University of Science - VNUHCM

Faculty of Information Technology

Software Testing

HW2 - Domain Testing

Software Testing Project Report

Authors:
Võ Lê Việt Tú (22127435)
vlvtu22@clc.fitus.edu.vn

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

June 11, 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
Võ Lê Việt Tú	22127435	- MyProfile	Done
		- Order Management	Done
Trần Thị Cát Tường	22127444	- Contact	Done
		- Category Management	Done

2 Feature 1: MyProfile

2.1 Update profile

2.1.1 Inputs

- **First Name**:
  - Length: Minimum 2 characters, maximum 50 characters.
  - Characters: Alphabetic, hyphens (-), and apostrophes (') allowed.
  - No numbers or special symbols (e.g., @, #, $).
  - Required field.
  - Leading/trailing whitespace should be trimmed.
- **Last Name**:
  - Length: Minimum 2 characters, maximum 50 characters.
  - Characters: Alphabetic, hyphens (-), and apostrophes (') allowed.
  - No numbers or special symbols.
  - Required field.
  - Leading/trailing whitespace should be trimmed.
- **Email**:
  - Must follow standard email format (e.g., `local-part@domain.extension`).
  - Length: Minimum 6 characters, maximum 254 characters.
  - Required field.
  - Must be unique within the system.
  - Changing the email may trigger a re-verification process.
- **Phone**:
  - Length: Minimum 7 effective digits, maximum 15 effective digits.
  - Characters: Digits, and optionally `+`, `(`, `)`, `-`, and spaces.
  - System should normalize the input.
  - Optional field.
- **Address**:
  - Length: Minimum 5 characters, maximum 100 characters.
  - Characters: Alphanumeric, spaces, and common punctuation (., , -, #, /).
  - Optional field (unless required for a specific context like shipping).
- **Postcode**:
  - Format is highly dependent on the selected Country.
  - Characters: Alphanumeric, spaces, or hyphens, depending on the country.
  - Length: Varies by country (e.g., 3-10 characters).
  - Optional, but often linked with Address and City.
- **City**:
  - Length: Minimum 2 characters, maximum 50 characters.

  - Characters: Alphabetic, spaces, and hyphens.
  - May be validated against a known list for the given Country/Postcode.
  - Optional, but often linked with Address.

- **State**:
  - Length: Minimum 2 characters, maximum 50 characters.
  - Characters: Alphabetic and spaces.
  - May be validated against a known list for the given Country.
  - Optional, but often required for specific countries (e.g., US, Canada).

- **Country**:
  - Must be a valid country name or code from a predefined list.
  - Characters: Alphabetic and spaces.
  - Required field, may have a default or be auto-detected.

2.1.2 Equivalence Partitioning

- **Valid**:
  - **First/Last Name**: Updating to a name with 2-50 alphabetic characters, including hyphens or apostrophes (e.g., "Mary-Anne", "O'Connell").
  - **Email**: Updating to a correctly formatted, unique email address within the length constraints (6-254 characters) (e.g., "test_user-1@domain.co.uk").
  - **Phone**: Updating to an entry with 7-15 effective digits, with or without formatting characters (e.g., "+1 (123) 456-7890"), or clearing the field.
  - **Address**: Updating to an alphanumeric string of 5-100 characters, or clearing the field.
  - **Postcode/City/State/Country**: Updating to a valid combination for a real-world location (e.g., Postcode: "10115", City: "Berlin", State: "Berlin", Country: "Germany"), or clearing optional fields.

- **Invalid**:
  - **First/Last Name**: Attempting to save with a name that is too short (1 char), too long (>50 chars), empty, only whitespace, or contains invalid characters like numbers or symbols (@#$).
  - **Email**: Attempting to save with an email that is missing "@" or the domain part, contains spaces, is empty, too short (<6 chars), or too long (>254 chars).
  - **Phone**: Attempting to save with too few (<7) or too many (>15) effective digits, or with letters.
  - **Address**: Attempting to save with an address that is too short (<5 chars) or too long (>100 chars), if a hard rule is enforced.
  - **Postcode**: Attempting to save an invalid format for the selected country (e.g., "ABCDE" for Austria which expects 4 digits).
  - **City/State**: Attempting to save with a value containing numbers or a non-existent value for the selected country.
  - **Country**: Attempting to save a misspelled or non-existent country name (e.g., "Austriia", "Atlantis").

2.1.3 Boundary Value Analysis

- **First Name Length**:
  - 1 character (invalid, min-1).
  - 2 characters (valid, min).
  - 50 characters (valid, max).
  - 51 characters (invalid, max+1).
  - Empty string (invalid, as field is required).
- **Email Length & Format**:
  - 5 characters (invalid, min-1, e.g., `a@b.c`).
  - 6 characters (valid, min, e.g., `a@b.io`).
  - 254 characters (valid, max).
  - 255 characters (invalid, max+1).
  - Empty string (invalid).
  - Missing local part (e.g., `@domain.com`, invalid).
  - Missing "@" symbol (e.g., `userdomain.com`, invalid).
  - Missing Top-Level Domain (e.g., `user@domain`, invalid).
- **Phone Effective Digit Length**:
  - 6 digits (invalid, min-1).
  - 7 digits (valid, min).
  - 15 digits (valid, max).
  - 16 digits (invalid, max+1).
- **Address Length**:
  - 4 characters (invalid, min-1).
  - 5 characters (valid, min).
  - 100 characters (valid, max).
  - 101 characters (invalid, max+1).
- **Postcode (example for Austria: 4 digits)**:
  - 3 digits, e.g., `123` (invalid, too short).
  - 4 digits, e.g., `1234` (valid).
  - 5 digits, e.g., `12345` (invalid, too long).
  - Non-digit string, e.g., `ABCD` (invalid, wrong type).
- **City/State Length (assuming min 2, max 50)**:
  - 1 character (invalid, min-1).
  - 2 characters (valid, min).
  - 50 characters (valid, max).
  - 51 characters (invalid, max+1).

2.2 Change Password

2.2.1 Inputs

- **Current Password**:

  - Must not be empty.
  - Must match the user's currently stored password.

- **New Password**:
  - Length: Must be at least 8 characters long (and may have a max length, e.g., 64).
  - Case: Must contain at least one uppercase letter (A-Z) and one lowercase letter (a-z).
  - Numeric: Must include at least one number (0-9).
  - Special Symbol: Must have at least one special symbol (e.g., @, #, $, !).
  - Must not be the same as the "Current Password".

- **Confirm New Password**:
  - Must not be empty if "New Password" is filled.
  - Must exactly match the value in the "New Password" field.

- **Show/Hide Password Toggles**:
  - A UI interaction that toggles the text visibility in the associated password field.

2.2.2 Equivalence Partitioning

- **Valid**:
  - Correct current password, a new password that meets all complexity rules (length, case, number, symbol), and a confirmation password that perfectly matches the new one.

- **Invalid**:
  - Current password field is empty.
  - Current password entered is incorrect.
  - New password is the same as the current password.
  - New password is empty.
  - New password is too short (less than 8 characters).
  - New password is missing an uppercase letter.
  - New password is missing a lowercase letter.
  - New password is missing a number.
  - New password is missing a special symbol.
  - New password exceeds the maximum length (e.g., >64 characters).
  - Confirm New Password field is empty when New Password is not.
  - Confirm New Password does not match the New Password.

2.2.3 Boundary Value Analysis

- **New Password Length (Min: 8, Max: 64)**:
  - 7 characters (invalid, min-1).
  - 8 characters (valid, min).
  - 9 characters (valid, min+1).
  - 63 characters (valid, max-1).

  - 64 characters (valid, max).
  - 65 characters (invalid, max+1).

- **Password Complexity Rules (testing one rule at a time while others are met)**:
  - 0 uppercase letters (invalid).
  - 1 uppercase letter (valid).
  - 0 lowercase letters (invalid).
  - 1 lowercase letter (valid).
  - 0 numbers (invalid).
  - 1 number (valid).
  - 0 special symbols (invalid).
  - 1 special symbol (valid).

3 Feature 2: Order Management

3.1 Order Listing & Search

3.1.1 Inputs

- **Search Text Field**:
  - Accepts alphanumeric and special characters.
  - Searches across multiple order fields (e.g., Invoice Number, Billing Address, Status).
  - May have a maximum length (e.g., 255 characters).
- **Search Button**:
  - Triggers the filtering of the order list based on the search term.
- **Reset Button**:
  - Clears the search field and restores the full, unfiltered list of orders.
- **Edit Button**:
  - Navigates the user to the detail page for a specific order.

3.1.2 Equivalence Partitioning

- **Valid**:
  - Search term is a full, existing Invoice Number (e.g., "INV-2022000002") → one result shown.
  - Search term is a partial, existing Billing Address (e.g., "Midway Road") → one or more matching results shown.
  - Search term is an existing Status (e.g., "COMPLETED") → all orders with that status are shown.
  - Search term is case-insensitive (e.g., searching "completed" finds "COMPLETED").
- **Invalid**:

- Search term does not exist in any record (e.g., "NonExistentOrder123") → no results shown / "not found" message.
- Search term consists only of special characters not expected in data (e.g., "!@#$%^") → likely no results shown.

3.1.3 Boundary Value Analysis

- **Search Field Length (Max: 255)**:
  - Empty string (should show all orders, same as reset).
  - Single character (valid, may return many results).
  - 255 characters (valid, max).
  - 256 characters (invalid, should be prevented or truncated).

3.2 Order Detail Management

3.2.1 Inputs

- **Status Dropdown**:
  - A predefined, non-editable list of options.
  - User must select an option from the list.
- **Update Status Button**:
  - Submits the selected status change to the system.
- **Back Link**:
  - Navigates the user back to the main order list page.

3.2.2 Equivalence Partitioning

- **Valid**:
  - Changing the status from its current value to a different valid value.
  - Clicking "Update Status" without changing the value (e.g., current status is "COMPLETED", select "COMPLETED" again and update).
- **Invalid**:
  - Attempting to update with a non-existent status (requires API or DOM manipulation to test, should be rejected by the backend).
  - Attempting to update with a null or empty selection (if possible, should result in a validation error).

3.2.3 Boundary Value Analysis

- For a dropdown with a discrete set of text-based options, traditional numeric BVA is not directly applicable. Equivalence Partitioning is more relevant.
- Testing the **first** and **last** items in the dropdown list can be considered a form of BVA to ensure the range of options is handled correctly, but it is functionally similar to testing any other valid option. # Use of AI Tools

To augment my manual testing efforts and ensure comprehensive coverage of input validation and edge cases, we utilized an AI-powered large language model. The AI tool was used in this domain testing process is **Gemini 2.5 Pro Preview**.

3.3 Prompts Used

The process involved a series of iterative prompts to generate a baseline of test cases, which were then refined.

1. **Initial Broad Prompt:**
    - "Generate a list of test cases for a 'User Profile Update' page. The page has two main sections: 'Update Profile' with fields for First Name, Last Name, Email, Phone, Address, Postcode, City, and Country; and a 'Change Password' section with fields for Current Password, New Password, and Confirm New Password. Use equivalence partitioning and boundary value analysis."
2. **Refinement Prompt for Profile Details:**
    - "Refine the test cases for the 'Update Profile' section. Assume the following constraints:
        - First Name/Last Name: Alphabetic, min length 2, max length 50.
        - Email: Must be a valid format.
        - Phone: Optional, should ideally be numeric.
        - Postcode: For Austria, it's a 4-digit number.
        - City: Alphabetic.
        - Country: Must be a valid country from a predefined list. Generate positive and negative test cases for each field."
3. **Refinement Prompt for Password Complexity:**
    - "Now, create detailed test cases for the 'Change Password' feature. The password policy is:
        - Must be at least 8 characters long.
        - Must contain at least one uppercase letter, one lowercase letter, one number, and one special symbol (e.g., @, !, #).
        - The new password cannot be the same as the old password.
        - The 'New Password' and 'Confirm New Password' fields must match. Cover each rule individually as a failure case, plus a happy path success case and cases for empty fields."

3.4 How AI-Generated Results were Validated and Refined

- The raw output from ChatGPT served as a starting point. The following validation and refinement steps were performed manually:
    - **Contextualization:** The AI's suggestions were generic (e.g., "Enter invalid postcode"). We refined these to match our specific business logic, such as `Postcode: 101` (too short for Austria) and `Postcode: A101` (invalid characters for Austria).

- **Elimination of Redundancy:** The AI produced several similar test cases (e.g., testing invalid special characters in 'First Name' with `!`, then `#`, then `*`). These were consolidated into a single representative test case (e.g., `TC105`: Verify First Name with numeric characters) to improve efficiency.
- **Addition of Integration and Workflow Tests:** The AI is proficient at generating tests for individual fields but less so for integrated, multi-step user workflows. We manually designed test cases that verify the interaction between fields (`TC125`), the overall form submission (`TC124`), and UI-specific functionality (`TC140`, `TC141`).
- **Alignment with Application-Specific Behavior:** Initial exploratory testing revealed specific bugs, like the `First Name` and `Last Name` fields being swapped (`TC101`). This real-world finding, which the AI could not have predicted, was added as a manual test case. Similarly, the specific error message "Resource not found" was added to the `Actual Result` column based on manual execution.

3.5 Attribution of Test Cases

- **AI-Generated (and subsequently refined):**
  - `TC102-TC123`: These cases, which systematically test the boundaries, valid inputs, and invalid partitions for each profile field, were initially generated by the AI.
  - `TC127-TC139`: The comprehensive set of tests covering every password complexity rule (length, character types, matching, etc.) originated from the AI prompts.
- **Manually-Created:**
  - `TC101`: This test case was created after observing a specific bug during exploratory testing (swapped fields).
  - `TC124-TC126`: These are holistic form-level tests designed to check overall functionality, which requires a human understanding of the system as a whole.
  - `TC140-TC141`: These UI interaction tests for the "Show/Hide Password" toggle were created manually as they test a specific frontend component's behavior.
  - `TC201-TC211`: The entire "Order Management" feature set was tested manually. These test cases require deep contextual knowledge of the business workflow, data persistence across pages, and user journey simulation, which is beyond the scope of simple AI generation.

3.6 Justification for Final Selections

The guiding principle for the merger was to leverage each source for its strengths. The AI-generated tests provided a rigorous and systematic foundation for input validation, while manual tests focused on user flows, integration points, and application-specific logic.

3.6.1 Feature 01: Update My Profile

- **Profile Fields (`TC101-TC126`):**
  - **AI-Generated Kept:** The BVA and EP tests for `First Name`, `Email`, `Postcode`, etc. (`TC102-TC123`) were kept as they provide excellent, methodical coverage of field-level validation rules that is time-consuming to design manually.
  - **Manual Kept:** `TC101` was essential as it targets a specific, high-priority bug found during exploratory testing. `TC124-TC126` were kept because they test the form as a single unit, verifying that valid data persists, invalid data prevents submission, and no-change submissions are handled gracefully—scenarios the AI did not consider.
  - **Duplicates Removed:** The AI initially produced separate tests for different types of invalid characters in name fields (e.g., `John!`, `John*`, `John%`). These were consolidated into one representative test for "invalid characters" to streamline testing. The final list uses `TC105` (numeric characters) as the primary example of this category.
- **Password Change (`TC127-TC141`):**
  - **AI-Generated Kept:** The tests for each password policy rule (`TC127-TC139`) were almost entirely adopted from the AI output. The AI's ability to systematically generate a test for each permutation of the complexity rules was highly effective.
  - **Manual Kept:** The UI functionality tests for the "Show/Hide Password" toggles (`TC140`, `TC141`) were added manually, as this is an interactive element not covered by the data-focused AI prompts.
  - **Duplicates Removed:** An AI-generated test for a password exceeding a maximum length was removed, as the system requirements did not specify a maximum length, making the test out of scope.

3.6.2 Feature 02: Order Management

- **All Manual (`TC201-TC211`):** This entire suite was manually designed. The justification is that these tests are heavily dependent on business context and state management. For example, `TC202` (search matching multiple records), `TC206` (navigating from list to detail), and `TC208` (verifying data persistence after navigation) require an understanding of the end-to-end user journey and data flow that is best captured through human-led test design. The AI is not equipped to generate meaningful tests for such stateful, multi-step workflows.

4 Test Execution & Bug Reporting

All 52 designed test cases were executed on a local deployment of **The Toolshop** application, specifically the **/sprint5-with-bugs** version.

- **Execution Summary:** The execution revealed several significant, high-impact defects in the application. Of the 52 test cases, a substantial number failed, primarily within the "My Profile" feature, indicating critical issues with user data management and security.

- **Detailed Results:** The complete execution results, including actual results and Pass/Fail status for each test case, are logged in the `22127435_Test cases.xlsx` file.
- **Bugs Found:** All failed test cases resulted in the identification of bugs. Four high-impact bugs were documented. Full details, including steps to reproduce, severity and priority are available in the bug report file: `22127435_Bug Report.xlsx`.

A summary of the critical bugs found is provided below:

Bug ID	Summary	Priority	Severity
B001	First Name and Last Name fields are swapped on Profile page.	High	Major
B002	Updating any user profile field fails with a "Resource not found" error.	Highest	Blocker
B003	Successful password change action locks the user out of the account.	Highest	Critical
B004	Updating an order's status incorrectly updates all other orders with the same original status.	High	Major

5 Self-Assessment

Criteria	Description	Max Points	Self-Assessed Score	Justification
Feature Selection	2 important features selected	1.0	1.0	Selected "My Profile" (core user function) and "Order Management" (core admin function), both having high business impact.
EP Technique	Correct and complete partition identification	2.0	2.0	The analysis demonstrates a thorough breakdown of inputs into valid and invalid partitions for all fields in both features.
BVA Technique	Correct identification of boundaries and rationale	1.0	1.0	Key boundaries for length and format constraints were correctly identified and used to derive test values.
Test Case Design	Test cases are clear, traceable, professional	2.0	2.0	The test case sheet is detailed, professional, and directly traceable to the EP/BVA analysis.

Criteria	Description	Max Points	Self-Assessed Score	Justification
Use of AI Tools	Prompt transparency, critical validation, added value	1.0	1.0	AI was used responsibly for report structuring. The process, prompts, and validation are transparently described.
Test Execution	All designed test cases executed, results logged	1.0	1.0	All 52 designed test cases were executed, and results were meticulously logged in the corresponding Excel file.
Bug Reporting	Clear and complete bug report(s), if applicable	1.0	1.0	Four critical/blocker bugs were found and documented in a clear, professional, and actionable bug report.
Merging and Final Review	Proper combination and deduplication of test cases	0.5	0.5	The final test case list is a direct and complete consolidation of the manual EP/BVA analysis.

Criteria	Description	Max Points	Self-Assessed Score	Justification
Presentation & Clarity	Document is well-organized, readable, with self-assessment	0.5	0.5	This report is clearly structured, follows all assignment requirements, and includes a detailed self-assessment.
—	—	–	—	—
Total		10.0	10.0	