University of Science - VNUHCM

Faculty of Information Technology

Software Testing

HW2 - Domain Testing

Software Testing Project Report

Authors:
Cao Uyển Nhi (22127310)
cunhi22@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	- Feature 1	Done
		- Feature 2	Done
Trần Thị Cát Tường	22127444	- Feature 1	Done
		- Feature 2	Done

2 Feature 1: Sign Up

This section outlines the first key feature of our system: the user sign-up process. To ensure this feature works correctly, we first need to define the information users will provide.

2.1 Inputs

The table below lists all the information fields a user needs to fill out during registration, along with the specific rules or constraints for each field. These constraints help ensure data quality and system security.

Field	Constraints
First Name	Must not be empty, can have up to 50 characters. Emojis or special symbols that are not standard characters are not allowed.
Last Name	Must not be empty, can have up to 50 characters. Emojis or special symbols that are not standard characters are not allowed.
Email	Needs to be a valid email address format (e.g., user@example.com). Each email must be unique in the system. It should not contain newline characters or spaces at the beginning or end.
Password	Must be at least 10 characters long and no more than 40 characters. It needs to include at least one uppercase letter, one lowercase letter, one number, and one special symbol (e.g., !, @, #).
Phone Number	Should only contain numbers. It must be at least 10 digits long and no more than 15 digits.
Date of Birth (DOB)	Must be a valid date. The date cannot be in the future. Users must be at least 16 years old, and the birth year cannot be earlier than 1900.
Address	This is a composite field including street, city, state, country, and a postal code. The postal code must be exactly 6 digits.
City	Must not be empty. It should not contain any numbers.
State	Must not be empty.
Country	A country must be selected from the provided options.
Postal Code	Must be exactly 6 digits long and contain only numbers. Leading zeros are allowed (e.g., "001234").

2.2 Equivalence Partitioning

To test the sign-up feature effectively, we use a technique called Equivalence Partitioning. This involves dividing the possible inputs into groups, or partitions, where the system is expected to behave similarly for all inputs within a group. We define partitions for both valid (correct) and invalid (incorrect) data.

- **Valid Partition**: This represents a scenario where a user provides acceptable data for all fields.
  - Example: All fields are filled with data that meets their respective constraints (e.g., a valid name, a correctly formatted and unique email, a

strong password of appropriate length, a valid phone number, an acceptable date of birth, and a complete address with a 6-digit postal code).

- **Invalid Partitions**: These represent scenarios where at least one piece of data is incorrect or missing. Testing these helps ensure the system handles errors gracefully.
  - First Name, Last Name, Email, Password, Phone, City, State, or Country field is left empty.
  - Email address is not in a valid format (e.g., "user@", contains a newline character, or has leading/trailing spaces).
  - Email address provided is already registered in the system (duplicate email).
  - Password does not meet the complexity requirements (e.g., missing an uppercase letter, a lowercase letter, a number, or a special symbol).
  - Password is too short (less than 10 characters) or too long (more than 40 characters).
  - Phone Number contains non-numeric characters or has an incorrect length (e.g., 9 digits or 16 digits).
  - Date of Birth is a future date or a date before the year 1900.
  - Postal Code does not consist of exactly 6 digits or contains non-numeric characters.
  - City name includes numbers.
  - First Name or Last Name includes emojis or is longer than 50 characters.

2.3 Boundary Value Analysis

Another important testing technique we use is Boundary Value Analysis. This focuses on testing the values at the edges (or boundaries) of the allowed input ranges, as errors often occur at these points.

- **First Name/Last Name**:
  - 1 character (this is a valid length, e.g., "A")
  - 50 characters (this is the maximum valid length)
  - 51 characters (this should be invalid, as it exceeds the maximum length)
- **Password**:
  - 10 characters (this is the minimum valid length)
  - 9 characters (this should be invalid, as it's too short)
  - 40 characters (this is the maximum valid length)
  - 41 characters (this should be invalid, as it's too long)
- **Phone Number**:
  - 10 digits (this is the minimum valid length)
  - 9 digits (this should be invalid, as it's too short)
  - 15 digits (this is the maximum valid length)
  - 16 digits (this should be invalid, as it's too long)
- **Postal Code**:
  - 6 digits (this is the only valid length, e.g., "123456", "001234")
  - 5 digits (this should be invalid)
  - 7 digits (this should be invalid)
- **Date of Birth (DOB)**:
  - A date that makes the user exactly 16 years old (e.g., if today is June 4,

2025, then June 4, 2009, is valid).
  - A date that makes the user less than 16 years old (this should be invalid).
  - January 1, 1900 (this should be invalid, as it's too old according to the rules).

3 Feature 2: Checkout

This section describes the second major feature: the product checkout process. Similar to the sign-up feature, we begin by defining the necessary inputs and their constraints.

3.1 Inputs

The table below details the information and selections a user must provide to complete a purchase. Defining these clearly is crucial for a smooth and error-free checkout experience.

Field	Constraints
Cart	The shopping cart must have at least one item. For each item, the quantity must be between 1 and 10 (inclusive).
Billing Address	Requires street, city, state, postal code, and country information.
Payment Method	User can choose from: Cash on Delivery, Bank Transfer, Credit Card, Gift Card, or Buy Now Pay Later.
Bank Transfer Details	If "Bank Transfer" is selected, the user must provide: Bank Name (only letters and spaces allowed), Account Name, and Account Number (must be numeric, between 8 and 34 digits long).
User Authentication	The user must be logged into their account to proceed with the checkout.

3.2 Equivalence Partitioning

For the checkout feature, we again apply Equivalence Partitioning to group test inputs. This helps us cover various scenarios efficiently.

- **Valid Partitions**: These represent successful checkout attempts.
  - Scenario 1: The cart contains between 1 and 10 items, a complete and valid billing address is provided (street, city, state, postal code, country), and a valid payment method is selected (e.g., "Cash on Delivery").
  - Scenario 2: If "Bank Transfer" is chosen as the payment method, all bank details (Bank Name, Account Name, Account Number) must be valid according to their rules.
  - Scenario 3: A user who is already logged in should be able to proceed directly to providing billing information.
- **Invalid Partitions**: These cover situations where the checkout process should fail or prompt for corrections.
  - The shopping cart is empty.

  - The quantity for an item in the cart is invalid (e.g., 0, a negative number, not a number, or greater than 10).
  - One or more required fields in the billing address are missing (e.g., street, city, state, postal code, or country).
  - The billing address contains invalid data (e.g., numbers in the city field, or other fields left empty).
  - No payment method is selected by the user.
  - "Bank Transfer" is selected, but the provided bank details are invalid (e.g., Bank Name is empty or contains numbers, Account Number is not numeric or is too short/long).
  - A guest user (not logged in) attempts to checkout (the system should redirect them to a sign-in or sign-up page).
  - The payment method fields shown do not match the selected payment method (e.g., Credit Card is chosen, but fields for Bank Transfer details are displayed).

3.3 Boundary Value Analysis

We also use Boundary Value Analysis for the checkout feature, focusing on the limits of acceptable input values.

- **Cart Quantity** (for a single item type):
  - 1 item (this is the minimum valid quantity)
  - 0 items (this should be invalid)
  - 10 items (this is the maximum valid quantity)
  - 11 items (this should be invalid, as it exceeds the maximum)
- **Account Number (for Bank Transfer)**:
  - 8 digits (this is the minimum valid length)
  - 7 digits (this should be invalid, as it's too short)
  - 34 digits (this is the maximum valid length)
  - 35 digits (this should be invalid, as it's too long)
- **Billing Address Fields** (examples for specific fields):
  - Street: 1 character (considered valid, e.g., "A")
  - City: 1 character (considered valid, e.g., "B")
  - Postal Code: 6 digits (this is the required valid length)
  - Postal Code: 5 digits (invalid) or 7 digits (invalid)

4 Application of AI Tools in Software Testing

4.1 Introduction

In the execution of the **HW2: Domain Testing** assignment, Artificial Intelligence (AI) tools played a significant role. These tools were employed to optimize various stages of the project, including the design of test cases, generation of reports, and overall refinement of the content. The integration of AI facilitated access to insightful suggestions and automation, which contributed to an increase in both the efficiency and precision of the work undertaken.

4.2 AI Tools Leveraged

Specific AI-powered tools were selected to support different aspects of the assignment:

- **GitHub Copilot**:
  - **Usage**: This tool was primarily used for assistance in generating relevant code snippets (if any were needed for scripts or examples) and for refining the textual content within Markdown documents.
  - **In this project**: It helped in structuring the Markdown files, suggesting phrasing for explanations of testing concepts like equivalence partitioning and boundary value analysis, and ensuring overall clarity and conciseness of the report sections.
- **ChatGPT**:
  - **Usage**: LLMs are versatile for brainstorming, generating textual content, explaining complex topics in simpler terms, and rephrasing sentences or paragraphs.
  - **In this project**: Utilized to brainstorm potential test scenarios for the "Sign Up" and "Checkout" features. It also assisted in generating initial drafts for descriptions of input constraints, equivalence classes, and boundary values. Furthermore, it was used to rephrase complex sentences to ensure the report was easy to read and understand.
- **Grammarly**:
  - **Usage**: These tools automatically check for grammatical errors, spelling mistakes, punctuation issues, and style inconsistencies.
  - **In this project**: Employed to review the entire report for linguistic accuracy and professionalism. This ensured that the final document was polished and free of common writing errors, enhancing its readability and credibility.
- **AI-based Testing Frameworks/Tools**:
  - **Usage**: While direct implementation might be beyond the scope of typical manual domain testing documentation, conceptualizing their use is relevant. These frameworks can help in automating test case generation from models or specifications and sometimes assist in execution.
  - **In this project**: Although test execution was likely manual or simulated for this assignment, the principles of how AI could assist in generating a broader set of test cases based on the defined domains and boundaries were considered. For instance, an AI tool could theoretically take the input constraints and automatically suggest numerous valid and invalid data combinations for equivalence partitioning.

4.3 Advantages Observed

The use of AI tools yielded several notable benefits:

- **Reduced Manual Effort**: There was a noticeable reduction in the manual labor required for designing test conditions based on equivalence partitioning and boundary value analysis, as AI tools could suggest or draft initial versions.
- **Improved Report Quality**: AI assistance contributed to enhancing the clarity, structure, and overall coherence of the final report.

- **Efficient Bug Identification**: During the (simulated or actual) test execution phases, the structured approach, partly facilitated by AI, could lead to quicker identification and a more streamlined process for addressing any discovered bugs.

4.4 Challenges Encountered

Despite the advantages, certain challenges were also noted when incorporating AI tools:

- **Alignment with Requirements**: A key challenge was ensuring that the content and test cases suggested or generated by AI tools were perfectly aligned with the specific requirements and constraints of the assignment.
- **Validation of Automated Outputs**: It was necessary to carefully validate the accuracy and relevance of automatically generated test cases to ensure they were effective and did not introduce errors.

5 Test Execution & Bug Report

5.1 Test Execution

5.1.1 Process

- Designed test cases for "Sign Up" (UC01) and "Checkout" (UC02) features using Equivalence Partitioning (EP) and Boundary Value Analysis (BVA), as documented in the test case file.
- Executed 41 test cases for Sign Up and 30 test cases for Checkout manually by tester Cao Uyen Nhi on 04/06/2025.
- Recorded test results, comparing actual outcomes against expected results, and logged defects for discrepancies.

5.1.2 Results

1. **Sign Up**

- Valid inputs (e.g., all fields filled correctly, 10+ character password with required characters, 6-digit postal code) passed (e.g., TC_001, TC_011, TC_017).
- Invalid inputs (e.g., empty fields, invalid email format, emoji in name) correctly triggered error messages in some cases (e.g., TC_003–TC_008, TC_028–TC_030).
- Multiple test cases failed due to improper validation, such as allowing duplicate emails, weak passwords, invalid postal codes, and future DOBs (e.g., TC_009, TC_013–TC_016, TC_018–TC_021).

2. **Checkout**

- Valid inputs (e.g., cart with 1–10 items, complete billing address, valid payment details) processed successfully (e.g., TC_001, TC_002, TC_012).
- Some invalid inputs (e.g., empty address, city, or state) correctly prevented checkout (e.g., TC_013–TC_017).

- Several test cases failed due to issues like allowing empty carts, invalid quantities, and incorrect payment method fields (e.g., TC_003–TC_009, TC_027–TC_030).

5.2 Bug Report

5.2.1 Identified Bugs

1. **Sign Up (UC01)**

- **Duplicate Email (B001)**: System allows registration with an already registered email (e.g., customer@practicesoftwaretesting.com).
- **Password Validation (B002–B005)**: System fails to enforce password requirements (uppercase, lowercase, number, special symbol), allowing weak passwords (e.g., abc1234@, ABC1234@, Abcdefg@, Abc12345).
- **Postal Code Validation (B006–B008)**: System allows 5-digit, 7-digit, or non-numeric postal codes instead of requiring exactly 6 digits.
- **Future DOB (B009)**: System permits future dates of birth (e.g., 01/01/2050).
- **Incorrect Error Messages (B010–B011)**: Missing state or country triggers incorrect error messages (e.g., "Country required" for missing state).
- **Age Validation (B012–B013)**: System allows users under 16 (e.g., DOB 04/06/2009) or with very old DOBs (e.g., 01/01/1900).
- **Multiple Empty Fields (B014)**: System may proceed with multiple empty required fields.
- **Name Length Validation (B015–B016)**: System allows first/last names exceeding 50 characters.
- **Phone Number Validation (B017–B019)**: System rejects valid 10-digit phone numbers, allows 9-digit or 16-digit numbers.
- **City Validation (B020)**: System allows city names with numbers (e.g., 12345).

2. **Checkout (UC02)**

- **Quantity Validation (B021–B026)**: System allows checkout with quantities above 10, zero, negative, non-numeric, or empty, or resets quantity to 10 without error (e.g., quantity=100, 0, -1, "abc", "").
- **Cart Removal (B027)**: System does not allow removing items from the cart via the Remove button.
- **Bank Transfer Validation (B028–B031)**: System allows bank names with numbers, account numbers with letters, or account numbers shorter/longer than 8–34 digits.
- **Payment Method Fields (B032–B035)**: Selecting Credit Card, Cash on Delivery, Gift Card, or Buy Now Pay Later displays incorrect Bank Transfer input fields or fails to enable Confirm button.

5.2.2 Resolutions

1. **Sign Up**

- Implement unique email validation to reject duplicate emails.

- Enforce password requirements (min 10 characters, uppercase, lowercase, number, special symbol).
- Restrict postal codes to exactly 6 numeric digits.
- Validate DOB to reject future dates and ensure users are at least 16 years old, with a reasonable historical range (e.g., post-1900).
- Fix error messages for missing state/country to display correct prompts.
- Ensure multiple empty fields trigger appropriate errors and prevent registration.
- Limit first/last names to 50 characters and reject invalid characters.
- Validate phone numbers to accept 10–15 digits and reject shorter/longer or non-numeric inputs.
- Restrict city field to letters and spaces only.

2. **Checkout**

- Enforce cart quantity validation (1–10) and disable checkout for invalid quantities (0, negative, non-numeric, empty).
- Fix Remove button functionality to clear items from the cart.
- Validate Bank Transfer fields: restrict Bank Name to letters/spaces, Account Number to 8–34 numeric digits.
- Ensure correct input fields are displayed for each payment method (Credit Card, Cash on Delivery, Gift Card, Buy Now Pay Later) and enable Confirm button appropriately.

6 Self-Assessment

Criteria	Description	Max	Self-Assessed	Justification
Feature Selection	2 important features selected	1.0	1.0	Selected "Sign Up" and "Checkout" 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.

Criteria	Description	Max	Self-Assessed	Justification
Test Case Design	Test cases are clear, traceable, professional	2.0	2.0	Designed 66 thorough test cases (41 Sign Up, 30 Checkout) 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.
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	