

# Test Design Specification

## ParaBank Demo Application

---

### Document Information

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

### Document Changes

Version	Date	Author	Changes
1.0	2025-08-30	Senior Test Analyst	First version - 200 test cases design

---

## 1. What is this document?

### 1.1 Why we need this document

This document explains **WHAT we need to test** in ParaBank application. It shows which parts of the app we will check and how we will do it. We will create 200 test cases to cover the most important features.

### 1.2 What this document covers

We will test these areas:

- **Login and registration** - how users get into the app
- **Bank accounts** - checking balances, opening new accounts
- **Money transfers** - moving money between accounts
- **Bill payments** - paying bills through the app
- **Security** - making sure the app is safe

- **Performance** - checking if the app works fast
- **Different browsers** - testing in Chrome, Firefox, Safari

## 1.3 Related documents

- **Test Strategy Document:** TS-PARABANK-001
  - **Master Test Plan:** MTP-PARABANK-001
  - **ParaBank Requirements:** REQ-PARABANK-001
- 

## 2. What ParaBank should do

### 2.1 Main Features to Test

#### 2.1.1 Login System

What it should do	How important	Test priority
Users can log in with username and password	Very important	Must test
Show error when login details are wrong	Important	Must test
Stop hackers from guessing passwords	Very important	Must test
Log out users after 30 minutes	Medium	Should test
Users can log out properly	Important	Must test
Prevent harmful code in login form	Very important	Must test

#### 2.1.2 New User Registration

What it should do	How important	Test priority
New users can create accounts	Very important	Must test
Check all required information is filled	Important	Must test
Each username must be unique	Important	Must test
Password must be strong enough	Medium	Should test
Phone numbers must have correct format	Low	Nice to test
Social Security Numbers must be valid	Medium	Should test

#### 2.1.3 Bank Accounts

What it should do	How important	Test priority
Users can see all their accounts	Very important	Must test
Account balances show correct amounts	Very important	Must test
Users can open checking accounts	Very important	Must test
Users can open savings accounts	Very important	Must test
Money moves from old account to new one	Important	Must test
Each account gets unique number	Important	Must test

#### 2.1.4 Moving Money

What it should do	How important	Test priority
Users can move money between their accounts	Very important	Must test
Cannot move more money than available	Very important	Must test
Users can pay bills	Important	Must test
Users can add people to pay bills to	Important	Must test
Users can ask for loans	Medium	Should test
System saves all money movements	Important	Must test

#### 2.1.5 Reports and Search

What it should do	How important	Test priority
Users can search for transactions	Medium	Should test
Users can see account history	Important	Must test
Users can get monthly statements	Medium	Should test
Can search by date ranges	Low	Nice to test

### 2.2 Other Important Things

#### 2.2.1 Security

What it should do	Test priority
Stop hackers from entering bad code	Must test
Keep user sessions safe	Must test
Protect data when sending	Should test
Check user permissions properly	Must test

## 2.2.2 Speed

What it should do	Test priority
Pages load in less than 3 seconds	Should test
Money transfers finish in 2 seconds	Must test
App works with 100 people at same time	Should test

## 2.2.3 Easy to Use

What it should do	Test priority
Works in different browsers	Must test
Works on mobile phones	Should test
Easy for disabled people to use	Should test

## 3. What We Need to Test

### 3.1 Login Tests (35 test cases)

#### 3.1.1 Basic Login (12 tests)

1. **Login with correct username and password**
2. **Try wrong username with right password**
3. **Try right username with wrong password**
4. **Try wrong username and wrong password**
5. **Leave username empty**
6. **Leave password empty**
7. **Leave both fields empty**
8. **Try to hack with SQL code in username**
9. **Try to hack with bad code in password**
10. **Use very long username and password**
11. **Use special symbols like @#\$%**
12. **Check if username is case sensitive (John vs john)**

#### 3.1.2 Session Management (8 tests)

13. **Logout button works correctly**
14. **System logs out user after 30 minutes**
15. **Can login from two different browsers**
16. **Protect against session stealing**
17. **Cannot access account pages after logout**
18. **Back button doesn't work after logout**
19. **Login stays active in different tabs**
20. **Remember me checkbox works**

### **3.1.3 Security Tests (15 tests)**

21. **Stop people from trying many passwords**
22. **Lock account after too many wrong tries**
23. **Password shows as .....**
24. **Don't show password in website address**
25. **Error messages don't give away secrets**
26. **Protect against fake form submissions**
27. **Force users to use https not http**
28. **Check password is strong enough**
29. **Cannot skip login somehow**
30. **Limit how often someone can try to login**
31. **Cookies are secure**
32. **Check login tokens are real**
33. **Prevent clickjacking attacks**
34. **Save log when someone fails to login**
35. **Don't let browser remember password automatically**

### **3.2 Registration Tests (25 test cases)**

#### **3.2.1 Form Validation (20 tests)**

36. **Fill all required fields successfully**
37. **First name - empty, too long, special characters**
38. **Last name - empty, too long, special characters**

39. **Address - check different address formats**
40. **City - check city names work**
41. **State - check valid state codes**
42. **Zip code - check format like 12345 or 12345-6789**
43. **Phone - check formats like 555-1234**
44. **SSN - check format like 123-45-6789**
45. **Username must be unique**
46. **Password must be strong**
47. **Password confirmation must match**
48. **Fields must be long enough**
49. **Fields cannot be too long**
50. **Handle special characters**
51. **Handle numbers in name fields**
52. **Prevent HTML code injection**
53. **Remove extra spaces**
54. **Support international characters**
55. **Optional fields can be empty**

### **3.2.2 Business Rules (5 tests)**

56. **Cannot register same user twice**
57. **Cannot use existing username**
58. **New account created after registration**
59. **Welcome email sent (if feature exists)**
60. **User redirected to login after registration**

## **3.3 Account Management Tests (40 test cases)**

### **3.3.1 Account Overview (15 tests)**

61. **Show all user accounts after login**
62. **Display correct balance for each account**
63. **Format money correctly (\$1,234.56)**
64. **Show negative balances properly**

65. Handle users with no accounts
66. Sort accounts by type
67. Show account type (checking, savings)
68. Show account numbers
69. Update balances when money moves
70. Click account to see details
71. Show last transaction date
72. Display available vs total balance
73. Show multiple currencies (if supported)
74. Handle very large account numbers
75. Show account status (active, closed)

### 3.3.2 Opening New Accounts (15 tests)

76. Open checking account successfully
77. Open savings account successfully
78. Choose which account to take money from
79. Put different amounts of starting money
80. Try to open account with no money
81. Minimum deposit amount required
82. Maximum deposit limit
83. Cannot take more money than available
84. New account gets unique number
85. New account appears in account list
86. Starting money moves from old account
87. Choose from different account types
88. Cancel account opening process
89. Error handling when opening fails
90. Confirm account opening details

### 3.3.3 Account Details (10 tests)

91. View account transaction history
92. See account opening date

- 93. Show account type and status**
- 94. Display current and available balance**
- 95. Show recent activity**
- 96. Filter transactions by date**
- 97. Export account statement**
- 98. Search transactions in account**
- 99. Show account interest rate (savings)**
- 100. Display account fees and charges**

### **3.4 Money Transfer Tests (55 test cases)**

#### **3.4.1 Basic Transfers (25 tests)**

- 101. Move money between own accounts**
- 102. Transfer different dollar amounts**
- 103. Move money from checking to savings**
- 104. Move money from savings to checking**
- 105. Transfer exact account balance**
- 106. Try to transfer more than balance**
- 107. Transfer negative amount**
- 108. Transfer zero dollars**
- 109. Transfer to same account**
- 110. Use decimal amounts like \$12.34**
- 111. Transfer very small amount (\$0.01)**
- 112. Transfer large amounts**
- 113. Handle rounding for cents**
- 114. Check balance updates immediately**
- 115. Transaction appears in history**
- 116. Confirm transfer before processing**
- 117. Cancel transfer process**
- 118. Transfer with special characters in amount**
- 119. Transfer with letters instead of numbers**
- 120. Transfer with very long number**

- 121. Multiple transfers in quick succession**
- 122. Transfer while another transfer processing**
- 123. Transfer with insufficient funds fee**
- 124. Transfer between different account types**
- 125. Verify transfer completion message**

### **3.4.2 Bill Pay (20 tests)**

- 126. Add new payee successfully**

- 127. Fill all payee required information**

- 128. Try to add payee with missing info**

- 129. Edit existing payee details**

- 130. Delete payee from list**

- 131. Pay bill to existing payee**

- 132. Pay different bill amounts**

- 133. Pay bill with insufficient funds**

- 134. Schedule future bill payment**

- 135. Set up recurring bill payments**

- 136. Cancel pending bill payment**

- 137. View payment history**

- 138. Search for specific payments**

- 139. Pay multiple bills at once**

- 140. Verify payee account numbers**

- 141. Handle payee name validation**

- 142. Check payment confirmation**

- 143. Test payment processing time**

- 144. Handle payment failures**

- 145. Export payment records**

### **3.4.3 Loan Requests (10 tests)**

- 146. Request personal loan**

- 147. Request auto loan**

- 148. Request mortgage loan**

- 149. Enter different loan amounts**
- 150. Provide down payment information**
- 151. Submit loan application**
- 152. Check loan approval process**
- 153. Handle loan rejection**
- 154. View loan application status**
- 155. Cancel loan application**

### **3.5 Search and Reports (20 test cases)**

#### **3.5.1 Transaction Search (15 tests)**

- 156. Find transaction by ID number**
- 157. Search by transaction date**
- 158. Search by amount range**
- 159. Search by transaction type**
- 160. Search by description keywords**
- 161. Use multiple search criteria**
- 162. Search with no results found**
- 163. Search with invalid criteria**
- 164. Search in date range**
- 165. Search for large amounts**
- 166. Search for small amounts**
- 167. Search recent transactions**
- 168. Search old transactions**
- 169. Clear search results**
- 170. Export search results**

#### **3.5.2 Account Statements (5 tests)**

- 171. Generate monthly statement**
- 172. Create statement for date range**
- 173. Download statement as PDF**
- 174. Email statement to user**

175. Generate statement with no transactions

### 3.6 API Testing (30 test cases)

#### 3.6.1 REST API (20 tests)

176. Get customer information

177. Create new customer

178. Update customer details

179. Delete customer

180. Get account list

181. Get specific account details

182. Create new account

183. Get transaction history

184. Make transfer via API

185. Handle API authentication

186. Test API response format

187. Check API error codes

188. Test API with wrong data

189. Test API performance

190. Verify data consistency between UI and API

191. Test concurrent API calls

192. Check API rate limiting

193. Test API with missing parameters

194. Verify API security

195. Test API data validation

#### 3.6.2 SOAP Services (10 tests)

196. Test SOAP login service

197. Test SOAP account service

198. Test SOAP transaction service

199. Handle SOAP errors

200. Test SOAP data format

---

## 4. Test Design Techniques

### 4.1 Equivalence Partitioning

We group similar test data together:

#### Example - Transfer amounts:

- **Valid amounts:** \$0.01 to \$999,999.99
- **Invalid amounts:** negative numbers, zero, over limit
- **Invalid formats:** letters, special characters

### 4.2 Boundary Value Analysis

We test the edges of data ranges:

#### Example - Username length:

- **Minimum:** 1 character (boundary)
- **Just above minimum:** 2 characters
- **Maximum:** 50 characters (boundary)
- **Just over maximum:** 51 characters

### 4.3 Decision Tables

For complex business rules:

#### Example - Account opening:

Has existing account	Has money	Minimum met	Result
Yes	Yes	Yes	<input checked="" type="checkbox"/> Open account
Yes	Yes	No	<input checked="" type="checkbox"/> Error: Minimum deposit
Yes	No	No	<input checked="" type="checkbox"/> Error: No funds
No	No	No	<input checked="" type="checkbox"/> Error: Cannot open

### 4.4 State Transition

For account states:

**Account States:** New → Active → Suspended → Closed

- Test each transition

- Test invalid transitions
  - Test edge cases
- 

## 5. Risk-Based Priority

### 5.1 High Risk Areas (Must Test - 80 test cases)

- Login security
- Money transfers
- Account balances
- Data validation

### 5.2 Medium Risk Areas (Should Test - 80 test cases)

- Bill payments
- Account opening
- Search functionality
- Performance

### 5.3 Low Risk Areas (Nice to Test - 40 test cases)

- UI improvements
  - Help documentation
  - Advanced features
  - Edge cases
- 

## 6. Coverage Goals

Area	Target Coverage	Test Cases
Critical Functions	100%	80 cases
Important Functions	90%	80 cases
Nice-to-have Functions	70%	40 cases
Browser Compatibility	3 browsers	Built into each test
API Coverage	All endpoints	30 cases



---

## 7. Next Steps

After this document is approved:

1. **Create detailed test cases** - Write 200 specific test cases
  2. **Build traceability matrix** - Link tests to requirements
  3. **Set up test data** - Create users, accounts, sample data
  4. **Prepare test environment** - Configure browsers, tools
  5. **Start test execution** - Run tests and report results
- 

### Document Approval

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