System Test Plan

Project Name:	
SyncGallery+	
Project Manager: Will Goreshi This test plan was done by: Hithagna Dinesh (SQA Engineer from the Testing Team)	**Revision:** 2.0

Table of Contents

Reference Material

1. IEEE Std 829-1998, IEEE Test Plan, IEEE 1998

2. Software Quality Assurance/Test Plan (SQATP)
Project – SyncGallery +
(From Phase A)

1. Objectives

Testing is done on the system as all the parts of the SyncGallery+ application are now integrated into one system the testing phase will now have to be done on the system to check and remove any errors or bugs.

- Test the overall functionality of the SyncGallery+ application.
- Cause failures involving the entire system when running on a single platform.
- Report these failures to the software development team so that the underlying defects can be identified and fixed.
- Help the software development team to stabilize the software so that it can be successfully distributed prior to system testing.
- Minimize the number of low-level defects that will prevent effective system and launch testing.

2. Testing responsibility

SQA Engineers (Testing team):

- Hithagna Dinesh: Test cases, Test procedures.
- Sarah Yang: Test cases, Test Procedures.
- Will Goreshi: Document Results.

3. Requirements

- The functionality, delivered by the development team, is as specified by the Requirements Documentation.
- The software is of high quality and the software will support the intended functions and achieves the standards required by the group for the development of the new application.
- The software delivers interfaces correctly with existing systems.

4. Defect Tracking

Integration Testing will lose its edge if the defects are not tracked correctly. Each defect should be documented and tracked. Information should be captured as to how the defect was fixed. This is valuable information. It can help in future integration and deployment processes.

5. Milestones

Feature Complete Mar. 1st
Alpha Release Mar. 2st

Beta Release Mar. 5th
Release Candidate Mar. 9th

6. Test procedures

5.1 Test Approach

- One system test will be for usability testing – Test how well the user can access the different features in the system and how easy it is to use
- Another system test will be for GUI software testing – this is to check if graphically that the program looks how was intended and the GUI works as intended.
- Black box testing as the testing team only really deals with the output of the system and the documenting of any problems due to the output.

If there is sufficient time load and volume testing will also be considered as probable system tests.

5.2 Test Environment

A grey-box tester will be permitted to set up his testing environment observe the state of the product being tested after performing certain actions.

For example, to test the directory of the image, the tester can create temp files and folders and check if the code returns the right directory.

5.3 How will the test be completed?

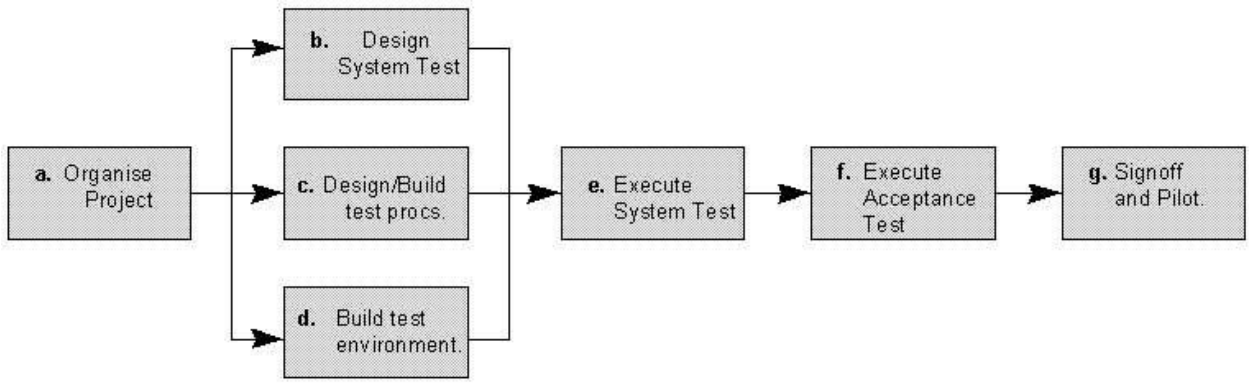

Figure1. The diagram above outlines the basic Test Process that will be followed by the testing team.

If the integration stage was done accurately then most of the test plan and test cases would already have been done and simple testing would only have to be done in order to ensure there are no bugs because this will be the final product.

If else, additional test cases for integrated components or different units may be created.

5.4 What platform will we used to ensure changes are not made during our tests?

The platforms that will be used are both Hardware and software packages which include the android ICS and the phone baseline release candidate along with the latest candidate from development team.

5.5 How many 'tests' make an acceptable level of test?

Two to three good test cases to test the entire system as a whole will be efficient considering the about of time we have for the testing.

7. Exit criteria (Acceptance)

The system tests indicate that the application satisfies the required functionality stated in the requirements document. Also, there are no bugs/errors and it is runs as specified.

8. Testing Tools

JUnit: Built in android testing tools will be sufficient to test the completed application software.

9. Testing Facilities

- Equipment – desks, books, computers, network
- Location – conducive to testing, easily accessible for the team, possibly consider an off-site testing facility for customers or users.
- Process simulations – If there are any process changes that have been accomplished during the development of the system be sure to incorporate those changes into the Acceptance testing.

10. Risks

- Integration test cases fail and the system does not run on a single platform
- Programmers fail to debug errors.
- No time; deadline approaching