Integration Test Strategy

The integration test strategy is based on the bottom-up concept, testing first of all single components and taking into account the various dependences between each other. Then we will test the groups of component integrated together and finally the entire subsystem of every component. In this way, first we could work on a specific functionality of the application and then, steps by steps, we could have the global view of the system. Obviously, there are some modules that are more important than the other and this force us even to have an order in considering and testing components.

Here we show all the dependences between the main components of the application in order to highlight which component is necessary to develop and test before.

We also need the so called “drivers”, a software which manage the testing models creating the rights input for each one.

* NavigatorDriver: it guides tests invoking the methods of the Navigate Interface and input from Itinerary
* AccountDriver: it will invoke the profile Management Interface and input from Navigator
* AppointmentDriver: it will invoke the Calendar Management’s methods
* ItineraryDriver: it will invoke the Itinerary Management Interface’s methods
* PaymentDriver it will invoke the Buy Ticket Interface’s methods
* NotificationDriver it will invoke the calls from Itinerary Manager

TOOLS

qTEST

It is scalable test management and automation tools. qTest has proven to make every step of the QA process faster, simpler and more efficient:

Manage Requirements, Test case repository, Test Execution, Defect Tracking, Reporting, and Integrations.

Furthermore qTEST is very common in Agile testing and development teams which follows our philosophy principle in terms of testing work and it is one of the best tool in this field.

APACHE TOMCAT

Apache Tomcat, is an open-source [Java Servlet Container](https://en.wikipedia.org/wiki/Servlet_container) which implements several [Java EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) specifications including Java Servlet  and provides a "pure [Java](https://en.wikipedia.org/wiki/Java_(programming_language))" [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can run. This is really important because the application needs a Servlet in the web-based interaction and Apache is really reliable.

JUNIT

As a Unit testing framework: Junit is the de facto standard library for unit testing in Java and it is supported out of the box by all major IDEs following all our project choices.