test Case 2.5.3	Not passed	Moved to next IT2Rail release
	Test Case 2.5.4	Not passed	Moved to next IT2Rail release
	Test Case 2.5.5	Passed	
2.6 Complete test WP2	**Test Case 2.6.1**	Not passed	Not possible to perform - Moved to next IT2Rail release