# Software Test Report (STR)

## ParaBank QA Automation: Customer Login Feature

This document summarizes the testing activities and results for the Customer Login feature of the ParaBank application. The tests were executed as defined in the Software Test Plan (STP) v1.1.

|  |  |
| --- | --- |
| Attribute | Details |
| Project | ParaBank QA Automation |
| Feature Under Test | Customer Login |
| Test Cycle | Sprint 2 - Manual & Automated Execution |
| Build Version | Build 2025.08.15.1 |
| Report Prepared By | Ali |
| Date of Report | August 16, 2025 |

## 1. Executive Summary

This test cycle covered 15 test cases for the Customer Login feature, comprising a mix of automated API tests (via JMeter) and manual UI tests. The results show significant quality issues, including critical security vulnerabilities and major usability defects.

The overall pass rate for this cycle was 60%. Of the 15 tests executed, 6 failed. The failures are concentrated in the Security and Usability/Accessibility areas, which represent blockers for a production release.

Key Findings:

* Critical Security Vulnerabilities: The application is vulnerable to unauthorized access via whitespace-only credentials (C5), exposes server information during security probe attempts (C8, C9), and lacks basic brute-force protection (C10).
* Major Usability & Accessibility Failures: The login form is not accessible via keyboard navigation (C12) and the application is not responsive on mobile devices (C13), failing to meet fundamental modern web standards.

Recommendation:  
Based on the severity and number of failures, the recommendation is to REJECT this build. The identified critical defects must be addressed by the development team before the feature can be approved for another round of regression testing.

## 2. Test Scope

The scope of this test cycle included functional, security, usability, accessibility, and backend integration testing of the Customer Login feature.

* In Scope:
  + API-level validation of login logic using JMeter.
  + Manual validation of UI elements, navigation, and responsiveness.
  + Verification of all 15 test cases defined in the test-cases-parabank.csv document.
* Out of Scope:
  + Formal performance or load testing.
  + Testing of the "Forgot login info?" and "Register" features beyond initial navigation.

## 3. Test Results Summary

The test execution was divided into two test runs: one for automated API tests and one for manual UI tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Suite | Total Cases | Passed | Failed | Pass Rate |
| Functional Tests | 7 | 6 | 1 | 85.7% |
| Security Tests | 3 | 0 | 3 | 0.0% |
| Usability & Accessibility | 3 | 1 | 2 | 33.3% |
| Backend & Integration | 2 | 2 | 0 | 100.0% |
| Total | 15 | 9 | 6 | 60.0% |

## 4. Detailed Analysis of Failures

The following section provides a detailed analysis of each failed test case, including the observed behavior and its impact.

### Automated Test Failures (JMeter Run)

Test Case C5 / TC05: Submit login form with whitespace-only input

* Status: FAILED
* Finding: The API accepted a username and password consisting only of whitespace (%20/%20) and returned a 200 OK status with a full user data payload.
* Expected Result: The system should have rejected the input with a validation error message.
* Impact / Severity: Critical. This is a major security flaw that could allow unauthorized access to an account without valid credentials.

Test Case C8 / TC08: Attempt login with SQL injection payload

* Status: FAILED
* Finding: The application responded with a 403 Forbidden error. While it correctly blocked the request, the response body contained application source code and server information. The assertion failed because it expected a standard "Invalid username" error, not a server error page.
* Expected Result: The application should sanitize the input and return a generic, user-friendly error message without exposing any backend information.
* Impact / Severity: High. Information leakage of this nature helps attackers profile the system for more sophisticated attacks.

Test Case C9 / TC09: Attempt login with XSS script payload

* Status: FAILED
* Finding: The request failed with a java.net.URISyntaxException before a proper HTTP response could be generated. This indicates that the input was not sanitized, causing the URL itself to be malformed.
* Expected Result: The application should sanitize the input on the server side and return a generic error message.
* Impact / Severity: High. This indicates a lack of basic input validation, which is the root cause of XSS vulnerabilities.

Test Case C10 / TC13: Account lockout after 5 failed login attempts

* Status: FAILED
* Finding: After six consecutive failed login attempts with an invalid password, the system continued to respond with a standard 400 Bad Request and did not lock the account.
* Expected Result: After 5 failed attempts, the system should display an "Account locked" message and temporarily prevent further login attempts.
* Impact / Severity: Medium. The absence of a lockout mechanism makes the application vulnerable to automated brute-force password guessing attacks.

### Manual Test Failures (UI Run)

Test Case C12 / TC11: Navigate form using only keyboard

* Status: FAILED
* Finding: It was not possible to navigate between the Username field, Password field, and Login button using the Tab key.
* Expected Result: All interactive form elements should be focusable and navigable in a logical order using only the keyboard.
* Impact / Severity: Major. This is a critical accessibility failure, making the application unusable for users who rely on keyboard navigation due to motor disabilities.

Test Case C13 / TC12: Responsive design on mobile device

* Status: FAILED
* Finding: When viewed on a mobile-sized viewport, the login page layout was broken, with elements overlapping and becoming difficult to use.
* Expected Result: The page layout should adapt cleanly to smaller screen sizes, ensuring all content is readable and all controls are usable.
* Impact / Severity: Major. A non-responsive design provides a poor user experience for the significant number of users who access services via mobile devices, potentially leading to customer frustration and abandonment.

## 5. Defect Summary

The following defects should be logged in Jira to track the resolution of the issues found in this test cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| Defect ID (Example) | Title | Severity | Related Test Case |
| PB-101 | Critical Security: Whitespace-only input grants account access. | Critical | C5 |
| PB-102 | High Security: SQL injection attempt leaks server information. | High | C8 |
| PB-103 | High Security: Unsanitized XSS payload causes server exception. | High | C9 |
| PB-104 | Medium Security: Brute-force protection (account lockout) is not implemented. | Medium | C10 |
| PB-105 | Major Accessibility: Login form cannot be navigated using the keyboard. | Major | C12 |
| PB-106 | Major Usability: Login page is not responsive on mobile devices. | Major | C13 |

## 6. Final Assessment and Recommendation

The Customer Login feature, in its current state, does not meet the quality criteria for release. The presence of multiple critical and high-severity security vulnerabilities, combined with major accessibility and usability defects, poses a significant risk to both the business and its customers.

Recommendation: REJECT BUILD

Next Steps:

1. All defects listed above must be assigned to the development team for immediate remediation.
2. Once fixes are deployed to the test environment, a full regression test of all 15 test cases (both automated and manual) must be performed to verify the fixes and ensure no new issues have been introduced.
3. The feature cannot proceed to a production release until all listed defects are resolved and a subsequent test run achieves a 100% pass rate.