**Testing Plan**

**<Internet Banking System>**

**QA TRACK ITI**

**SMART VILLAGE ITI**

**GIZA, SMART VILLAGE 12511**

**07/05/2023**

Contents

[**Introduction:** 2](#_Toc134318098)

[**Objective of Testing:** 2](#_Toc134318099)

[**Scope of Testing:** 2](#_Toc134318100)

[**-In Scope:** 2](#_Toc134318101)

[**-Out Of Scope:** 2](#_Toc134318102)

[**Testing Levels:** 2](#_Toc134318103)

[**Testing Types:** 2](#_Toc134318104)

[**Testing Environment:** 2](#_Toc134318105)

[**Entry Criteria:** 2](#_Toc134318106)

[**-For Write Test Cases:** 2](#_Toc134318107)

[**-For Executing Test Cases:** 2](#_Toc134318108)

[**Exit Criteria:** 3](#_Toc134318109)

[**-From Write Test Cases:** 3](#_Toc134318110)

[**-From Executing Test Cases:** 3](#_Toc134318111)

[**Test Case Lifecycle:** 3](#_Toc134318112)

[**Review Strategy:** 3](#_Toc134318113)

# **Introduction:**

The purpose of this testing plan is to ensure that the internet banking system meets all of the required specifications and functions as intended. The testing plan will identify the scope of testing, test levels, objective, entry & exit criteria, test case lifecycle, test types, and the testing environment. The goal of this plan is to ensure that the system is reliable, functional, and user-friendly.

# **Objective of Testing:**

* Ensure the system under test conforms to functional software requirements specification (SRS).
* Bugs/issues are identified and fixed before go live.

# **Scope of Testing:**

## **-In Scope:**

Defines the features, functional requirements of the software that will be tested

## **-Out Of Scope:**

Nonfunctional requirements and anything else outer from features in software requirements specification (SRS).

# **Testing Levels:**

**System testing:** verify that the system is meeting all of the specified functional requirements

And we will not tested any interaction or integration with other systems.

**Acceptance Testing:** ensure the software meets customer’s needs and expectations.

# **Testing Types:**

**Functional Testing:** to ensure that the system is functioning as intended and that it meets all of the specified requirements.

# **Testing Environment:**

* Windows 10 and above.
* Chrome version 106.0.5249.103 and above.

# **Entry Criteria:**

## **-For Write Test Cases:**

Completeness of writing and revision SRS and other document related.

## **-For Executing Test Cases:**

Completeness of code, design and other document related.

# **Exit Criteria:**

## **-From Write Test Cases:**

* Writing test cases, finishing reviewing and documentation of test cases document.
* 100% feature linked by test cases at least one test case for each feature.
* Working software as expected without crash.

## **-From Executing Test Cases:**

* Execute all test cases written in test case report.
* Finishing bug report for any bug found within executing.

# **Test Case Lifecycle:**

## **-Test Case Generation:**

In this phase, testing team documents the test cases.

## **-Test Case Documentation:**

1- Testers will write test cases report in excel file on drive and this file will uploaded daily to our GitHub repository this sheet include these parameters (Test Case, ID Name, Description, steps, Expected Result, Actual result, Designed By, Status)

2- Excel file for test cases separated into sheets every sheet named by one feature and include test cases relate this feature.

3- Tester will write ID manually depend on feature’s name and number of test cases.

## **-Test Case Review Strategy:**

Test cases that are prepared can be moved to other team member to review the test cases,

And may be review by coach or team’s assessor and write Review comments

## **-Rework:**

Test cases with Review comments are reworked.

## **-Test Case Finalization:**

Upon successful review, test cases are ready to execute. This is an important step. Once test cases are finalized, we can version the test case artifacts and baseline them.

# **Bug Lifecycle:**

If we have test case fail when execute test cases it is detect as a bug and we will write bug report in excel file which is same as test case report and its criteria and mention related Test case but in this step status will be defined as (open or closed) if test case fail it will follow next steps:

1. Assign bugs to developer to solve this defect with status (open)
2. After fixing bugs tester is doing confirmation test and verified if bug fixed true tester will change status to close, else reopen bug and return it to develop team.
3. If this bug is duplicated or rejected it will closed.