

Chapter 1

1.1 Overview

The Login feature allows users to access the application using valid credentials. The primary purpose of this feature is to authenticate users and provide secure access to the system. It includes input fields for username and password, a login button, and appropriate error messages for invalid credentials.

This test strategy defines the objectives, approach, and scope of testing for the Login feature to ensure it functions correctly under various conditions.

1.2 Objective

The objective of testing the Login feature is to:

- Verify that users with valid credentials can log in successfully.
- Ensure invalid or empty credentials are handled with proper error messages.
- Confirm consistent behavior across supported browsers and devices.
- Provide traceability and evidence of test results using screenshots and reports.

1.3 Test Approach

The testing approach includes:

1. **Functional Testing:** Validate all functional requirements of the Login feature.
2. **Positive and Negative Testing:**
 - Positive: Login with valid credentials.
 - Negative: Invalid username, invalid password, empty fields, and security checks.
3. **UI Verification:** Confirm all input fields, buttons, and messages display correctly.
4. **Automation Testing:** Implement automated test cases using Playwright with JUnit 5.
5. **Reporting:** Use Allure to generate test reports with screenshots for failed and passed steps.
6. **Regression Testing:** Ensure Login continues to function after system updates.

1.4 System Considerations

- Supported browsers: Chrome, Firefox, Edge (latest versions)
- Supported devices: Desktop (Windows, macOS), Tablet
- Network requirement: Stable internet connection for backend API calls
- Security: Passwords must be encrypted; login attempts rate-limited

1.5 Items Not to Be Tested

- Third-party authentication (OAuth via Google, Facebook, etc.)
- Backend database validation (covered by API/unit tests)
- Performance and load testing (handled separately)

1.6 Abbreviations

Abbreviation	Description
UI	User Interface
API	Application Programming Interface
SQL	Structured Query Language
JWT	JSON Web Token
UX	User Experience

1.7 Reference

1. System Requirement Specification (SRS) – Authentication Module - Mohamed, v1.0, [https://example.confluence.com/Software/LZ1112/SRS/Login-user-requirement-specification-\(SRS\)](https://example.confluence.com/Software/LZ1112/SRS/Login-user-requirement-specification-(SRS))
2. Login test strategy - Cuong Le, v1.2, <https://example.confluence.com/Quality-Aussrance/LZ1112/Test-strategy/Login-test-strategy>

1.8 Test case naming convention

Example: LOGIN_VALID_SPEC_CHAR_01

Where:

- **LOGIN**: Feature name
- **VALID**: Functional test for LOGIN feature
- **01**: Test case ID

Revision History

Version	Owner	Change Reason	Date
0.1	Cuong Le	Initial class for Login feature	2025-09-13
0.2	Cuong Le	Update Chapter 1 – Test Strategy	2025-09-14
0.3	Cuong Le	Add 2 test cases <code>testValidLogin</code> , <code>testValidLoginWithSpecialCharacters</code>	2025-09-15
0.4	Cuong Le	Update test case names in 1.8	2025-09-16
0.5	Cuong Le	Add screenshot strategy in test approach	2025-09-17
0.6	Cuong Le	Add System Considerations section	2025-09-18
0.7	Cuong Le	Change test procedure of <code>testInvalidLoginWithEmptyUsername</code>	2025-09-19
0.8	Cuong Le	Update test case example details	2025-09-20
0.9	Cuong Le	Revise Items Not to Be Tested section	2025-09-21
1.0	Cuong Le	Final review and consolidation	2025-09-22

Revision History

Version	Reviewer	Comment	Date
0.1	Bob	Here is Bob's comment for test plan	2025-09-13