Integration Test Plan

myTaxiService

Petar Korda
Krishnan Ranjithkumar

Politecnico di Milano, 21.01.2016

Contents

1 Introduction

1.1 Revision History

This is the first document for integration testing completed on 1/21/2016.

1.2 Purpose and Scope

The purpose of the document is to test the interaction among the different components by integrating it together. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing.

The scope of the document is only to test block of the components together (i.e.) only for integration testing not for other testing such system testing, regression testing in the system myTaxiService - a taxi service, providing a mobile and web application for requesting and reserving taxi rides.

1.3 List of Definitions and Abbreviations

Definitions:

- Request - and action that user takes when requesting a taxi ride
- Reservation - and action that user takes when reserving a taxi ride for the specific time and date
- Proposal - and overview of the request or reservation that is send to the user, and used for confirmation

Abbreviations:

- DD - Design Document
- IT - Integration Test
- API - Application Programming Interface

1.4 List of Reference Documents

[1] Requirement analysis and design document of myTaxiService.
[2] Design document of myTaxiService.
[3] Integration test plan document by spin grid
[4] Website Jboss.org - testing tools.
[5] Verification and validation slides from the lecture.

2 Integration Strategy

2.1 Entry Criteria

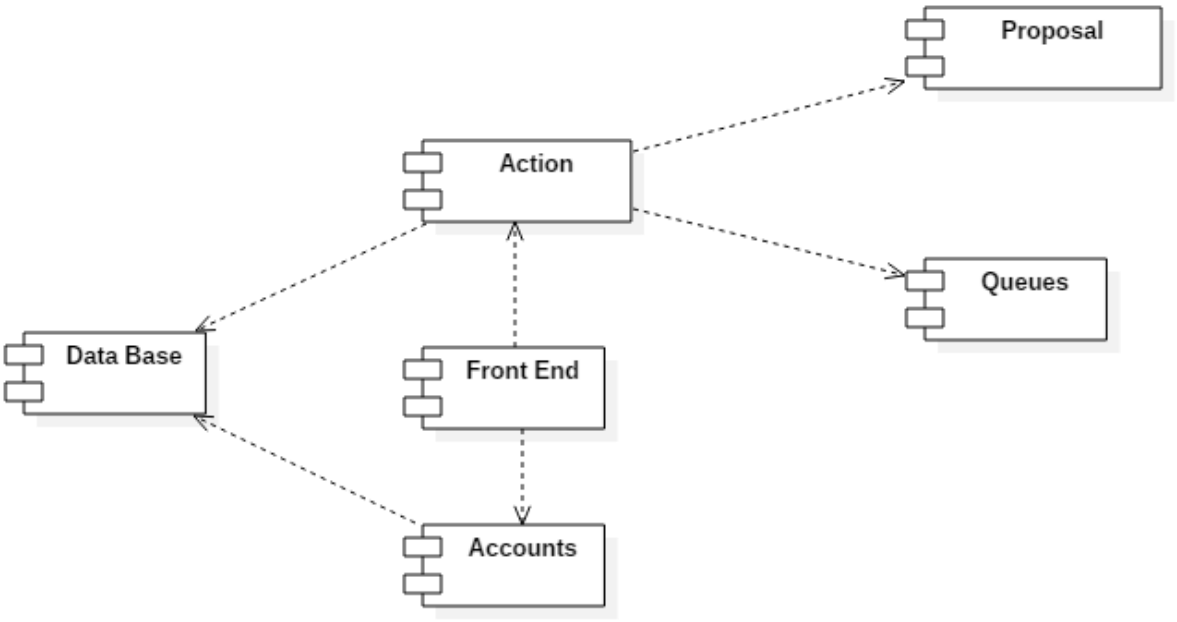
Components have to be unit tested before their integration tests. Also if certain drivers and stubs are required, they need to be developed before the integration tests.

2.2 Elements to be Integrated

The system is divided to 3 subsystems:

- Core System - business logic
- User Application - user management and front - end applications
- Driver Application - taxi application

The whole system is composed of high level components (section 2.2 in DD) shown below:

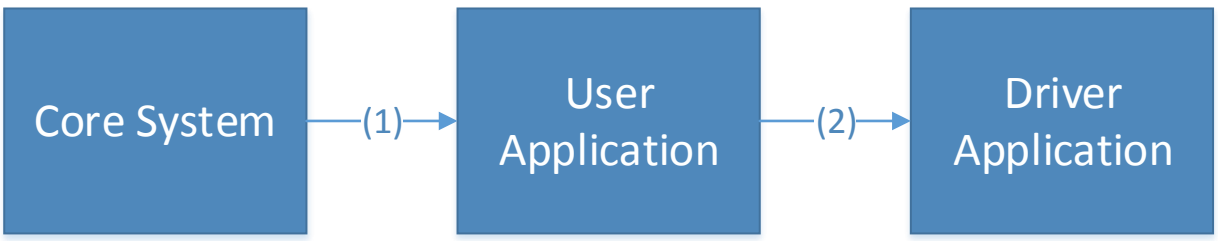


Each of these high level components is composed of other lover level components (sections 2.3 of the DD):

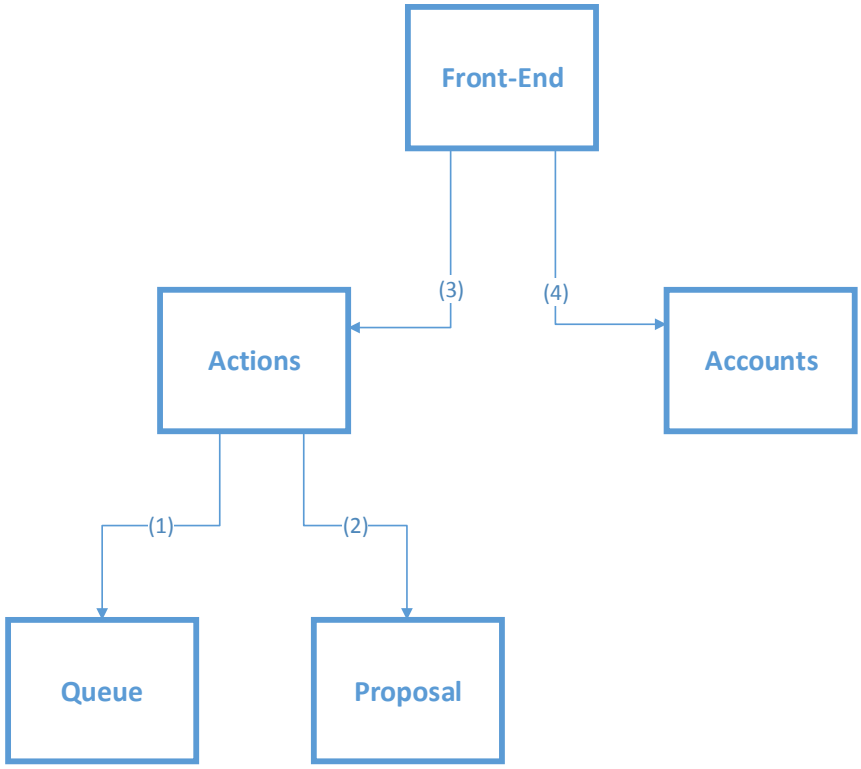
- Queues - Queue, ZoneManager, TaxiManager
- Proposal - ProposalManager, FareCalculator, WaitingCalculator, PaymentManager
- Actions - RequestManager, ReservationManager, NotificationManager
- Accounts - UserManager
- Front End - AndroidApp, iOSApp, WebSite

2.3 Integration Testing Strategy

In the diagram below the bottom - up order of integration of subsystems is shown. Core System is composed of the Queue and Zone management, Requests and Reservation for a ride, as well as Proposal generation. User Application is the front-end of the user app as well as Accounts management. Finally, Driver Application is the taxi application.

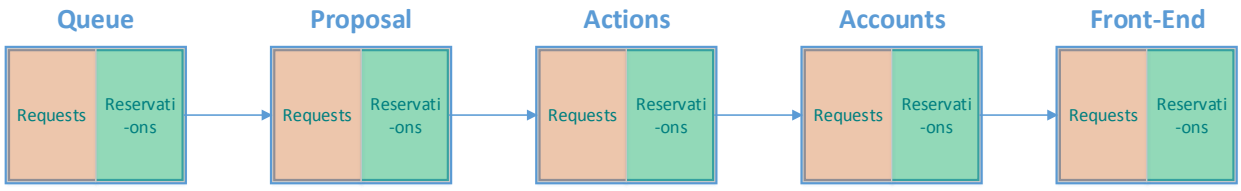


The order of integration sequence in regard to the high level view components is shown in the next diagram.



As it can be seen, here the bottom - up approach is being used. Whole system can be divided into two based on the type of action a user is performing: Request or Reservation. That's why it is convenient to perform the thread integration tests. First all of the components are integrated but only the parts required for Requests (requesting a ride) are developed and tested, and then after this is completed the Reservations (reserving a ride) functionality is added to the system.

With regard to the sequence of components that have to be developed, the diagram bellow is concerned about this. First the Requests parts are developed and integrated, and after that Reservations functionality is added and integrated.



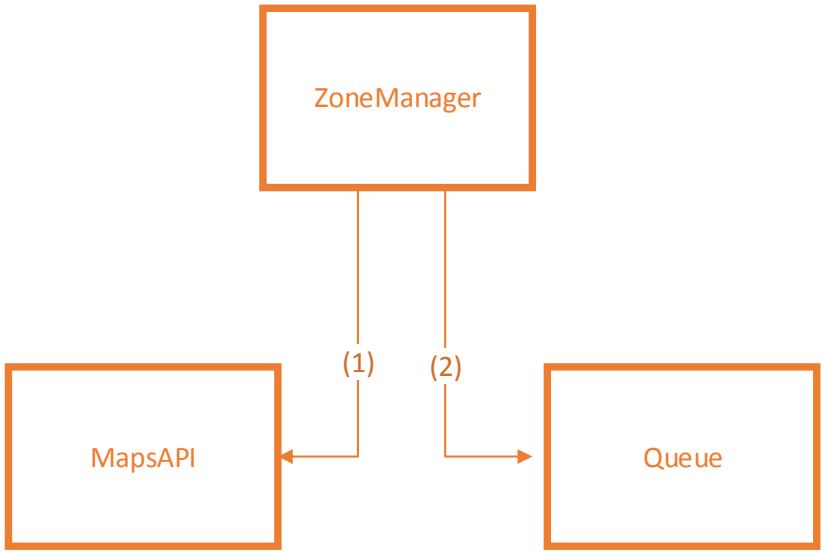
2.4 Sequence of Component/Function Integration

2.4.1 Software Integration Sequence

Core System:

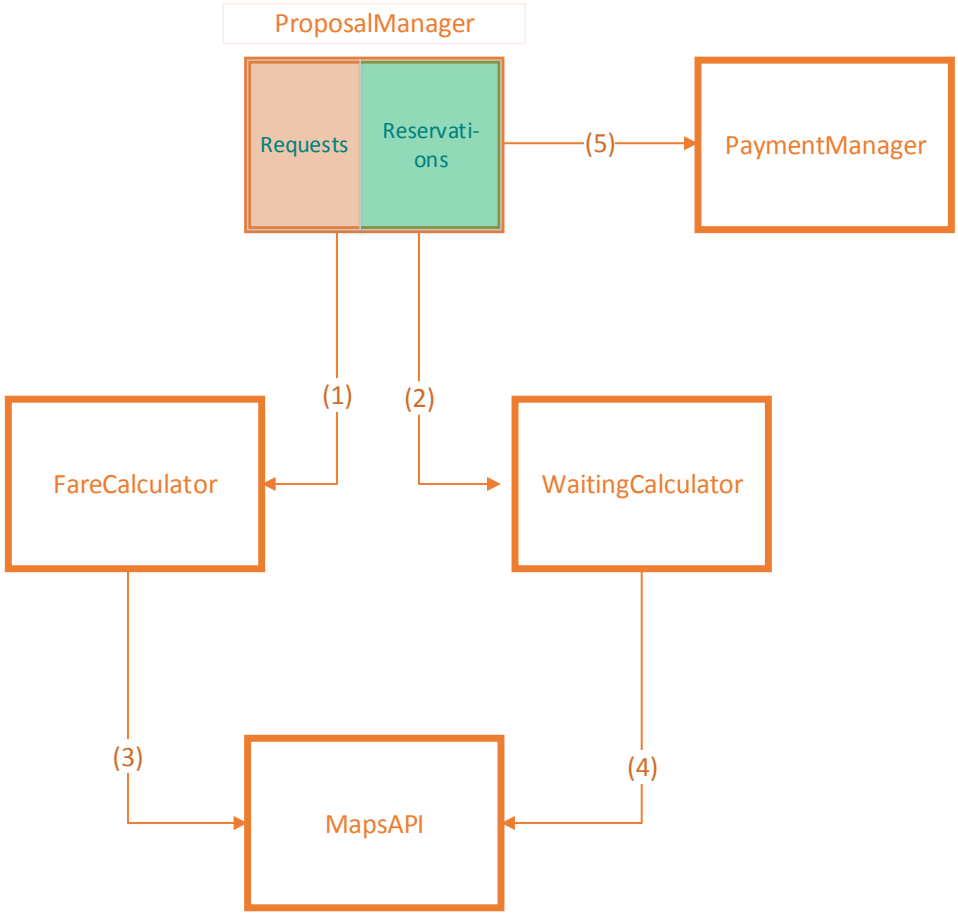
Components of core system and their integration order is:

- Queue (High level component) - consisted of (lower level components): Queue, ZoneManager



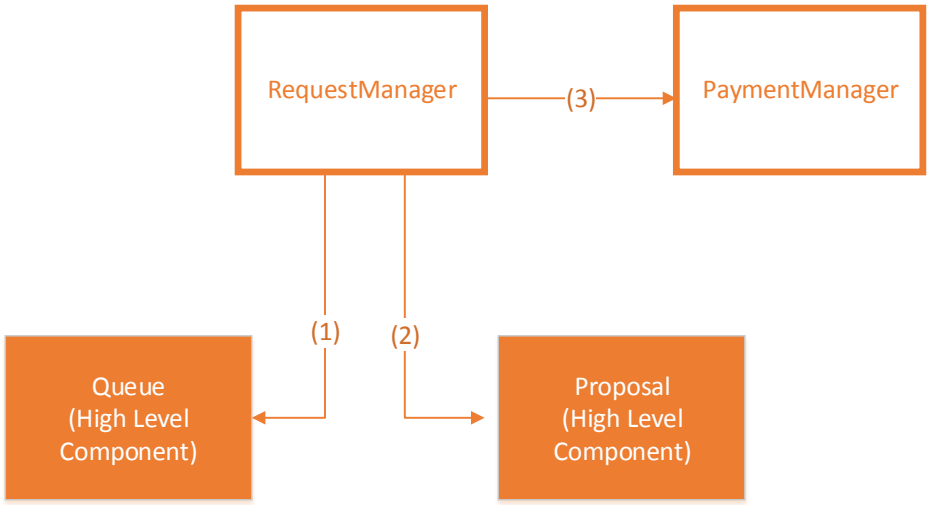
First the ZoneManager is developed and tested with external component MapsAPI, then Queue is developed and tested with ZoneManager.

- Proposal (High level component) - consisted of (lower level components): ProposalManager, FareCalculator, WaitingCalculator

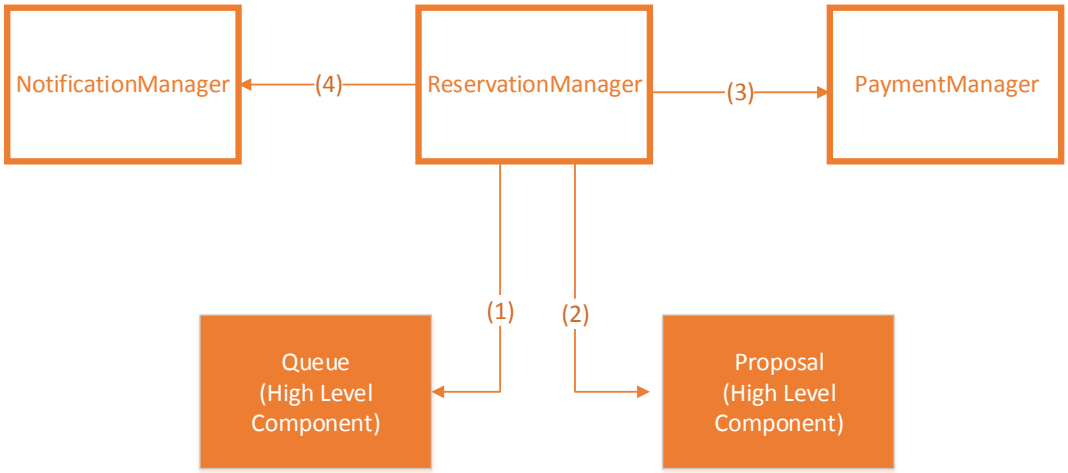


These tests are done is separate integration threads for Requests and Reservations. First the ProposalManager is developed and tested with stubs of FareCalculator, WaitingCalculator and PaymentManager, which are then replaced with real components. So the top - down strategy is used in this case.

- Actions (High level component) - consisted of (lower level components): RequestManager, ReservationManager



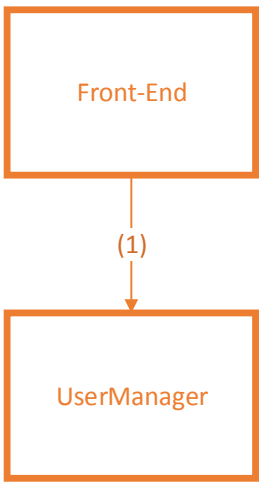
These tests are performed after Queue and Proposal high level components have been tested. By the time when the reservations functionality is being added, also these tests are done for the ReservationManager.



User Application:

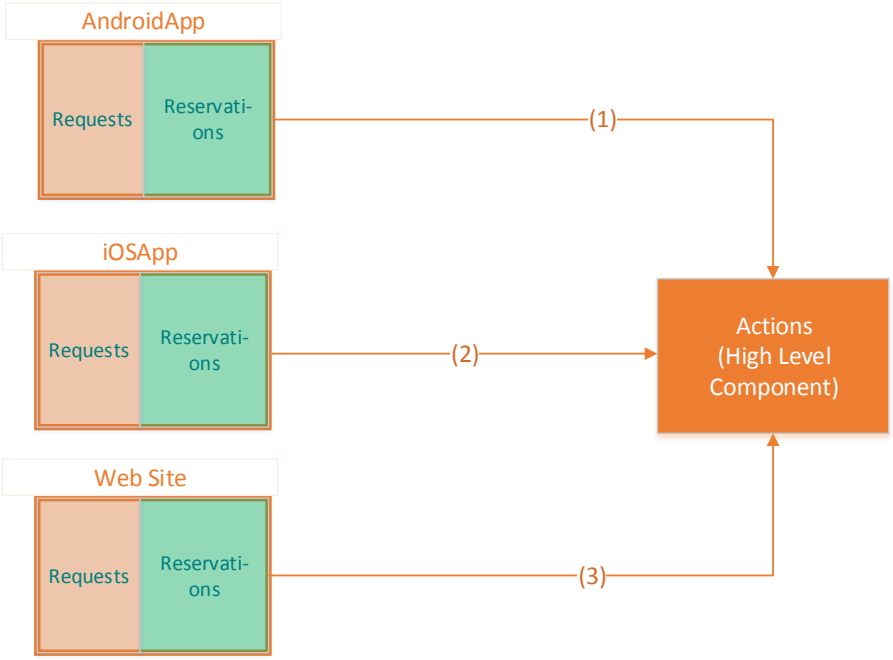
Components of user application and their integration order is:

- Accounts (High level component) - consisted of (lower level components): UserManager



UserManager is being tested with a Front-End driver. Bottom - up is used here.

- Front-End (High level component) - consisted of (lower level components): AndroidApp, iOSApp, WebSite



These tests are done is separate integration threads for Requests and Reservations.

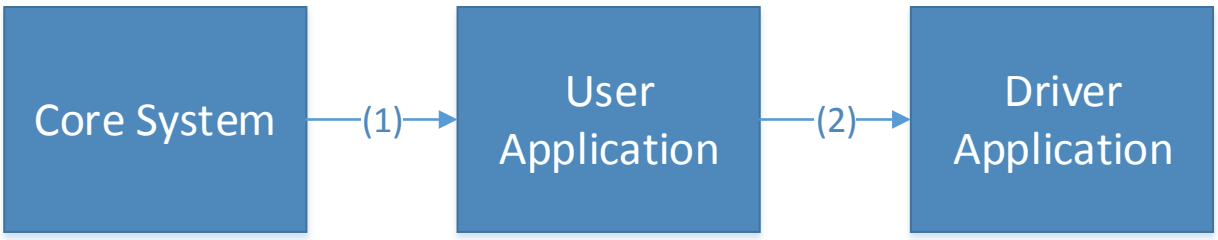
Driver Application:

Components of driver application and their integration order is:

- TaxiManager

2.4.2 Subsystem Integration Sequence

If we refer again to the first diagram, the order of subsystem integration is shown.



3 Individual Steps and Test Description

INTEGRATION TEST CASE IT1

Test case Id	IT1
Test case components	Zone Manager → Maps
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check if the zone value is correct which is returned by maps
Environmental Needs	External API - Google maps

INTEGRATION TEST CASE IT2

Test case Id	IT2
Test case components	Zone Manager →Queue
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the zone manager calling the corresponding queue
Environmental Needs	IT1 Succeeded

INTEGRATION TEST CASE IT3

Test case Id	IT3
Test case components	Proposal Manager → Fare calculator
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the fare value returned is appropriate
Environmental Needs	Fare calculator Stub

INTEGRATION TEST CASE IT4

Test case Id	IT4
Test case components	Proposal Manager →Waiting time calculator
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the returned output is appropriate
Environmental Needs	Waiting time calculator stub

INTEGRATION TEST CASE IT5

Test case Id	IT5
Test case components	Fare calculator→Maps
Input specifications	Check the typical starting point and destination (both may or may not be in the

	same zone).
Output Specifications	Check the value of the distance returned by Maps is appropriate
Environmental Needs	External API such as Google maps

INTEGRATION TEST CASE IT6

Test case Id	IT6
Test case components	Waiting time calculator→Maps
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the value of the distance returned by Maps is appropriate
Environmental Needs	External API such as Google maps

INTEGRATION TEST CASE IT7

Test case Id	IT7
Test case components	Proposal Manager → Payment Manager
Input specifications	Typical credit card details (some valid - it exists and is not expired, some of invalid cards)
Output Specifications	Check the payment is made successfully, or the system is informed otherwise
Environmental Needs	External Payment API

INTEGRATION TEST CASE IT8

Test case Id	IT8
Test case components	Fare calculator→Maps
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the value of the distance returned by Maps is appropriate
Environmental Needs	External API such as Google maps

INTEGRATION TEST CASE IT9

Test case Id	IT9
Test case components	Request manager→Queue(high level component)
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the corresponding queue is called and taxi is found.
Environmental Needs	IT1,IT2,IT3 succeeded

INTEGRATION TEST CASE IT10

Test case Id	IT10
Test case components	Request Manager → Proposal
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone).
Output Specifications	Check the proposal generated consist of waiting time and fare is appropriate
Environmental Needs	IT4-IT8 succeeded

INTEGRATION TEST CASE IT11

Test case Id	IT11
Test case components	Request Manager → Payment Manager
Input specifications	Typical credit card details (some valid - it exists and is not expired, some of invalid cards)
Output Specifications	Check the payment is made successful, also check refund amount in case of cancellation.
Environmental Needs	External payment API, IT4-IT8 succeeded

INTEGRATION TEST CASE IT12

Test case Id	IT12
Test case components	Reservation manager→Queue(high level

	component)
Input specifications	Check the typical starting point and destination, (both may or may not be in the same zone), Date and Time.
Output Specifications	Check the corresponding queue is called and taxi is found.
Environmental Needs	IT1,IT2,IT3 succeeded

INTEGRATION TEST CASE IT13

Test case Id	IT13
Test case components	Reservation Manager → Proposal
Input specifications	Check the typical starting point and destination (both may or may not be in the same zone), Date and Time.
Output Specifications	Check the proposal generated consist of waiting and fare is appropriate
Environmental Needs	IT4-IT8 succeeded

INTEGRATION TEST CASE IT14

Test case Id	IT14
Test case components	Reservation Manager → Payment Manager
Input specifications	Typical credit card details (some valid - it exists and is not expired, some of invalid cards)
Output Specifications	Check the payment is made successful, also check refund amount in case of cancellation.
Environmental Needs	External payment API, IT4-IT8 succeeded

INTEGRATION TEST CASE IT15

Test case Id	IT15
Test case components	Reservation Manager → Notification Manager
Input specifications	Various reservation (different times, same time etc.)
Output Specifications	Check the reminder/notification is generated and sent to the user
Environmental Needs	IT4-IT8 succeeded

INTEGRATION TEST CASE IT16

Test case Id	IT16
Test case components	Front End → User Manager
Input specifications	Various types of users (registered users, not registered users)
Output Specifications	Check the user will be able to Register, login.
Environmental Needs	Front end driver.

INTEGRATION TEST CASE IT17

Test case Id	IT17
Test case components	Android app → Action Manager
Input specifications	Typical requests (different starting points and destinations) and reservations (different times and dates also)
Output Specifications	Check the user will be able to Request and reserve a taxi.
Environmental Needs	IT9 - IT15 succeeded

INTEGRATION TEST CASE IT18

Test case Id	IT18
Test case components	IOS app → Action Manager
Input specifications	Typical requests (different starting points and destinations) and reservations (different times and dates also)
Output Specifications	Check the user will be able to Request and reserve a taxi.
Environmental Needs	IT9 - IT15 succeeded

INTEGRATION TEST CASE IT19

Test case Id	IT19
Test case components	Website → Action Manager

Input specifications	Typical requests (different starting points and destinations) and reservations (different times and dates also)
Output Specifications	Check the user will be able to Request and reserve a taxi.
Environmental Needs	IT9 - IT15 succeeded

INTEGRATION TEST CASE IT20

Test case Id	IT20
Test case components	Request Manager→ Taxi Manager
Input specifications	Typical request different time and starting point
Output Specifications	Check that the taxi Manager dispatches the aforementioned taxi in the proposal, Informs taxi driver about the cancellation of the request.
Environmental Needs	IT9 - IT15 succeeded

INTEGRATION TEST CASE IT21

Test case Id	IT21
Test case components	Reservation Manager→ Taxi Manager
Input specifications	Typical request different time, date and starting point.
Output Specifications	Check that the taxi Manager dispatches the taxi related to the Time mentioned in the reservation request, Informs taxi driver about the cancellation of the reservation.
Environmental Needs	IT9 - IT15 succeeded

INTEGRATION TEST CASE IT22

Test case Id	IT22
Test case components	Taxi Manager→ Queue
Input specifications	Different taxi ids from various locations in zones
Output Specifications	Check whether the taxi is being put to the

	correct queue.
Environmental Needs	IT1, IT2, IT3 succeeded

INTEGRATION TEST PROCEDURE

Test procedure Identifier	ITP1
Purpose	This procedure verifies whether system satisfies taxi request • It checks whether the system finds appropriate zone manager for the specified starting point. • It checks whether the zone manager finds free the taxi in the queue for the request. • It checks whether the system generates appropriate taxi proposal for the received request. • It checks whether the system handles taxi payment using external payment API.
Procedure Sequence	IT1-IT9

Test procedure Identifier	ITP2
Purpose	This procedure verifies whether system satisfies taxi reservation • It checks whether the system finds appropriate zone manager for the specified starting point. • It checks whether the zone manager finds free the taxi in the queue for the reservation. • It checks whether the system generates appropriate taxi proposal for the received reservation request. • It checks whether the system handles taxi payment using external payment API.

	• It also checks whether the system sends notification to the passenger accordingly.
Procedure Sequence	IT1-IT8,IT10

Test procedure Identifier	ITP3
Purpose	This procedure verifies whether front end satisfies the following • It checks whether the user able to register a new account. • It checks whether the user able to login into his/her account with the credentials. • It checks whether the user able to request a taxi by specifying the required parameters. • It checks whether the user be able to reserve a taxi by specifying the required parameters.
Procedure Sequence	IT18-IT22

Test procedure Identifier	ITP4
Purpose	This procedure verifies whether driver app satisfies the following • It checks whether the driver app able to forward the taxi request/reservation to the driver. • It checks whether the taxi manager is able to dispatch the appropriate taxi for the request/reservation. • It checks whether the app is able to inform the driver about the cancellation of request/reservation.
Procedure Sequence	IT18-IT22

4 Tools and Test Equipment Required

Tool used for the tests is Arquillian. Arquillian is an integration testing framework for containers, that is used to execute test cases against containers. We are going to use this tool because the application will be written in Java EE, and Arquillian is a proven solution to testing Java EE applications.

Arquillian combines a unit testing framework, ShrinkWrap, and one or more supported target containers (Java EE container, servlet container, Java SE CDI environment, etc) to provide a simple, flexible and pluggable integration testing environment.

For testing purposes we will need some Android and iOS devices (both tablets and phones), with different versions of OSs. For server we don't need any special hardware, since we are going to use Amazon Cloud for the architecture, and the testing will be done there. For the Web app testing it should be tested on different browsers and screen sizes (can be done on the same machine).

5 Program Stubs, Drivers and Test Data Required

Stubs and Drivers that need to be developed are:

- FareCalculator stub - for the test IT3
- WaitCalculator stub - for the test IT4
- FrontEnd driver - for the test IT16