© [**www.SoftwareTestingHelp.com**](http://www.softwaretestinghelp.com)

Mobile Bank App Demo Automation Test Plan

KATALON STUDIO COURSES - HACKTIV8

13/07/2022

Hafidz Firmansyah Ghufara

**Version:** 1.0

**Created:** 13/07/2022

**Last Updated:** 13/07/2022

**Status:** DRAFT

**Revision and Signoff Sheet**

**Document History** - To maintain a list of changes being made

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description of Change** |
| 1 | 13/07/2022 | Hafidz Firmansyah G. | Draft |
|  |  |  |  |
|  |  |  |  |

**Approvers List** - To track who has reviewed and signoff on the Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Approver / Reviewer | Approval / Review Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reference Documents** - Clearly mark the document used as an input to create the test plan

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Document Name** |
| 1.0 |  | Mobile Automation Test Scenario |
|  |  |  |

**Table of Contents**

[1. INTRODUCTION 5](#_30j0zll)

[1.1. Purpose 5](#_1fob9te)

[1.2. Project Overview 5](#_3znysh7)

[1.3. Audience 5](#_2et92p0)

[2. TEST STRATEGY 5](#_tyjcwt)

[2.1. Test Objectives 5](#_3dy6vkm)

[2.2. Test Assumptions 6](#_1t3h5sf)

[2.3. Test Principles 7](#_4d34og8)

[2.4. Data Approach 7](#_2s8eyo1)

[2.5. Scope and Levels of Testing 7](#_17dp8vu)

[2.5.1. Exploratory 7](#_3rdcrjn)

[2.5.2. Functional Test 8](#_26in1rg)

[TEST ACCEPTANCE CRITERIA 8](#_35nkun2)

[TEST DELIVERABLES 8](#_1ksv4uv)

[MILESTONE LIST 9](#_44sinio)

[2.5.3. User Acceptance Test (UAT) 9](#_z337ya)

[TEST DELIVERABLES 9](#_3j2qqm3)

[2.6. Test Effort Estimate 10](#_1y810tw)

[3. EXECUTION STRATEGY 10](#_2xcytpi)

[3.1. Entry and Exit Criteria 10](#_1ci93xb)

[3.2. Test Cycles 11](#_3whwml4)

[3.3. Validation and Defect Management 11](#_2bn6wsx)

[3.4. Test Metrics 12](#_qsh70q)

[3.5. Defect tracking & Reporting 13](#_3as4poj)

[4. TEST MANAGEMENT PROCESS 13](#_1pxezwc)

[4.1. Test Management Tool 13](#_49x2ik5)

[4.2. Test Design Process 14](#_2p2csry)

[4.3. Test Execution Process 15](#_147n2zr)

[4.4. Test Risks and Mitigation Factors 16](#_3o7alnk)

[4.1. Communications Plan and Team Roster 17](#_23ckvvd)

[4.2. Role Expectations 17](#_ihv636)

[4.2.1. Project Management 18](#_32hioqz)

[4.2.2. Test Planning (Test Lead) 18](#_1hmsyys)

[4.2.3. Test Team 18](#_41mghml)

[4.2.4. Test Lead 18](#_2grqrue)

[4.2.5. Development Team 18](#_vx1227)

[5. TEST ENVIRONMENT 19](#_3fwokq0)

# INTRODUCTION

## Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the Mobile Bank App Demo Automation Test. The document introduces:

* Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
* Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
* Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

## Project Overview

This project is a mobile UI automation test project with a subject of an open source Bank App Demo provided by [Mike-Banks](https://github.com/Mike-Banks). The project will be done using tools such as Katalon Studio, Appium, and Android 9.0.

The scope of the project includes testing some functionality of the application, including user credentials (register/create a profile, login, logout), creating bank accounts, and payments. No actual money will be involved in any of the test.

# TEST STRATEGY

## Test Objectives

The objective of the test is to verify that the functionality of Bank App Demo works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via Developer

The final product of the test is:

* A Katalon Studio project, uploaded to a github repository.

## Test Assumptions

**Key Assumptions**

* The application version and functionality will be treated as is during the testing period, any further update to the application outside this project period may or might not reflect the result of the test.
* No real money will or can be used during testing.
* The project is done individually by the author.

**General**

* Exploratory Testing will be carried out as author sees fit.
* Performance testing is not considered for this estimation.
* Test scenario, test case design will be owned by the author.
* Automation test design will be carried out by the author.
* Git/Github will be used as version control.
* Test environment will be carried using Katalon Studio.
* JIRA integration will be used for at least a single test case.
* Changes to test environment is owned by the author.
* The APK for Bank App Demo is provided by Hacktiv8’s team.

**Functional Testing**

* The scope of functional testing includes user credentials (profile creation and login), bank accounts, and transactions.
* The author will perform Functional testing only on Bank App Demo provided apk.

**UAT**

* No UAT planned yet.

## Test Principles

* Testing will be focused on meeting the business objectives, cost efficiency, and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing environment and data will emulate a production environment as much as possible.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.

## Data Approach

* In functional testing, Bank App Demo will contain pre-loaded test data and which is used for testing activities.

## Scope and Levels of Testing

### Exploratory

**PURPOSE**: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

**SCOPE**: First level navigation, dealer and admin modules

**TESTERS**: Testing team.

**METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation

**TIMING**: at the beginning of each cycle.

### Functional Test

**PURPOSE:**  Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

**Scope:** The below excel sheet details about the scope of Functional test. Note: The scope is high level due to changes in the requirement.

To keep the document easily fragmented and categorized, the scope has been embedded as separate document. If you prefer you can insert a table here itself. The scope is created based on the Test scenarios that were identified in the previous article.

**TESTERS**: Testing Team.

**METHOD**: The test will be performed according to Functional scripts, which are stored in JIRA.

**TIMING**: after Exploratory test is completed.

#### TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
2. Test cases approved and signed-off prior to start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with application installed, configured and ready to use state

Sign-off

Approved Functional Specification Document

Approved Use cases

Approved Test cases

Readiness

Development completed & unit tested

Application deployed and system ready for testing on Test environment

Production like data is available to test all functionalities.

Defect fixes planned based on Defect triage (Unit Testing) and evaluation criteria

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Hafidz Firmansyah G | M. Fariz Agati |
| 2. | Test Scenario | Hafidz Firmansyah G | M. Fariz Agati |
| 3. | Functional Test Cases | Hafidz Firmansyah G | M. Fariz Agati |
| 4. | Jira integration | Hafidz Firmansyah G | M. Fariz Agati |
| 5. | Katalon project committed to Git | Hafidz Firmansyah G | M. Fariz Agati |
| 6. | Documentation on Github README | Hafidz Firmansyah G | M. Fariz Agati |

#### MILESTONE LIST

The milestone list is tentative and may change due to below reasons

1. Any issues in the System environment readiness
2. Any change in scope/addition in scope
3. Any other dependency that impacts efforts and timelines

Testing generally is not carried out in one cycle. Based on the testing scope, we can estimate how much time it takes and establish the time lines as you can see in the below embedded excel sheet.

### User Acceptance Test (UAT)

**PURPOSE**: this test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

**TESTERS**: the UAT is performed by the end users (L1, L2 and L3).

**METHOD**: Since the business users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End user (L1,L2 and L3 users) and Business Analyst’s.

**TIMING**: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

#### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | UAT Test Cases | Test Team | Business Analyst’s Sign off |

# EXECUTION STRATEGY

## Entry and Exit Criteria

* The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
* The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
* Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final “go-no go” decision.
* Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
* Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| 100% Test Scripts executed |  |  |  |
| 95% pass rate of Test Scripts |  |  |  |
| No open Critical and High severity defects |  |  |  |
| 95% of Medium severity defects have been closed |  |  |  |
| All remaining defects are either cancelled or documented as Change Requests for a future release |  |  |  |
| All expected and actual results are captured and documented with the test script |  |  |  |
| All test metrics collected based on reports from JIRA |  |  |  |
| All defects logged in JIRA |  |  |  |
| Test Closure Memo completed and signed off |  |  |  |
| Test environment cleanup completed and a new back up of the environment |  |  |  |



## Test Cycles

* + There will be two cycles for functional testing. Each cycle will execute all the scripts .
  + The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
  + The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.

## Validation and Defect Management

* It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the Business analyst’s join the TCOE in the execution of the test, since the BUSINESS ANALYSTs have a deeper knowledge of the business processes. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
* The defects will be tracked through Jira only. The technical team will gather information on a daily basis from JIRA, and request additional details from the Defect Coordinator. The technical team will work on fixes.
* It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review Jira on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool “Jira” and the categories are:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This bug is critical enough to crash the system, cause file corruption, or cause potential data loss * It causes an abnormal return to the operating system (crash or a system failure message appears). * It causes the application to hang and requires re-booting the system. |
| 2 (High) | * It causes a lack of vital program functionality with workaround. |
| 3 (Medium) | * This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. * This bug prevents other areas of the product from being tested. However other areas can be independently tested. |
| 4 (Low) | * There is an insufficient or unclear error message, which has minimum impact on product use. |
| 5(Cosmetic) | * There is an insufficient or unclear error message that has no impact on product use. |

## Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **Report** | **Description** | **Frequency** |
| Test preparation & Execution Status | To report on % complete, %WIP, % Pass, % Fail  Defects severity wise Status – Open, closed, any other Status | Weekly / Daily (optional) |
| Daily execution  status | To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects | Daily |
| Project Weekly Status report | Project driven reporting (As requested by PM) | Weekly – If project team needs weekly update apart from daily and there is template available with project team to use. |

## Defect tracking & Reporting

Not planned yet.

# TEST MANAGEMENT PROCESS

## Test Management Tool

HP Application Lifecycle Management is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the JIRA Software tool.

* Project specific folder structure will be created in JIRA to manage the status of this DFRT project.
* Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in JIRA.
* During the Test Design phase, all test cases are written directly into JIRA. Any change to the test case will be directly updated in JIRA.
* Each Tester will directly access their respective assigned test cases and update the status of each executed step in JIRA directly.
* Any defect encountered will be raised in JIRA linking to the particular Test case/test step.
* During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in JIRA.
* Various reports can be generated from JIRA to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

## Test Design Process

Understanding Requirements

Establishing Traceability Matrix in JIRA

Preparation of Test cases

SME /Peer Review of Test cases

Incorporating Review comments in test cases

* The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
* Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
* Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
* During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
* Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
* Sign-off for the test cases would be communicates through mail by Business Analyst’s.
* Any subsequent changes to the test case if any will be directly updated in JIRA.

## Test Execution Process

Execute each of the test step in test case

Mark Status as Pass/Fail in JIRA

Raise defects for the failed test cases in JIRA

Send the daily status report to Test Lead

Participate in Defect Triage cycle and explain the defects

Complete the test execution of all the test cases

* Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.
* Each Tester is assigned Test cases directly in JIRA.
* Testers to ensure necessary access to the testing environment, JIRA for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
* If any showstopper during exploratory testing will be escalated to the respective development SPOCs for fixes.
* Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in JIRA.
* Tester will prepare a Run chart with day-wise execution details
* If any failures, defect will be raised as per severity guidelines in JIRA tool detailing steps to simulate along with screenshots if appropriate.
* Daily Test execution status as well as Defect status will be reported to all stakeholders.
* Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
* If there are any defects that are not part of steps but could be outside the test steps, such defects need to be captured in JIRA and map it against the test case level or at the specific step that issue was encountered after confirming with Test Lead.
* This process is repeated until all test cases are executed fully with Pass/Fail status.
* During the subsequent cycle, any defects fixed applied will be tested and results will be updated in JIRA during the cycle.

As per Process, final sign-off or project completion process will be followed

## Test Risks and Mitigation Factors

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Prob.** | **Impact** | **Mitigation Plan** |
| **SCHEDULE**  Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | * The testing team can control the preparation tasks (in advance) and the early communication with involved parties. * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **RESOURCES**  Not enough resources, resources on boarding too late (process takes around 15 days. | Medium | High | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing. |
| **DEFECTS**  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Medium | Medium | Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution. |
| Delayed Testing Due To new Issues | Medium | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

## Communications Plan and Team Roster

## Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

|  |  |  |  |
| --- | --- | --- | --- |
| **SN0.** | **Roles** | **Name** | **Contact Info** |
| 1. | Project Manager | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 2. | Test Lead | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 3. | Business Analyst | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 4. | Development Lead | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 5. | Testing Team | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 6. | Development Team | Hafidz Firmansyah G. | hafidzfg@gmail.com |
| 7. | Technical Lead | Hafidz Firmansyah G. | hafidzfg@gmail.com |

### Project Management

* Project Manager: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

### Test Planning (Test Lead)

* Ensure entrance criteria are used as input before start the execution.
* Develop test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
* Provide guidelines on how to manage defects.
* Attend status meetings in person or via the conference call line.
* Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
* Provide on premise or telecommute support.
* Provide functional (Business Analysts) and technical team to test team personnel (if needed).

### Test Team

* Develop test conditions, test cases, expected results, and execution scripts.
* Perform execution and validation.
* Identify, document and prioritize defects according to the guidance provided by the Test lead.
* Re-test after software modifications have been made according to the schedule.
* Prepare testing metrics and provide regular status.

### Test Lead

* Acknowledge the completion of a section within a cycle.
* Give the OK to start next level of testing.
* Facilitate defect communications between testing team and technical / development team.

### Development Team

* Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
* Assist in the validation of results (if requested).
* Support the development and testing processes being used to support the project.
* Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
* Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
* Define processes/tools to facilitate the initial and ongoing migration of components.
* Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate defects.
* Implement fixes to defects according to schedule.

# TEST ENVIRONMENT

Bank App Demo will be installed in each tester’s device/virtual device used for testing.

Bank App Demo APK will be provided by Hacktiv8, it will also be included in the Katalon root project folder.

A windows environment Katalon Studio 8.3.5, Appium 12+, Java 8, and an Android 9.0 version will be used.

# APPROVALS

The Names and Titles of all persons who must approve this plan.

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** | **M. Fariz Agati** |
| **Role:** |  |
| **Date:** |  |

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |