University of Science - VNUHCM

Faculty of Information Technology

Software Testing

HW2 - Domain Testing

Software Testing Project Report

Authors:
Trần Thị Cát Tường
(22127444)
ttctuong22@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 Equivalence Partitioning and Boundary Value Analysis Design Process

2.1 Feature 1: Contact

2.1.1 Inputs and Constraints

The Contact form includes the following fields, with variations depending on user authentication status, all constraints were identified through manual interaction with the web UI and observing real-time validation errors on the form:

- **If the user is logged in:**
  - **Subject**: dropdown selection, required, must select a valid (non-default) option.
  - **Message**: required textarea, must be between 50 and 250 characters.
  - **Attachment**: optional, only `.txt`, `.pdf`, or `.jpg` files allowed, size $\leq$ 500KB.

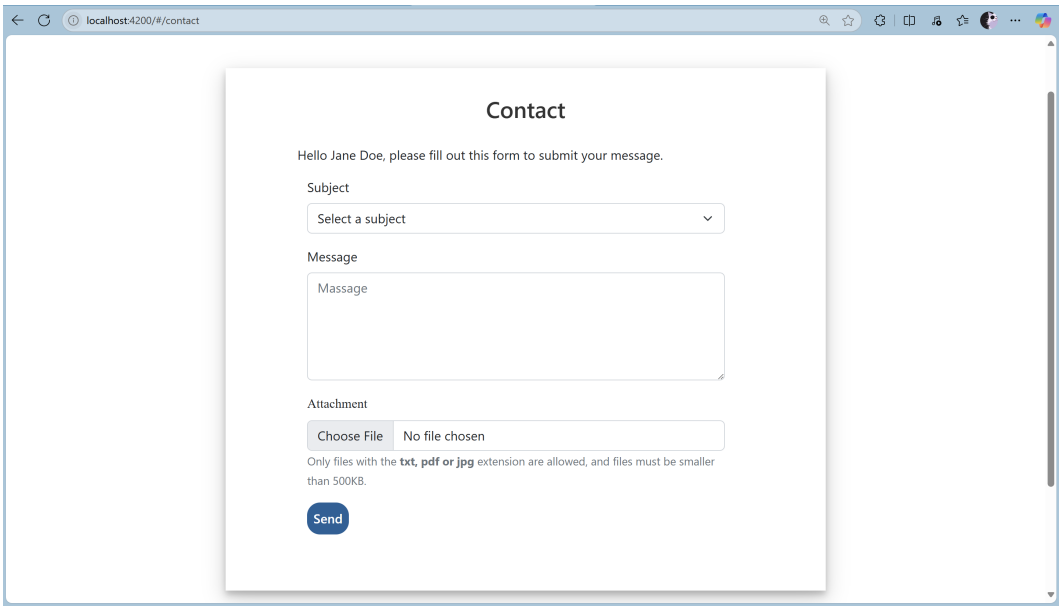

Figure 1: Contact function interface for user

- **If the user is a guest (not logged in):**
  - **First Name**: required text input, max length 120 characters.
  - **Last Name**: required text input, max length 120 characters.
  - **Email**: required, max length 120 characters, must be in valid email format (e.g., name@example.com).
  - **Subject**: dropdown selection, required, must select a valid (non-default) option.
  - **Message**: required textarea, must be between 50 and 250 characters.
  - **Attachment**: optional, only `.txt`, `.pdf`, or `.jpg` files allowed, size $\leq$ 500KB.

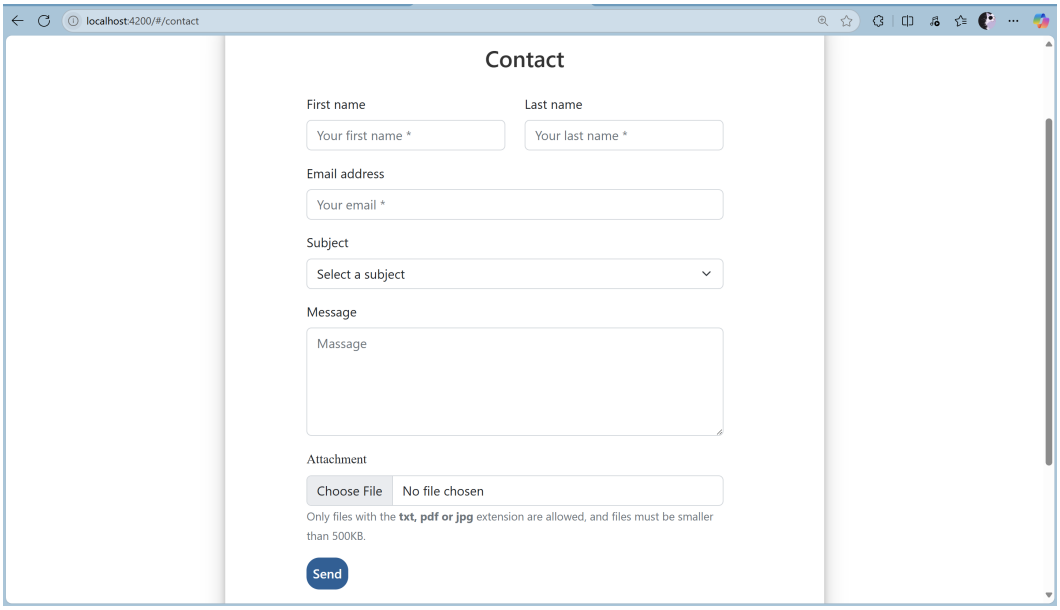

Figure 2: Contact function interface for guest

2.1.2 Equivalence Partitioning (EP)

Field	Valid Partition	Invalid Partition
First Name	Non-empty, $\leq$ 120 characters	Empty, $>$ 120 characters
Last Name	Non-empty, $\leq$ 120 characters	Empty, $>$ 120 characters
Email	Valid email format (e.g. `a@b.com`)	Missing `@`, missing domain, empty, malformed format, $>$ 120 characters
Subject	A valid selected option (not default/empty)	Not selected (empty/default value)
Message	Non-empty, $\geq$ 50 and $\leq$ 250 characters	Empty, $<$ 50 characters, $>$ 250 characters
Attachment	`.txt`, `.pdf`, `.jpg`, file size $\leq$ 500KB	Wrong type (e.g. `.docx`, `.exe`), size $>$ 500KB

2.1.3 Boundary Value Analysis (BVA)

Field	Valid Values	Invalid Values
First Name	Length: 1, 120	Length: 0, 121
Last Name	Length: 1, 120	Length: 0, 121
Email	Format: `a@b.com`, `a@b.c`; Length: 1, 120	Format: `a.com`, `a@`, `a@b`, `a@b..com`; Length: 0, 121
Subject	A selected option from dropdown	Default (empty), not in list
Message	Length: 50, 250	Length: 0, 49, 251
Attachment	Type: `.txt`, `.pdf`, `.jpg`; Size: 0KB, 500KB	Type: `.docx`, `.exe`; Size: 501KB

2.2 Feature 2: Category Management

2.2.1 Inputs and Constraints

Category Management allows admin to perform the following operations, all constraints were identified through manual interaction with the web UI and observing real-time validation errors on the form:

1. **Add Category**

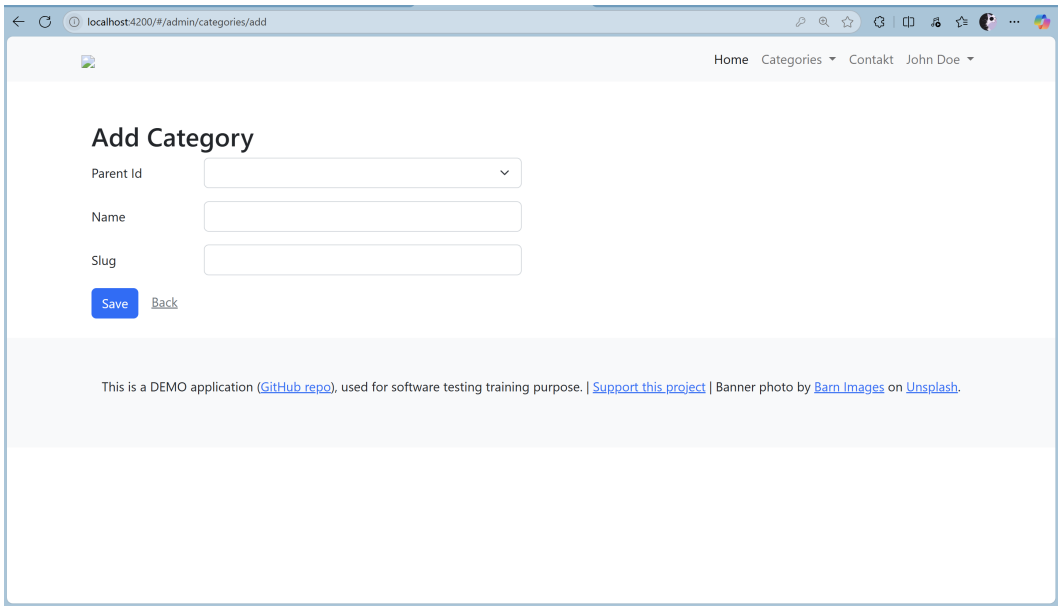

Figure 3: Add category function interface for admin

- **Parent Id** (Dropdown):
  - Optional.
  - Must be the `id` of an existing category.
  - Cannot be the same as the new category's own `id` (if known).
- **Name** (Text):
  - Required.
  - Maximum length: 120 characters.
  - Cannot be the same as an existing category.
- **Slug** (Text):
  - Required.
  - Must be lowercase, URL-safe (hyphens instead of spaces).
  - Must be unique (no duplicate slugs allowed in the database).

2. **Edit Category**

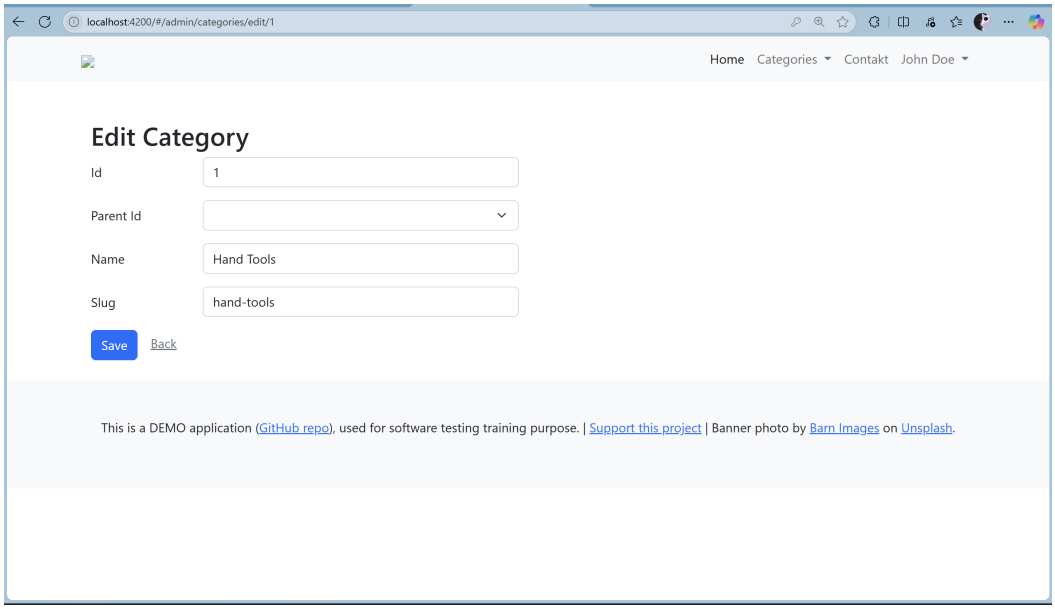

Figure 4: Edit category function interface for admin

- Same input rules as **Add**.
- Cannot change to have itself as its own parent.

3. **Delete Category**

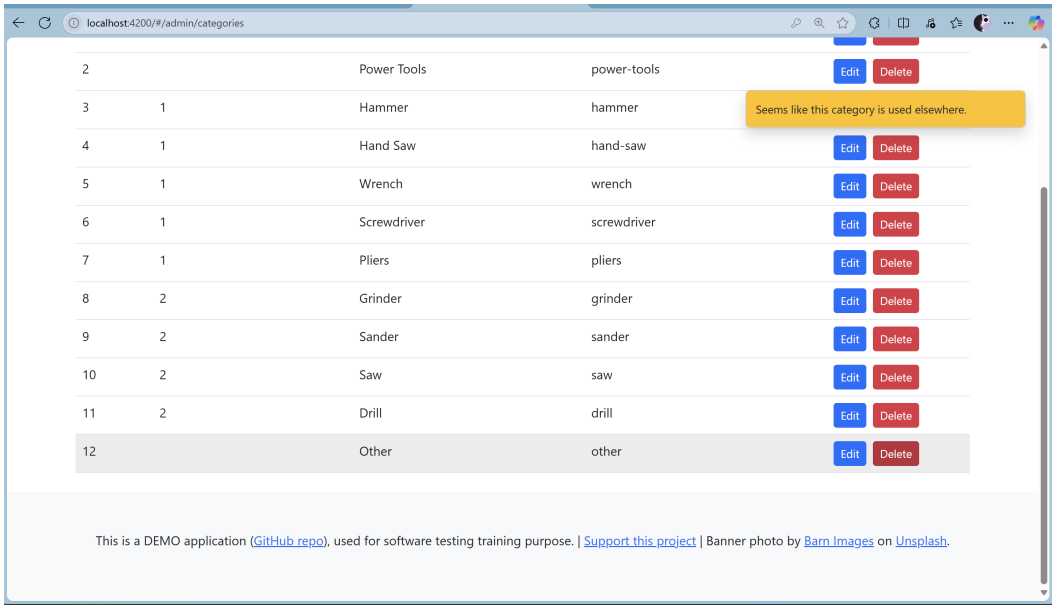

Figure 5: Delete category function interface for admin

- Allowed only if:
  - The category is not currently set as `Parent Id` of other categories or is having products.
  - Otherwise, a warning is shown: *"Seems like this category is used elsewhere."*

4. **Search Category**

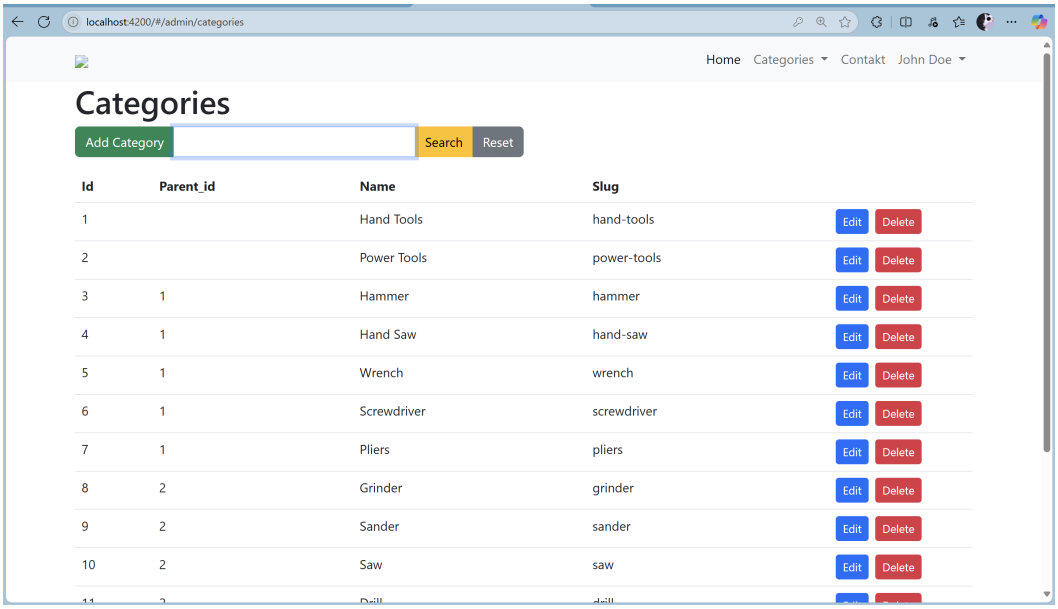

Figure 6: Search category function interface for admin

- Search by keyword in the **Name** field.
- Case-insensitive and supports partial match.
- Empty search returns full list.

2.2.2 Equivalence Partitioning (EP)

Operation	Field	Valid Partition	Invalid Partition
Add/Edit	Parent Id	Empty or existing category `id` (own id)	Non-existing `id`, or same as own `id`
Add/Edit	Name	Non-empty string 120 characters	Empty string, > 120 characters
Add/Edit	Slug	Unique, lowercase, hyphenated, 120 characters	Empty, contains spaces/special chars, duplicate in database
Delete	Category	Category with no child categories	Category used as parent elsewhere
Search	Keyword	Empty or string that matches existing category name (partial)	Strings that don't match any name

2.2.3 Boundary Value Analysis (BVA)

Field	Valid Values	Invalid Values
Name	Length: 1, 120	Length: 0, 121
Slug	Length: 1, 120; Format: `hand-tools`	Length: 0, 121; Format: `Hand Tools`, `tool!`

Field	Valid Values	Invalid Values
Parent Id	Existing IDs current id	Own `id`, non-existing ids
Search	Length: 0 (returns all), Length: 1–120	— (no functional invalid case for input size)

3 Use of AI Tools

Tools Used: ChatGPT

Purpose:

An AI tool was employed to generate a list of test cases based on manually constructed **Equivalence Partitioning (EP)** and **Boundary Value Analysis (BVA)** tables for the *Category Management* feature.

This approach helped:

- Accelerate the test case creation process.
- Ensure full coverage of valid/invalid partitions and boundary values.

Prompt Used to Generate Test Cases:

```
I have manually created EP and BVA tables for a feature called [feature
 name]. Below is the detailed input constraints, followed by the EP and
 BVA tables.

Please generate at least 40 high-quality test cases using both EP and
BVA logic. Cover both positive and negative cases, including field-level
 and logic-level scenarios (e.g., invalid parent ID, duplicate slugs,
 invalid deletions).

For each test case, provide:
- Test Case ID
- Title
- Preconditions (if any)
- Input values
- Test steps
- Expected result
- Type (EP or BVA)

[Inputs and Constraints]

[EP and BVA tables]
```

The full EP and BVA tables (as shown in Part 2 of the report) were directly pasted into the prompt. This provided ChatGPT with a complete understanding of input constraints, value ranges, and test boundaries, resulting in a high-quality, logic-driven test case list.

Review and Refinement Process:

- Each test case was reviewed to verify coverage of all identified EP classes and BVA points.
- Titles, test steps, and expected outcomes were adjusted where necessary to improve clarity and consistency.
- Redundant or low-value cases were removed to keep the final set concise and executable.

Test Case Categorization:

- **AI-generated**: The initial test case set was produced directly from the provided prompt and tables.
- **Manually refined**: Selected cases were edited for logic accuracy, completeness, and formatting quality.

Reusability:

The prompt is designed to be reusable for any web-based feature or form. By supplying well-structured EP and BVA tables along with clear input constraints, this method can consistently generate 30–40 high-quality test cases in a single session while maintaining test logic and coverage control.

4 Self-Evaluation

Criteria	Self-Evaluation	Notes
Feature Selection	1.0 / 1.0	Selected two important and realistic features with high business impact.
EP Technique	2.0 / 2.0	Clearly identified valid/invalid partitions with justified reasoning.
BVA Technique	1.0 / 1.0	Properly identified boundary values with rationale.
Test Case Design	2.0 / 2.0	Well-structured and professional test cases based on EP and BVA results.
Use of AI Tools	1.0 / 1.0	Provided tool name, prompt, validation method, and highlighted AI usage.
Test Execution	1.0 / 1.0	All test cases executed and actual results documented clearly.
Bug Reporting	1.0 / 1.0	Detailed bug reports with reproduction steps and necessary information.
Merging and Final Review	0.5 / 0.5	Removed duplicates, combined cases logically and ensured full coverage.
Presentation & Clarity	0.5 / 0.5	Report is well-organized with clear formatting and honest reflection.