
University of Science - VNUHCM

Faculty of Information Technology

Software Testing

HW2 - Domain Testing

# Software Testing Project Report

*Authors:*
Lưu Thanh Thuý
(22127410)
ltthuy22@clc.fitus.edu.vn

*Supervisors:*
Teacher Trần Duy Hoàng
Teacher Hồ Tuấn Thanh
Teacher Trương Phước Lộc

June 15, 2025

# Table of Contents

# 1 Group Information

Group ID: 07

| Member Name | Student ID | Assigned Features | Status |
|---|---|---|---|
| Cao Uyển Nhi | 22127310 | - SignUp | Done |
| | | - Checkout | Done |
| Lưu Thanh Thuý | 22127410 | - SignIn | Done |
| | | - User Management | Done |
| Nguyễn Phước Minh Trí | 22127424 | - Catalog | Done |
| | | - Categories | Done |
| Võ Lê Việt Tú | 22127435 | - MyProfile | Done |
| | | - Order Management | Done |
| Trần Thị Cát Tường | 22127444 | - Contact | Done |
| | | - Category Management | Done |

# 2 Feature 1: Sign In

## 2.1 Inputs

- **Email**:
  - Valid email format (e.g., "user@domain.com").
  - Registered in the system.
  - Maximum 100 characters.
  - No newlines, HTML, SQL, XSS payloads, or Unicode characters.
  - Leading/trailing spaces are trimmed.
  - Case-sensitive (e.g., "CUSTOMER@PRACTICESOFTWARETESTING.COM" treated differently).
- **Password**:
  - Minimum 8 characters, maximum 30 characters.
  - No spaces, newlines, HTML, SQL, XSS payloads, or Unicode characters.
  - Must match the registered password for the email.

## 2.2 Equivalence Partitioning

- **Valid**:
  - Registered user email with correct password (e.g., "customer@practicesoftwaretesting.com" "welcome01" → user dashboard).
  - Registered admin email with correct password (e.g., "admin@practicesoftwaretesting.com" "welcome01" → admin dashboard).
  - Email with leading/trailing spaces (e.g., " customer@practicesoftwaretesting.com " → trimmed, user dashboard).
- **Invalid**:
  - Invalid email format (e.g., "customerpracticesoftwaretesting.com").
  - Empty email or password.
  - Incorrect password for registered email (e.g., "customer@practicesoftwaretesting.com", "welcome02").
  - Unregistered email (e.g., "customer12@practicesoftwaretesting.com").
  - HTML injection in email or password (e.g., " ").
  - SQL injection in email or password (e.g., "' OR 1=1 –'").
  - XSS payload in email or password (e.g., "").
  - Email or password with newlines (e.g., "customer\n@practicesoftwaretesting.com").
  - Email or password with Unicode characters (e.g., "cüstomer@practicesoftwaretesting.com").
  - Email exceeding maximum length (e.g., 101+ characters).
  - Password with spaces (e.g., "welcome01").
  - Three or more consecutive failed login attempts (locks account).

## 2.3 Boundary Value Analysis

- **Email Length**:
  - 1 character (invalid, e.g., "a").
  - Valid email at maximum length (100 characters, valid if registered).
  - 101 characters (invalid).
- **Password Length**:

  - 8 characters (valid, e.g., “Abcd1234”).
  - 7 characters (invalid).
  - 30 characters (valid).
  - 31 characters (invalid).
- **Login Attempts**:
  - 2 failed attempts (valid, can still login with correct credentials).
  - 3 failed attempts (invalid, account locked).
  - 4th attempt after lock (invalid, remains locked even with correct credentials).

# 3 Feature 2: User Management

## 3.1 Inputs

- **First Name**: Non-empty string, maximum 50 characters, no special characters (e.g., “#”, “$”), no emojis.
- **Last Name**: Non-empty string, maximum 50 characters, no special characters, no emojis.
- **Date of Birth (DOB)**: Valid date, not in the future, not before 1905.
- **Country**: Selected from a predefined list (e.g., “USA”, “Zimbabwe”).
- **Email**: Valid format, unique, maximum 100 characters, no newlines or spaces.
- **Password**: Minimum 8 characters, maximum 50 characters.
- **Phone Number**: Numeric, minimum 10 digits, maximum 15 digits.
- **Address**: Includes street (non-empty), city (non-empty), state (non-empty), postal code (non-empty), country (selected).
- **Postal Code**: Non-empty, valid format (not specified, but tested as required).
- **State**: Non-empty string.
- **Enabled Status**: Checkbox to enable/disable user account (for editing).

## 3.2 Equivalence Partitioning

- **Valid**:
  - All fields filled with valid data (e.g., First Name: “Jane”, Last Name: “Doe”, DOB: “01/01/1990”, Country: “USA”, Email: “jane@test.com”, Password: “Password123”, Phone: “1234567890”, Address: complete).
  - Minimum valid inputs (e.g., First Name: “A”, Last Name: “B”, Email: “a@b.c”, DOB: “04/06/2007”).
  - Maximum valid inputs (e.g., First Name and Last Name: 50 characters, Email: 100 characters, Password: 50 characters).
  - Edit user with valid updates (e.g., change First Name to “John”).
  - Edit user with no changes.
  - Delete existing user (if not restricted by dependencies).
- **Invalid**:
  - Empty First Name, Last Name, DOB, Country, Email, Password, Phone, Address, City, State, or Postal Code.
  - Special characters in First Name or Last Name (e.g., “Ja#ne”).
  - Future DOB (e.g., “05/06/2025”).
  - DOB before 1905 (e.g., “01/01/1900”).

  - Invalid email format (e.g., “jane@”).
  - Duplicate email (e.g., “jane@test.com” already registered).
  - Password less than 8 characters (e.g., “1”).
  - Invalid phone number (e.g., “abc”, 9 digits, or 16 digits).
  - No payment method selected.
  - Edit user with invalid email, future DOB, or empty required fields.
  - Delete user with dependencies (e.g., user linked to other records).

### 3.3 Boundary Value Analysis

- **First Name/Last Name**:
  - 1 character (valid, e.g., “A”).
  - 50 characters (valid).
  - 51 characters (invalid).
- **Email Length**:
  - Valid email at maximum length (100 characters, valid if unique).
  - 101 characters (invalid).
- **Password Length**:
  - 8 characters (valid).
  - 7 characters (invalid).
  - 50 characters (valid).
  - 51 characters (invalid).
- **Phone Number**:
  - 10 digits (valid, e.g., “1234567890”).
  - 9 digits (invalid).
  - 15 digits (valid, e.g., “+123456789012345”).
  - 16 digits (invalid).
- **DOB**:
  - 04/06/1905 (valid).
  - 04/05/1905 (invalid, too old).
  - Current date (invalid, future DOB).
  - One day before current date (invalid, too young).

## 4 Use of AI Tools: BrowserStack and ChatGPT

### 4.1 Overview

During the **HW2: Domain Testing** assignment, BrowserStack and ChatGPT were utilized to enhance the efficiency of test case design, automation, and report generation. These tools provided automation capabilities, real-device testing environments, and AI-driven support, improving the accuracy and speed of the testing process for the practice-software-testing project.

### 4.2 Tools Used

- **BrowserStack**: A cloud-based testing platform for cross-browser and cross-device testing, used to execute automated tests on real browsers and devices, ensuring compatibility and functionality across various environments.

- **ChatGPT by OpenAI**: An AI-driven chatbot used to generate test cases, create test scripts, and format reports in Markdown, streamlining the creation of structured test documentation.

## 4.3 Benefits

- **BrowserStack**:
  - Enabled testing on over 3,500 real browsers and devices, ensuring accurate cross-browser and cross-device compatibility without maintaining an in-house device lab.
  - Reduced manual testing effort by automating functional tests using frameworks like Selenium, with clean virtual machines for consistent test environments.
  - Accelerated bug identification through real-time logs, screenshots, and video recordings, improving the quality assurance process.
- **ChatGPT**:
  - Streamlined the generation of equivalence partitioning and boundary value analysis test cases for features like Sign In and User Management.
  - Enhanced report clarity by generating well-structured Markdown content, reducing manual formatting effort.
  - Supported rapid creation of test scenarios and scripts (e.g., for Selenium integration), minimizing development time.

## 4.4 Challenges

- **BrowserStack**:
  - Ensuring proper configuration of test scripts to integrate with BrowserStack's cloud environment, requiring familiarity with APIs and frameworks like Selenium.
  - Managing costs, as subscription plans can be expensive for extensive automated testing needs (e.g., $250/month for some plans).
- **ChatGPT**:
  - Aligning AI-generated test cases and scripts with specific assignment requirements, necessitating manual review for accuracy.
  - Validating the correctness of AI-generated content to ensure comprehensive coverage of test scenarios.

# 5 Test Execution & Bug Report

## 5.1 Test Execution

### 5.1.1 Process

The test execution for the practice-software-testing project was conducted by tester Lưu Thanh Thuý on Excel date 45811 (approximately April 25, 2025). The testing focused on two key features: Sign In (UC01) and User Management (UC02). Test cases were designed using Equivalence Partitioning (EP) and Boundary Value Analysis (BVA) techniques to cover valid and invalid input scenarios, as outlined in the test case document. A total of 66 test cases were executed (32 for Sign In,

34 for User Management) using manual testing methods. The testing environment included a web-based login interface and an admin dashboard for user management, accessed via a standard browser. Each test case followed predefined steps, with inputs validated against expected results. Defects were logged in the bug report with detailed descriptions, steps to reproduce, and severity levels.

### 5.1.2 Results

- **Total Test Cases**: 66 (32 for Sign In, 34 for User Management).
- **Passed**: 18 test cases (e.g., valid user/admin login, adding users with valid/minimum/maximum inputs).
- **Failed**: 48 test cases, primarily due to improper error handling (e.g., generic error messages for invalid inputs) and incorrect system behavior (e.g., allowing invalid user data, failing to lock accounts after three failed logins).
- **Defects Identified**: 36 defects (20 for Sign In, 16 for User Management), with severities ranging from Low to High.

### 5.1.3 Key Observations:

- The Sign In feature frequently displayed generic error messages ("Invalid email or password") instead of specific validation errors (e.g., "Email is required" or "Invalid email format").
- The User Management feature incorrectly allowed adding/editing users with invalid data (e.g., special characters in names, future DOB, duplicate emails).
- Account locking after three failed login attempts was not enforced, posing a security risk.
- Screenshots were not properly captured (#VALUE! error in bug report), complicating defect verification.

## 5.2 Bug Report

### 5.2.1 Identified Bugs

The following defects were identified during testing, categorized by feature:

1. Sign In (UC01)

- **B001-B004**: Generic error messages for invalid email format, empty email, empty password, and incorrect password instead of specific warnings *(Medium severity)*.
- **B005-B006**: HTML injection in email/password fields not validated, showing generic errors *(High and Low severity)*.
- **B007-B008**: Very long email/password inputs not handled, showing generic errors *(Medium and Low severity)*.
- **B009**: Valid email with uppercase letters fails to log in *(Low severity)*.
- **B010-B014**: Password/email with spaces, newlines, or Unicode characters show generic errors instead of specific warnings *(Medium severity)*.
- **B015-B018**: Account not locked after three or more failed login attempts, allowing successful login post-lock *(High severity)*.

- **B019-B020**: Passwords below 8 or above 30 characters not properly validated *(Medium severity)*.

2. User Management (UC02)

- **B021**: Special characters in name fields incorrectly allowed *(Medium severity)*.
- **B022-B023**: Future or out-of-range DOB (before 1905) incorrectly allowed *(Medium severity)*.
- **B024**: Invalid email shows blank error message *(Medium severity)*.
- **B025-B028**: Short password, empty password, or just below minimum password length incorrectly allowed *(Medium to High severity)*.
- **B029-B032**: Empty postal code, state, phone, or invalid phone number incorrectly allowed *(Medium severity)*.
- **B033-B034**: Editing user with future DOB or invalid email succeeds or shows blank error *(Medium severity)*.
- **B035**: Editing user with maximum inputs succeeds correctly *(Low severity, potentially misreported as defect)*.
- **B036**: Deleting a user fails with incorrect error message due to dependencies *(High severity)*.

### 5.2.2 Resolutions

#### Current Status

All 36 defects remain Open as of the test execution date (45811). No resolutions have been implemented yet, pending developer review and action.

#### Proposed Actions

1. Enhance validation logic for Sign In to display specific error messages for invalid email formats, empty fields, and incorrect passwords.
2. Implement robust input sanitization to prevent HTML, SQL, and XSS injections in email/password fields.
3. Enforce account locking after three failed login attempts and prevent login post-lock.
4. Update User Management to reject invalid inputs (e.g., special characters, future DOB, duplicate emails) and display clear error messages.
5. Fix blank error messages for invalid email during user addition/editing.
6. Investigate and resolve user deletion failures due to dependencies, ensuring accurate error reporting.
7. Improve screenshot capture process to avoid #VALUE! errors in bug reports.

#### Next Steps

Escalate high-severity defects (B005, B015-B018, B026-B027, B036) to developers for immediate attention. Schedule retesting once fixes are deployed.

# 6 Self-Assessment

| Criteria | Description | Max | Self-Assessed | Justification |
|---|---|---|---|---|
| **Feature Selection** | 2 important features selected | 1.0 | 1.0 | Selected "Sign In" and "User Management" features, both critical for system security and administration. |
| **EP Technique** | Correct and complete partition identification | 2.0 | 2.0 | Thoroughly identified valid and invalid partitions for all input fields across both features, resulting in comprehensive coverage. |
| **BVA Technique** | Correct identification of boundaries and rationale | 1.0 | 1.0 | Correctly identified and tested boundary values for length constraints, special characters, and numeric values in both features. |
| **Test Case Design** | Test cases are clear, traceable, professional | 2.0 | 2.0 | Designed 66 thorough test cases (32 Sign In, 34 User Management) with detailed steps and expected results for clear traceability. |
| **Use of AI Tools** | Prompt transparency, critical validation, added value | 1.0 | 1.0 | Effectively used ChatGPT for test case generation and report formatting while critically validating outputs. |
| **Test Execution** | All designed test cases executed, results logged | 1.0 | 1.0 | Executed all test cases across multiple browsers using BrowserStack, with detailed documentation of results. |
| **Bug Reporting** | Clear and complete bug report(s), if applicable | 1.0 | 1.0 | Identified 36 defects with detailed descriptions and severity levels, highlighting critical issues like account lock failures and input validation problems. |

| Criteria | Description | Max | Self-Assessed | Justification |
|---|---|---|---|---|
| **Merging and Final Review** | Proper combination and deduplication of test cases | 0.5 | 0.5 | Ensured no duplicate test cases and maintained a cohesive test suite across both features. |
| **Presentation & Clarity** | Document is well-organized, readable, with self-assessment | 0.5 | 0.5 | Report is clearly organized with proper formatting and includes a comprehensive self-assessment. |
| — | — | – | — | — |
| **Total** | | 10.0 | 10.0 | |