

# Test Case Specification

## ParaBank Demo Application - 200 Test Cases

---

### Document Information

Field	Value
Document ID	TCS-PARABANK-001
Version	1.0
Date	August 30, 2025
Author	Senior QA Engineer
Reviewed by	Test Lead
Approved by	Test Manager
Status	Active

### Document Changes

Version	Date	Author	Changes
1.0	2025-08-30	Senior QA Engineer	Initial 200 test cases created

---

## 1. Introduction

### 1.1 Purpose

This document contains **200 detailed test cases** for ParaBank Demo Application. Each test case has step-by-step instructions, expected results, and links to requirements.

### 1.2 Test Case Format

Each test case includes:

- **Test Case ID** - unique number for each test
- **Test Case Name** - clear description of what we test
- **Priority** - how important this test is
- **Type** - functional, security, performance, etc.
- **Requirements Link** - which requirement this test checks

- **Preconditions** - what must be ready before test
- **Test Steps** - detailed actions to take
- **Expected Results** - what should happen
- **Test Data** - what information to use
- **Postconditions** - cleanup after test

## 1.3 Related Documents

- **Test Strategy:** TS-PARABANK-001
  - **Test Design Specification:** TDS-PARABANK-001
  - **Master Test Plan:** MTP-PARABANK-002
  - **Traceability Matrix:** TM-PARABANK-001
- 

## 2. Test Case Categories Overview

### 2.1 Test Cases Distribution

Category	Test Cases	Automation	Manual	Priority
Authentication	35 cases	28 auto	7 manual	Critical
Registration	25 cases	20 auto	5 manual	High
Account Management	40 cases	32 auto	8 manual	Critical
Transactions	55 cases	45 auto	10 manual	Critical
Search & Reports	20 cases	15 auto	5 manual	Medium
API Testing	25 cases	25 auto	0 manual	High
<b>TOTAL</b>	<b>200 cases</b>	<b>165 auto</b>	<b>35 manual</b>	Mixed

## 3. Authentication Test Cases (TC\_001 - TC\_035)

### TC\_001: Successful Login with Valid Credentials

Field	Value
Test Case ID	TC_001
Test Case Name	Login Successfully with Correct Username and Password
Priority	Critical
Type	Functional - Positive

Field	Value
Requirements	REQ_AUTH_001
Automation	Yes
Estimated Time	2 minutes

### Preconditions:

1. ParaBank application is accessible at <https://parabank.parasoft.com/parabank/>
2. Valid user account exists: Username "john", Password "demo"
3. Browser is open and internet connection is working

Step	Action	Expected Result
1	Open browser and go to <a href="https://parabank.parasoft.com/parabank/">https://parabank.parasoft.com/parabank/</a>	ParaBank homepage loads successfully
2	Verify login form is displayed	Username field, Password field, and "Log In" button are visible
3	Click in Username field and type "john"	Username "john" appears in the field
4	Click in Password field and type "demo"	Password shows as dots (****) for security
5	Click "Log In" button	Page processes the login request
6	Wait for page to load	Page redirects to Accounts Overview page
7	Check welcome message	"Welcome John Smith" message is displayed
8	Verify navigation menu	Account Services menu is visible with all options
9	Check page URL	URL contains "overview.htm"

### Test Data:

- Username: john
- Password: demo

### Postconditions:

- User is logged into the system
- User session is active
- User can access all account features

---

## TC\_004: Login with Wrong Username

Field	Value
Test Case ID	TC_004
Test Case Name	Login Fails with Invalid Username
Priority	High
Type	Functional - Negative
Requirements	REQ_AUTH_002
Automation	Yes
Estimated Time	2 minutes

#### Preconditions:

1. ParaBank application is accessible
2. Login form is displayed

Step	Action	Expected Result
1	Open ParaBank homepage	Login form is visible
2	Type "wronguser" in Username field	Text appears in username field
3	Type "demo" in Password field	Password shows as dots
4	Click "Log In" button	Login attempt is processed
5	Check for error message	Error message "Please enter a valid username and password." appears
6	Verify page location	User stays on login page (URL still contains index.htm)
7	Check form fields	Username and password fields keep their values
8	Verify no unauthorized access	Accounts overview page is not accessible

#### Test Data:

- Username: wronguser
- Password: demo

#### Postconditions:

- User remains logged out
- No session created
- Login form ready for new attempt

## TC\_016: SQL Injection Prevention in Login

Field	Value
Test Case ID	TC_016
Test Case Name	Prevent SQL Injection Attack in Login Form
Priority	Critical
Type	Security
Requirements	REQ_SEC_001
Automation	Yes
Estimated Time	3 minutes

### Preconditions:

1. ParaBank application is accessible
2. Application has SQL injection protection

Step	Action	Expected Result
1	Navigate to ParaBank login page	Login form displayed
2	Enter "' OR '1'='1" in Username field	Malicious SQL code entered
3	Enter "' OR '1'='1" in Password field	Malicious SQL code entered
4	Click "Log In" button	System processes input safely
5	Verify security protection	Error message shown, no unauthorized login
6	Check system integrity	Application continues working normally
7	Verify no database damage	No data corruption occurred
8	Check security logs	Attack attempt logged (if logging available)

### Test Data:

- Username: ' OR '1'='1
- Password: ' OR '1'='1

### Postconditions:

- No unauthorized access granted
- System security maintained
- Application remains functional

## 4. Registration Test Cases (TC\_036 - TC\_060)

### TC\_036: Successful User Registration

Field	Value
Test Case ID	TC_036
Test Case Name	Register New User with All Valid Information
Priority	Critical
Type	Functional - Positive
Requirements	REQ_REG_001
Automation	Yes
Estimated Time	4 minutes

#### Preconditions:

1. ParaBank application is accessible
2. Registration page is available
3. Test data is prepared

Step	Action	Expected Result
1	Go to ParaBank homepage	Homepage loads successfully
2	Click "Register" link	Registration form opens
3	Fill "Jane" in First Name field	First name entered correctly
4	Fill "Doe" in Last Name field	Last name entered correctly
5	Fill "123 Main Street" in Address field	Address entered correctly
6	Fill "New York" in City field	City entered correctly
7	Fill "NY" in State field	State entered correctly
8	Fill "12345" in Zip Code field	Zip code entered correctly
9	Fill "555-123-4567" in Phone field	Phone number entered correctly
10	Fill "123-45-6789" in SSN field	SSN entered correctly
11	Fill "janedoe2025" in Username field	Username entered correctly
12	Fill "SecurePass123!" in Password field	Password shows as dots
13	Fill "SecurePass123!" in Confirm Password	Confirmation password shows as dots
14	Click "Register" button	Registration is processed
15	Verify success message	"Your account was created successfully" message shown
16	Check redirect to login	User redirected back to login page
17	Test new account login	Can login with new username/password

### Test Data:

- First Name: Jane
- Last Name: Doe
- Address: 123 Main Street
- City: New York
- State: NY
- Zip Code: 12345
- Phone: 555-123-4567
- SSN: 123-45-6789
- Username: janedoe2025
- Password: SecurePass123!

### Postconditions:

- New user account created in system

- User can login with new credentials
  - Account appears in user database
- 

## TC\_039: Empty Required Field Validation

Field	Value
Test Case ID	TC_039
Test Case Name	Registration Fails When Required Fields Are Empty
Priority	High
Type	Functional - Negative
Requirements	REQ_REG_002
Automation	Yes
Estimated Time	3 minutes

### Preconditions:

1. Registration form is open and displayed
2. All fields are empty

Step	Action	Expected Result
1	Leave First Name field empty	Field remains empty
2	Fill all other required fields with valid data	Other fields filled correctly
3	Click "Register" button	Registration attempt processed
4	Check for validation error	Error message appears for empty First Name
5	Verify specific error text	Message says "First name is required" or similar
6	Check form submission	Registration does not complete
7	Verify user stays on form	Registration page still displayed
8	Check other field values	Other filled fields keep their data

### Test Data:

- First Name: (empty)
- Last Name: Doe
- Address: 123 Main St
- Other fields: valid data

**Postconditions:**

- No user account created
  - Registration form ready for correction
  - Error state clearly shown
- 

## 5. Account Management Test Cases (TC\_061 - TC\_100)

### TC\_061: View Accounts Overview

Field	Value
Test Case ID	TC_061
Test Case Name	Display All User Accounts in Overview Page
Priority	Critical
Type	Functional - Positive
Requirements	REQ_ACC_001
Automation	Yes
Estimated Time	2 minutes

**Preconditions:**

1. User "john" is logged into ParaBank
2. User has at least one bank account
3. Accounts Overview page is accessible

Step	Action	Expected Result
1	Login successfully as user "john"	User logged in and on overview page
2	Verify page title	Page shows "Accounts Overview"
3	Check accounts table	Table with account information is displayed
4	Verify account columns	Table shows Account#, Type, Balance, Available Amount
5	Check account numbers	All account numbers are displayed correctly
6	Verify account types	Account types (Checking, Savings) shown correctly
7	Check balance format	Balances show in currency format (\$1,234.56)
8	Verify available amounts	Available amounts match or are less than balance
9	Test account links	Clicking account number opens account details
10	Check balance totals	Total balance calculated correctly if shown

#### Test Data:

- Username: john
- Password: demo
- Expected: At least 1 account visible

#### Postconditions:

- User remains logged in
- Account information displayed correctly
- Page ready for further navigation

## TC\_076: Open New Checking Account

Field	Value
Test Case ID	TC_076
Test Case Name	Successfully Open New Checking Account
Priority	Critical
Type	Functional - Positive
Requirements	REQ_ACC_003
Automation	Yes
Estimated Time	3 minutes

## **Preconditions:**

1. User is logged in with existing account
2. Existing account has sufficient funds (\$100+)
3. Open New Account page is accessible

<b>Step</b>	<b>Action</b>	<b>Expected Result</b>
1	Navigate to "Open New Account" page	Open Account form displayed
2	Verify account type options	"Checking" and "Savings" options available
3	Select "Checking" account type	Checking account selected
4	Check minimum deposit field	Minimum deposit amount shown
5	Select funding account from dropdown	Existing account selected as funding source
6	Verify funding account balance	Balance of selected account displayed
7	Click "Open New Account" button	Account creation processed
8	Wait for confirmation page	"Account Opened!" confirmation displayed
9	Check new account number	New unique account number shown
10	Verify account in overview	New account appears in Accounts Overview
11	Check funding account balance	Money deducted from funding account
12	Verify new account balance	New account shows minimum deposit amount

## **Test Data:**

- Account Type: Checking
- Minimum Deposit: \$100.00
- Funding Account: Existing account with \$500+ balance

## **Postconditions:**

- New checking account created
- Funds transferred correctly
- Account visible in user's account list

## 6. Transaction Test Cases (TC\_101 - TC\_155)

### TC\_101: Transfer Money Between Accounts

Field	Value
Test Case ID	TC_101
Test Case Name	Successfully Transfer Money Between Own Accounts
Priority	Critical
Type	Functional - Positive
Requirements	REQ_TXN_001
Automation	Yes
Estimated Time	3 minutes

#### Preconditions:

1. User logged in with minimum 2 accounts
2. From account has balance of \$500+
3. Transfer Funds page is accessible

Step	Action	Expected Result
1	Navigate to "Transfer Funds" page	Transfer form displayed
2	Check default values	Amount field empty, account dropdowns populated
3	Enter "100.00" in Amount field	Amount entered correctly
4	Select source account from "From" dropdown	Source account selected
5	Verify source account balance	Current balance displayed
6	Select destination account from "To" dropdown	Destination account selected
7	Verify accounts are different	Cannot select same account for both
8	Click "Transfer" button	Transfer processing starts
9	Wait for confirmation page	"Transfer Complete!" message shown
10	Check transfer details	Amount, from/to accounts, date/time displayed
11	Verify source account balance	\$100 deducted from source balance
12	Verify destination balance	\$100 added to destination balance
13	Check transaction history	Transfer appears in both account histories

#### Test Data:

- Transfer Amount: \$100.00
- From Account: Account with \$500+ balance
- To Account: Different account (checking or savings)

**Postconditions:**

- Money transferred successfully
- Balances updated correctly
- Transaction recorded in history

---

**TC\_106: Transfer More Money Than Available**

Field	Value
Test Case ID	TC_106
Test Case Name	Transfer Fails When Amount Exceeds Available Balance
Priority	Critical
Type	Functional - Negative
Requirements	REQ_TXN_002
Automation	Yes
Estimated Time	2 minutes

**Preconditions:**

1. User logged in with account having \$50 balance
2. Transfer Funds page is open

Step	Action	Expected Result
1	Open Transfer Funds page	Transfer form ready
2	Check source account balance	Balance shows \$50.00
3	Enter "100.00" in Amount field	Amount greater than balance entered
4	Select source account (with \$50 balance)	Account selected
5	Select different destination account	Destination selected
6	Click "Transfer" button	Transfer attempt processed
7	Check for error message	Error appears: "Insufficient funds" or similar
8	Verify transfer did not happen	No confirmation page shown
9	Check source account balance	Balance remains \$50.00 (unchanged)
10	Check destination balance	Destination balance unchanged
11	Verify no transaction created	No new transaction in history

#### Test Data:

- Transfer Amount: \$100.00
- Account Balance: \$50.00

#### Postconditions:

- No money transferred
- Balances remain unchanged
- No transaction recorded
- User shown clear error message

## 7. API Test Cases (TC\_176 - TC\_200)

### TC\_176: Get Customer Information via API

Field	Value
Test Case ID	TC_176
Test Case Name	Retrieve Customer Data Using REST API
Priority	High
Type	API - Functional
Requirements	REQ_API_001
Automation	Yes

Field	Value
Estimated Time	2 minutes

### Preconditions:

1. ParaBank API is accessible
2. Valid customer ID exists (12212)
3. API testing tool is configured

Step	Action	Expected Result
1	Send GET request to /parabank/services/bank/customers/12212	Request sent successfully ➔
2	Check response status code	HTTP 200 OK received
3	Verify response format	JSON or XML format returned
4	Check customer ID in response	Customer ID 12212 matches request
5	Verify customer name	First name and last name present
6	Check address information	Address, city, state, zip code included
7	Verify phone number	Phone number in correct format
8	Check SSN field	SSN present and properly formatted
9	Verify response time	Response received within 2 seconds
10	Check data consistency	Data matches what's shown in UI ➔

### Test Data:

- Customer ID: 12212
- API Endpoint: /parabank/services/bank/customers/{id}
- Expected Response: Customer details in JSON/XML

### Postconditions:

- Customer data retrieved successfully
- No impact on system performance
- Data integrity maintained

## TC\_189: Transfer Money via API

Field	Value
Test Case ID	TC_189
Test Case Name	Transfer Funds Using REST API Call
Priority	Critical
Type	API - Functional
Requirements	REQ_API_003
Automation	Yes
Estimated Time	3 minutes

### Preconditions:

1. Valid customer with 2+ accounts
2. Source account has \$200+ balance
3. API authentication working

Step	Action	Expected Result
1	Prepare API request with transfer data	Request payload ready
2	Send POST to /parabank/services/bank/transfer	Request sent successfully
3	Include fromAccountId in request	Source account ID included
4	Include toAccountId in request	Destination account ID included
5	Include amount (50.00) in request	Transfer amount specified
6	Check response status	HTTP 200 OK or 201 Created
7	Verify transaction ID returned	New transaction ID in response
8	Check source account balance via API	Balance reduced by \$50.00
9	Check destination balance via API	Balance increased by \$50.00
10	Verify transaction in history API	Transfer appears in transaction list
11	Check UI consistency	Changes visible in web interface

### Test Data:

- From Account ID: 13344
- To Account ID: 13355
- Amount: \$50.00

## **Postconditions:**

- Transfer completed successfully
  - Balances updated correctly
  - Transaction recorded properly
- 

## **8. Performance Test Cases (TC\_191 - TC\_198)**

### **TC\_191: Page Load Performance**

Field	Value
Test Case ID	TC_191
Test Case Name	Homepage Loads Within Performance Requirements
Priority	Medium
Type	Performance
Requirements	REQ_PERF_001
Automation	Yes
Estimated Time	2 minutes

### **Preconditions:**

1. Stable internet connection
2. Performance testing tools ready
3. Clear browser cache

Step	Action	Expected Result
1	Clear browser cache and cookies	Cache cleared successfully
2	Start performance timer	Timer started
3	Navigate to ParaBank homepage	Homepage loading begins
4	Wait for page fully loaded	Page completely rendered
5	Stop performance timer	Load time measured
6	Check load time result	Load time is under 3 seconds
7	Verify all elements loaded	Login form, images, text all visible
8	Check for JavaScript errors	No console errors present
9	Test form interaction	Can click and type in login form
10	Measure time to interactive	Interactive within 2 seconds

#### Test Data:

- Target Load Time: < 3 seconds
- Target Interactive Time: < 2 seconds

#### Postconditions:

- Performance baseline established
- Page fully functional after load
- No performance degradation detected

## 9. Browser Compatibility Test Cases (TC\_021 - TC\_024)

### TC\_021: Chrome Browser Compatibility

Field	Value
Test Case ID	TC_021
Test Case Name	All Main Features Work in Chrome Browser
Priority	High
Type	Compatibility
Requirements	REQ_UI_001
Automation	Yes
Estimated Time	10 minutes

## **Preconditions:**

1. Google Chrome browser installed (latest version)
2. ParaBank application accessible
3. Test user account available

<b>Step</b>	<b>Action</b>	<b>Expected Result</b>
1	Open Chrome browser	Chrome opens successfully
2	Navigate to ParaBank homepage	Homepage displays correctly
3	Test login functionality	Login works as expected
4	Check accounts overview page	All accounts display properly
5	Test transfer funds feature	Money transfer completes successfully
6	Test bill pay feature	Bill payment works correctly
7	Test account opening	New account creation successful
8	Check search functionality	Transaction search works
9	Test logout function	Logout works properly
10	Verify UI elements	All buttons, forms, text display correctly
11	Check responsive design	Page adjusts to window resizing
12	Test JavaScript functions	All interactive features work

## **Test Data:**

- Browser: Google Chrome (latest version)
- Test User: john/demo
- Test Functions: All major features

## **Postconditions:**

- All features confirmed working in Chrome
- No browser-specific issues found
- Performance acceptable in Chrome

## 10. Test Case Execution Tracking

### 10.1 Test Execution Summary Template

Test ID	Test Name	Status	Execution Date	Tester	Result	Notes
TC_001	Successful Login	Not Run	-	-	-	Ready for execution
TC_002	Login UI Display	Not Run	-	-	-	Ready for execution
TC_003	Login Data Entry	Not Run	-	-	-	Ready for execution
...	...	...	...	...	...	...

### 10.2 Status Legend

- **Not Run** - Test case not executed yet
- **In Progress** - Test case currently being executed
- **Pass** - Test case passed all steps
- **Fail** - Test case failed one or more steps
- **Blocked** - Cannot run due to blocking issue
- **Skipped** - Deliberately not executed

### 10.3 Execution Rules

1. **Critical tests first** - Run P0 tests before others
2. **Dependencies matter** - Login tests before account tests
3. **Document everything** - Record all results and issues
4. **Retest after fixes** - Verify bug fixes work
5. **Update status daily** - Keep execution status current

---

## 11. Test Data Requirements

### 11.1 User Accounts Needed

User Type	Username	Password	Purpose	Accounts
Standard User	john	demo	Basic functionality testing	2 accounts (Checking, Savings)
New User	testuser1	TestPass123!	Registration testing	Will be created during test
Rich User	richuser	password	High balance testing	3 accounts with \$10,000+
Poor User	pooruser	password	Low balance testing	1 account with \$5.00

User Type	Username	Password	Purpose	Accounts
Loan User	loanuser	password	Loan testing	2 accounts + 1 loan

## 11.2 Test Transaction Data

Transaction Type	Amount	Purpose
Small Transfer	\$10.00	Basic transfer testing
Large Transfer	\$1,000.00	High amount validation
Decimal Transfer	\$123.45	Decimal handling
Minimum Transfer	\$0.01	Boundary value testing

## 11.3 Bill Pay Test Data

Payee Name	Account Number	Purpose
Electric Company	12345-67890	Standard bill payment
Water Department	98765-43210	Government utility
Credit Card Company	1111-2222-3333-4444	Credit payment

## 12. Test Case Categories Summary

### 12.1 By Test Type

Test Type	Count	% of Total	Automation
Functional Positive	80 cases	40%	70 automated
Functional Negative	60 cases	30%	55 automated
Security	25 cases	12.5%	20 automated
API	25 cases	12.5%	25 automated
Performance	10 cases	5%	8 automated

### 12.2 By Priority

Priority	Count	% of Total	Must Pass
Critical (P0)	80 cases	40%	100%
High (P1)	70 cases	35%	95%
Medium (P2)	35 cases	17.5%	90%

Priority	Count	% of Total	Must Pass
Low (P3)	15 cases	7.5%	80%

## 12.3 By Feature Area

Feature	Count	Automation %	Manual %
Authentication	35 cases	80%	20%
Registration	25 cases	80%	20%
Accounts	40 cases	80%	20%
Transactions	55 cases	82%	18%
Search/Reports	20 cases	75%	25%
API	25 cases	100%	0%

## 13. Test Case Maintenance

### 13.1 Update Rules

- **Requirement changes** - Update related test cases within 2 days
- **UI changes** - Update affected test steps and expected results
- **New features** - Add new test cases following same format
- **Bug fixes** - Review and update test cases if needed

### 13.2 Review Schedule

- **Weekly reviews** during active testing
- **Monthly reviews** during maintenance
- **Quarterly full review** of all test cases
- **Annual cleanup** - remove obsolete test cases

### 13.3 Version Control

- **Track all changes** in document history
- **Approve changes** with Test Lead
- **Backup old versions** before major updates
- **Sync with automation scripts** when changes made

## Document Approval

Role	Name	Signature	Date
Author	Senior QA Engineer	_____	_____
Test Lead	Sarah Johnson	_____	_____
Test Manager	Test Manager	_____	_____

## Appendix A: Complete Test Case List (1-200)

### Authentication (TC\_001 - TC\_035)

- TC\_001-003: Successful login variations
- TC\_004-007: Invalid credentials
- TC\_008-010: Brute force protection
- TC\_011-015: Session management
- TC\_016-020: Security injection prevention
- TC\_021-024: Browser compatibility
- TC\_025-030: Advanced security features
- TC\_031-035: Login edge cases

### Registration (TC\_036 - TC\_060)

- TC\_036-038: Successful registration
- TC\_039-048: Field validation
- TC\_049-054: Format validation
- TC\_055-060: Registration edge cases

### Account Management (TC\_061 - TC\_100)

- TC\_061-075: Account overview
- TC\_076-090: Open new accounts
- TC\_091-100: Account details

### Transactions (TC\_101 - TC\_155)

- TC\_101-125: Money transfers
- TC\_126-145: Bill payments

- TC\_146-155: Loan requests

## **Search & Reports (TC\_156 - TC\_175)**

- TC\_156-170: Transaction search
- TC\_171-175: Account statements

## **API Testing (TC\_176 - TC\_200)**

- TC\_176-185: Customer API
- TC\_186-195: Account API
- TC\_196-200: Transaction API

*Detailed steps for all 200 test cases are available in the complete specification document.*