

Parabank Testing Documentation

1. Introduction

This document summarizes the testing process performed on the online banking web application. It includes test objectives, strategies, test cases, test results, and defects found during execution. The goal is to ensure the application's functionality, reliability, and performance meet the specified requirements.

2. Test Objectives

- Ensure correct behavior of login, registration, and account management modules.
 - Validate all form input fields and data integrity.
 - Verify that transaction modules handle fund transfers and payments correctly.
 - Identify bugs and inconsistencies for developer review.
-

3. Scope of Testing

In Scope:

- Login & Logout
- Registration
- Forgot Login Info
- Account Opening
- Transfer Funds
- Bill Payment

- Find Transactions
- Update Contact Info
- Loan Request

Out of Scope:

- Performance / Load Testing
- Third-party API integrations
- Mobile responsiveness

4. Test Environment

Item	Details
Environment	Web Application (Parabank System)
Server	Test Server
Database	MySQL
Browser Used	Chrome 142.0.0.0, Edge 129
Operating System	Windows 11
Testing Type	Manual Functional Testing
Tools Used	Excel for test cases, Postman for API validation

5. Test Strategy

- **Functional Testing:** Checked each module against requirements.
- **Negative Testing:** Used invalid or blank inputs to test validations.

- **Regression Testing:** Retested working features after new changes.
 - **Boundary Testing:** Tested limits (e.g., deposit < \$100).
 - **Integration Testing:** Ensured data consistency across modules.
-

6. Test Case Summary

Module	Total Tests	Passed	Failed	Pass %
Login	8	8	0	100%
Register	7	4	3	57%
Open Account	3	2	1	66%
Transfer Funds	7	4	3	57%
Bill Pay	5	2	3	40%
Find Transactions	6	4	2	66%
Update Contact Info	5	4	1	80%
Loan Request	6	3	3	50%

7. Defect Summary

Bug ID	Module	Description	Severity	Status
BUG-01	Registration	SSN not validated	Medium	Open
BUG-02	Registration	Duplicate account allowed	High	Open
BUG-03	Transfer Funds	Negative balance allowed	Critical	Open
BUG-04	Bill Pay	Invalid payee not detected	High	Open
BUG-05	Loan Request	Negative amount causes crash	High	Open
BUG-06	Update Info	Phone accepts letters	Medium	Open
BUG-07	Find Transaction	Internal error on valid ID	High	Open

8. Defect Analysis

- Lack of backend validation in several forms.
 - Missing exception handling in transaction-related modules.
 - UI and server validation not synchronized.
 - Some business rules (e.g., minimum deposit, loan interest) are not enforced.
-

9. Recommendations

- Implement server-side validation for all numeric and unique fields.
 - Apply database constraints to prevent negative balances.
 - Add input sanitization for SSN, Zip, and Phone fields.
 - Conduct full regression testing after fixes.
 - Perform unit tests for calculation-based modules (Loan, Transfer).
-

10. Conclusion

Most core functionalities (login, logout, navigation) work correctly.

However, critical logic and validation flaws were found in financial transactions, posing risks of inconsistent data and incorrect account states.

The application should not move to production until high-severity defects are resolved and retested.
