**Java Air Test Plan**

Version 1.1

Prepared by: Matthew D Moscatel and Xu Wu

**Team: Avian Limited**

# Test Plan Identifier

This is Java Air Project Software Test Plan Version 1.1, in short identified as ‘STD\_Plan\_1.1’.

# Introduction

This document contains the Software Test Plan for the Java Air Project. The categories of testing addressed in this document include unit, integration, system, acceptance, and installation testing. This document describes the test items, approaches and criteria. The test tasks and activities are scheduled in this document along with Software Project Management Plan, Software Quality Assurance Plan, and Software Verification & Validation Plan.

# Test Items

The classes and methods in the Java Air project code base

The test items (Java code files) shall be committed and uploaded to online Git repository before the tests run.

# Features to be tested

All features of Java Air Project, as defined in the Java Air Software Requirement Specification Version 4.2 Section 2.2. ‘Product Functions’ shall be tested.

# Features not to be tested

All features of Java Air Project shall be tested.

# Approach

There are two major approaches for testing the Java Air Project.

For system-level testing, manual execution and validation approach is used.

For unit-level and module-level testing, test code shall be written, manually and repeatedly executed to automatically validate unit correctness.

# Item pass/fail criteria

Item pass/fail criteria are defined in the Test Case Specification documents. All test case steps must be considered acceptable for a passing classification.

# Suspension criteria and resumption requirements

When an issue is identified during testing that prevent testing to be continued, the affected tests shall be suspended for further investigation while the rest of the tests shall be continuously executed. After the investigation is finished and remedies are implemented, the affected tests shall be resumed to be executed.

# Test deliverables

The following documents shall be delivered:

* Test Plan
* Test design specification
* Test case specification
* Test procedure specification
* Test item transmittal reports
* Test logs
* Test incident reports
* Test summary reports
* Test input & output data
* Module drivers & stubs
* JUnit Test Code

# Testing Tasks

The following testing tasks shall be performed in the sequence as defined below (from top to bottom):

* Compose Test Plan, Design, and Test Case Documents
* Build Unit Test Code
* Build Module Test Code
* Execute Unit Test Code
* Execute Module Test Code
* Execute System Tests
* Execute Acceptance Tests
* Execute Installation Tests
* Record Test Logs and Incidents and compose documents
* Record Test Item Transmittal and compose reports
* Summarize test activities, results and compose report.

# Environmental Needs

The following facilities shall be available for testing:

* Personal Computer or Laptop with Internet Connection

A computer is needed to write and execute tests. Internet is needed to exchange project artifacts between the software development teams.

* Java JDK

Java JDK is needed to execute the software and the test code.

* Java IDE

Java IDE is needed to write and execute test code.

* JUnit Framework

JUnit is needed to write unit and module test code.

* Microsoft Word

Word is need to view and write test documentations.

# Responsibilities

Related roles include the project manager (PM), requirement engineers (RE), software architect (SA), integration engineer (IE), testing engineers (TE), code developer (CD), and user/product director (UPD): Major Professor (MP).

The PM is responsible for overseeing and managing the whole testing process.

The REs are responsible for providing SRS which is needed to write test plan and test design documents.

The SAs are responsible for providing SDD which is needed to write test case specification documents.

The TEs are responsible for providing all deliverables defined in Section 10. ‘Testing Tasks’

The CDs are responsible for providing Java code which will be tested.

The MP is responsible for executing the acceptance test.

# Staffing and training needs

All Testing Engineers have already mastered the necessary knowledge needed for the Java Air Project, and thus need no further training.

# Schedule

The following milestones shall be secured during the testing process:

* System Test Plan Finished

Estimation: 10/31/2016 ~ 11/04/2016

* Unit & Module Test Code Composed

Estimation: 11/07/2016 ~ 11/11/2016

* Unit & Module Test Code Executed

Estimation: 11/14/2016 ~ 11/18/2016

* System Test Executed

Estimation: 11/21/2016 ~ 11/25/2016

* Installation Test Executed

Estimation: 11/28/2016 ~ 12/03/2016

* Acceptance Test Executed

Estimation: 11/28/2016 ~ 12/03/2016

According to ‘Java Air Project Schedule V1.2’, the whole testing activities shall be finished within three and a half weeks. The deadline is December, 3rd, 2016.

# Risks and contingencies

The main risk of the test plan is the on-time delivery of the test items (Java code). If the test items cannot be delivered on-time, unit tests and module tests will be delayed since they depend on the code. Test case discrepancies that are out of the scope of the project or time allotment, as well as deviations that improve the project, must be signed off by the project manager.

To mitigate the risk, unit and module test code shall be written in a continuous and iterative pattern. Once a Java class or module is written, the unit / module test code for the class / module shall be immediately composed and executed. This way, at least the test process can catch up with the code development speed instead of delaying further. And all existing codes are tested.

The same strategy applies to system tests. As soon as a system level feature is implemented, it shall be fully tested.

# Approvals

This Test Plan document shall be approved by Java Air Project Manager and Major Professor.