SE2 Integration Test Plan Document

Edoardo Giacomello Mattia Fontana

January 21, 2016

Contents

1 Introduction

1.1 Revision History

1.2 Purpose and Scope

1.2.1 Purpose

The purpose of this document is to provide a plan for integration testing for the MyTaxiServiceSystem.
This document is intended to all the persons which are in charge to write and execute the tests, and it's intended to be a reference for the tests that are to be scheduled.

1.2.2 Scope

The MyTaxiServiceapplication is a system for managing taxi for a municipal environment. The product is composed of several parts and components that will be described in this document and must cooperate as expected in order to not run into malfunctions and unforeseen behaviours.

1.3 List of Definitions and Abbreviations

1.4 List of Reference Documents

- Assigment 1: Project Description
- MyTaxiServiceRequirement and Specification Analysis Document
- MyTaxiServiceDesign Document

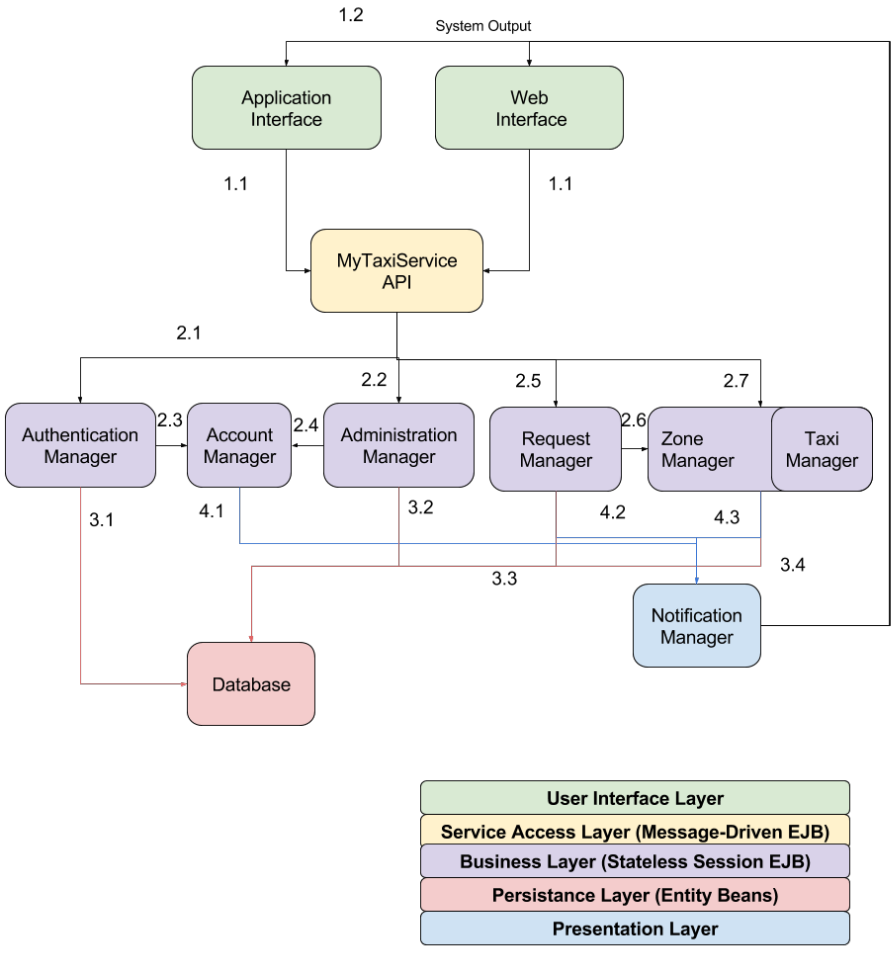
2 Integration Strategy

2.1 Entry Criteria

The entry criteria for the integration phase are the following:

- All method groups have already been unit-tested, with particular attention to the taxi and request management algorithm;
- All Interfaces implementation belongs to the classes as specified in the design document, therefore unit-tests also includes the implemented members;
- The classes that are not detailed in the section Element to be integrated must have been already unit tested since they present a high level of coupling and can work as a single component;
- The data model for the database has been already tested.

2.2 Elements to be Integrated



2.2.1 User Interface Layer

Web Interface / Application Interface

- Passengers
  - Registration
  - Login
  - Map
  - Request
  - Reservation
  - Pending Reservation
  - Information
  - Profile
  - Logout
- Taxi Drivers
  - Login
  - Map
  - Request Incoming
  - Location Update
  - Active Ride
  - Ride Confirmation
  - Logout
- System Administrators
  - Login
  - Taxi Management
  - Account Management
  - Logs
  - Backup/Restore
  - Logout

2.2.2 Service Access Layer

- MyTaxiService API

2.2.3 Business Layer

- AuthenticationManager (Unit tested as a single component)
- AdministrationManager
  - AdministrationController
- AccountManager
  - AccountController
- RequestManager
  - RequestResolver
  - ReservationController
- ZoneManager and TaxiManager (Referenced as a single unit, all TaxiManager interfaces are implemented by ZoneManager by design)
  - ZoneController

2.2.4 Presentation Layer

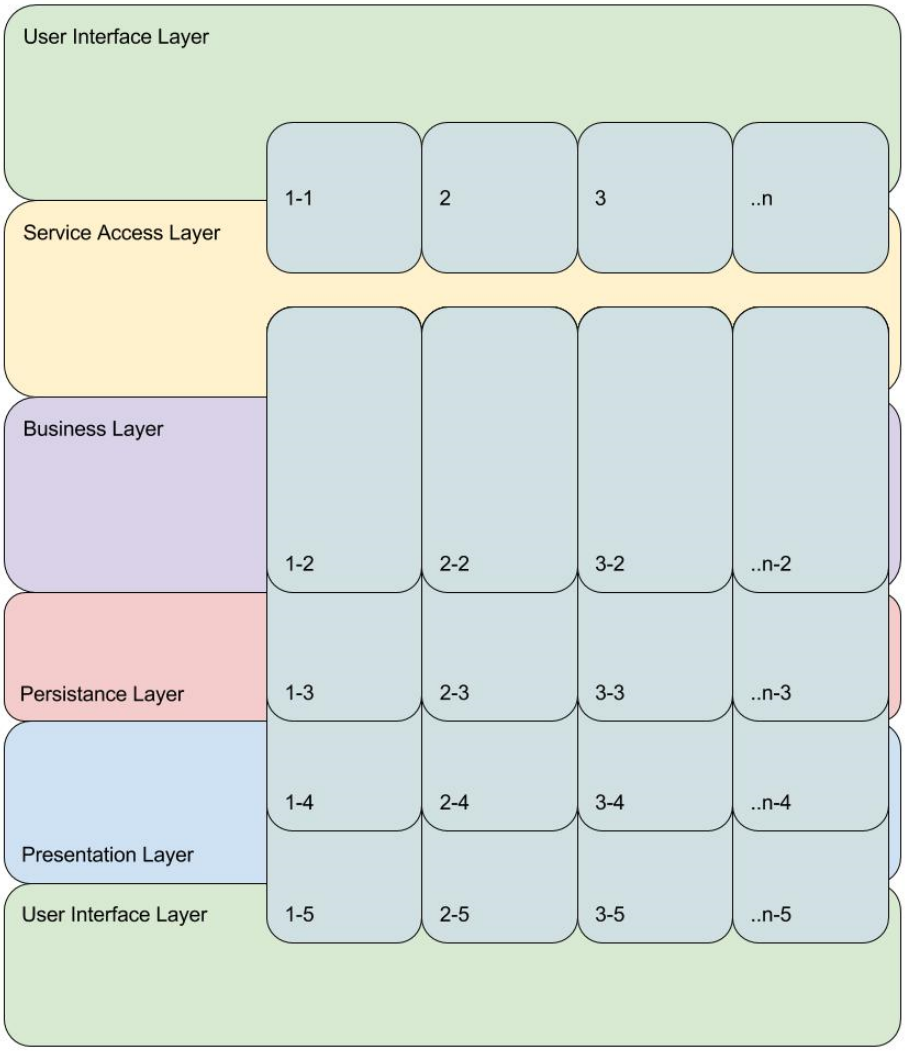
- Notification Manager

2.2.5 Persistance Layer

- Database

2.3 Integration Testing Strategy

The adopted test strategy will be **Top-Down** with **functional grouping**. In particular, the first phase will comprehend the user interface and the issue of messages toward the server APIs. Once that the messages are proven to be consistent with the specifications of the APIs for each functionality, it will be possible to test the interaction between the API container and the other business components by testing each functionality separately, stepping down to the persistence layer. Then it will be possible to test the presentation layer, that is the set of components that provides the output to the client. A diagram describing the process, given n functionalities, is given below:



2.3.1 Rationale

This strategy has been adopted because of the hierarchical structure of the project, that allows to modularize the testing process. In particular, the system API interface allows to separate the user interface testing from the backend testing by writing appropriate stubs for the former and a client driver and backend stubs for the latter. Another advantage of this strategy is the possibility to run the tests in parallel if more than one team is working together, by assigning each functionality to different teams.

2.4 Sequence of Component/Function Integration

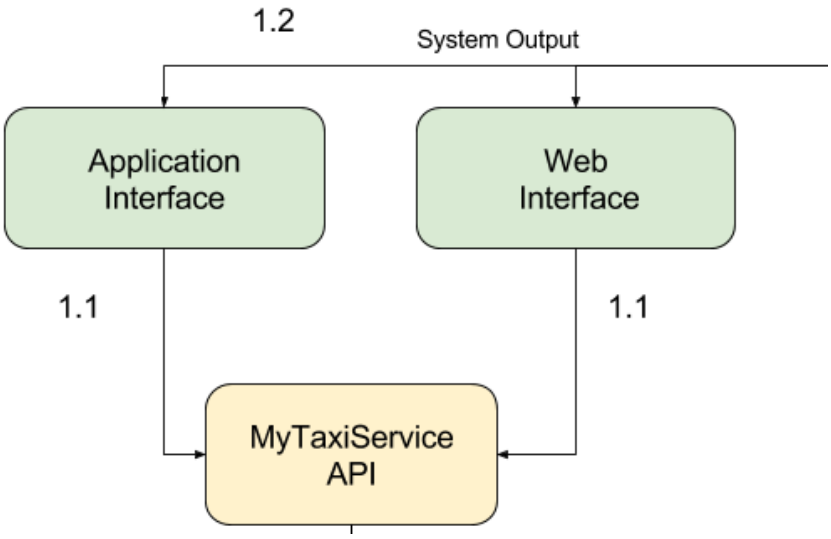
The first phase of the integration testing will consider the User Interface Layer against the Service Access Layer (MyTaxiService APIs). In particular, the User Interface Layer is composed by the following subsystems:

- Passenger Mobile/Web Application
- Taxi Driver Mobile Application
- Administrator Web Interface

For the test concerning the Passenger application there will be noted only a single instance; it is intended that both the web and mobile application test must be runned.

2.5 Software Integration Sequence

2.5.1 Phase 1: User Interface I/O



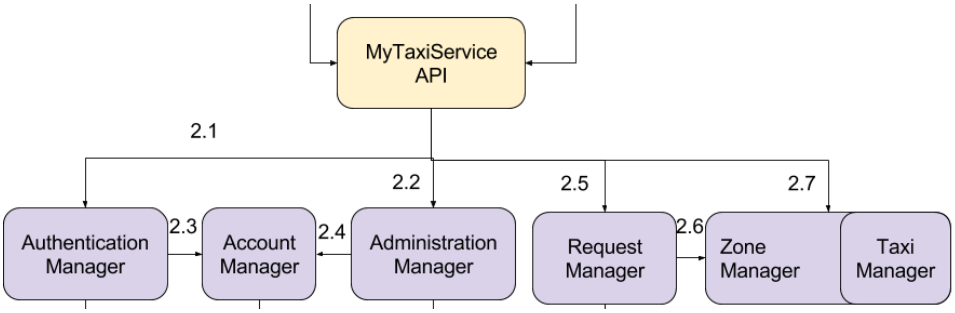
Source Component	Destination Component	Functionality	Reference
Passenger Interface	APIs	Passenger Registration	3.1.2
Passenger Interface	APIs	Passenger Login	3.1.1
Taxi Interface	APIs	Taxi Login	3.1.10
Passenger Interface	APIs	Logout	3.1.6
Taxi Interface	APIs	Logout	3.1.12
Passenger Interface	APIs	Map Request	3.1.3
Passenger Interface	APIs	Request and Reservation	3.1.2, 3.1.5
Passenger Interface	APIs	Pending Reservations	3.1.7
Passenger Interface	APIs	Profile	3.1.8
Taxi Interface	APIs	Driver Response	3.1.14
Taxi Interface	APIs	Driver Notification	3.1.14
Administrator Interface	APIs	Taxi Management	3.1.15
Administrator Interface	APIs	Account Management	3.1.16
Administrator Interface	APIs	Database Management	3.1.18, 3.1.19

For the reference of this set of Tests, please refer to the corresponding inverse test in the previous table

Source Component	**Destination Component**	**Functionality**	**Reference**
Administrator Interface	MyTaxiService APIs	Log Management	3.1.17
MyTaxiService APIs	Passenger Interface	Login Logout Registration TaxiProbeResponse TaxiConfirmation Notification RequestList Account Information.	3.1.1 3.1.6 3.1.2 3.1.14 3.1.14 3.1.7 3.1.9 3.2.9
MyTaxiService APIs	Taxi Interface	Login Logout DriverRequest.	3.1.10 3.1.12 3.1.14 3.2.15
MyTaxiService APIs	Administrator Interface	Login Logout TaxiList AccountList LogList.	3.1.15 3.1.16 3.1.17

Once this set of test is passed, it is possible to set up the integration tests for the backend.

2.5.2 Phase 2:Business Layer Integration tests

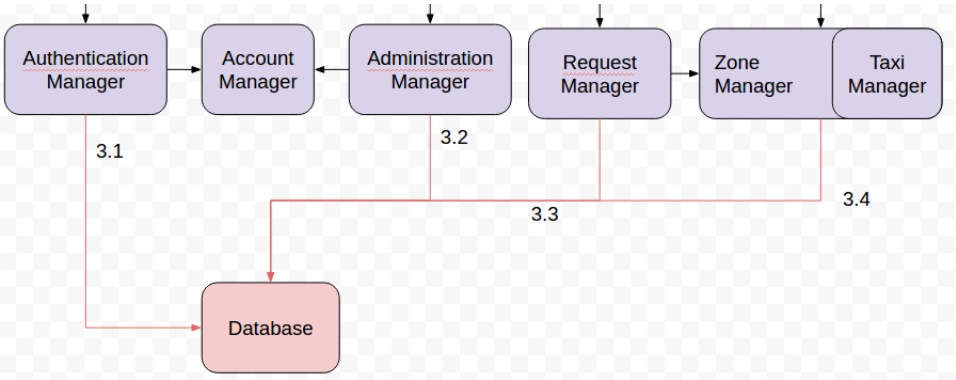


All of these tests require a Client Driver that simulates the client requests intercepted by the APIs. An appropriate Stub has to be written in order to emulate the behaviour of lower layers, in particular the business components that are still not tested and the database.

Source Component	**Destination Component**	**Functionality**	**Reference**
MyTaxiService APIs	Authentication Manager	Login Logout Account Registration APIRegistration.	3.2.1 PLog2.1 3.2.1 SLog2.1 3.2.1 TLog2.1 3.2.2 PReg2.1 3.2.2 AReg2.1
MyTaxiService APIs	Administration Manager	AddTaxi RemoveTaxi EditTaxi AddAccount EditAccount Backup Restore Logs.	3.2.3 TAdd2.2 3.2.3 TRem2.2 3.2.3 TEdi2.2 3.2.4 AAdd2.2 3.2.4 AEdi2.2 3.2.5 ALis2.2 3.2.6 SBac2.2 3.2.6 SRes2.2 3.2.7 SLogs2.2
Authentication Manager	Account Manager	GetAccount EditAccount AddAccount RemoveAccount	3.1.16 SAcc1 3.2.8 SAcc2.3 3.2.10 SAcc2.4
API	Account Manager	RequestList	3.2.9 PReqList2.3 3.3.3 Req3.3

Source Component	**Destination Component**	**Functionality**	**Reference**
Administration Manager	Account Manager	EditAccount AddAccount RemoveAccount	3.2.9 PReqList2.3 3.3.3 Req3.3
MyTaxiService APIs	RequestManager	TaxiRequest TaxiReservation RemoveRequest TaxiProbe.	3.2.15 TReqEvents2.7
RequestManager	ZoneManager	getTaxiInZone getZones getAdjacents	3.2.14 ZTaxi2.6 3.2.14 ZZone2.6 3.2.14 ZAdj2.6
MyTaxiService APIs	TaxiManager	OnRequestAccepted OnRequestRefused TaxiAvailable UpdateLocation OnZoneEnter OnZoneExit	3.4.1 TReq4.2 3.4 RConf4.2 3.4 RNotf4.2 3.2.17 TLoc2.7 3.2.16 TNotif2.7

2.5.3 Phase 3: Persistence Level Integration

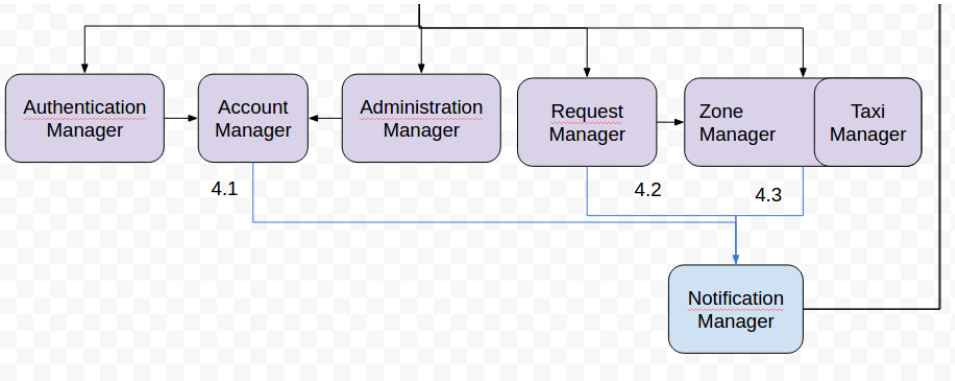


Now every component that accesses the business data have to be tested against the database.

Source Component	**Destination Component**	**Functionality**	**Reference**
Autenthication Manager	Database	Account Retrieval for Login	3.3 Aut3.1
Account Manager	Database	Account Data Editing	3.3.1 Acc3.5
Administration Manager	Database	Log Access Taxi Data Access	3.3.2 Adm3.2
Request Manager	Database	Request and Reservation data access	3.3.3 Req3.3
Zone Manager	Database	Zone Data Access Taxi Data Access	3.3.4 Zon3.4

2.5.4 Phase 4: Presentation Layer Integration

This layer checks if the component that generates the system output use a format that is compliant with specifications. The test data is sent to a Client Interface Stub since the actual system output visualization should have been already tested in Phase 1.



Source Component	Destination Component	Functionality	Reference
Request Manager	Notification Manager	Request Confirmation, Request Notification, Driver Request, TaxiProbeResponse	3.4 Rconf4.23.4
Account Manager	Notification Manager	Registration Confirmation, Password Reset	3.4.3 TReg4.1
Notification Manager	APIs	Output Message Generation	3.4.4 SOut4.4
Zone Manager	APIs	GetZone for 3rd-party applications	3.4.5 SDev4.3

3 Individual Steps and Test Description

3.1 User Interface vs API tests

3.1.1 Passenger Login

Test Case Identifier	PLog1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Login → AutenticationRequests
Input Specification	Login email and password
Output Specification	The data passed to the AuthenticationRequests are consistent and fulfils specifications. The HomePage is shown after successful login. An error is shown if the login fails.
Environmental Needs	AuthenticationManager Stub

3.1.2 Passenger Registration

Test Case Identifier	PReg1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Registration → AutenticationRequests
Input Specification	User Registration Data
Output Specification	The data passed to the AuthenticationRequests are consistent and fulfils specifications. A confirmation is shown if the registration is successful. An error is shown if the registration fails (duplicate account or missing data).
Environmental Needs	AuthenticationManager Stub

3.1.3 Passenger Map

Test Case Identifier	PMap1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Map → PassengerRequests
Input Specification	TaxiProbe Specifications
Output Specification	The TaxiProbe request fulfills the specifications of the API. The list of taxi positions is retrieved if the request is successful (TaxiProbeResponse). A default map position is sent if the request fails.
Environmental Needs	RequestInteface Stub PLog1 test has to be successful

3.1.4 Passenger Taxi Request

Test Case Identifier	PReq1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Request → PassengerRequests
Input Specification	Request Specifications
Output Specification	The Request object fulfills the specifications of the taxiRequest method. A taxiConfirmation message is received if the request is successful. An error message is shown if the request fails.
Environmental Needs	RequestInteface Stub PLog1 test has to be successful

3.1.5 Passenger Taxi Reservation

Test Case Identifier	PRes1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Reservation → PassengerRequests
Input Specification	ReservationData specification (Email, currentTime, meetingTime, meetingLocation, arrivalLocation)
Output Specification	The Reservation object fulfills the specifications of the taxiReservation method. A taxiConfirmation message is received if the reservation is successful. An error message is shown if the reservation fails.
Environmental Needs	RequestInteface Stub PLog1 test has to be successful

3.1.6 Passenger Logout

Test Case Identifier	PLog1.2
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Logout → AutenticationRequests
Input Specification	/
Output Specification	The page of Login is shown after successful logout. An error is shown if the logout fails.
Environmental Needs	AuthenticationManager Stub PLog1 Test has to be successful

3.1.7 Passenger Pending Reservation

Test Case Identifier	PPen1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	PendingReservation → PassengerRequest
Input Specification	/
Output Specification	The list of reservation is displayed. An error message is shown if the visualization fails.
Environmental Needs	RequestInteface Stub PLog1 test has to be successful.

3.1.8 Passenger Profile

Test Case Identifier	PPro1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Profile → PassengerRequest
Input Specification	/
Output Specification	User Account information. An error message is shown if the visualization fails.
Environmental Needs	RequestInteface Stub PLog1 test has to be successful.

3.1.9 Passenger Information

Test Case Identifier	PInf1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Information → PassengerRequest
Input Specification	/
Output Specification	Information about the application are visualized. An error message is shown if the visualization fails.
Environmental Needs	RequestInteface Stub

3.1.10 Taxi Driver Login

Test Case Identifier	TLog1
Integrated Components	Application Interface → MyTaxiService API
Test Items	Login → AutenticationRequests
Input Specification	Login email and password
Output Specification	The data passed to the AuthenticationRequests are consistent and fulfills specifications. The HomePage is shown after successful login. An error is shown if the login fails.
Environmental Needs	AuthenticationManager Stub

3.1.11 Taxi Driver Active Ride

Test Case Identifier	TAct1
Integrated Components	Application Interface → MyTaxiService API
Test Items	ActiveRide → TaxiRequests
Input Specification	Complete or Release ride.
Output Specification	Information about the ride are visualized along with two options for completing or releasing the ride. An error message is shown if the visualization fails.
Environmental Needs	TaxiManager stub. TLog1 test has to be successful.

3.1.12 Taxi Driver Logout

Test Case Identifier	TLog1.2
Integrated Components	Application Interface → MyTaxiService API
Test Items	Logout → AutenticationRequests
Input Specification	/
Output Specification	The page of Login is shown after successful logout. An error is shown if the logout fails .
Environmental Needs	AuthenticationManager Stub TLog1 Test has to be successful

3.1.13 Taxi Driver Map

Test Case Identifier	TMap1
Integrated Components	Application Interface → MyTaxiService API
Test Items	Map → TaxiRequests
Input Specification	TaxiProbe Specifications
Output Specification	The location of the current incoming request is shown on the map.
Environmental Needs	ZoneManager Stub TLog1 test has to be successful.

3.1.14 Taxi Driver Request Incoming

Test Case Identifier	TReq1
Integrated Components	Application Interface → MyTaxiService API
Test Items	RequestIncoming → TaxiRequests
Input Specification	Accept or Refuse ride.
Output Specification	Information about the ride are visualized along with two options for completing or releasing the ride. An error message is shown if the visualization fails.
Environmental Needs	TaxiManager stub. TLog1 test has to be successful.

3.1.15 System Administrator Taxi Management

Test Case Identifier	STax1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	TaxiManagement → AdministratorRequests
Input Specification	Add and Remove options are available.
Output Specification	The list of registered taxi drivers is shown. An error message is shown if the visualization fails.
Environmental Needs	AdministratorManager stub.

3.1.16 System Administrator Account Management

Test Case Identifier	SAcc1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	AccountManagement → AdministratorRequests
Input Specification	Add and Remove options are available.
Output Specification	The list of registered accounts is shown
Environmental Needs	AdministratorManager stub.

3.1.17 System Administrator Logs

Test Case Identifier	SLog1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Logs→ AdministratorRequests
Input Specification	/
Output Specification	The system logs are shown. An error message is shown if the visualization fails.
Environmental Needs	AdministratorManager stub.

3.1.18 System Administrator Backup

Test Case Identifier	SBac1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Backup → AdministratorRequests
Input Specification	/
Output Specification	An error message is shown if the backup fails.
Environmental Needs	AdministratorManager stub.

3.1.19 System Administrator Restore

Test Case Identifier	SRes1
Integrated Components	Web Interface → MyTaxiService API Application Interface → MyTaxiService API
Test Items	Restore → AdministratorRequests
Input Specification	A previously stored backup
Output Specification	An error message is shown if the restore of a backup fails.
Environmental Needs	AdministratorManager stub.

3.2 Business Layer Integration

3.2.1 AuthenticationManager Integration

This component receives requests from the API interface on behalf of the client. Its scope is to manage the registration, login and logout processes for all kind of users.

Test Case Identifier	PLog2.1, SLog2.1, TLog2.1
Integrated Components	API → Authentication Manager
Test Items	Authentication Request → System Authenticator (apiLogin)
Input Specification	Well-formed Account objects and their subclasses
Output Specification	The component should generate a valid token on login if the user is authorized to access the resources The component should invalidate active tokens on logout
Environmental Needs	Phase 1 Passed, Client Driver, AccountManager Stub

3.2.2

Test Case Identifier	PReg2.1, AReg2.1
Integrated Components	API → Authentication Manager
Test Items	Authentication Requests → System Authenticator (userRegistration, apiSubscribe)
Input Specification	Well-formed Account objects and their subclasses
Output Specification	The component should request the AccountManager to store the new account and generate a new valid token, or launch an exception instead
Environmental Needs	Phase 1 Passed, Client Driver, AccountManager Stub

3.2.3 AdministrationManager integration

This component manages the System Administrator requests and fulfils all the maintenance and administrative operations. All these operations can be done only as logged as system administrator; for this reason the AuthenticationManager should be tested first.

Test Case Identifier	TAdd2.2, TRem2.2, TEdi2.2
Integrated Components	API → Administration Manager
Test Items	Administrator Requests → Administration Controller
Input Specification	Well-Formed administrator Account, Well-Formed Taxi Object
Output Specification	The TaxiManager Should be requested to Add, remove or Edit the Taxi data if the administrator is logged and the new data are well-formed or throw an exception instead
Environmental Needs	Phase 1, SLog2.1, TLog2.1 passed Client Driver, TaxiManager Stub

3.2.4

Test Case Identifier	AAdd2.2, AEdi2.2
Integrated Components	API → Administration Manager
Test Items	Administrator Requests → Administration Controller
Input Specification	Well-Formed Administrator Account, Well-Formed Account to Edit
Output Specification	The AccountManager Should be requested to Add, Remove or Edit the account data if the administrator is logged and the new data are well-formed or throw an exception instead
Environmental Needs	Phase 1, SLog2.1 passed Client Driver, AccountManager Stub

3.2.5

Test Case Identifier	ALis2.2
Integrated Components	API → Administration Manager
Test Items	Administrator Requests → Administration Controller (List Account)
Input Specification	Well-Formed Administrator Account
Output Specification	The AccountManager Should be requested to retrieve the account list if the administrator is logged or throw an exception instead
Environmental Needs	Phase 1, SLog2.1 passed Client Driver, AccountManager Stub

3.2.6

Test Case Identifier	SBac2.2, SRes2.2
Integrated Components	API → Administration Manager
Test Items	Administrator Requests → Administration Controller (DB Backup/Restore)
Input Specification	Well-Formed Administrator Account
Output Specification	The DatabaseManager Should be requested to backup/Restore the account list if the administrator is logged and the database parameters are compliant with the system or throw an exception instead
Environmental Needs	Phase 1, SLog2.1 passed Client Driver, AccountManager Stub

3.2.7

Test Case Identifier	SLogs2.2
Integrated Components	API → Administration Manager
Test Items	Administrator Requests → Administration Controller (showLogs)
Input Specification	Well-Formed Administrator Account
Output Specification	The DatabaseManager Should be requested to show system logs if the administrator is logged and the log files available or throw an exception instead
Environmental Needs	Phase 1, SLog2.1 passed Client Driver, DatabaseManager Stub

3.2.8 AccountManager integration

This component manages all the user Accounts that are stored in the database and it's accessed from the AuthenticationManager and the Administrator-Manager. For this reason, it requires a DatabaseManager stub component.

Test Case Identifier	SAcc2.3
Integrated Components	Authentication Manager → Account Manager
Test Items	Authentication Controller → Account Controller (getAccount, addAccount, editAccount, removeAccount)
Input Specification	Well-Formed Account object
Output Specification	The Account Manager should add, edit or remove the specified account if the data are well-formed. The Account Manager should raise an exception on add if the account is alredy present, or on remove and edit if it is not present in the database.
Environmental Needs	Phase 1, SLog2.1 passed, Client Driver, DatabaseManager Stub

3.2.9

Test Case Identifier	PReqList2.3
Integrated Components	API → Account Manager
Test Items	Passenger Requests → Account Controller (requestList)
Input Specification	Well-Formed Account object
Output Specification	The Account Manager should retrieve the list of reservations for the specified account if it is present on database and the requester is authorized to access that account data or raise an exception otherwise.
Environmental Needs	Phase 1, SLog2.1, passed, Authentication Manager integrated Client Driver, DatabaseManager Stub

3.2.10

Test Case Identifier	SAcc2.4
Integrated Components	Administration Manager → Account Manager
Test Items	Administrator Controller → Account Controller (Edit, Add, Remove Account)
Input Specification	Well-Formed Account object
Output Specification	The Account Manager edit, add or remove the specified account, or raise an exception if the account does not exists (for editing or removal) and if already exists in the case of adding.
Environmental Needs	Phase 1, SLog2.1, passed, Authentication Manager and Administration Manager integrated Client Driver, DatabaseManager Stub

3.2.11 RequestManager Integration

The RequestManager is the host for the core business algorithm. It receives and manages user inputs for every taxi request or reservation. It is accessed from the API and requires a valid system state to operate, for this reason it requires the DatabaseManager Stub and a Client Driver.

Test Case Identifier	PReq2.5, PRes2.5
Integrated Components	API → RequestManager (IncomingRequest, IncomingReservation)
Test Items	PassengerRequest → RequestInterface
Input Specification	Reservation and Request Data
Output Specification	An exception is raised if data are not well-formed An exception is raised if the user is not logged and authorized The request is stored in the database
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub, ZoneManager Stub,

3.2.12

Test Case Identifier	PRemReq2.5
Integrated Components	API → RequestManager (IncomingRequest, IncomingReservation)
Test Items	PassengerRequest → RequestInterface
Input Specification	Reservation and Request Data
Output Specification	An exception is raised if data are not well-formed An exception is raised if the user is not logged and authorized An exception is raised if the request does not exist The request is removed from the database and the queues if meets the requirements (see RASD)
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub, ZoneManager Stub,

3.2.13

Test Case Identifier	PPro2.5
Integrated Components	API → RequestManager (IncomingRequest, IncomingReservation)
Test Items	PassengerRequest → RequestInterface (TaxiProbe)
Input Specification	Location Data
Output Specification	If the Location Data are malformed or not covered by the service, the result for the default location is returned. The ZoneManager result is returned otherwise An exception is raised if the user is not logged and authorized
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub, ZoneManager Stub,

3.2.14 ZoneManager/TaxiManager Integration

This component manages the state for all the zones and the taxis that are associated with them. On runtime it relies on data stored in the database, for that reason it requires that stub.

Test Case Identifier	ZTaxi2.6, ZZone2.6, ZAdj2.6
Integrated Components	RequestManager → ZoneManager/TaxiManager
Test Items	RequestResolver → ZoneController (GetTaxiInZone, GetZones, GetAdjacents)
Input Specification	Zone Data
Output Specification	If the Zone Data are malformed an exception is raised The requested data is returned
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub

3.2.15

This component belongs to the TaxiManager component and it's implemented by the RequestResolver. This two functionalities are requested from the taxi application when the driver accepts or refuses a request.

Test Case Identifier	TReqEvents2.7
Integrated Components	API → ZoneManager/TaxiManager
Test Items	TaxiRequests(driverResponse) → RequestManager(OnRequestAccepted, OnRequestRefused)
Input Specification	Taxi Data and Request Data
Output Specification	The corresponding code implemented in the RequestManager component is executed on a taxi event. An exception is raised if the taxi is not logged, if the Request is malformed or not assigned to that driver
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated, RequestResolver integrated Client Driver, Database Stub

3.2.16

Test Case Identifier	TNotif2.7
Integrated Components	API → ZoneManager/TaxiManager
Test Items	TaxiRequests(driverNotifications) → RequestManager(OnAvailable, OnLogin, OnLogout)
Input Specification	Taxi Data and Request Data
Output Specification	The corresponding code implemented in the ZoneManager component is executed on a taxi event. An exception is raised if the taxi is not logged, if the Request is malformed or not assigned to that driver. The OnAvailable, OnLogin and OnLogout events should be triggered according to the input data.
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub

3.2.17

Test Case Identifier	TLoc2.7
Integrated Components	API → ZoneManager/TaxiManager
Test Items	TaxiRequests(locationUpdate) → RequestManager(OnLocationUpdate, OnZoneEnter, OnZoneExit)
Input Specification	Taxi Data and Location Data
Output Specification	The corresponding code implemented in the ZoneManager component is executed on a taxi event. An exception is raised if the taxi is not logged, or the Location is not valid. The OnLocationUpdate, OnZoneEnter and OnZoneExit events should be triggered according to the input data.
Environmental Needs	Phase 1 Passed, Authentication Manager integrated, Account Manager Integrated Client Driver, Database Stub

3.3 Persistence Layer Integration

In this section all the components that access database data will be tested. In particular it is important that the data model is consistent to each component contract.

Test Case Identifier	Aut3.1
Integrated Components	Authentication Manager → Database
Test Items	ApiToken, SystemAuthenticator → Account Model
Input Specification	Account Data
Output Specification	The component should raise an exception if the data is malformed or if the database connectivity is absent All data that is inserted, removed or updated should be consistent with the specifications
Environmental Needs	Phase2 Passed, Entity Beans running, Database Running

3.3.1

Test Case Identifier	Acc3.5
Integrated Components	Account Manager → Database
Test Items	Account Controller → Account Model
Input Specification	Account Data
Output Specification	The component should raise an exception if the data is malformed or if the database connectivity is absent All data that is inserted, removed or updated should be consistent with the specifications
Environmental Needs	Phase2 Passed, Entity Beans running, Database Running

3.3.2

Test Case Identifier	Adm3.2
Integrated Components	Administration Manager → Database
Test Items	Administration Controller (showLogs, taxi account management) → Log Data Model, Taxi Data
Input Specification	Log Data, Taxi Data
Output Specification	The component should raise an exception if the data is malformed or if the database connectivity is absent All data that is inserted, removed or updated should be consistent with the specifications
Environmental Needs	Phase2 Passed, Entity Beans running, Database Running

3.3.3

Test Case Identifier	Req3.3
Integrated Components	Request Manager → Database
Test Items	RequestResolver → Request and Reservation Model
Input Specification	Request and Reservation Data
Output Specification	The component should raise an exception if the data is malformed or if the database connectivity is absent All data that is inserted, removed or updated should be consistent with the specifications
Environmental Needs	Phase2 Passed, Entity Beans running, Database Running

3.3.4

Test Case Identifier	Zon3.4
Integrated Components	Zone Manager → Database
Test Items	Zone Manager → Zone and Taxi Model
Input Specification	Zone and Taxi Data
Output Specification	The component should raise an exception if the data is malformed or if the database connectivity is absent All data that is inserted, removed or updated should be consistent with the specifications
Environmental Needs	Phase2 Passed, Entity Beans running, Database Running

3.4 Presentation Layer Integration

This section will integrate the system output presentation. In particular every kind of output toward the users should comply with the specifications and the expected formats.

Test Case Identifier	RConf4.2, RNotf4.2
Integrated Components	Request Manager → Notification Manager
Test Items	RequestResolver → Passenger Notification (request confirmation, request notification)
Input Specification	Request Data, Passenger Account
Output Specification	The Confirmation should be triggered upon an accepted request, containing data as stated in the specification document. The notification should occur at the time specified in the Specification and Design documents Only the passenger associated with the request are suitable to receive this messages
Environmental Needs	Phase 2 and 3 passed. Client Driver for Passenger and Taxi, Client Interface or Email Recipient

3.4.1

Test Case Identifier	TReq4.2
Integrated Components	Request Manager → Notification Manager
Test Items	Request Resolver (driver Request) → Driver Notification Interface
Input Specification	Valid Request Data, Active Taxi Data
Output Specification	This taxi request is triggered upon a request that is being served by the Request-Manager. It should be issued to the specified taxi. The taxi driver should be online and available in order to this request being issued
Environmental Needs	Phase 2 and 3 passed. Client Driver for Passenger, Connected Taxi Driver Interface

3.4.2

Test Case Identifier	TPro4.2
Integrated Components	Request Manager → Notification Manager
Test Items	Request Resolver (TaxiProbeResponse) → Passenger Notification Interface
Input Specification	Valid Location Data
Output Specification	This response is triggered upon a successful TaxiProbeRequest issued by an user. The data provided should be compliant with the specifications
Environmental Needs	Phase 2 and 3 passed. Client Driver for Passenger, Connected Passenger Interface

3.4.3

Test Case Identifier	TReg4.1
Integrated Components	Account Manager → Notification Manager
Test Items	Account Controller → Email Server
Input Specification	Pre Defined Registration confirmation messages, Account data
Output Specification	The message should be issued to the email recipient of the specified user
Environmental Needs	Phase 2 and 3 passed. Client Email Recipient

3.4.4

Test Case Identifier	SOut4.4
Integrated Components	Notification Manager → API
Test Items	NotificationController → API Components (Multiple)
Input Specification	Data output from previous tests
Output Specification	The API component should generate Message-Driven Beans or JSON messages according to the specified function and targeting the specified client passed as parameter
Environmental Needs	Phase 2 and 3 passed. Client Driver, Clients Connected

3.4.5

Test Case Identifier	SDev4.3
Integrated Components	Zone Manager → API
Test Items	ZoneController → API Requests
Input Specification	Database Model
Output Specification	The API component should provide business data to authorized system developers The component must throw an exception if the associated APIToken is not valid
Environmental Needs	Database Stub, Developer interface stub

4 Tools and Test Equipment Required

- **Manual Testing** The phase 1 integration testing is done by manual testing. This assumption has been done because of the limited number of pages and functionalities in the system interface and the complexity of some outputs, such as the taxi positions on map.
- **Arquilian** for the phase 2, 3, 4. It is particular useful for testing containers and it provides a module for glassfish.

5 Program Stubs and Test Data Required

This testing strategy both needs of Stubs and Drivers. A short description is provided for each element that has to be implemented in order to start the integration testing phase.

5.1 Drivers

5.1.1 Client Driver

This driver is needed in phases 2, 3 and 4. This component directly calls the API methods and should emulate the user interface by injecting typical user input for each functionality as well any exceptional input for testing all the cases for each integration step.
It has to be composed of 4 subcomponents:

- **Passenger Client**: This component should emulate the behaviour of a passenger client.
- **System Administrator Client**: This component should emulate the behaviour of a system administrator.
- **Taxi Driver Client**: This component should emulate the behaviour of a Taxi Driver client
- **System Developer/External API Client**: This component should emulate an external API client or system developer that request access to system resources by means of API calls.

5.2 Stubs

All integration phases uses the stubs to emulate the behaviour of one or more system components. The stubs should emulate the response of particular user inputs or system states. The developer of these components should consider, for each functionality, both a positive, negative or exceptional response for the driver inputs.

5.2.1 AuthenticationManager Stub

This component should emulate the behaviour of the AuthenticationManager when testing the Client Interface or the API integration.
The typical response should comprehend both successful and unsuccessful logins and registrations and every eventual exception.

5.2.2 RequestIntefrace Stub

This component emulates the RequestInterface component, which manages all requests relative to the service.

5.2.3 ZoneManager/TaxiManager Stub

This component emulates the ZoneManager and must implement the TaxiManager interface. At runtime, this component should emulate a possible state of the taxi zones and provide requested data.

5.2.4 AdministratorManager Stub

This component emulates the AdministrationManager component. It will receive calls from the API when testing the Client Interface for the System Administrator Interface.

5.2.5 AccountManager Stub

This component should emulate the behaviour of the Account data part, that consists in editing, adding or removing client accounts and in fetching Accounts in the login phase. The testing data should consider every possible exceptional situation in emulating the accounting capabilities.

5.2.6 DatabaseManager Stub

This component should emulate the business data model. In particular it should be a stub of the database pre-configured with test data that is used during the integration tests at phases 2 and 4.

5.3 Client Interface Stub

This component is used in the phase 4 of the integration tests. Since the system output should be already tested in phase 1, this component is useful to emulate the presence of a client interface for receiving the test data in the context of system outputs.

6 Work Hours

- **Edoardo Giacomello**: 25 Hours
- **Mattia Fontana**: 23 Hours