**4. Tools and Test Equipment Required**

In this section we recommend some tools that might be used during the integration test phase. These advices are not mandatory since the RASD document and the Design Document to which this document refers do not enter in technical details and do not point towards the choice of a particular language/platform, leaving it to the implementation phase.

However, we think that it is very likely that the final choice will fall on the Java EE platform. For this reason we suggest some possibly useful tools to be used based on it.

For what concerns unit testing, for example unit testing necessary to fulfil the entry criteria of chapter 2.1, Mockito could be an interesting choice together with Junit.

Documentation for Mockito:

<http://docs.mockito.googlecode.com/hg/1.9.5/org/mockito/Mockito.html>

Documentation for JUnit:

<http://junit.org/javadoc/latest/>

Instead, for what concerns the most relevant part of this document, which is integration test, Arquillian might be an easy choice, always together with JUnit.

Documentation for Arquillian:

<http://arquillian.org/guides/>

In addition to these “core” tools, we find that it might be useful to look for specific frameworks concerning the handling of messages, since the remote communication among the subsystems identified in the previous chapters is not trivial at all, and it’s a central part of the whole system.

For example, Citrus is a framework that provides this kind of service.

Documentation for Citrus:

<http://www.citrusframework.org/documentation.html>

Since we decided to include an email service in MyTaxiService application, it could be useful to have a tool specifically devoted to the testing of this part. Such a tool might be, for example, GreenMail.

Documentation for GreenMail:

<http://www.icegreen.com/greenmail/javadocs/index.html>

As a final suggestion, we recognize that for some parts of the software there might be the need to apply manual testing, for example during the testing of web client /web pages.

As we said at the beginning, almost all these suggestions relies on the choice of Java EE as platform for the implementation. If that won’t be the case, a good strategy would be to look for similar tools that accomplish as much as possible the same purposes of the ones described above, recurring to manual testing when it is not possible.