



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 11, 2025

Table of Contents

| | | |
|----------|--|-----------|
| 1 | Group Information | 1 |
| 2 | Equivalence Partitioning and Boundary Value Analysis Design Process | 2 |
| 2.1 | Feature 1: Contact | 2 |
| 2.1.1 | Inputs and Constraints | 2 |
| 2.1.2 | Equivalence Partitioning (EP) | 2 |
| 2.1.3 | Boundary Value Analysis (BVA) | 3 |
| 2.2 | Feature 2: Category Management | 4 |
| 2.2.1 | Inputs and Constraints | 4 |
| 2.2.2 | Equivalence Partitioning (EP) | 6 |
| 2.2.3 | Boundary Value Analysis (BVA) | 6 |
| 3 | Use of AI Tools | 8 |
| 4 | Self-Evaluation | 10 |

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 | - Feature 1 | Done |
| | | - Feature 2 | 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.

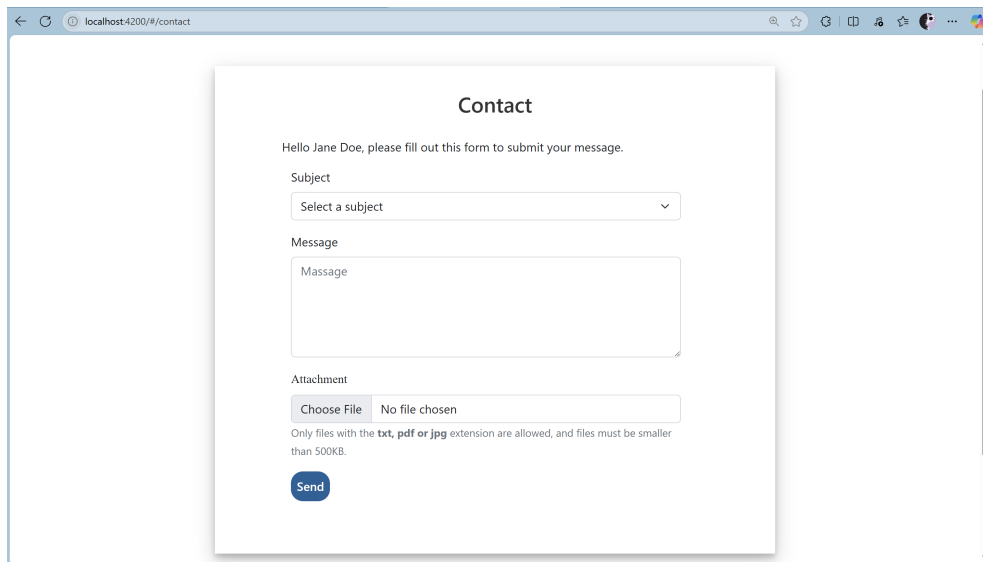
A screenshot of a web browser showing a contact form titled "Contact". The browser's address bar shows "localhost:4200/#/contact". The form is centered on the page and has a white background with a subtle shadow. It contains the following elements: a greeting "Hello Jane Doe, please fill out this form to submit your message.", a "Subject" dropdown menu with the placeholder text "Select a subject", a "Message" text area with the placeholder text "Message", an "Attachment" section with a "Choose File" button and the text "No file chosen", and a "Send" button at the bottom. Below the attachment section, there is a small note: "Only files with the .txt, .pdf or .jpg extension are allowed, and files must be smaller than 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.

2.1.2 Equivalence Partitioning (EP)

The image shows a web browser window with the address 'localhost:4200/#/contact'. The page title is 'Contact'. The form contains the following elements:

- First name:** Input field with placeholder 'Your first name *'.
- Last name:** Input field with placeholder 'Your last name *'.
- Email address:** Input field with placeholder 'Your email *'.
- Subject:** Dropdown menu with 'Select a subject'.
- Message:** Text area with placeholder 'Message'.
- Attachment:** File upload area with a 'Choose File' button and 'No file chosen' text. Below it, a note states: 'Only files with the txt, pdf or jpg extension are allowed, and files must be smaller than 500KB.'
- Send:** A blue button at the bottom of the form.

Figure 2: Contact function interface for guest

| Field | Valid Partition | Invalid Partition |
|------------|--|--|
| First Name | Non-empty, ≤ 120 characters | Empty, > 120 characters |
| Last Name | Non-empty, ≤ 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, ≥ 50 and ≤ 250 characters | Empty, < 50 characters, > 250 characters |
| Attachment | .txt, .pdf, .jpg, file size ≤ 500 KB | Wrong type (e.g. .docx, .exe), size > 500 KB |

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

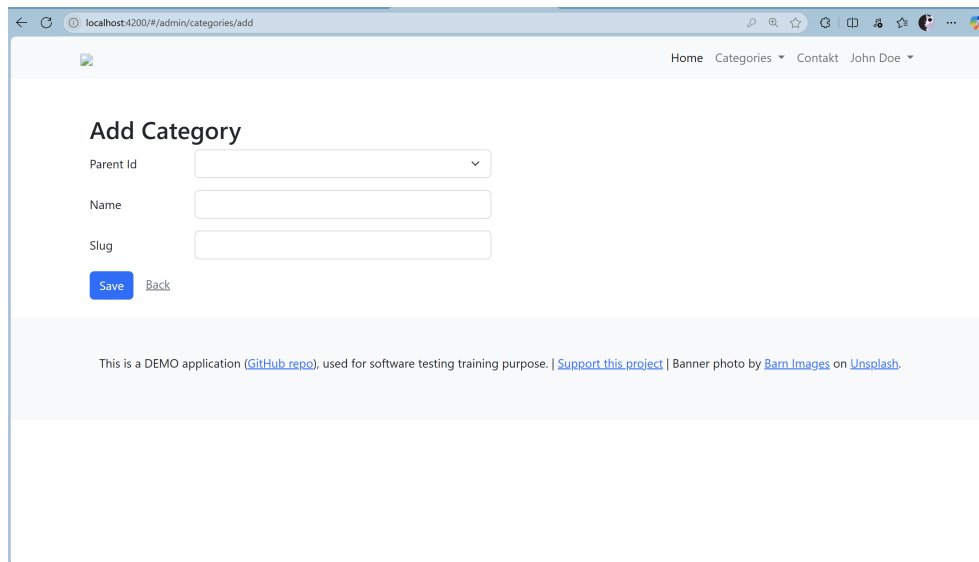
A screenshot of a web browser showing the 'Add Category' form. The browser's address bar shows 'localhost:4200/#/admin/categories/add'. The page has a navigation bar with 'Home', 'Categories', 'Contact', and 'John Doe'. The form is titled 'Add Category' and contains three input fields: 'Parent Id' (a dropdown menu), 'Name' (a text input), and 'Slug' (a text input). Below the inputs are two buttons: 'Save' (in blue) and 'Back' (in grey). At the bottom of the page, there is a footer text: 'This is a DEMO application (GitHub repo), used for software testing training purpose. | Support this project | Banner photo by Barn Images on Unsplash.'

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

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

3. Delete Category

- 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.”*

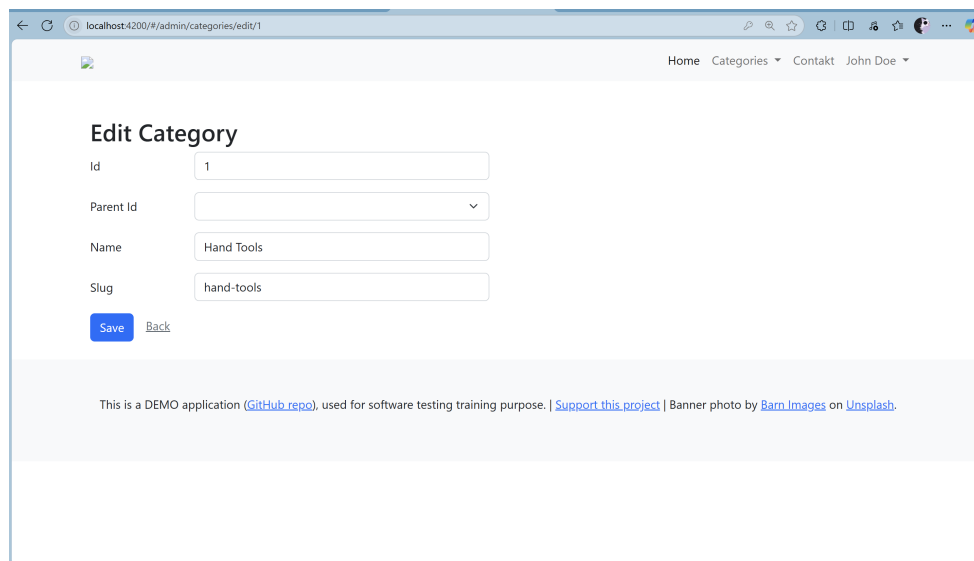


Figure 4: Edit category function interface for admin

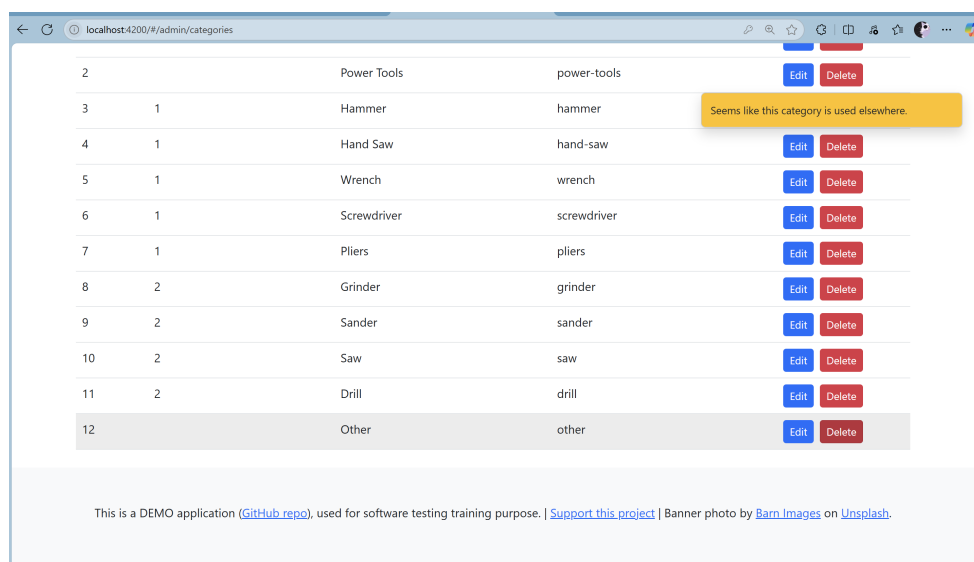


Figure 5: Delete category function interface for admin

4. Search Category

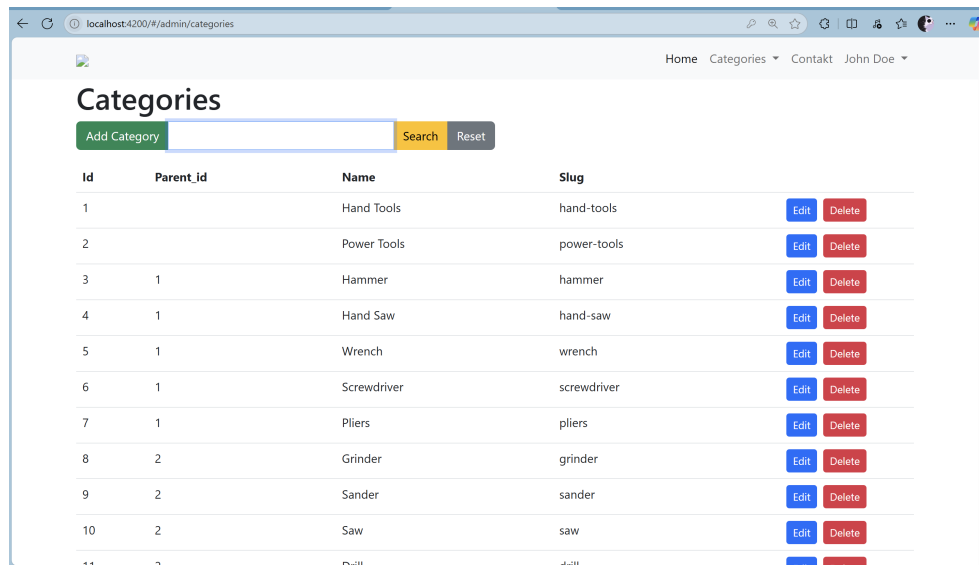


Figure 6: Search category function interface for admin

- 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! |
| Parent Id | Existing IDs current id | Own id, non-existing ids |

| Field | Valid Values | Invalid Values |
|--------|---|---|
| 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 |
|--------------------------------------|-----------------|--|
| Chose appropriate features | 100% | Features are clearly divided and relevant to real-world scenarios. |
| Applied EP/BVA correctly | 100% | Input constraints are well justified with systematic partitioning. |
| Designed clear test cases | 100% | Excel sheet provided with distinct EP and BVA cases. |
| Correct use of AI | 100% | Prompt and validation steps included in report. |
| Executed testing properly | 100% | Full test results documented, including actual vs expected outcomes. |
| Reported bugs accurately | 100% | Bug report includes clear reproduction steps. |
| Merged and refined test cases | 100% | Duplicate/overlapping cases removed; coverage ensured. |
| Followed submission instructions | 100% | All required files included and properly named. |
| Maintained professional presentation | 100% | Report well-organized with consistent formatting and explanations. |
| Performed honest self-assessment | 100% | Criteria reviewed based on actual contribution and completeness. |