5/6 chapters

1.Introduction:

.**purpose and scope**: the aim of this document is to clarify which are the integration of the components identified in the Component View of the DD to guarantee a well tested final software.

.**list of definition and abbreviations:**

- Server Database: data layer

- Server Application: application layer

- Client: client layer

- Mobile App: myTaxiService mobile application, in Client

- Web App: myTaxiService web application, in Client

- System: the union of software and hardware to be developed and implemented Acronyms

- RASD: Requirements Analysis and Specification Document

- DD: Design Document

- API: application programming interface

- DBMS: DataBase Management System

.**list of references**

2.

. **Integration strategy**: Before starting the integration testing of any software component that has been

designed for myTaxiService system, the internal functions of the considered com-

ponent (i.e. public or protected methods that are exposed within the package of

the component but are not part of any external public interface) must be unit

tested using an appropriate framework.

. **Elements to be integrated:** PowerEnJoy, as shown in Design Document, is a three-tier system:

- DBMS, PowerEnJoy server, Client application, Car System (????)

Each tier is composed by interacting modules:

1. Application (mobile), 2. Data manager, 3. Account manager, 4. Ride manager ,5. Bill manager, 6. Map services ,7. Notification,8. Car manager, 9. Zone manager, 10. Problem manager

.**integration testing strategy:** The integration testing strategy, conducted in this project, is a bottom-up approach. This strategy tests the lower level components and start testing a way upwards to higher level components. The advantage of this strategy is that it's easier to maintain code, smaller modules have unit tests and there is a clearer structure of how to do things. The disadvantage is that when releasing a prototype it's impossible to see a working prototype until nearly all the program has been completed so that may take a long time before this happens. In early development, testing tools as Mockito and Arquillian (described in Chapter 4) allow us to test components which depend on incomplete ones through stubs and drivers (Chapter 5).

**.sequence of component / function integration**

**-**2.4.1 foto e 2.4.2 uguale a Diffolo

|  |  |
| --- | --- |
| Item | Data Manager **⬄** Account Manager |
| Input Specification | A set of methods calls on Data Manager to retrieve user  information |
| Output Specification | Check that the user information are correct |
| Environmental needs | Glassfish Server, a test Database, I1 successful |
| Purpose | Verify that the user information are retrieved from the  Data Manager |

|  |  |
| --- | --- |
| Item | Data Manager **⬄** Ride Manager |
| Input Specification | A set of methods calls on Data Manager to retrieve reserved ride information |
| Output Specification | Check that the user information are correct |
| Environmental needs | Glassfish Server, a test Database, I1 successful |
| Purpose | Verify that the reserved ride info are retrieved from the  Data Manager |

|  |  |
| --- | --- |
| Item | Data Manager **⬄** Map Service |
| Input Specification | A set of methods calls on Data Manager to retrieve informations about coordinates |
| Output Specification | Check that the coordinates information are correct |
| Environmental needs | Glassfish Server, a test Database, I1 successful |
| Purpose | Verify that the info of coordinates are retrieved from the  Data Manager |

|  |  |
| --- | --- |
| Item | Ride Manager **⬄** Car Manager |
| Input Specification | A set of methods calls on Car Manager |
| Output Specification | Check if an available Car is returned |
| Environmental needs | Ride/Notification and Data/Account satisfied |
| Purpose | Retrieve an available Car |

|  |  |
| --- | --- |
| Item | Ride Manager **⬄** Zone Manager |
| Input Specification | A set of methods calls on Zone Manager |
| Output Specification | Verify that the returned zone is the correct one |
| Environmental needs | Map Service/EXTApi and Car mng/Ride satisfied |
| Purpose | Find the position in a certain zone |

|  |  |
| --- | --- |
| Item | Ride Manager **⬄** Bill Manager |
| Input Specification | A set of methods calls on Bill Manager |
| Output Specification | Verify that the total amount is correct |
| Environmental needs | Map/EXTApi, CarManager/System and Data/Account satisfied |
| Purpose | Verify that the Bill Manager calculate the correct amount of the ride |

|  |  |
| --- | --- |
| Item | Ride Manager **⬄** Notification |
| Input Specification | A set of methods calls in order to create a notification |
| Output Specification | Check if the correct notification is created |
| Environmental needs | Data/Account and Ride/Car satisfied |
| Purpose | Verify that the Notification Manager creates the notification from the Ride Manager |

|  |  |
| --- | --- |
| Item | Ride Manager **⬄** Problem Manager |
| Input Specification | A set of methods calls on Problem Manager |
| Output Specification | Verify that the problem belong to the provided list |
| Environmental needs | Data/Account and Notification/Mobile satisfied |
| Purpose | Check if Problem Manager can handle correctly the calls, returning an element belong to the provided listbox |

|  |  |
| --- | --- |
| Item | Map Service**⬄** External API |
| Input Specification | Create a typical set of methods calls by Map Service on External API |
| Output Specification | Check that all the methods of External APIs Manager produce the expected results |
| Environmental needs | N/A |
| Purpose | Verify that the External APIs Manager works with the Map Services Manager |

|  |  |
| --- | --- |
| Item | Bill Manager **⬄** External API |
| Input Specification | Create a typical set of methods calls by Bill Manager on External API |
| Output Specification | Check that all the methods of External APIs Manager produce the expected results |
| Environmental needs | N/A |
| Purpose | Verify that the External APIs Manager works with the Bill Manager |

|  |  |
| --- | --- |
| Item | Problem Manager **⬄** External API |
| Input Specification | Create a typical set of methods calls by Problem Manager on External API |
| Output Specification | Check that all the methods of External APIs Manager produce the expected results |
| Environmental needs | N/A |
| Purpose | Verify that the External APIs Manager works with the Problem Manager |

|  |  |
| --- | --- |
| Item | Car Manager **⬄** Map Service |
| Input Specification | A set of methods calls on Map Services Manager |
| Output Specification | Verify that the position of the car is correct |
| Environmental needs | Map/EXT and Data/Account succesful |
| Purpose | Retrieve the position of a car in a certain position on the map |

|  |  |
| --- | --- |
| Item | Car Manager **⬄** Car System |
| Input Specification | A set of methods calls Car Manager |
| Output Specification | Verify that the state of the car is correct |
| Environmental needs | Car with GPS and Data Connection enabled |
| Purpose | Verify that the information exchanged between car and system work |

|  |  |
| --- | --- |
| Item | Database Server **⬄** Application Server |
| Input Specification | Queries on the DBMS for the table Driver, Car, Ride and Zone |
| Output Specification | The queries return the expected results |
| Environmental needs | Glassfish Server, a test Database |
| Purpose | Verify that the typical queries to the DBMS works |

|  |  |
| --- | --- |
| Item | Application Server **⬄** Client |
| Input Specification | A set of methods calls on both Server Application and Client |
| Output Specification | Check that methods calls mentioned in Input Specification produce the expected results |
| Environmental needs | Glassfish, a test Database |
| Purpose | Verify the interaction between Server Application and Client work |

|  |  |
| --- | --- |
| Item | Mobile App **⬄** Account Manager |
| Input Specification | Create a typical set of methods calls performed by Mobile App on Account Manager |
| Output Specification | Check that methods calls mentioned in Input Specification produce the expected results |
| Environmental needs | A device that can run Mobile App |
| Purpose | Verify if Account Manager can handle correctly Mobile App methods calls |

|  |  |
| --- | --- |
| Item | Mobile App **⬄** Ride Manager |
| Input Specification | Create a typical set of methods calls performed by Mobile App on Ride Manager |
| Output Specification | Check that methods calls mentioned in Input Specification produce the expected results |
| Environmental needs | A device that can run Mobile App |
| Purpose | Verify if Ride Manager can handle correctly Mobile App methods calls |

|  |  |
| --- | --- |
| Item | Mobile App **⬄** Notification |
| Input Specification | Create a typical set of methods calls performed by Mobile App on Notification |
| Output Specification | Check that methods calls mentioned in Input Specification produce the expected results |
| Environmental needs | A device that can run Mobile App |
| Purpose | Verify if Notification can handle correctly Mobile App methods calls |