Test Strategy for Guru99 Banking Application

---

1. Introduction

This Test Strategy document outlines the approach and resources required to test the Guru99 Banking Application. The main objective of the testing process is to ensure that all functionalities meet the specified requirements, the application is user-friendly, and there are no critical defects before deployment.

---

2. Objectives

- Verify that all functional requirements are met.
- Identify and report defects early to minimize risks.
- Validate the application's usability and performance.
- Ensure all integrated modules work seamlessly.

---

3. Scope

The testing covers the following modules and functionalities:

- **User Authentication**: Login and logout functionality for both Manager and Customer.
- **Account Management**:
  - Create, Edit, and Delete Customer Accounts.
  - Open and Close Bank Accounts.
- **Banking Transactions**:
  - Fund Transfers.
  - Deposits and Withdrawals.
  - Balance Enquiry.
- **Statements**:
  - Customized Statements.
  - Mini Statements.
- **Other Functionalities**:
  - Navigation between modules.
  - Error handling and validation messages.

Note: Non-functional testing such as performance and load testing is not included in this phase.

---

4. Test Approach

The testing will follow a structured approach with the following phases:

1. **Requirement Analysis**:
    - Understand the application's requirements from the provided documentation or walkthroughs.
    - Create a traceability matrix to ensure full coverage.
2. **Test Design**:
    - Develop test scenarios and detailed test cases for all functionalities.
    - Define test data for scenarios like account creation, invalid logins, and transactions.
3. **Test Execution**:
    - Perform functional testing of all modules.
    - Execute positive and negative test cases to validate behavior.
    - Record defects in a defect tracking tool.
4. **Regression Testing**:
    - Retest fixed defects and ensure they don’t impact existing functionalities.
5. **Defect Reporting**:
    - Log defects with severity and priority in Jira or any defect tracking tool.
    - Provide detailed steps to reproduce, screenshots, and logs.

---

5. Test Environment

- **Operating Systems**: Windows 10/11.
- **Browsers**:
    - Google Chrome (Version 27 and above).
    - Mozilla Firefox (Latest Version).
    - Microsoft Edge (Latest Version).
- **Database**: MySQL or the equivalent backend database.
- **Test Tools**:
    - Jira for test management and defect tracking.
    - Postman for API testing (if applicable).

---

6. Entry and Exit Criteria

Entry Criteria:

- Test environment is set up and accessible.
- All modules are deployed and functional.
- Test cases and test data are reviewed and finalized.

Exit Criteria:

- All critical and high-priority defects are resolved.
- Test execution is completed, and results are documented.
- Stakeholders approve the test summary report.

---

7. Roles and Responsibilities

- **Test Manager**:
  - Develop the test strategy and plan.
  - Manage testing schedules and resources.
- **Test Engineers**:
  - Execute test cases.
  - Log defects and perform retesting.
- **Developers**:
  - Fix defects and assist in recreating issues.

---

8. Risks and Mitigation

- **Risk**: Limited access to the test environment.
  - **Mitigation**: Plan environment setup in advance.
- **Risk**: Incomplete requirements leading to missed scenarios.
  - **Mitigation**: Conduct requirement walkthroughs with stakeholders.

---

9. Deliverables

- Test Scenarios and Test Cases.
- Test Execution Report.
- Defect Report.
- Final Test Summary Report.

---

10. Approval

This Test Strategy document must be approved by:

- Project Manager.
- QA Lead.
- Client Representative.