



Politecnico di Milano

A.Y. 2015-2016

Software Engineering 2: “myTaxiService”

Integration Test Plan Document

version 1.0

Adrian Mihai Berbieru #854698

Attilio D’Onofrio #789614

21/01/2016

INDEX

1 Introduction

1.1 Revision History

1.2 Purpose and Scope

1.3 List of Definitions and Abbreviations

1.4 List of Reference Documents

2 Integration Strategy

2.1 Entry Criteria

2.2 Elements to be Integrated

2.3 Integration Testing Strategy

2.4 Sequence of Component/Function Integration

2.4.1 Software integration sequence

2.4.2 Subsystem integration sequence

3 Individual Steps and Test Description

3.1 Long Term Booking→Request Management

3.1.1 Long term booking creation

3.2 Status Checking→Update Management

3.2.1 Change status from unavailable to available

3.3 Status Checking→Location Management

3.4 Web Application→Account Management

3.4.1 User registration via web application

3.4.2 Login via web application

3.4.3 Lost password via web application

3.5 Customer app→Account Management

3.5.1 User registration via smartphone app

3.5.2 Login via smartphone app

3.5.3 Lost password via smartphone app

3.6 Taxi Driver app→Account Management

- 3.6.1 Driver's login via app*
- 3.7 Web Application→Taxi Management*
 - 3.7.1 Instant Booking with available taxi*
 - 3.7.2 Instant Booking with unavailable taxi with booking deletion*
- 3.8 Customer app→Taxi Management*
 - 3.8.1 Instant Booking with available taxi*
 - 3.8.2 Instant Booking with unavailable taxi with booking deletion*
- 3.9 Taxi Driver app→Taxi Management*
 - 3.9.1 Accepted Call*
 - 3.9.2 Declined Call*
- 3.10 Taxi Management→Zone Management*
 - 3.10.1 Taxi driver call decline*
- 3.11 Data Access Utilities*
 - 3.11.1 Account Management→Data Access Utilities*
 - 3.11.2 Taxi Management→Data Access Utilities*
 - 3.11.3 Zone Management→Data Access Utilities*
 - 3.11.4 System Administration→Data Access Utilities*
- 4 Tools and Test Equipment Required*
- 5 Program Stubs and Test Data Required*
 - 5.1 Long Term Booking→Request Management*
 - 5.2 Status Checking→Update Management*
 - 5.3 Status Checking→Location Management*
 - 5.4 Web Application→Account Management*
 - 5.4.1 User registration via web application*
 - 5.4.2 Login via web application*
 - 5.4.3 Lost password via web application*
 - 5.5 Customer app→Account Management*
 - 5.5.1 User registration via smartphone app*
 - 5.5.2 Login via smartphone app*

- 5.5.3 Lost password via smartphone app*
- 5.6 Taxi Driver app→Account Management*
 - 5.6.1 Driver's login via app*
- 5.7 Web Application→Taxi Management*
 - 5.7.1 Instant Booking with available taxi*
 - 5.7.2 Instant Booking with unavailable taxi with booking deletion*
- 5.8 Customer app→Taxi Management*
 - 5.8.1 Instant Booking with available taxi*
 - 5.8.2 Instant Booking with unavailable taxi with booking deletion*
- 5.9 Taxi Driver app→Taxi Management*
 - 5.9.1 Accepted Call*
 - 5.9.2 Declined Call*
- 5.10 Taxi Management→Zone Management*
 - 5.10.1 Taxi driver call decline*
- 5.11 Data Access Utilities*
 - 5.11.1 Account Management→Data Access Utilities*
 - 5.11.4 System Administration→Data Access Utilities*

1 Introduction

1.1 Revision History

In this document there will be references to the RASD and DD documents (see paragraph 1.4 to read more).

- RASD version: 1.1 - which is similar to 1.0 version but with a couple of assumptions regarding the GPS signal and the tables containing the binding between city addresses and zones and with the system administrator figuring among the actors.
- DD version: 1.0 - the document is the original one.

1.2 Purpose and Scope

This document describes how the integration among all the components will be done. In particular, we will focus on testing the components interfaces.

The software manages the process of taxi booking of a large city. By adopting the component division explained in the Design Document, the interactions between the different actors that come into play will be made possible.

1.3 List of Definitions and Abbreviations

- Visitor: also called unregistered user, is a person who is not registered to the system.
- Customer: also called user or client, is the person who books the ride via the web or smartphone app.
- Device: term used to refer to PCs and smartphones in general.
- City zone: the part of the city the taxi is assigned to. Each zone has an area of about 2km².
- Availability: referred to a taxi, it says if the driver is able to take calls or not.
- Queue: the list of taxis associated to a city zone.
- Code: the identifier associated to a taxi.
- Booking: also called reservation, the act of asking for a taxi, done by a customer through the system. It may be instant or long term.

- Instant booking: also called instant reservation, it is done when a customer requests a taxi, sending his current position.
- Long term booking: also called long term reservation, it is done when a customer makes a reservation for a taxi in a specific time in the near future.
- User: whomever interacts with the system; the terms is used referring to customers and taxi drivers alike.
- Request: the message sent between client and server or between server side components.
- Active: referred to a device, it is active if it is turned on and myTaxiService application is running.
- Gn: n-goal.
- DBMS: database management system.
- AS: application server.
- PC: personal computer

1.4 List of Reference Documents

The RASD and DD documents can be found on Github at the following address:

- Project description and rules: <https://beep.metid.polimi.it/documents/ProjectDescription>;
- RASD: <https://github.com/AdrianBerbieru/INGSW2/Deliveries/RASD.pdf>;
- DD: <https://github.com/AdrianBerbieru/INGSW2/Deliveries/Design.pdf>;
- Integration Test Plan Example:
<https://beep.metid.polimi.it/documents/3343933/5b3768d0>;

2 Integration Strategy

2.1 Entry Criteria

Before the integration testing begins every single component that our system is composed of will be tested alone, which means that each class that compose our system will have gone through an extensive phase of unit testing.

We've set a minimum of 95% coverage of unit tests of the code in general so that any major flaws in the classes structure or in the algorithms implementation have been discovered and dealt with, at the same time we aim to reach 100% coverage of the system. Code inspection should have also been done in order to ensure maintainability, greater efficiency and easy to read code in order to facilitate the tester jobs. Finally the following documents must be done before the integration test can start:

- JavaDoc;
- RASD;
- DD;

We assume the Data Access Utilities and DBMS have been thoroughly tested on their own by the vendor, since it's a off-the-shelf software.

2.2 Elements to be Integrated

The main components (also known as subsystems) that our system is composed of have been identified in the Design Document at paragraph 2.2, while the main subsystems subcomponents have been described in section 2.3.

<i>Subsystem Name</i>	<i>Functional role</i>
Taxi Management	Is in charge for handling taxi rides.
Zone Management	Is in charge of managing taxi queues.
Account Management	Is responsible for all operations regarding user accounts.
System Administration	Is responsible with some system configurations (such as insertion, update, delete of vehicles and drivers and redefinition of the city boundaries).
Data Access Utilities	Offers a specific interface to every other component that wants to store or retrieve data from the database.

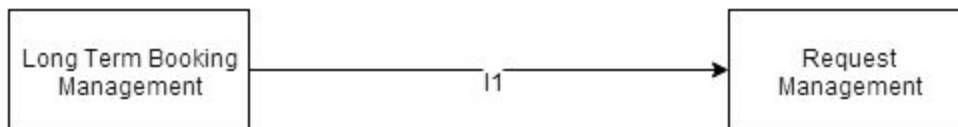
<i>Parent Subsystem</i>	<i>Name</i>	<i>Functional role</i>
Taxi Management	Long Term Booking Management	The Long Term Booking Management subcomponent handles every long term reservation a user makes.
	Request Management	The Request Management subcomponent handles every instant booking or long term that should be served.
Zone Management	Update Management	The Update Management subcomponent has to update the queues of every city zone.
	Status Checking	The Status Checking subcomponent has to create queue of new available taxis where it saves the taxi codes of the vehicles and the zone which it is located in.
	Location Management	The Location Management subcomponent is in charge with computing to which zone the coordinates of a vehicle belong to and thus add them at the end of the queue.
Account Management	User Registration	The User Registration subcomponent allows Visitors to create an account in order for them to use myTaxiService services.
	Login	The Login subcomponent allows all users to access their accounts and from then on interact with the system.
	Lost Password	The Lost Password subcomponent permits only to registered users to regain access to their account in the case they forgot the password.

2.3 Integration Testing Strategy

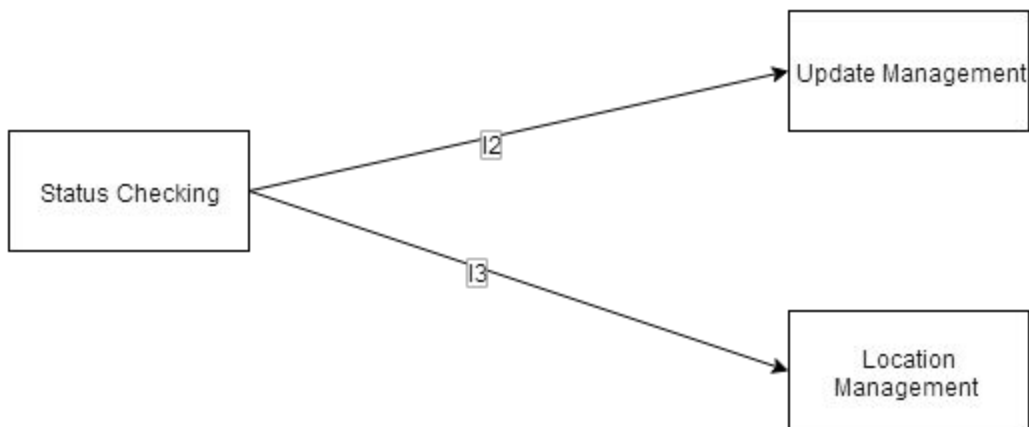
The integration strategy chosen is the sandwich approach. This choice has been made considering the relative big size of the project so that the different teams that will uncover can find issues more easily and at the same time keep track of logical ties between the modules. We will also start from testing how the different subcomponents of the main subsystems interact with each other and then move on to test how the subsystem interact with each other.

2.4 Sequence of Component/Function Integration

2.4.1 Software integration sequence



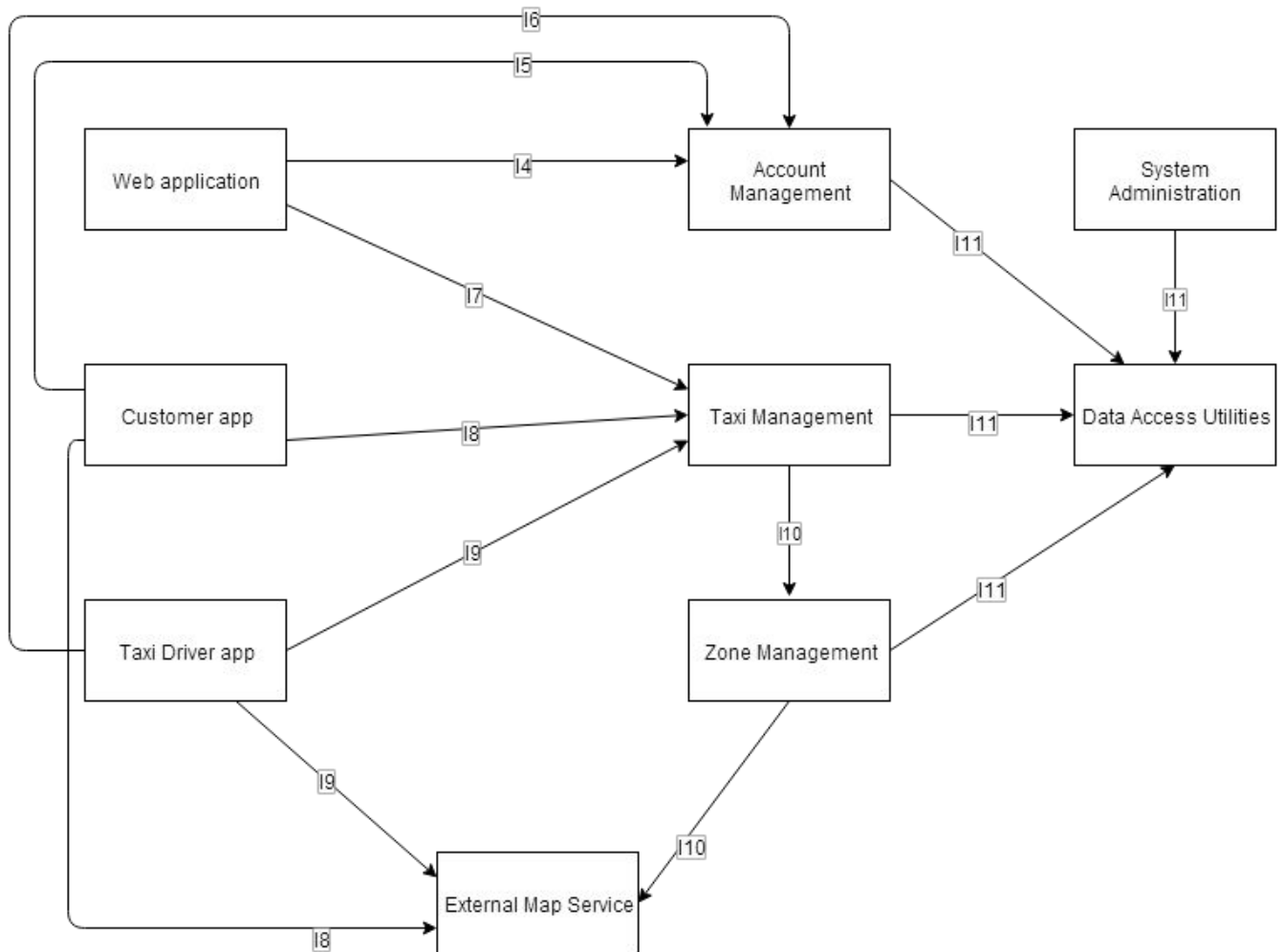
<i>ID</i>	<i>Integration test</i>	<i>Paragraphs</i>	
I1	Long Term Booking→Request Management	3.1	5.1



<i>ID</i>	<i>Integration test</i>	<i>Paragraphs</i>	
I2	Status Checking→Update Management	3.2	5.2
I3	Status Checking→Location Management	3.3	5.3

Note that the **Account Management** Components *User Registration, Login, Lost Password* do not communicate with each other therefore the unit testing is sufficient and we can test them directly in the subsystems integration tests.

2.4.2 Subsystem integration sequence



<i>ID</i>	<i>Integration test</i>	<i>Paragraphs</i>	
I4	Web Application→Account Management	3.4	5.4
I5	Customer app→Account Management	3.5	5.5
I6	Taxi Driver app→Account Management	3.6	5.6
I7	Web Application→Taxi Management	3.7	5.7
I8	Customer app→Taxi Management	3.8	5.8
I9	Taxi Driver app→Taxi Management	3.9	5.9
I10	Taxi Management→Zone Management	3.10	5.10
I11	Account Management→Data Access Utilities, Taxi Management→Data Access Utilities, Zone Management→Data Access Utilities, System Administration→Data Access Utilities	3.11.1 3.9 3.9 3.11.4	5.11.1 5.11.4

3 Individual Steps and Test Description

As the testing phase goes on integrating higher level components, the granularity of the description will become coarse, since we will be focusing on the process as a whole and not on the low-level steps that will be tested first.

The external map service will be tested together with the other components in steps I8,I9 and I10. We assume the steps are consecutive, which means that I8 cannot be done until the previous integration testing steps (I7,I6 and so on up to I1) have been done.

3.1 Long Term Booking→Request Management

3.1.1 Long term booking creation

Purpose	To ensure only valid long term bookings are served
External Dependencies	The device used to make the booking is connected to the Web
Test Description	1)Log in myTaxiService application 2)Receive ride details ten minutes before the scheduled time
Expected Results	As long as web connectivity exists, the user is able to read the details of the booking

3.2 Status Checking→Update Management

3.2.1 Change status from unavailable to available

Purpose	To ensure taxi driver's status switch works properly
External Dependencies	The driver's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Log in myTaxiService application 2)Set the status from off to on
Expected Results	As long as web connectivity exists, the system puts the taxi at the bottom of the queue

3.3 Status Checking→Location Management

Purpose	To ensure taxi zone changes according to its position
External Dependencies	The driver's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Log in myTaxiService application 2)Set the status from on to off
Expected Results	As long as web connectivity exists, the system extracts the address and puts the taxi in the appropriate queue

3.4 Web Application→Account Management

3.4.1 User registration via web application

Purpose	To ensure a visitor is able to register to the system
External Dependencies	The user's PC is connected to the Web
Test Description	1)Open myTaxiService web application 2)Click on the signup button 3)Fill out the form with required data 4)Send the request
Expected Results	As long as data provided by user are correct, the system creates the new account

3.4.2 Login via web application

Purpose	To ensure a registered user can login
External Dependencies	The user's PC is connected to the Web
Test Description	1)Open myTaxiService web application 2)Fill out the form with email address and password 3)Send the request
Expected Results	As long as the couple email and password hash provided by user is correct, he is able to log in into the web application

3.4.3 Lost password via web application

Purpose	To ensure the password recovery process works correctly
External Dependencies	The user's PC is connected to the Web
Test Description	1)Open myTaxiService web application 2)Click on password recovery button 3)Fill out the form with the email address 4)Send the request
Expected Results	As long as email address is already registered, the system sends an email containing a link through which user is able to reset the password

3.5 Customer app→Account Management

3.5.1 User registration via smartphone app

Purpose	To ensure a visitor is able to register to the system
External Dependencies	The user's smartphone is connected to the Web
Test Description	1)Open myTaxiService app 2)Click on the signup button 3)Fill out the form with required data 4)Send the request
Expected Results	As long as data provided by user are correct, the system creates the new account

3.5.2 Login via smartphone app

Purpose	To ensure a registered user can login
External Dependencies	The user's smartphone is connected to the Web
Test Description	1)Open myTaxiService app 2)Fill out the form with email address and password 3)Send the request
Expected Results	As long as the couple email and password hash provided by user is correct, he is able to log in into the web application

3.5.3 Lost password via smartphone app

Purpose	To ensure the password recovery process works correctly
External Dependencies	The user's smartphone is connected to the Web
Test Description	1)Open myTaxiService app 2)Click on password recovery button 3)Fill out the form with the email address 4)Send the request
Expected Results	As long as email address is already registered, the system sends an email containing a link to a page where a user is able to reset the password

3.6 Taxi Driver app→Account Management

3.6.1 Driver's login via app

Purpose	To ensure a registered taxi driver can login
External Dependencies	The driver's smartphone is connected to the Web
Test Description	1)Open myTaxiService app 2)Fill out the form with taxi ID and password 3)Send the request
Expected Results	As long as the couple taxi ID and password hash provided by user is correct, he is able to log in into the web application

3.7 Web Application→Taxi Management

3.7.1 Instant Booking with available taxi

Purpose	To ensure a registered user can make an instant booking
External Dependencies	The user's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Select to book a taxi now 3)Fill out the form with required data 4)Send the request
Expected Results	As long as the provided address is valid, the system looks for a queue in that city zone

3.7.2 Instant Booking with unavailable taxi with booking deletion

Purpose	To ensure a registered user can make an instant booking
External Dependencies	The user's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Select to book a taxi now 3)Fill out the form with required data 4)Send the request 5)Receive notification of taxi unavailability 6>Delete taxi request
Expected Results	As long as the provided address is valid, the system looks for a queue in that city zone and there are no taxis available. During the wait, user decides to delete the booking.

3.7.3 Long term booking deletion

Purpose	To ensure existing reservation may be deleted
External Dependencies	The PC is connected to the Web
Test Description	1)Log in myTaxiService application 2)Select see pending reservations 3)Delete one of the existing reservations
Expected Results	As long as web connectivity exists, the system checks if the selected reservation is two hours away and in that case deletes it

3.8 Customer app→Taxi Management

3.8.1 Instant Booking with available taxi

Purpose	To ensure a registered user can make an instant booking
External Dependencies	The user's PC is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Select to book a taxi now 3)Fill out the form with required data and use the GPS to detect the current position 4)Send the request
Expected Results	As long as the provided address is valid, the system looks for a queue in that city zone

3.8.2 Instant Booking with unavailable taxi with booking deletion

Purpose	To ensure a registered user can make an instant booking
External Dependencies	The user's PC is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Select to book a taxi now 3)Fill out the form with required data 4)Send the request 5)Receive notification of taxi unavailability 6>Delete taxi request
Expected Results	As long as the provided address is valid, the system looks for a queue in that city zone and there are no taxis available. During the wait, user decides to delete the booking.

3.8.3 Long term booking with taxi deletion

Purpose	To ensure existing reservation may be deleted
External Dependencies	The smartphone used to login into the account is connected to the Web
Test Description	1)Log in myTaxiService application 2)Select see pending reservations 3)Delete one of the existing reservations
Expected Results	As long as web connectivity exists, the system checks if the selected reservation is two hours away and in that case deletes it

3.9 Taxi Driver app→Taxi Management

3.9.1 Accepted Call

Purpose	To ensure taxi drivers can accept calls
External Dependencies	The driver's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Switch status to available 3)Accept the incoming call 4)Switch status to unavailable 5)Use the map to reach the customer 6)Finish the ride
Expected Results	The process described above must be flawless so that the driver is able to reach the customer in an acceptable amount of time

3.9.2 Declined Call

Purpose	To ensure taxi drivers can decline calls
External Dependencies	The driver's smartphone is connected to the Web and GPS tracking is active
Test Description	1)Open and log in myTaxiService app 2)Switch status to available 3)Decline the incoming call
Expected Results	The process described above must be flawless so that the act of declining is promptly communicated to the system and the request is sent to another taxi

3.10 Taxi Management→Zone Management

3.10.1 Taxi driver call decline

Purpose	To ensure queues are up to date
External Dependencies	Driver's device is connected to the Web and GPS tracking is active
Test Description	1)Taxi driver opens and login into the app 2)He receives a call 3)He declines the call
Expected Results	The system puts the driver in the last position of the queue

3.11 Data Access Utilities

3.11.1 Account Management→Data Access Utilities

Purpose	Actions related to registered users' accounts are properly managed
External Dependencies	Customer's device is connected to the Web
Test Description	1)A visitor completes the signup 2)The visitor forgets the password and resets it 3)The visitor logs into the application
Expected Results	The user must eventually login

3.11.2 Taxi Management→Data Access Utilities

(Testing already done in paragraph 3.9)

3.11.3 Zone Management→Data Access Utilities

(Testing already done in paragraph 3.9)

3.11.4 System Administration→Data Access Utilities

Purpose	The addition of taxi drivers works properly
External Dependencies	System administrator is connected to the Web
Test Description	1)System administrator authenticates to the system 2)He adds to the DB the new taxi driver
Expected Results	There is a new driver in the driver's table

4 Tools and Test Equipment Required

For the integration testing we've used the following tools in order to automate the procedure:

- ***JUnit***, is the most popular tool used for unit testing java classes but also offers integration testing features;
- ***Mockito***, is an open source framework that generates mock objects, stubs and drivers. It is designed as a open source testing framework for Java which is available under a MIT License. Mockito allows programmers to create and test double objects (mock objects) in automated unit tests for the purpose of Test-driven Development (TDD).
- ***Arquillian***, is a highly innovative and extendible testing platform for JVM that allows developers to easily create automated integration, functional and acceptance tests for Java. Arquillian allows you to run test in the run-time so you don't have to manage the run-time from the test. It can be used to manage the lifecycle of the container (or containers), bundling test cases, dependent classes and resources. Arquillian integrates with familiar testing frameworks such as JUnit, TestNG and allows tests to be launched using existing IDE, and because of its modular design it is capable of running Ant and Maven test plugins.
- ***TestNG***, is a Java testing framework, inspired by JUnit. It overcomes the limitations and drawbacks of JUnit and introduces a whole new set of features, making TestNG more powerful and user-friendly. From simple unit testing to complex integrated testing, it is designed to simplify all your testing requirements such as functional testing, regression, end-to-end, and more.

5 Program Stubs and Test Data Required

5.1 Long Term Booking→Request Management

Stub(s)	<ul style="list-style-type: none">• Customer device• DBMS
Test data	<ul style="list-style-type: none">• Taxi queue• Customer reservation data
Description	The aim of this test is to ensure the customer gets a notification of the ride before the scheduled time

5.2 Status Checking→Update Management

Stub(s)	<ul style="list-style-type: none">• Taxi driver device• DBMS
Test data	<ul style="list-style-type: none">• Taxi driver data and location• Taxi queue
Description	The aim of this test is to ensure the switch of the status from unavailable to available adds the taxi at the end of the queue

5.3 Status Checking→Location Management

Stub(s)	<ul style="list-style-type: none">• Taxi driver device• DBMS
Test data	<ul style="list-style-type: none">• Taxi driver data and location• Taxi queue
Description	The aim of this test is to ensure the switch of the status puts the taxi in the appropriate queue

5.4 Web Application→Account Management

5.4.1 User registration via web application

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer data and email address
Description	The aim of this test is to ensure the signup process done via PC works

5.4.2 Login via web application

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer email address and password
Description	The aim of this test is to ensure the login process done via PC works

5.4.3 Lost password via web application

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer email address
Description	The aim of this test is to ensure the password recovery process done via PC works

5.5 Customer app→Account Management

5.5.1 User registration via smartphone app

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer data and email address
Description	The aim of this test is to ensure the signup process done via smartphone app works

5.5.2 Login via smartphone app

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer email address and password
Description	The aim of this test is to ensure the login process done via smartphone app works

5.5.3 Lost password via smartphone app

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer email address
Description	The aim of this test is to ensure the password recovery process done via smartphone app works

5.6 Taxi Driver app→Account Management

5.6.1 Driver's login via app

Stub(s)	<ul style="list-style-type: none">• Taxi driver smartphone• DBMS
Test data	<ul style="list-style-type: none">• Taxi driver data
Description	The aim of this test is to ensure the login process done by drivers works

5.7 Web Application→Taxi Management

5.7.1 Instant Booking with available taxi

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Taxi queue• Customer position• Reservation data
Description	The aim of this test is to ensure instant booking process done by customers via web application works when taxis are available

5.7.2 Instant Booking with unavailable taxi with booking deletion

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Taxi queue• Customer position• Reservation data
Description	The aim of this test is to ensure that during the instant booking process done by customers via web application, when taxis are unavailable the customer is able to delete the request

5.7.3 Long term booking deletion

Stub(s)	<ul style="list-style-type: none">• Customer PC• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Long term reservation
Description	The aim of this test is to ensure that a long term reservation via the web application can be deleted by the customer

5.8 Customer app→Taxi Management

5.8.1 Instant Booking with available taxi

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Taxi queue• Customer position• Reservation data
Description	The aim of this test is to ensure instant booking process done by customers via the app works when taxis are available

5.8.2 Instant Booking with unavailable taxi with booking deletion

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Taxi queue• Customer position• Reservation data
Description	The aim of this test is to ensure that during the instant booking process done by customers via the app, when taxis are unavailable the customer is able to delete the request

5.8.3 Long term booking deletion

Stub(s)	<ul style="list-style-type: none">• Customer smartphone app• DBMS
Test data	<ul style="list-style-type: none">• Customer data• Long term reservation
Description	The aim of this test is to ensure that a long term reservation can be deleted by the customer via the app

5.9 Taxi Driver app→Taxi Management

5.9.1 Accepted Call

Stub(s)	<ul style="list-style-type: none">• Taxi driver app• DBMS
Test data	<ul style="list-style-type: none">• Reservation request data• Taxi driver data• Taxi queue
Description	The aim of this test is to ensure that a reservation can be accepted by a taxi driver

5.9.2 Declined Call

Stub(s)	<ul style="list-style-type: none">• Taxi driver app• DBMS
Test data	<ul style="list-style-type: none">• Reservation request data• Taxi driver data• Taxi queue
Description	The aim of this test is to ensure that a declined ride request is sent to another taxi driver

5.10 Taxi Management→Zone Management

5.10.1 Taxi driver call decline

Stub(s)	<ul style="list-style-type: none">• Taxi driver app• DBMS
Test data	<ul style="list-style-type: none">• Reservation request data• Taxi driver data• Taxi queue
Description	The aim of this test is to ensure that when a taxi driver declines a ride request its taxi is put at the end of the queue

5.11 Data Access Utilities

5.11.1 Account Management→Data Access Utilities

Stub(s)	<ul style="list-style-type: none">• Customer device (smartphone or PC)
Test data	<ul style="list-style-type: none">• Customer email address, name and password
Description	The aim of this test is to ensure that queries are sent to the DBMS and served with a response (e.g. a set of tuples)

5.11.4 System Administration→Data Access Utilities

Stub(s)	<ul style="list-style-type: none">• Administrator PC
Test data	<ul style="list-style-type: none">• Taxi driver data• Administrator authentication data
Description	The aim of this test is to ensure that a system administrator is able to add a new taxi driver to the system and alter the city's subdivisions table

6 Appendix

Hours spent

- Adrian Mihai Berbieru: ~8 hours
- Attilio D'Onofrio: ~8 hours