SE2 Integration Test Plan Document

Edoardo Giacomello Mattia Fontana

January 20, 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

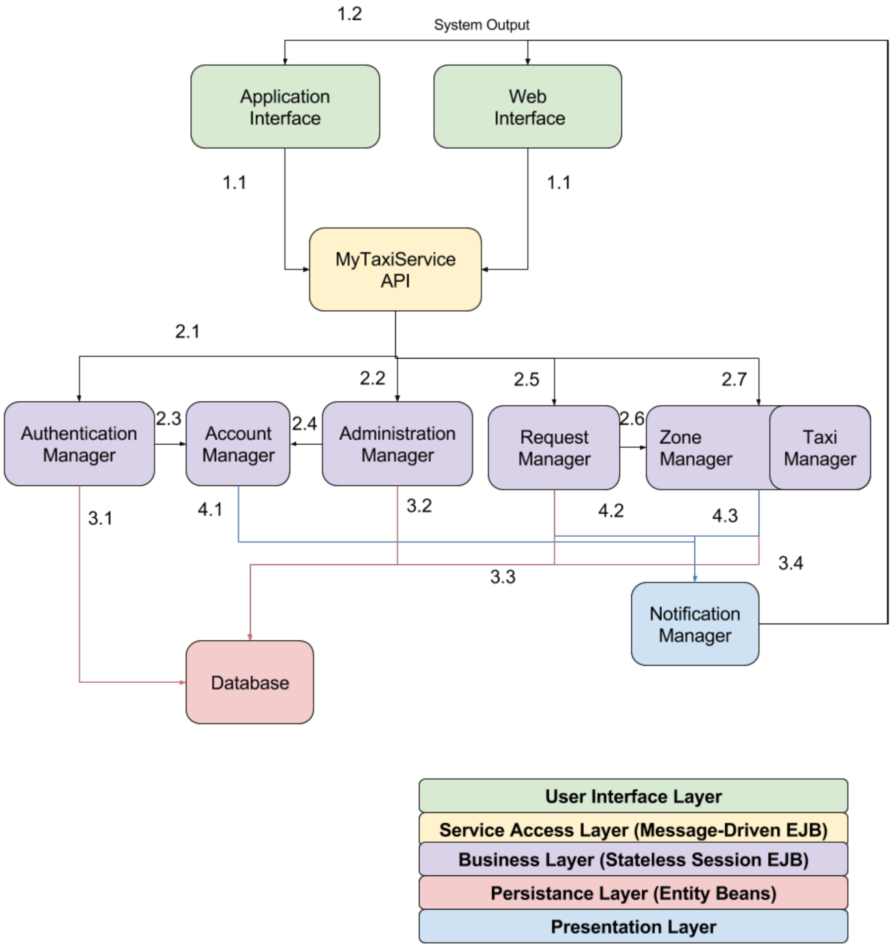
2 Integration Strategy

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

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

2.2.3 Presentation Layer

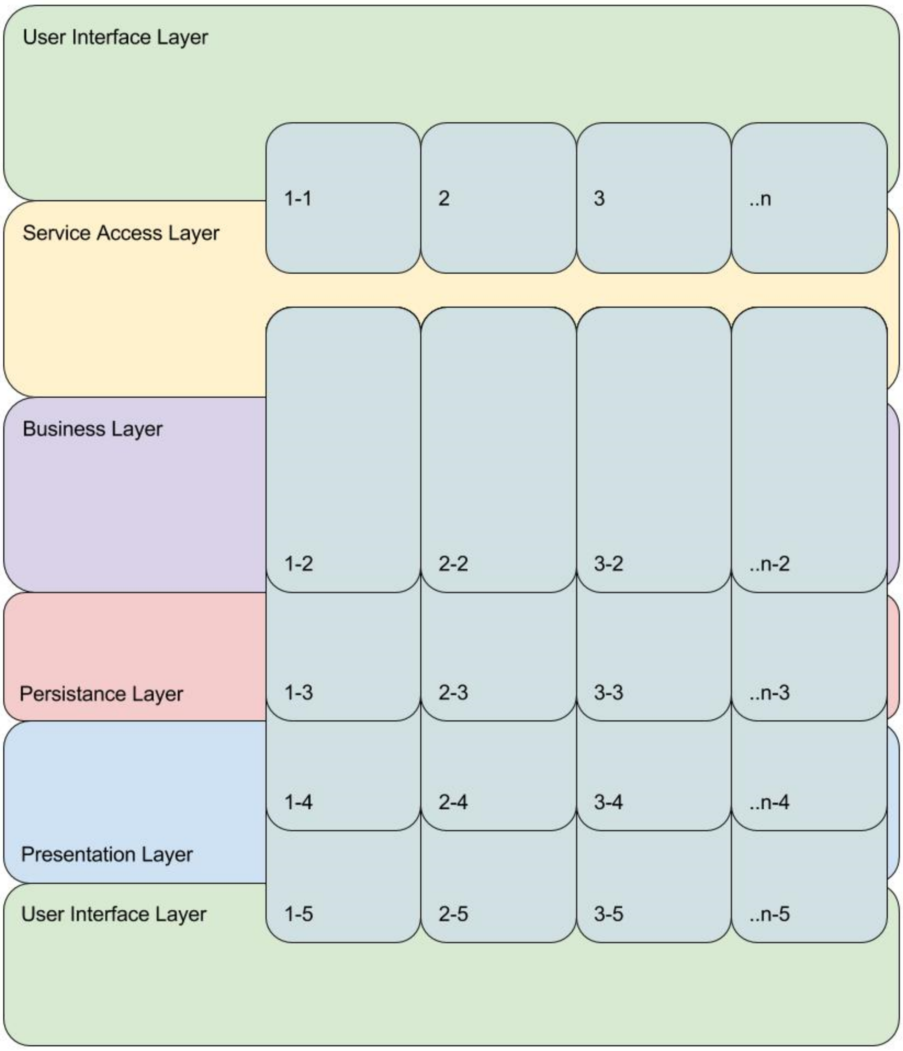
- Notification Manager

2.2.4 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

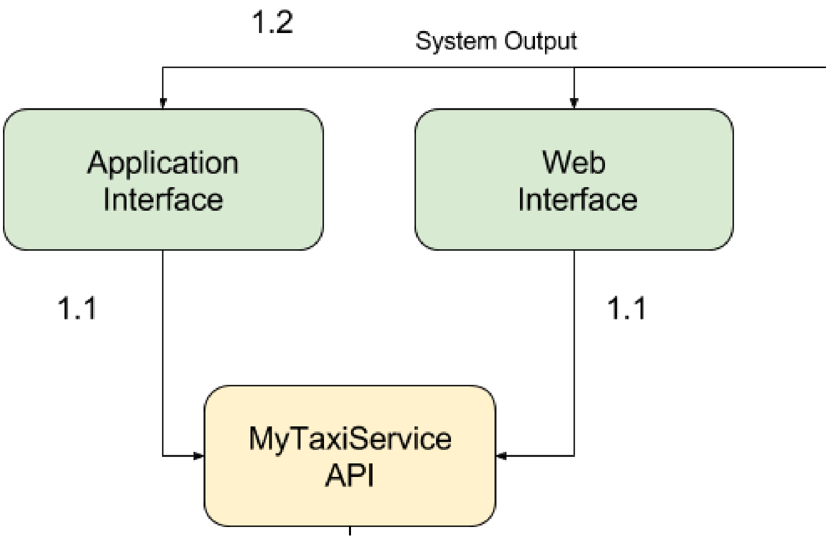
The first phase of the integration testing will consider the User Interface Layer against the Service Access Layer (MyTaxiService APIs). In particular, the User Interface Layer is composed by the following subsystems:

- 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

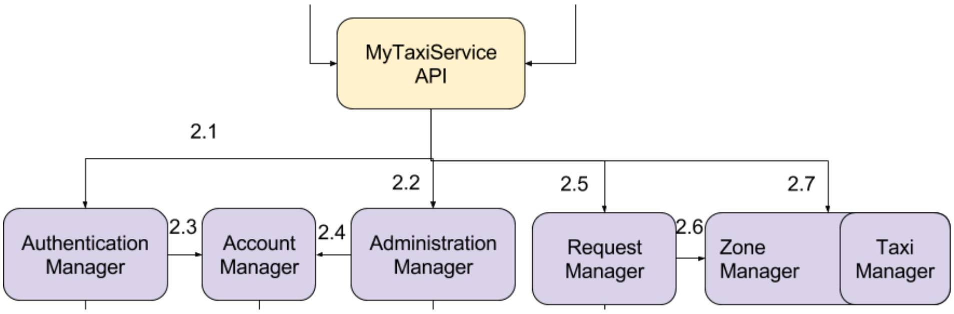


Source Component	**Destination Component**	**Functionality**	**Reference**
Passenger Interface	MyTaxiService APIs	Passenger Registration	
Administrator Interface	MyTaxiService APIs	Taxi Registration	
Passenger Interface	MyTaxiService APIs	Passenger Login	
Administrator Interface	MyTaxiService APIs	Administrator Login	
Taxi Interface	MyTaxiService APIs	Taxi Login	
Passenger Interface	MyTaxiService APIs	Logout	
Taxi Interface	MyTaxiService APIs	Logout	
Administrator Interface	MyTaxiService APIs	Logout	
Passenger Interface	MyTaxiService APIs	Map	
Passenger Interface	MyTaxiService APIs	Request	
Passenger Interface	MyTaxiService APIs	Reservation	
Passenger Interface	MyTaxiService APIs	Pending Reservations	
Passenger Interface	MyTaxiService APIs	Profile	
Taxi Interface	MyTaxiService APIs	Driver Response	
Taxi Interface	MyTaxiService APIs	Location Update	
Taxi Interface	MyTaxiService APIs	Driver Notification	
Administrator Interface	MyTaxiService APIs	Taxi Management	
Administrator Interface	MyTaxiService APIs	Account Management	
Administrator Interface	MyTaxiService APIs	Database Management	

Source Component	**Destination Component**	**Functionality**	**Reference**
Administrator Interface	MyTaxiService APIs	Log Management	
MyTaxiService APIs	Passenger Interface	Login Logout Registration TaxiProbeResponse TaxiConfirmation Notification RequestList Account Information.	
MyTaxiService APIs	Taxi Interface	Login Logout DriverRequest.	
MyTaxiService APIs	Administrator Interface	Login Logout TaxiList AccountList LogList.	

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

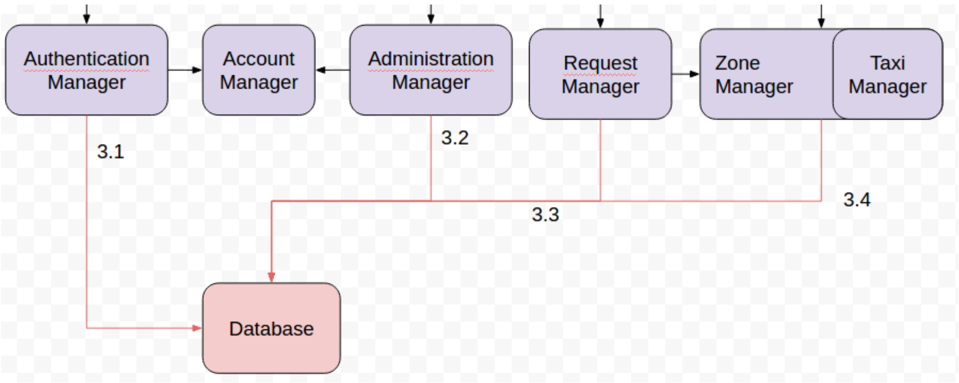


All of these tests require a Client Driver that simulates the client requests intercepted by the APIs. An appropriate Stub has to be written in order to emulate the behaviour of lower layers, in particular the business components that are still not tested and the database.

Source Component	**Destination Component**	**Functionality**	**Reference**
MyTaxiService APIs	Authentication Manager	Login Logout AccountRegistration APIRegistration.	
MyTaxiService APIs	Administration Manager	AddTaxi RemoveTaxi EditTaxi AddAccount EditAccount Backup Restore Logs.	
Authentication Manager	Account Manager	GetAccount EditAccount AddAccount RemoveAccount	
API	Account Manager	RequestList	

Source Component	**Destination Component**	**Functionality**	**Reference**
Administration Manager	Account Manager	EditAccount AddAccount RemoveAccount	
MyTaxiService APIs	RequestManager	TaxiRequest TaxiReservation RemoveRequest TaxiProbe.	
RequestManager	ZoneManager	getTaxiInZone getZones getAdjacents	
MyTaxiService APIs	TaxiManager	OnRequestAccepted OnRequestRefused TaxiAvailable UpdateLocation OnZoneEnter OnZoneExit	

2.5.3 Phase 3: Persistence Level Integration

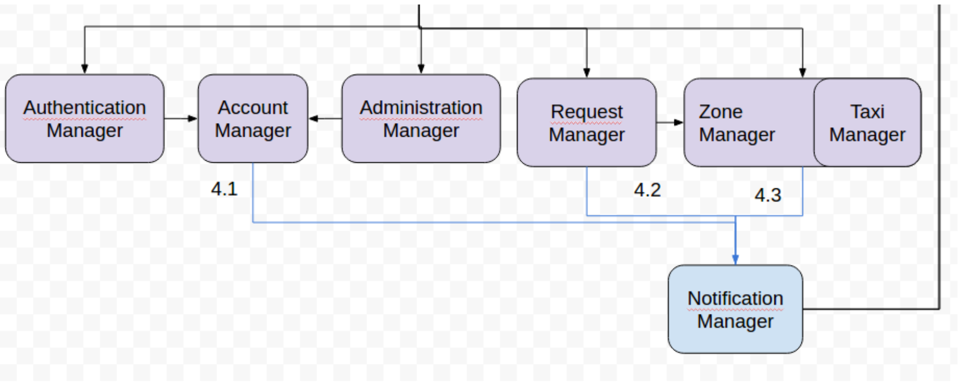


Now every component that accesses the business data have to be tested against the database.

Source Component	Destination Component	Functionality	Reference
Autenthication Manager	Database	Account Retrieval for Login	
Account Manager	Database	Account Data Editing	
Administration Manager	Database	Log Access Taxi Data Access	
Request Manager	Database	Request and Reservation data access	
Zone Manager	Database	Zone Data Access Taxi Data Access	

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.



Source Component	**Destination Component**	**Functionality**	**Reference**
Request Manager	Notification Manager	RequestConfirmation, RequestNotification, DriverRequest, TaxiProbeResponse	
Account Manager	NotificationManager	RegistrationConfirmation, Password Reset	
Notification Manager	MyTaxiService APIs	OutputMessageGeneration	
Zone Manager	MyTaxiService APIs	GetZone for 3rd-party applications	

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 consistant and fulfills specifications. The HomePage is shown after successful login. An error is shown if the login fails.
Environmental Needs	AuthenticationManager Stub Plog1 Test has to be successful.

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 consistant and fulfills specifications. A confirmation is shown if the registration is successful. An error is shown if the registration fails (duplicate account or mising 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	PLog2
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	Data insert by user when he registered to the application. 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	Informations about the application are visualizzed. An error message is shown if the visualization fails.
Environmental Needs	RequestInteface Stub

3.1.10 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	Informations about the ride are visualizzed with two options for complete or release the ride. An error message is shown if the visualization fails.
Environmental Needs	TaxiManager stub. TLog1 test has to be successful.

3.1.11 Taxi Driver Logout

Test Case Identifier	TLog2
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

3.1.12 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 consistant and fulfills specifications. The HomePage is shown after successful login. An error is shown if the login 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	On map is showed the position of the request that taxi driver recived.
Environmental Needs	TaxiManager 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	Informations about the ride are visualizzed with two options for accept or refuse 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	Informations about the list of taxi driver are visualizzed. 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	Informations about the list of account are visualizzed. An error message is shown if the visualization fails.
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	Informations about the list of logs are visualizzed. 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 visualization 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	/
Output Specification	An error message is shown if the visualization fails.
Environmental Needs	AdministratorManager stub.

3.2 Business Layer Integration

3.2.1 AuthenticationManager Integration

Test Case Identifier	PLog2, SLog2, TLog2
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

Test Case Identifier	PReg2, AReg2
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

AdministrationManager integration

Test Case Identifier	TAdd2, TRem2, TEdi2
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, TLog2 passed Client Driver, TaxiManager Stub

Test Case Identifier	AAdd2, AEdi2
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 passed Client Driver, AccountManager Stub

Test Case Identifier	ALis2
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 passed Client Driver, AccountManager Stub

Test Case Identifier	SBac2, SRes2
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 passed Client Driver, AccountManager Stub

Test Case Identifier	SLog2
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 passed Client Driver, DatabaseManager Stub

3.2.2 AccountManager integration

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 passed, Client Driver, DatabaseManager Stub

Test Case Identifier	PReqList2
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, passed, Authentication Manager integrated Client Driver, DatabaseManager Stub

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, passed, Authentication Manager and Administration Manager integrated Client Driver, DatabaseManager Stub

3.2.3 RequestManager Integration

Test Case Identifier	PReq2, PRes2
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,

Test Case Identifier	PRemReq2
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,

Test Case Identifier	PPro2
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,

3.2.4 ZoneManager/TaxiManager Integration

Test Case Identifier	ZTaxi2, ZZone2, ZAdj2
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

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

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

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

3.4 Presentation Layer Integration

4 Tools and Test Equipment Required

5 Program Stubs and Test Data Required

6 Work Hours