**Test Plan**

**October 5, 2010**

**Submitted By:**

Avni Shah

Manoj Bora

Sagar Mutha

Siddharth Razdan

**Table of Contents**

**1** **Introduction 3**

**2** **Test Objective 3**

**3** **Process Overview 3**

**4** **Testing Process 4**

**5** **Bug Tracking/ Bug Process 5**

**6** **Roles and Responsibilities 6**

6.1 Development Team 6

6.2 Testing Team 6

# **Introduction**

This document is a high-level overview of our testing strategy for the Objective-C application. Its objective is to showcase the quality standards and procedures. This document will address the different standards that will apply to the unit, integration and system testing of the specified application. Our testing methodology will include the testing criteria, methods, and test cases of the overall design.

## **Team Interaction**

The following describes the level of team interaction necessary to have a successful product.

* The Test Team will work closely with the Development Team to achieve a high quality design and user interface specifications based on customer requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.
* The Test Team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.
* Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

# **Test Objective**

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will be testing the objective C code count to see if it meets the Source Lines Of Code (SLOC) counting standards.

# **Process Overview**

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

# **Testing Process**

**a.** Organize Project

**b.** Design System Test

**c.** Design/Build Test Proc.

**d.** Organize Project

**e.** Design/Build Test Proc.

**f.** Signoff

**Figure 1: Test Process Flow**

The diagram above outlines the Test Process approach that will be followed.

**a.** **Organize Project** involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.

**b.** **Design/Build System Test** involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.

**c.** **Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.

**d.** **Build Test Environment** includes requesting/building hardware, software and data set-ups.

**e. Execute System Tests –** The tests identified in the Design/Build Test Procedures will be executed. All results will be documented and Bug Report Forms filled out and given to the Development Team as necessary.

**f.** **Signoff** - Signoff happens when all pre-defined exit criteria have been achieved.

## **Unit Testing**

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

## **System Testing**

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

### ***Performance testing***

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. It will be done using black-box testing method. And this will be performed by:

* Storing the maximum data in the file and trying to insert, and observe how the application will perform when it is out of boundary.
* Deleting data and check if it follows the right sorting algorithm to sort the resulting data or output.
* Trying to store new data and check if it over writes the existing once.
* Trying to load the data while they are already loaded

# **Bug Tracking/ Bug Process**

# 

During testing, the testing team members normally encounter behavior that goes against a specified or implied design requirement in the product. When this happens, we will document and reproduce the bugs for the developers.

**Expectation of a bug:**

* Keep track of what version of the application the bug is found
* Determine if bug has already been written up
* Indicate the steps to reproduce the bug – write enough details for others looking at the bug to be able to duplicate it; exclude unnecessary steps (i.e. If access point is irrelevant, be more general in your steps).
* Actual results – be specific on your findings.
* Expected results – how the product should behave based on the specified or implied requirements.
* Implications – How does the defect affect the quality of the product?

The following chart defines the impact levels to be used when entering bugs.

|  |  |
| --- | --- |
| **Impact** | **Definitions** |
| 1 – Fatal | **Test Stopper:** If you can’t access a function and need the bug to be fixed immediately. The defect prevents QA from testing the feature area, sub-area or functionality of the feature. |
| 2 – Serious | **Beta Stopper:** This is a bug that users would experience such as: data corruption, calculation errors, incorrect data, UE’s and system crash on common user scenarios, significant QA risk, and major UI defects. |
| 3 – Minor | **Live Release:** A bug that must be fixed before the product is officially completed, UE’s or crashes, content, and UI and graphic changes required for release. |

## **Various Roles in Bug Resolution**

* **Author –** The person who wrote the bug; this will be someone on the QA team
* **Resolver –** Normally an Engineer assigned to a specific area of the application.
* **Verifier –** normally a QA Engineer responsible for testing the fix and closing the bug.

# **Roles and Responsibilities**

## **Development Team**

**Code Development Team – Sagar Mutha & Manoj Bora**

* Ensure Phase 1 is delivered to schedule and quality
* Ensure exit criteria are achieved prior to system test signoff
* Regularly review testing progress with test controller.
* Raise and manage issues/risks relating to project or outside test teams control.
* Review and sign off test approach, plans and schedule.

## **Testing Team**

**Testers – Avni Shah & Siddharth Razdan**

* Identify test data
* Execute test conditions and mark-off results
* Prepare software error reports
* Administrate error measurement system
* Ensure test systems outages/problems are reported immediately and followed up.
* Ensure entrance criteria are achieved prior to system test start.
* Ensure exit criteria are achieved prior to system test signoff.