



Banking System

Test Plan

Section : I

Samay SR - PES1UG22CS510

Sandesh Kumatgi - PES1UG22CS522

Samrudh BA - PES1UG22CS516

1 Introduction

1.1 Purpose

The purpose of this test plan is to validate that the Banking Management System functions according to its defined functional and non-functional requirements. The objective is to ensure a secure, efficient, and user-friendly experience for managing banking transactions and services.

1.2 Background

The Banking Management System is designed to streamline financial transactions, account management, and customer communication within a banking environment. The backend is developed using Node.js and MySQL for data management, while the frontend uses React to deliver an interactive and accessible user interface.

1.3 Scope

This test plan includes unit testing, integration testing, system testing, and acceptance testing for primary modules within the Banking Management System, such as account management, transaction processing, and customer support. Functional and non-functional requirements will be validated to ensure system performance, security, and reliability.

2. Test Approach

2.1 Testing Stages

- Unit Testing: Testing individual modules, like transaction processing and customer profile management, for standalone functionality.
- Integration Testing: Verifying the interactions between modules, especially between frontend and backend components.
- System Testing: Conducting end-to-end tests to ensure the system meets all functional and non-functional requirements.
- Acceptance Testing: Ensuring the application meets business needs and is ready for deployment.

2.2 Testing Types

- GUI Testing: Ensuring the user interface behaves as expected across pages and screens.
- Functional Testing: Validating all specified functions, such as transaction approvals and account updates.
- Performance Testing: Assessing system performance, particularly under high transaction loads.
- Configuration Testing: Verifying application behavior across various environments and configurations.
- Browser Testing: Ensuring compatibility across major web browsers.

2.3 Testing Strategy

Test cases will be documented and executed based on the functional and non-functional requirements. Any issues identified will be logged, tracked, and resolved systematically.

3. Test Deliverables and References

- Test Cases
- Test Summary Report
- Software Requirements Specification (SRS)
- Project Plan

Test Case ID	Module Name	Test Case Description	Pre-condition	Test Steps	Test Data	Expected Result	Actual Result	Test Result
UT_01	Employee Login	Verify login with valid Employee credentials	Employee account exists	1. Navigate to employee login page 2. Enter valid username/password 3. Click login	Username: vasu, Password: vasu123	Employee is successfully logged in and redirected to dashboard	Successfully logged in, redirected to dashboard	PASS
UT_02	Employee Login	Verify login with invalid Employee credentials	Employee account doesn't exist	1. Navigate to employee login page 2. Enter invalid username/password 3. Click login	Username: vasu, Password: wrongpass	Employee fails to login and stays in the login page	Employee fails to login and login button turns red due to failure	PASS
UT_03	Dashboard	View all customer details	Employee logged in	1.Open Dashboard 2.Click Customer list	N/A	Display all the Customers	All the customer are displayed	PASS
UT_04	New customer Registration	Register a new customer	Employee logged in	1.Open Dashboard 2.Click register new customer	Name: John DOB: 25-2-1997 Add: PES Uni Phone:1234567890 Occupation: Student	Customer registered, Details saved	Customer added successfully	PASS
UT_05	Dashboard	View all account details	Employee logged in	1.Open Dashboard 2.Click Account list	N/A	Display all the Account	All the account are displayed	PASS
UT_06	New account Registration	Register a new account	Employee logged in, Customer already exists	1.Open Dashboard 2.Click register new account	Customer ID: 19 Account Type: Adult Account Initial Balance: 100000	Register a new account	New account registered	PASS
UT_07	Loan List	List all the loan	Employee logged in	1.Open Dashboard 2.Click Loan List	N/A	Display all the loan	All the loan are displayed	PASS
UT_08	Register new loan	Register a new loan	Employee logged in	1.Open Dashboard 2.Click register new loan	Customer ID: 19 Branch ID: 2	Register a new loan	New loan registered	PASS

					Amount: 1000 Duration: 6 Savings Account : 2			
UT_09	Withdrawal List	List all withdrawal	Employee logged in	1.Open Dashboard 2.Click Withdrawal List	N/A	Display all withdrawal	All the withdrawal are displayed	PASS
UT_10	New withdrawal	Add new withdrawal	Employee logged in, Account already exists	1.Open Dashboard 2.Click new withdrawal	Account ID: 2 Amount: 1000 Remark: Books	New withdrawal done	Added new withdrawal	PASS
UT_11	Deposit List	List all deposits	Employee logged in	1.Open Dashboard 2.Click Deposit List	N/A	Display all deposits	All the deposits are displayed	PASS
UT_12	New deposit	Add new deposit	Employee logged in, Account already exists	1.Open Dashboard 2.Click new deposit	Account ID: 2 Amount: 1000 Remark: Books	New deposit done	Added new deposit	PASS
UT_13	Customer Login	Verify login with valid Customer credentials	Customer account exists	1. Navigate to customer login page 2. Enter valid username/password 3. Click login	Username: AnjulaRox, Password: AnjulaRox123	Customer is successfully logged in and redirected to dashboard	Successfully logged in, redirected to dashboard	PASS
UT_14	Customer Login	Verify login with invalid Customer credentials	Customer account doesn't exists	1. Navigate to customer login page 2. Enter invalid username/password 3. Click login	Username: Ramu, Password: Ramu123	Customer fails to login and stays in the login page	Customer fails to login and login button turns red due to failure	PASS
UT_15	Online Banking	Transfer money	Account exists and required money is present	1. Open Dashboard 2. Click Online Banking	My Account ID: 1 To Account: 2 Amount: 1000 Remark: Test	Customer sends money to another account	Transfer of money is successfully done	PASS