Functional Test Plan
Project Name: Wonky Android Calculator
Product Owner:
Project Manager:
Document Date: 27/10/2017

1. Functional Test Plan Scope

In Scope	Out of Scope
In Scope: - *Testing of all functional, application performance, compatibility, security and use cases requirements listed in the Use Case document.* - *Quality requirements and fit metrics* - *End-to-end testing and testing of interfaces of all systems that interact with the application*	Out of Scope: - *Functional requirements testing for systems outside the application.* - *Testing of Business SOPs, disaster recovery and Business Continuity Plan.*

2. Functional Test Plan Assumptions and Constraints

Functional Test Plan Assumptions

Assumptions:

- *For User Acceptance testing, the Development team has completed unit, system and integration testing and met all the Requirements (including quality requirements) based on Requirement Traceability Matrix.*
- *User Acceptance testing will be conducted by End-users*
- *Test results will be reported on daily basis using a selected tool. Failed scripts and defect list from this tool with evidence will be sent to Developer directly.*
- *Test scripts are developed and approved.*
- *Test Team will support and provide appropriate guidance to Developers to conduct testing*
- *Major dependencies should be reported immediately after the testing kickoff meeting.*

Functional Test Plan Constraints

Constraints:

- *Developer will receive consolidated list of request for test environment setup.*
- *Developer will support ongoing testing activities based on priorities*
- *Test scripts must be approved by Test Lead prior test execution*
- *Test scripts, test environment and dependencies should be addressed during testing kickoff meeting in presence of a SME and request list should be submitted within 3 days of the kickoff meeting*
- *The Developer cannot execute the User Acceptance and End to End test scripts. After debugging, the developer can conduct their internal test, but no results from that test can be recorded / reported.*

3. Functional Test Team Roles & Responsibilities		
Name	**Roles**	**Responsibilities**
	Project Manager	*(a) Acts as a primary contact for development and QA team.* *(b) Responsible for Project schedule and the overall success of the project.*
	QA Lead	*(a) Participation in the project plan creation/update process.* *(b) Planning and organization of test process for the release.* *(c) Coordinate with QA analysts/engineers on any issues/problems encountered during testing.* *(d) Report progress on work assignments to the PM*
	QA Tester	*(a) Understand requirements* *(b) Writing and executing Test cases* *(c) Preparing RTM.* *(d) Reviewing Test cases, RTM.* *(e) Defect reporting and tracking.* *(f) Retesting and regression testing.* *(g) Bug Review meeting* *(h) Preparation of Test Data.* *(i) Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling.*

4. Functional Test Entry & Exit Criteria	
ID	**Criteria**
4.1	Entry Criteria: - *(a) Business Requirement Document complete* - *(b) Functional Specification* - *(c) High-Level Design Specifications* - *(d) Software Development Project Plan approved* - *(e) Software Verification Plan approved* - *(f) Software Configuration Management Plan* - *(g) Quality Assurance Plan approved.*
4.2	Exit Criteria: - *(a) All Show Stopper defects or Blockers are fixed and No known Critical / Severity 1 defect is in Open Status.*

	- *(b) The software runs on all the product's supported hardware and software configurations.* - *(c) If any medium or low-priority errors are outstanding – the implementation risk must be signed off as acceptable by Business Analyst and Business Expert.* - *(d) 100% Requirements coverage is achieved.* - *(e) Very few low priority open defects that do not impact software usage.* - *(f) Project Deadline or Test Finish deadline is reached.*

5. Functional Test Cases	
ID	**Test Cases**
5.1	Test Case: User can perform 'Add' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Add' button from the options on the screen **one time*** 3. *Check the 'Result' field under the last button* ***Expected Results:*** *Result should be equal to '1' if no operations were performed previously. 'Add' operation result should always be equal to **'result value'+1.***
5.2	Test Case: User can perform 'Subtract' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Subtract' button from the options on the screen **one time*** 3. *Check the 'Result' field under the last button* ***Expected Results:*** *Result should equal '-1' if no operations were performed previously. 'Subtract' operation result should always be equal to **'result value'-1.***
5.3	Test Case: User can perform 'Square root' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Add' button from the options on the screen **four times*** 3. *Check the 'Result' field under the last button, result should be equal to **4*** 4. *Tap 'Square root' button from the options on the screen **one time*** 5. *Check the 'Result' field under the last button* 6. *Repeat the calculation taking the square root for **'9'*** 7. *Check the 'Result' field under the last button* ***Expected Results:*** *Result 1 should equal '2' for square root from 4 operation.* *Result 2 should equal '3' for square root from 9 operation*
5.4	Test Case: User can perform 'Divide by 2'' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Add' button from the options on the screen **two times*** 3. *Check the 'Result' field under the last button, should be equal to '2'* 4. *Tap 'Divide by 2' button from the options on the screen one time*

	5. *Check the 'Result' field under the last button* ***Expected Results:*** *Result should equal to '1' if no operations were performed previously. 'Divide by 2' operation result should always be equal to '**result value' / 2***
5.5	Test Case: User can perform 'Multiply by 2'' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Add' button from the options on the screen **two times*** 3. *Check the 'Result' field under the last button, should be equal to '2'* 4. *Tap 'Multiply by 2' button from the options on the screen **two times*** 5. *Check the 'Result' field under the last button* ***Expected Results:*** *Result should equal to '**8**'.'Multiply by 2' operation result should always be equal to '**result value' * 2***
5.6	Test Case: User can perform 'Power by 2'' operation and receive the expected result ***Test Procedure:*** 1. *Start the application on the Android device with the OS version not lower than … (earliest supported version)* 2. *Tap 'Add' button from the options on the screen **two times*** 3. *Check the 'Result' field under the last button, result should be equal to **2*** 4. *Tap 'Power by 2'' button from the options on the screen **one time*** 5. *Check the 'Result' field under the last button* 6. *Repeat the calculation performing 'Power by 2' operation for number **3*** 7. *Check the 'Result' field under the last button* ***Expected Results:*** *Result 1 should equal '4' for 2 powered by 2.* *Result 2 should equal '9' for 3 powered by 2* *'Power by 2' operation result should always be equal to **'result value' * 'result value'***

6. Functional Test Results				
ID	**Test Cases**	**Pass/Fail**	**Tested By**	**Date Tested**
6.1	*User can perform 'Add' operation and receive the expected result.*	*Pass*	*QA Tester*	*27/10/2017*
6.2	*User can perform 'Subtract' operation and receive the expected result*	*Pass*	*QA Tester*	*27/10/2017*
6.3	*User can perform 'Square root' operation and receive the expected result*	*Pass*	*QA Tester*	*27/10/2017*
6.4	*User can perform 'Divide by 2'' operation and receive the expected result*	*Pass*	*QA Tester*	*27/10/2017*
6.5	*User can perform 'Multiply by 2'' operation and receive the expected result*	*Pass*	*QA Tester*	*27/10/2017*
6.6	*User can perform 'Power by 2'' operation and receive the expected result* *Corresponding defect **ID 7.1***	*Fail*	*QA Tester*	*27/10/2017*

7. Addendums & Appendices

7. Functional Defects				
ID	**Summary**	**Priority**	**Tested By**	**Date Tested**
7.1	'Power by 2'' operation returns incorrect result. ***Actual Results:*** *'Power by 2' for 3 returns 12, 'power by 2' for 4 returns 20.* ***The following calculation is performed instead of the correct formula: 'result value' * 'result value' + 'result value'*** ***Expected Result:*** *'Power by 2' operation result should always be equal to* ***'result value' * 'result value'***	*Critical*	*QA Tester*	*27/10/2017*
7.2	Combination of 'Power by 2' and 'Divide by 2' operations returns incorrect result ***Steps to reproduce:*** 1. *Tap 'Add' 4 times (Result: 4)* 2. *Then Tap 'Power by 2' one time (Result: 20)* 3. *Tap 'Divide by 2' one time* ***Actual Result:*** *1E+1* ***Expected Result:*** *20 divided by 2 should return 10* *Screenshot: Link*	*Hight*	*QA Tester*	*27/10/2017*