Sea Addicts

Brock Butler

Software Quality Assurance Plan

Version: (1) **Date: (01/24/2013)**

Document History and Distribution

1. Revision History

Revision #	Revision Date	Description of Change	Author
1	**01/24/2013**	Original Write-up	Lachlan Plant

2. Distribution

Recipient Name	Recipient Organization	Distribution Method
Vladimir Wojcik	Brock University	

TABLE OF CONTENTS

1. INTRODUCTION

(NOTE 1: THE SOFTWARE TEST PLAN GUIDELINES WERE DERIVED AND DEVELOPED FROM IEEE STANDARD FOR SOFTWARE TEST DOCUMENTATION (829-1998).

This Software Test Plan has been designed in order to provide the scope, approach resources, methodologies and schedule for the testing activities involved in the Brock Butler project. This plan is to be used to identify the items and features that are to be tested, as well as the type of testing to be preformed and the methodologies used in this testing.

1.1 Objectives

The overall objective of the testing to be performed on the Brock Butler System is to ensure a fully functional product. Testing is to be performed in a variety of different ways, as outlined below, to ensure the bug free nature of this software.

1.2 Testing Strategy

This test plan will provide a high level description of the methodologies and aspects of testing on the Brock Butler Application. This document will cover the following aspects of testing.

- Purpose of test,
- Items to be tested,
- Methods of testing
- Testing Documentation.
- Features to be tested,
- Features not to be tested,
- Pass / Fail criteria,
- Schedules, and
- Risk assumptions and constraints.

1.3 Reference Material

The following documents have been used, and will be used, for the creation of this document, as well as the creation of test cases.

- Phase Plan
- Standards Document

- Scope Document
- System Design Document
- Use Case Diagrams
- Use Case Documents
- UML Class Diagrams
- Test Plan Template[1]

2. TEST ITEMS

2.1 Program Modules

An initial test of each module is to be performed by the developer of said module. This test will be done to ensure the base functionality of the module so that it can be adequately tested by the testing team, without having to test for basic programming logic errors. Programs Modules includes integration testing as well as user interface testing.

2.2 Job Control Procedures

Not applicable.

2.3 User Procedures

All user documentation, such as installation and instruction manual will be tested in a step by step manner in accordance with the steps defined in those manuals. Testing performed will be to ensure the correctness and completeness of the instructions provided to the user.

2.4 Operator Procedures

Not applicable.

3. FEATURES TO BE TESTED

All aspects of the software, with the exceptions listed below, will be tested. Include in this is user documentation as well as testing to the acceptance criteria.

4. FEATURES NOT TO BE TESTED

The system will use the calendar implemented into the Android operating system by Google. Due to the heavily tested nature of this third party system this aspect of the application will not be tested further. The databaseing system SQL Lite will also not be tested outside of the scope of the software environment for the same reason as the calendar.

5. APPROACH

The following section describes the techniques that will be used in the various aspects of testing. It outlines what must be created for the tests as well as the method that will be used to perform the test. Also outlined is the condition that must be met for each test in each testing group to be considered a pass.

5.1 Unit Testing

Unit testing is to be performed on the individual modals of the program. This is done to ensure the logical correctness of all individual aspects of the system to ensure they will be able to be implemented into the system as a whole. A driver will be created for each unit and used to test that unit. Expected results for the unit test will be derived from the use case documents provided by the unit's developer. Failure to meet the requirements from the use case will result in a failure for the module.

5.2 Integration Testing

The style of integration testing to be used on this software is bottom up integration. The process of this integration testing technique is to first group these units into clusters that form a subsystem of the overall program. A driver is then written for this cluster and the cluster is tested. After the completion of the clusters test the cluster is combined with clusters of the same level and the process is repeated with this cluster. Due to the modular nature of this software, the bottom up approach is best suited to this environment. Pass criteria for this form of testing will be derived from the clusters use case diagram.

5.3 Conversion Testing

Not Applicable.

5.4 Job Stream Testing

Not Applicable.

5.5 Interface Testing

Interface testing will be performed on both the Android runtime environment of Eclipse, as well as a physical device running the android operating system. This testing will be done to ensure the functionality of the user interface, as well as ass input methods associated with the android environment. Pass criteria for interface testing is a fully functional user interface free of graphical errors and input errors.

5.6 Performance Testing

Not Applicable

5.7 Regression Testing

Any changes made to a module after the testing of that module has occurred will result in the retesting of this module. In addition to this any modules or subsystems which rely on the changed module will be retested to ensure no adverse effects have occurred within these modules.

5.8 Acceptance Testing

Acceptance testing will occur on the system as a whole. The expected results of this testing will be derived from the acceptance document. All aspects of the acceptance criteria will be tested to ensure full functionality as defined in the acceptance document has been achieved.

5.9 Beta Testing

Not Applicable due to time constraint.

6. PASS / FAIL CRITERIA

This section is used to determine the criteria used for the passing or failing of modules, as well as the procedures to be implemented in the case of a modules failure.

6.1 Suspension Criteria

If there is a failure within a specific module all testing on modules directly dependent on that module will be suspended to ensure the fix implemented does not have any side effects on other dependent modules.

6.2 Resumption Criteria

To resume testing on the specific modules the fix needs to have been successfully completed on the failed module. This module will then be repeated, as well as all modules directly dependent on the failed module.

6.3 Approval Criteria

For a module to receive a pass, it must all tests on that specific module must have results matching the expected results in the test case. Any deviations from this result will result in a fail for the module.

7. TESTING PROCESS

This Section will identify the methods and criteria used in performing test activities.

7.1 Test Deliverables

Upon the completion of a test a Test Case Report document will be submitted to the main testing repository as well as to the modules designer. In the event of a modules failure an incident report will also be submitted to the modules designer with the error and description of the steps performed to achieve the error.

7.2 Testing Tasks

The preparation of each test case is to be created using the use case diagrams created by the designer of that module. Testing documents will include all steps to be taken in the performance of the test as well as the expected results of the testing. Drivers to be used in the testing will be created using the relevant modules class diagram.

7.3 Responsibilities

Management, design, and preparation of all test documents are the responsibilities of the test leader Lachlan Plant. Testing is the responsibility of both the module designer, as well as the test leader. Resolution of incidents resulting from the testing is to be resolved by the creator of the module.

7.4 Resources

All testers are to have access to both the eclipse Android emulator as well as an Android device compatible with Android 4.0 in order to adequately preform all testing activities.

7.5 Schedule

March 24th, 2013	Unit Testing
March 27th, 2013	Integration Testing
March 18th, 2013	Interface Testing
March 31th, 2013	Acceptance Testing
March 31th, 2013	User Procedure Testing

8. ENVIRONMENTAL REQUIREMENTS

(Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. Also provide the levels of security required to perform test activities. Identify special test tools needed and other testing needs (space, machine time, and stationary supplies. Identify the source of all needs that is not currently available to the test group.)

8.1 Hardware

The testing will be performed on two separate devices. Initial testing of individual modules and some integration testing will be performed on a windows PC located in the Brock Computer Science laboratories. The secondary hardware device to be used for testing will be a Samsung Galaxy 10.1 Tab 2. This device will be used in system testing, to ensure the performance of the application on the intended device platform.

8.2 Software

The software required for in lab testing is the Android Eclipse IDE, specifically the Android operating system runtime emulator. The version of the Java runtime environment to be installed on the computers used for testing is Java 7. Software required for the Android device being tested on is Android 4.0 (Ice Cream Sandwich).

8.3 Security

All computers and devices used for this testing are to be password protected and locked when the user is away from the workspace. This is done to ensure plagiarism by other parties cannot occur.

8.4 Tools

No additional tools are required for the testing of this software.

8.5 Publications

The Standards document will be used to ensure that all tests comply with group standards. The test case for each unit test is to be derived from that specific units Use-Case report.

8.6 Risks and Assumptions

1) Schedule
   - The schedule for testing is to be found in section 7.5. In the event of the push back of a deadline prior to the testing of modules additional testers will be recruited in order to ensure the completion of testing.
2) Resources
   - Availability of devices that can run Android 4.0 may be a restraint. In the occurrence of a failure of testing device a replacement device will be located to ensure the completion of the testing according to the schedule.

9. CHANGE MANAGEMENT PROCEDURES

All potential changes must be submitted in writing and approved by the test leader.

10. PLAN APPROVALS

____

Lachlan Plant

____

Taras Mychaskiw

[1] North Carolina Enterprise Project Management Office www.epmo.scio.nc.gov/library/docs/**testplan**.doc