# **Test Strategy Document for Para Bank Application**

## **1. Introduction**

### **1.1 Purpose**

The purpose of this test strategy document is to outline the testing approach, objectives, scope, and methodologies for testing the Para Bank application. This document will guide the testing process to ensure the application meets the specified requirements and is free from defects.

### **1.2 Objectives**

The objectives of this test strategy are:

* To define the activities required to prepare for and conduct the test execution.
* To communicate the roles and responsibilities of the testing team.
* To define the sources of information used to prepare the plan.
* To identify the test tools and environment needed to conduct the testing process.

### **1.3 Scope**

The scope of testing includes the validation of functionalities, user interfaces, integration points, and performance of the Para Bank application. The testing types to be performed include:

* Functional Testing
* UI Testing
* Regression Testing
* Integration Testing
* System Testing

## **2. Test Strategy**

### **2.1 Test Case Development**

The test cases will be developed based on the functional requirements and user stories provided. A bottom-up approach will be applied:

* The application will be divided into features and functionalities.
* Test cases for each module will be designed, reviewed, and finalized.
* Any additional test cases identified during the execution phase will be incorporated and executed.

### **2.2 Testing Process**

The testing process will involve multiple levels of testing:

* **Unit Testing:** Conducted by the development team.
* **Component Testing:** Conducted by the QA team.
* **Integration Testing:** Conducted by the QA team to ensure modules work together.
* **System Testing:** Conducted by the QA team to validate the entire application.
* **User Acceptance Testing (UAT):** Conducted by the client.

### **2.3 Test Environment**

The application will be tested on the following environments:

* **Desktop:** Windows (various versions), macOS (various versions)
* **Browsers:** Chrome, Firefox, Edge, Safari
* **Mobile:** iOS and Android

## **3. Types of Testing Involved**

### **3.1 Functional Testing**

Validates that the application functions as expected according to the requirements.

### **3.2 UI Testing**

Ensures that the user interface meets the specified requirements, including layout, color, and font consistency.

### **3.3 Regression Testing**

Ensures that new code changes do not adversely affect existing functionality.

### **3.4 Integration Testing**

Verifies that different modules or services used by the application interact correctly.

### **3.5 System Testing**

Validates the complete and integrated software product.

## **4. Entry & Exit Criteria**

### **4.1 Entry Criteria**

* All required documentation is available.
* The test environment is set up and ready.
* Test data is prepared.

### **4.2 Exit Criteria**

* All planned test cases have been executed.
* All critical defects have been fixed and verified.
* UAT has been successfully completed.

## **5. Test Execution Strategy**

### **5.1 Understanding Requirements**

The QA team will review requirement specifications, raise queries, and get clarifications from the client.

### **5.2 Preparing Test Cases**

Test cases will be prepared based on the requirements and scenarios.

### **5.3 Traceability to Requirements**

An RTM will be created to ensure all requirements are covered by test cases.

### **5.4 Reviewing Test Cases**

Test cases will be peer-reviewed and approved.

### **5.5 Creating Test Data**

Test data will be created based on the scenarios and test cases.

### **5.6 Executing Test Cases**

Test cases will be executed, and results will be documented.

### **5.7 Defect Logging and Reporting**

Defects will be logged in Jira and tracked until resolution.

### **5.8 Retesting and Regression Testing**

Retesting of fixed defects and regression testing will be performed.

### **5.9 Deployment/Delivery**

The final tested build will be deployed to the client's test environment.

# **Test Scenarios for Para Bank Application**

### **Scenario 1: User Registration**

* Verify user can register with valid details.
* Verify error message for invalid input during registration.
* Verify email confirmation for successful registration.

### **Scenario 2: User Login**

* Verify user can log in with valid credentials.
* Verify error message for invalid login attempts.
* Verify account lockout after multiple failed login attempts.

### **Scenario 3:Open New Account**

* Verify Account Form Display.
* Verify Successful Submission.
* Verify Account Information.

### **Scenario 4: Account Overview**

* Verify account summary is displayed correctly.
* Verify account details are accurate.

### **Scenario 5: Transfer Funds**

* Verify user can transfer funds between accounts.
* Verify error message for invalid transfer details.
* Verify transaction history updates after transfer.

### **Scenario 6: Bill Pay**

* Verify user can add payees.
* Verify payment confirmation.

### **Scenario 7: Find Transaction**

* Verify transaction history is displayed correctly.
* Verify search functionality in transaction history.

### **Scenario 8: Update contact info**

* Verify user can update personal details.
* Verify changes are saved and reflected correctly.

#### **Scenario 9: Request loan**

* Verify user can apply for a loan with valid details.
* Verify user can view the status of their loan application.

### **Scenario 10: Logout**

* Verify user can logout successfully.
* Verify session is terminated after logout.

## 

## **Summary**

This document provides a comprehensive test strategy for the Para Bank application, including the scope, objectives, types of testing, test scenarios, detailed test cases, and a requirements traceability matrix (RTM) to ensure complete coverage and traceability of all functionalities. This structured approach aims to deliver a high-quality, defect-free banking application that meets all specified requirements.