

Master Test Plan v2.0

ParaBank Demo Application - 200 Test Cases

Document Information

Field	Value
Document ID	MTP-PARABANK-002
Version	2.0
Date	August 30, 2025
Author	QA Lead
Reviewed by	Test Manager
Approved by	Project Manager
Status	Active

Document Changes

Version	Date	Author	Changes
1.0	2025-08-30	QA Lead	First version - general plan
2.0	2025-08-30	QA Lead	Updated for 200 test cases strategy

1. What This Plan Covers

1.1 Why We Need This Plan

This plan explains how we will test ParaBank Demo Application with **200 carefully chosen test cases**. It shows who will do what, when we will do it, and how we will know if we are successful.

1.2 What We Will Test

We will test **5 main areas** of ParaBank:

- **Login and registration** (60 test cases)
- **Bank accounts management** (40 test cases)
- **Money transfers and bill pay** (55 test cases)
- **Search and reports** (20 test cases)

- **API and performance** (25 test cases)

1.3 Related Documents

- **Test Strategy Document:** TS-PARABANK-001
 - **Test Design Specification:** TDS-PARABANK-001
 - **ParaBank Requirements:** REQ-PARABANK-001
-

2. What We Will Test

2.1 Features We WILL Test

2.1.1 Critical Features (Must Test - 120 cases)

Login System (35 cases)

- Login with correct username/password
- Handle wrong login details
- Stop hackers from breaking in
- Logout works properly
- Session security

Money Management (45 cases)

- View account balances
- Transfer money between accounts
- Open new checking/savings accounts
- Check transaction history
- Pay bills

Registration (25 cases)

- New users can create accounts
- Check all form fields work
- Make sure data is valid
- Handle registration errors

API Testing (15 cases)

- Test backend services
- Check data is correct
- Make sure API is secure

2.1.2 Important Features (Should Test - 60 cases)

Advanced Money Features (30 cases)

-  Loan applications
-  Bill payee management
-  Scheduled payments
-  Account statements

Search and Reports (20 cases)

-  Find specific transactions
-  Generate monthly reports
-  Export account data

Performance and Security (10 cases)

-  App works fast enough
-  Can handle multiple users
-  Data is protected

2.1.3 Nice-to-Have Features (Good to Test - 20 cases)

User Experience (20 cases)

-  Works in different browsers
-  Mobile phone compatibility
-  Easy for everyone to use
-  Error messages are helpful

2.2 Features We Will NOT Test

Out of Scope:

-  Real bank connections
-  Actual money processing

- ✗ Database administration
 - ✗ Server hardware
 - ✗ Network infrastructure
 - ✗ Third-party integrations
-

3. How We Will Test

3.1 Testing Approach

We will use **smart testing strategy**:

- **80%** automated tests (160 cases) - run by computer
- **20%** manual tests (40 cases) - done by people
- **Focus on high-risk areas** first
- **Test in real browsers** (Chrome, Firefox, Safari)

3.2 Test Types

3.2.1 Functional Testing (150 cases)

- **Happy path tests** - normal user behavior
- **Error handling** - what happens when things go wrong
- **Data validation** - check forms work correctly
- **Business rules** - money rules work right

3.2.2 Security Testing (25 cases)

- **Login protection** - stop unauthorized access
- **Input validation** - prevent hacker attacks
- **Session security** - keep user data safe
- **Data protection** - encrypt sensitive information

3.2.3 Performance Testing (15 cases)

- **Speed tests** - pages load quickly
- **Load tests** - multiple users at same time
- **Stress tests** - what happens under pressure

3.2.4 Browser Testing (10 cases)

- **Chrome** - most popular browser
 - **Firefox** - second most used
 - **Safari** - for Mac users
 - **Mobile** - phones and tablets
-

4. Success Criteria

4.1 When We Pass

- 190 out of 200 tests pass** (95% pass rate)
- All critical features work** (120/120 pass)
- No high-priority bugs** remain unfixed
- App loads in under 3 seconds**
- Works in 3 main browsers**
- No security holes found**

4.2 When We Fail

- More than 10 tests fail** (less than 95% pass)
- Any critical feature broken**
- High-priority bugs not fixed**
- App too slow** (over 5 seconds)
- Security problems found**
- Doesn't work in main browsers**

4.3 When We Stop Testing

- More than 30% of tests fail** - something is very wrong
- Login completely broken** - users can't get in
- Money transfers don't work** - core feature failed
- Test environment down** for more than 4 hours
- Major requirements change** - need new plan

4.4 When We Continue Testing

- Stable app version** is available
- Critical bugs fixed** and ready to test
- Test environment working** properly

- Team members available** to do testing
 - Updated test plan approved** by management
-

5. Test Schedule

5.1 Timeline Overview

Total Time: 6 weeks

Start Date: September 1, 2025

End Date: October 12, 2025

5.2 Weekly Schedule

Week 1 (Sep 1-7): Setup and Preparation

Task	Days	Who does it	Result
Set up test environment	2 days	DevOps Engineer	Environment ready
Create test data	1 day	QA Engineer	Test users and accounts ready
Write automated tests	4 days	Senior QA Engineer	80 automated tests ready
Review test cases	1 day	Test Lead	All 200 cases approved

Week 2 (Sep 8-14): Core Function Testing

Task	Days	Who does it	Result
Test login system	2 days	QA Engineer	35 login tests done
Test account management	2 days	QA Engineer	40 account tests done
Test money transfers	3 days	Senior QA Engineer	55 transfer tests done
Fix any critical bugs	Ongoing	Developers	Bugs fixed same day

Week 3 (Sep 15-21): Additional Features

Task	Days	Who does it	Result
Test registration	1 day	QA Engineer	25 registration tests done
Test search and reports	2 days	QA Engineer	20 search tests done
Test API functions	2 days	Senior QA Engineer	30 API tests done
Browser compatibility	2 days	QA Team	3 browsers tested

Week 4 (Sep 22-28): Security and Performance

Task	Days	Who does it	Result
Security testing	3 days	Security Tester	25 security tests done
Performance testing	2 days	Performance Tester	Speed tests completed
Fix security issues	Ongoing	Developers	Security bugs fixed

Week 5 (Sep 29-Oct 5): Final Testing

Task	Days	Who does it	Result
Run all automated tests	1 day	QA Team	160 automated tests run
Manual testing of fixes	2 days	QA Engineers	Bug fixes verified
User acceptance testing	2 days	Business Analyst	Business rules checked
Test report preparation	1 day	Test Lead	Initial report ready

Week 6 (Oct 6-12): Sign-off

Task	Days	Who does it	Result
Final test run	1 day	QA Team	All tests completed
Bug verification	1 day	QA Engineers	All fixes confirmed
Final test report	1 day	Test Lead	Complete report ready
Project sign-off	1 day	Management	Project approved
Documentation handover	1 day	QA Team	All docs delivered

6. Team and Responsibilities

6.1 Core Test Team

Role	Person	Time	Main Tasks
Test Lead	Sarah Johnson	100%	Plan tests, manage team, report progress
Senior QA Engineer	Mike Chen	100%	Write automated tests, API testing, mentor others
QA Engineer #1	Lisa Wang	100%	Manual testing, registration, account features
QA Engineer #2	Tom Rodriguez	100%	Manual testing, money transfers, reports
Performance Tester	Alex Kim	25%	Speed tests, load testing

Role	Person	Time	Main Tasks
Security Tester	David Park	25%	Security testing, vulnerability checks

6.2 Supporting Team

Role	Person	Time	Main Tasks
DevOps Engineer	Emma Davis	20%	Set up test environment, fix environment issues
Business Analyst	John Smith	10%	Clarify requirements, user acceptance testing
Project Manager	Maria Garcia	10%	Project coordination, stakeholder communication

6.3 Daily Tasks

Test Lead does:

- Check team progress every day
- Report status to management
- Help solve blocking issues
- Review test results
- Plan next day activities

QA Engineers do:

- Run assigned test cases
- Report bugs they find
- Verify bug fixes
- Update test results
- Help each other with problems

Senior QA Engineer does:

- Create and maintain automated tests
- Review other people's test cases
- Help with difficult technical issues
- Test APIs and integrations
- Mentor junior team members

7. Test Environment

7.1 What We Need

7.1.1 Hardware

- **Computers:** 4 laptops with 16GB RAM
- **Internet:** Fast and stable connection
- **Backup:** Secondary internet connection
- **Storage:** Cloud storage for test results

7.1.2 Software

- **Browsers:** Chrome, Firefox, Safari (latest versions)
- **Test Tools:** Playwright for automation, Postman for API testing
- **Operating Systems:** Windows 11, Mac, Ubuntu Linux
- **Other Tools:** Git for code, Slack for communication

7.1.3 Test Data

- **Users:** 20 pre-made test accounts
- **Bank Accounts:** Different types (checking, savings)
- **Money:** Fake money for testing transfers
- **Bills:** Test companies for bill payments

7.2 Environment Rules

Access:

- Only test team can access test environment
- Each person has their own login
- Use VPN when working from home
- No production data allowed

Maintenance:

- Environment reset every Sunday night
- Backup created every day
- Monitor system health 24/7
- Fix issues within 2 hours

8. Risks and Solutions

8.1 Technical Risks

Risk	How Likely	Impact	What We Do
Browser issues	Medium	High	Test in 3 browsers, have backup plan
Environment problems	High	Medium	Daily monitoring, quick fix process
API changes	Low	High	Check with developers weekly
Performance issues	Medium	Medium	Test early and often
Security problems	Low	Critical	Hire security expert, test thoroughly

8.2 Project Risks

Risk	How Likely	Impact	What We Do
Team member sick	Medium	High	Cross-train team members
Requirements change	High	Medium	Change management process
Late delivery	Medium	High	Buffer time in schedule
Tool problems	Low	Medium	Have backup tools ready
Budget cuts	Low	High	Focus on most important tests first

8.3 Risk Monitoring

- **Check risks every week** in team meeting
 - **Update risk status** in project dashboard
 - **Escalate big risks** to project manager
 - **Keep solutions ready** for high-impact risks
-

9. Communication Plan

9.1 Daily Updates

Team Standup (9:00 AM)

- What did you do yesterday?
- What will you do today?

- Any problems blocking you?
- Quick wins to share?

Daily Status Email (5:00 PM)

- Tests completed today
- New bugs found
- Bugs fixed and verified
- Tomorrow's plan

9.2 Weekly Reports

Every Friday to Management:

- Total progress (X tests done out of 200)
- Pass/fail numbers
- Critical issues found
- Risks and concerns
- Next week priorities

9.3 Communication Channels

- **Daily work:** Slack channel #parabank-testing
 - **Urgent issues:** Phone call to Test Lead
 - **Bug reports:** Jira ticket system
 - **Formal updates:** Email to stakeholders
 - **Team meetings:** Zoom video calls
-

10. Deliverables

10.1 What We Will Create

10.1.1 Test Documentation

- **Test Strategy** (already done)
- **Test Design Specification** (already done)
- **Master Test Plan** (this document)
- **200 detailed test cases** (next week)

-  **Traceability matrix** (links tests to requirements)
-  **Test procedures** (how to run tests)

10.1.2 Test Execution Results

-  **Daily test reports** (progress updates)
-  **Bug reports** (problems found)
-  **Test evidence** (screenshots, logs)
-  **Performance results** (speed measurements)
-  **Final test summary** (overall results)

10.1.3 Automation Assets

-  **160 automated test scripts**
-  **Test data generators**
-  **CI/CD integration**
-  **Automated reports**
-  **Maintenance documentation**

10.2 Delivery Schedule

- **Week 1:** Test cases and automation scripts
- **Week 3:** Mid-project status report
- **Week 5:** Preliminary test results
- **Week 6:** Final test report and sign-off
- **After project:** Automated test suite handover

11. Success Metrics

11.1 Quality Metrics

Metric	Target	How We Measure
Test Pass Rate	95%	(Passed tests ÷ Total tests) × 100
Bug Find Rate	2-5 bugs/day	Count bugs found each day
Bug Fix Rate	Same day for critical	Time from bug report to fix
Test Coverage	90%	(Requirements tested ÷ Total requirements) × 100

11.2 Progress Metrics

Metric	Target	How We Measure
Test Completion	200/200	Count completed test cases
Schedule Performance	On time	Compare actual vs planned dates
Team Productivity	8 tests/person/day	Tests completed per team member
Automation Rate	80%	(Automated tests ÷ Total tests) × 100

11.3 Business Metrics

Metric	Target	How We Measure
User Satisfaction	4.5/5 rating	Survey after testing
Performance	< 3 sec load time	Measure page response times
Security Score	No critical issues	Count security vulnerabilities
Browser Support	3 browsers work	Test in Chrome, Firefox, Safari

12. Project Sign-off

12.1 Completion Criteria

Before we can say "testing is done":

Must Have (100% required):

- All 200 test cases executed
- 95% of tests passing (190 out of 200)
- All critical bugs fixed
- Security testing completed with no critical issues
- Performance requirements met
- Final test report approved by management

Should Have (90% required):

- All high-priority bugs fixed
- Browser compatibility verified
- API testing completed

- Automated test suite delivered
- Test documentation complete

 **Nice to Have (70% required):**

- All medium/low priority bugs documented
- User acceptance testing completed
- Knowledge transfer to support team
- Lessons learned document created

12.2 Sign-off Process

1. **Test Lead** confirms all criteria met
 2. **Test Manager** reviews final report
 3. **Project Manager** approves for release
 4. **Business Owner** accepts the application
 5. **Development Team** receives test assets
-

13. After Testing

13.1 Maintenance Plan

- **Automated tests** run every night
- **Test environment** maintained by DevOps
- **Test cases** updated when app changes
- **Team knowledge** transferred to support

13.2 Continuous Improvement

- **Retrospective meeting** to discuss what worked well
 - **Process improvements** for next project
 - **Tool evaluations** - can we do better?
 - **Team training** on new skills needed
-

Document Approval

Role	Name	Signature	Date
Test Lead	Sarah Johnson	_____	_____
Test Manager	Test Manager	_____	_____
Project Manager	Maria Garcia	_____	_____
Business Owner	Business Owner	_____	_____

This plan will be updated if requirements change or new risks are discovered. All changes must be approved by the project team.