Transaction Tracker

**Test Document**

Version 1.0.0

Prepared By

|  |  |  |
| --- | --- | --- |
|  | Student ID | Email Address |
| Shemar Henry | 180915 | Shemarhenry24@gmail.com |
| Mark-Anthony Jones | 180920 | Jmarkanthony.062@gmail.com |
| Shemar Lundy | 180916 | lundyshemar@yahoo.com |
| Dejeon Battick | 180915 | Dejeon.battick11@gmail.com |
|  |  |  |

Software Testing

Instructed by Thomas Canhao Xu

June 22, 2019

Table of Contents

[Table of Contents 1](#_Toc12119677)

Document Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Changes | Author | Approver |
| 1.0 | June 22, 2019 | Original Plan | Mark-Anthony Jones |  |

# Test Plan

## Test Plan Identifier:

This test plan is referred as Transition Tracker Test 1.0

## Introduction

## Background

Transaction Tracker, created by TPGD, is an application intended for the end user to track their expenses as well as create a budget and assist in making sound financial decisions. The application was developed using the Android Studio IDE with the Java programming language. The application features a Bottom Navigation Drawer with four menus, each representing Personal, Expenses, Income and Budget respectively. This application also features a SQLite database which stores user input. The end user can clear the database as they choose.

This test plan is intended to test the Transaction Tracker dynamic quality attributes, namely: reliability, correctness, completeness, consistency, usability and performance.

## Objectives of the Test Plan

The test plan encompasses the entirety of the Transaction Tracker application. What will be tested is the dynamic quality attributes of the software, focusing on specifics such as

* The probability of failure-free operation of the application
* How the application’s functionality corresponds to the documentation
* The presence of all the features laid out in the documentation
* How consistent the application performs
* Ease of use
* Performance under stress

For the testing process, four members will be recruited to test the application. A Software Requirements Specification document will be provided by the TPGD team. The Espresso framework in Android Studio will be used to extensively test the application. The testing method being used is white box testing as Espresso requires that the code be fully known to the tester to perform test cases. Expected deliverables include: a test incident report, a test log and a test summary report.

## References

The TPGD Transaction Tracker SRS Document

# Test Items

Testing will be done on the front end of the application using the Android Studio Espresso framework on Macintosh and Linux environments

## Features to Be Tested

* Daily Expenses and Transactions
* Income
* Expenses
* Chart
* Clear Expenses and Transactions

## Features Not to Be Tested

* “Spin the Wheel”
* Calculator
* Settings

# Approach

Domain testing, structural testing, stress testing, robustness testing and functional testing will be employed in this test procedure.

# Pass/Fail Criteria

No more than 95% of test cases should fail and there should exist no critical bugs

# Test Deliverables:

Test Incident Report, Test Summary Report, Test Logs, Test Cases

# Responsibilities:

Shemar does

Shemar does

Dejeon does

Mark does

# Schedule:

2 weeks

# Test Design Specification

Specify testing techniques, methods of analyzing results, common attributes of test cases (input constraints)

# Test Case Specification

Test Case Objective:

Input Specifications:

Output Specifications:

<do for each test case>

# Test Log

Test Case Objective:

Expected Results:

Actual Results:

Requirement:

<do for each test case>

# Test Incident Report

Summary:

Description:

* Feature tested
* Test case
* Status: New
* Tested by:

Severity

# Test Summary Report

<whatever>

# Acknowledgements