TEST PLAN FOR INTERNET BANKING SYSTEM

V1.0

20-MAY-2022

Prepared by: Aml Nasser

Contents

1. Introduction

1.1. Purpose of the document

The test plan describes the test approach and overall framework that will be used in testing. This document introduces:

- Test Strategy: rules the test will be based on, including the givens project (start /end, dates, assumptions) description of the process to set up a valid test.
- 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.

1.2. Project Overview

The internet banking system is relevant everywhere where banking exists. All banks will favor an online banking system over the traditional banking systems as it has more features and provides faster transfer methods. The scope of this project includes all development activities of the internet banking system. The internet banking system is a web-based system that offers clients access to multiple banking services through their banking accounts such as doing transfers, viewing their transfer history. The internet banking system allows clients transfer money in a secure and time saving manner.

1.3. Audience

- Project team members perform tasks specified in this document, provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants may take part in the user acceptance test to ensure the business is aligned with the results of the test.

2. Test Strategy

2.1. Test Objectives

The objective of the testing is to ensure that IBS (internet Banking system) is functioning as requested by the user and captured in the SRS document.

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 CR.

The final product of the test is twofold:

- A production-Ready software.
- A set of stable test scripts that can be reused for functional and UAT test execution.

2.2. Test Assumptions

2.2.1. Key Assumptions

Production like data required and be available in the system prior to start of Functional Testing.

2.2.2. General

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.
- All the defects would come along with a detailed description in the bug report and can have a snapshot of the error or the bug reported in the report.
- Defect fix plan should be included in this documentation.

2.2.3. User Acceptance testing (UAT)

UAT test will be executed by the user after providing the last release.

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- 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 will be a repeatable, quantifiable, and measurable activity.

2.4. Scope and Levels of testing

2.4.1. Unit Test:

PURPOSE: the purpose of this test is to validate that every function (unit) of the product is performing as it should be

SCOPE: Developing team make sure that the units are working as it should.

TESTERS: the developing team.

TIMING: after writing the code and done by reviews.

2.4.2. 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, Client, 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.

2.4.3. 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: functional testing should be done on all levels of the system (unit - component – system) most of the scope should be more focused on the functionality of the product.

- Types of testing provided:
    - Smoke Testing
    - Regression Testing
    - Integration Testing
    - System Testing
    - Black Box testing

TESTERS: Testing team

METHOD: the tests will be performed according to the functional scripts recorded in the test case sheet in Testing Folder.

TIMIING: after Exploratory testing is completed.

TEST ACCPTANCE CRITERIA:

1- Approved functional Specification document, Use Case diagram must be available.
2- Test cases should be approved before the Test execution
3- Development completed and unit tests are done (passed).

TEST DELIVERABLES:

ID	Name	Author	Reviewer
Test_deliver_01	Test Plan	Project Manager	Analyst
Test_deliver_02	Functional Test Cases	Test Team	Project Manager
Test_deliver_03	Bug Report	Test Team	Project Manager
Test_deliver_04	Test Closure report	Test Team	Project Manager

2.4.4. 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.

TESTER: The UAT will be performed by the end user

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.

TIMING: After all other levels of testing are done.

3. Execution strategy

3.1. 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.

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 test script			
Test Closer Memo Completed			

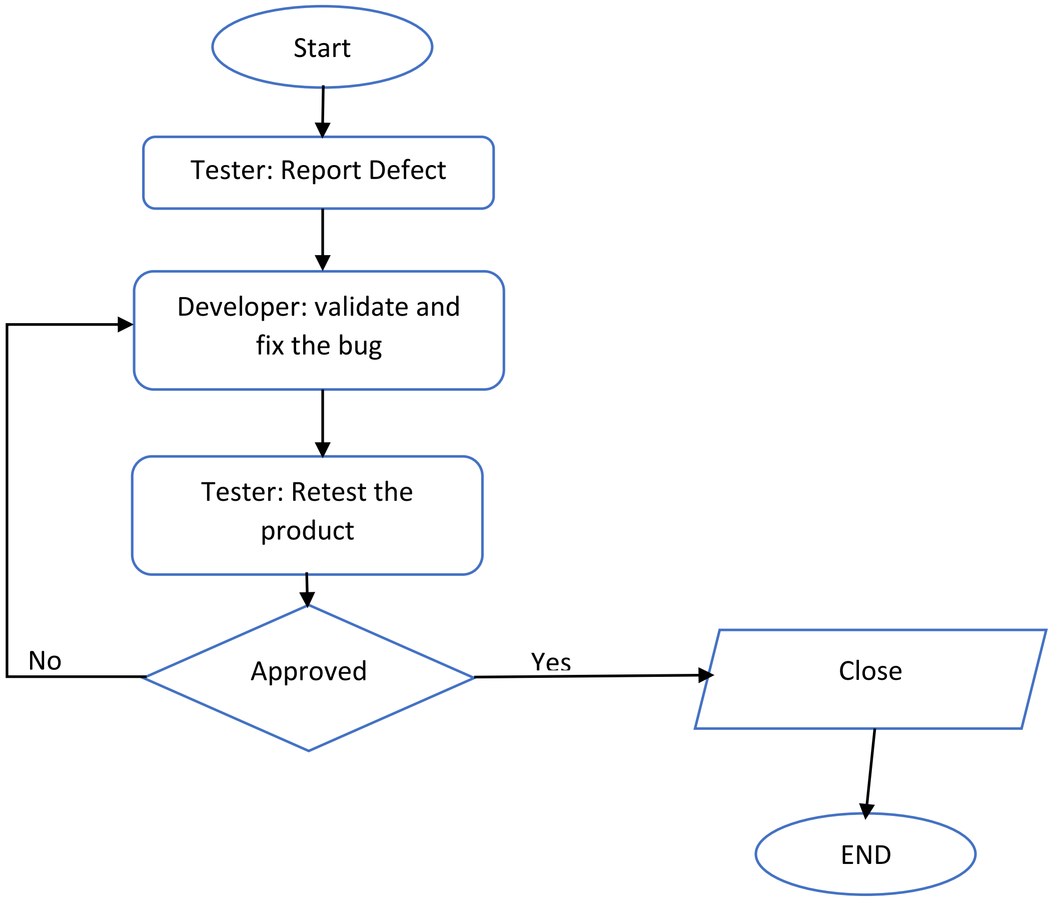
3.2. Validation and Defect Management

- It is expected that the testers execute all the scripts. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. 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 Traceability matrix and Trello. The technical team will gather information on a daily basis from Trello.
- 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.

Categories of Defects found during the Testing:

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.

3.3. Defect Tracking & Reporting


Start
Tester: Report Defect
Developer: validate and fix the bug
Tester: Retest the product
Approved
No
Yes
Close
END

4. Test Management process

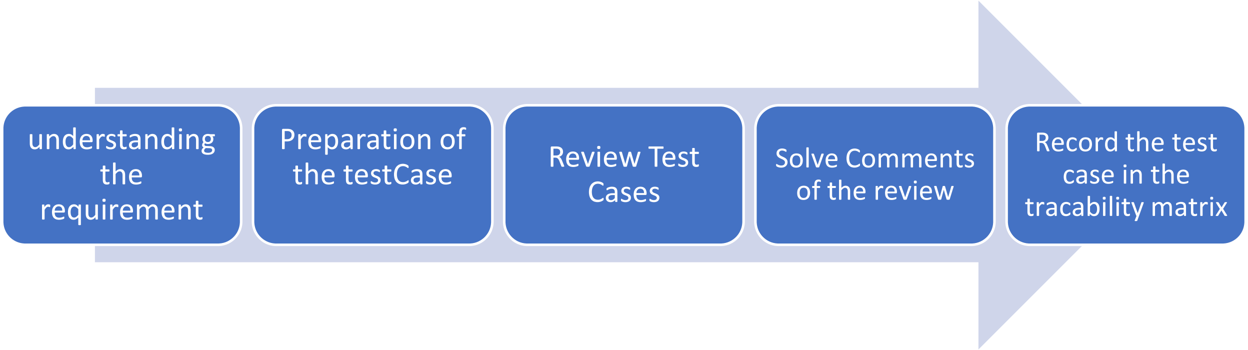
4.1. Test Management tool

Tasks: As every task in the project the tasks will be assigned to a member and to a specific deadline on TRELLO management tool.

Test Cases: The test cases are written in the test case template provided by the team and approved by the project manager by Excel Sheets.

Defects: The defect management system is done by writing a bug report with specific details that will be provided by the team by Excel Sheets.

4.2. Test Design process



understanding the requirement

Preparation of the testCase

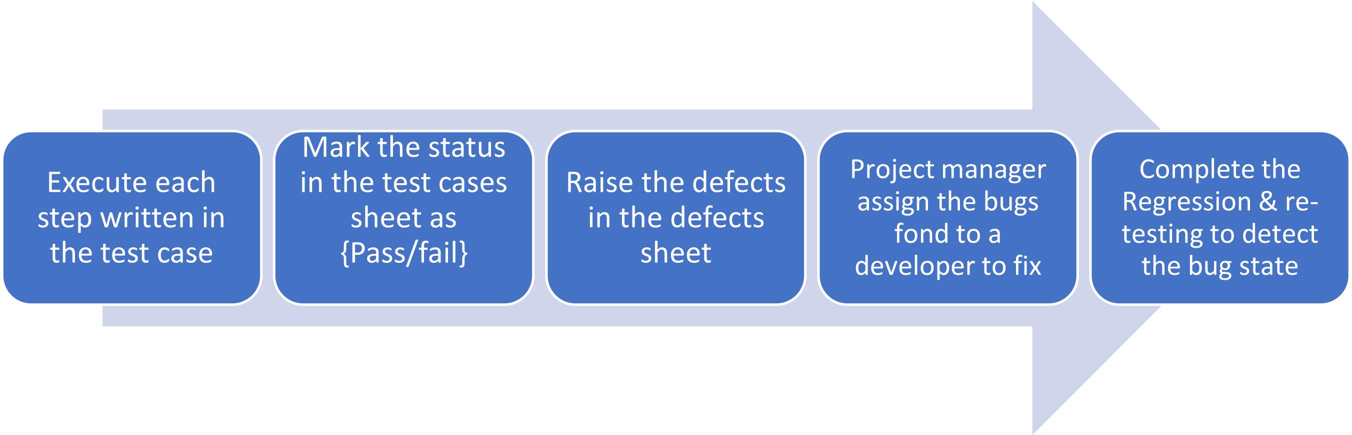
Review Test Cases

Solve Comments of the review

Record the test case in the tracability matrix

- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
- Each of the Test cases will undergo review a peer and the review defects are captured and shared to the Test team
- Each Test case will be mapped to Requirements as part of Traceability matrix.

4.3. Test execution process



Execute each step written in the test case

Mark the status in the test cases sheet as {Pass/fail}

Raise the defects in the defects sheet

Project manager assign the bugs fond to a developer to fix

Complete the Regression & re-testing to detect the bug state

- Once all Test cases are approved, tester will start an exploratory test of the application to ensure the application is stable for testing

- Each Tester is assigned Test cases directly
- If any showstopper during exploratory testing will be escalated to the respective development team member to fix
- Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each test case directly
- If any failures, defect will be raised as per severity guidelines detailing steps to simulate along with screenshots if appropriate.
- 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 Traceability matrix and map it against the test case level or at the specific step that issue was encountered after confirming with Test team
- 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

Revision History

Owner Name	Date	Status	Approved by	Version
Aml Nasser	20/5/2022	Draft		1.0