Vietnam National University- Ho Chi Minh City

University of Science

Faculty of Information Technology

HW03. DATA GENERATION & SCENARIO TESTING

Course Software Testing
Class 22KTPM3
Group Group10
Student **22127225 – Trần Thị Thiên Kim**

Ho Chi Minh City, 2025

Table of Contents

1 Task Allocation

Student ID	Student	Selected Feature
20127233	Huỳnh Thế Long	- User Profile Management - Product Details & Related Products - Rental Products System
22127225	Trần Thị Thiên Kim	- Product Search and Catalog - Category Management - Shopping Cart Operation
22127312	Nguyễn Thị Yến Nhi	- User Favorites/Wishlist - Product Filtering & Sorting - Contact Form with File Upload
22127316	Nguyễn Ngô Như Ngọc	- User Registration & Authentication - Customer Checkout - Brand Management

2 Introduction

2.1 Objectives

This report aims to apply scenario-based testing techniques to validate critical functionalities of the ToolShop system. The objectives include:

- Select two essential features from the system under test.
- Define realistic user-driven scenarios for each feature.
- Apply scenario testing to design detailed and meaningful test cases.
- Document the reasoning behind test design and testing strategy.

2.2 Overview of Scenario Testing

- Scenario testing is a black-box testing technique that focuses on user-centric interactions. Each scenario describes a real-world flow or usage pattern of the application, allowing testers to design test cases that mimic how actual users behave in the system.
- Unlike unit tests, which verify specific components in isolation, scenario-based tests assess how well different system components work together in realistic contexts. This helps uncover usability issues, integration bugs, and logical inconsistencies across modules.

3 Scenario Testing: Product Search & Catalog

3.1 Selected Feature

Product Search & Catalog is one of the core features of the ToolShop application. It allows users to explore the product catalog by:

- Typing in a keyword to search for products.
- Applying price range filters (minimum and maximum price).
- Combining both keyword and price filter to refine results.

The feature provides instant feedback through a user interface that displays matching results dynamically or informs users when no relevant products are found.

3.2 Scenario Description

Scenario: A user performs a product search by entering a keyword, applying price range filters, or using both methods simultaneously. The system should respond accurately by:

- Returning relevant product listings if available.
- Displaying appropriate validation messages for incorrect input.
- Showing an informative message when no matching products are found.

3.3 Applying Scenario Testing to Design Test Cases

To design test cases for this feature, I applied the principles of scenario testing by focusing on realistic user actions that combine search and filtering. The test design process included:

- **Keyword-based Search:** Entering a full or partial keyword that exists in the database, or an invalid one, and observing how the system handles it.
- **Price Filtering Only:** Applying a price range without entering a keyword, using various valid and invalid ranges, including edge cases like zero or negative values.
- **Combined Search and Filter:** Performing a search with both keyword and price filter to simulate real-life narrowing of results.
- **Dynamic Filter Adjustment:** Modifying filter conditions after an initial search and verifying that the product listing updates correctly.
- **Empty or No Match Results:** Searching for products with combinations that are expected to return no results, and validating the response message.

Each test case was written from a user perspective, following typical usage patterns. This helps ensure the system not only functions correctly but also handles unusual or invalid interactions gracefully.

4 Scenario Testing: Shopping Cart Operation

4.1 Selected Feature

Shopping Cart Operation is a core feature of the ToolShop system that enables users to add, update, or remove products from their virtual shopping cart. It plays a critical role in the purchase flow, providing real-time feedback on item selections and total cost before proceeding to checkout. It allows users to:

- Add products to the cart.
- Adjust product quantities.
- Remove products.
- View real-time updates of the cart's total price.

This functionality ensures that users can make informed purchasing decisions while the system maintains pricing accuracy, reflects inventory limits, and provides consistent feedback.

4.2 Scenario Description

Scenario: A user adds one or more products to the shopping cart, modifies the quantity of items, removes selected items, and verifies that the cart's total price reflects the correct value at all times. The scenario includes validating:

- The addition and removal logic for multiple products.
- The behavior when invalid or edge-case values (e.g., quantity $= 0$, quantity $>$ stock) are entered.
- The system's ability to handle duplicate additions or product updates.
- Visual and data integrity of the cart in all cases.

4.3 Applying Scenario Testing to Design Test Cases

Test cases for this feature were designed using scenario-based thinking to reflect how users interact with a shopping cart in a real online store:

- **Add Product to Cart:** Testing addition of single and multiple distinct products, including checking for duplicates.
- **Update Quantity:** Adjusting quantity of a product after it has been added to the cart, including increasing, decreasing, and setting it to zero.
- **Remove Product:** Verifying correct behavior when removing products and ensuring the total price is recalculated correctly.
- **Edge Cases:** Attempting to add out-of-stock products, exceeding inventory limits, or submitting non-numeric quantities.

- **Cart Persistence and Feedback:** Ensuring the cart reflects all updates in real-time and provides users with appropriate messages or alerts.

This approach ensures broad test coverage by capturing not only standard interactions but also user misbehavior and abnormal inputs, validating both functionality and user experience.

5 Use of AI Tools

In this assignment, I used the AI tool **ChatGPT (OpenAI)** to support the scenario-based test design process.

5.1 Tool Name

- **Tool:** ChatGPT (OpenAI)
- **Version:** GPT-4.0, accessed via web at `chat.openai.com`

5.2 Prompts Used

To generate realistic and complete test cases that align with scenario testing methodology, I crafted the following structured prompts:

- **Prompt for Product Search & Catalog:**

  > You are a QA engineer working on the ToolShop system — an e-commerce platform for tools. Based on the scenario: "A user searches for products by entering a keyword, using filters, or combining both," generate a set of scenario-based test cases for the 'Product Search & Catalog' feature. Include flows such as valid keyword search, price filtering only, combined filters, no result, and invalid inputs. Each test case should include: Test Case ID, Title, Preconditions, Inputs, Test Steps, and Expected Result.

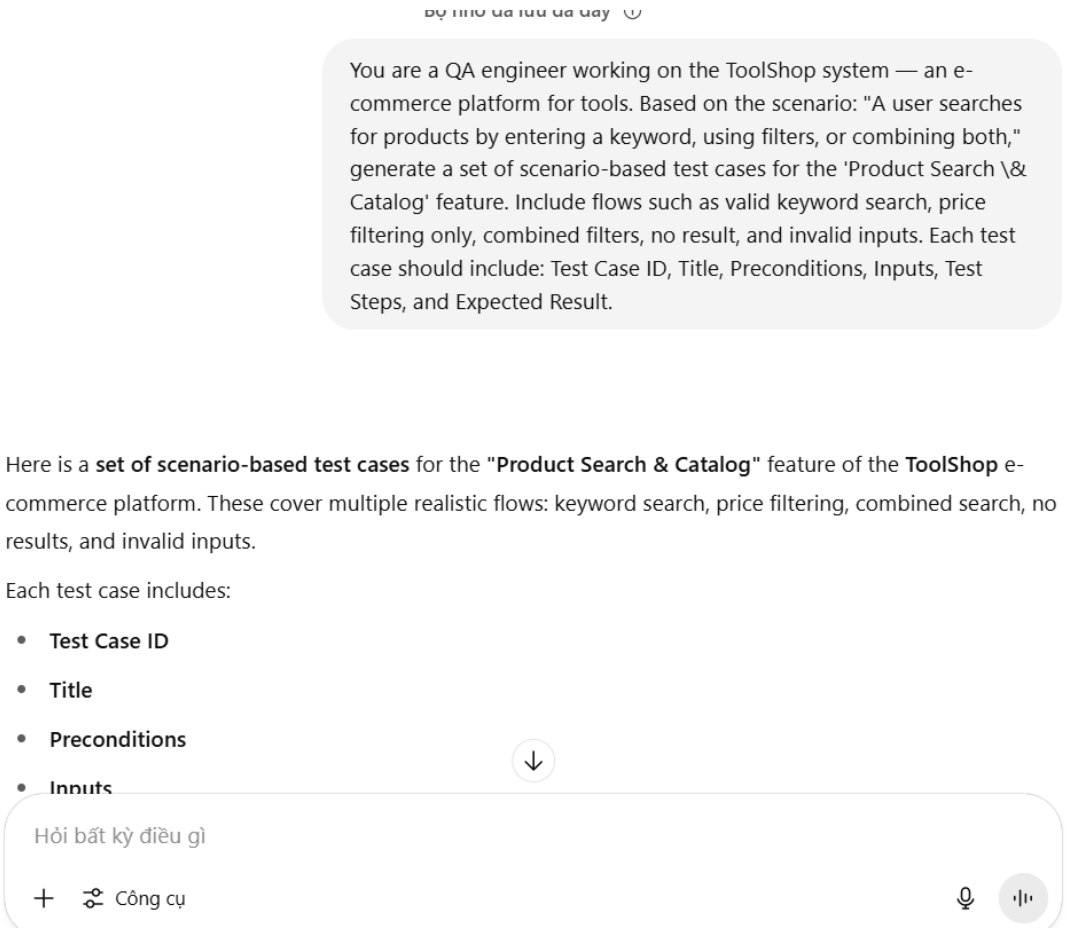

Figure 1: Prompting ChatGPT to generate testcase for ProductSearch&Catalog

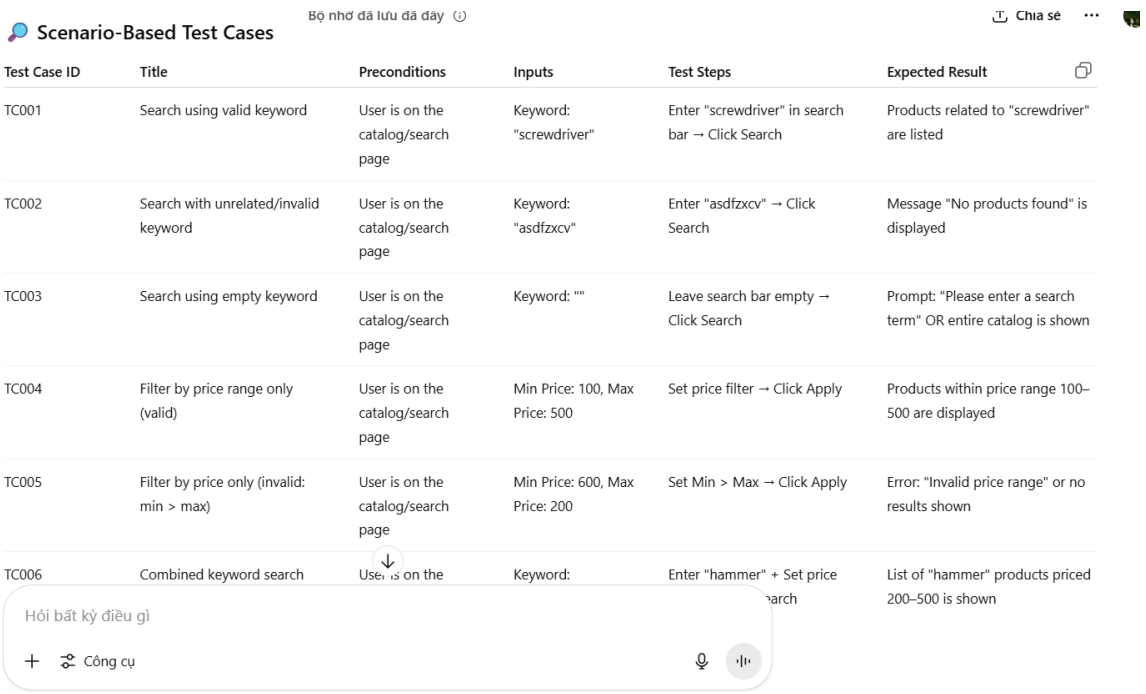

Figure 2: Detailed test cases for the Product Search & Catalog

- **Prompt for Shopping Cart Operation:**

  You are a QA engineer testing the Shopping Cart feature of the ToolShop system. Based on the scenario: "A user adds one or more products to the shopping cart, modifies quantity, removes items, and verifies the total price," generate a full set of test cases covering typical and edge user flows. Include test cases for normal add/remove/update actions, quantity updates, zero or negative quantities, exceeding stock, and verifying the cart total. For each test case, provide: Test Case ID, Title, Preconditions, Inputs, Test Steps, and Expected Result.

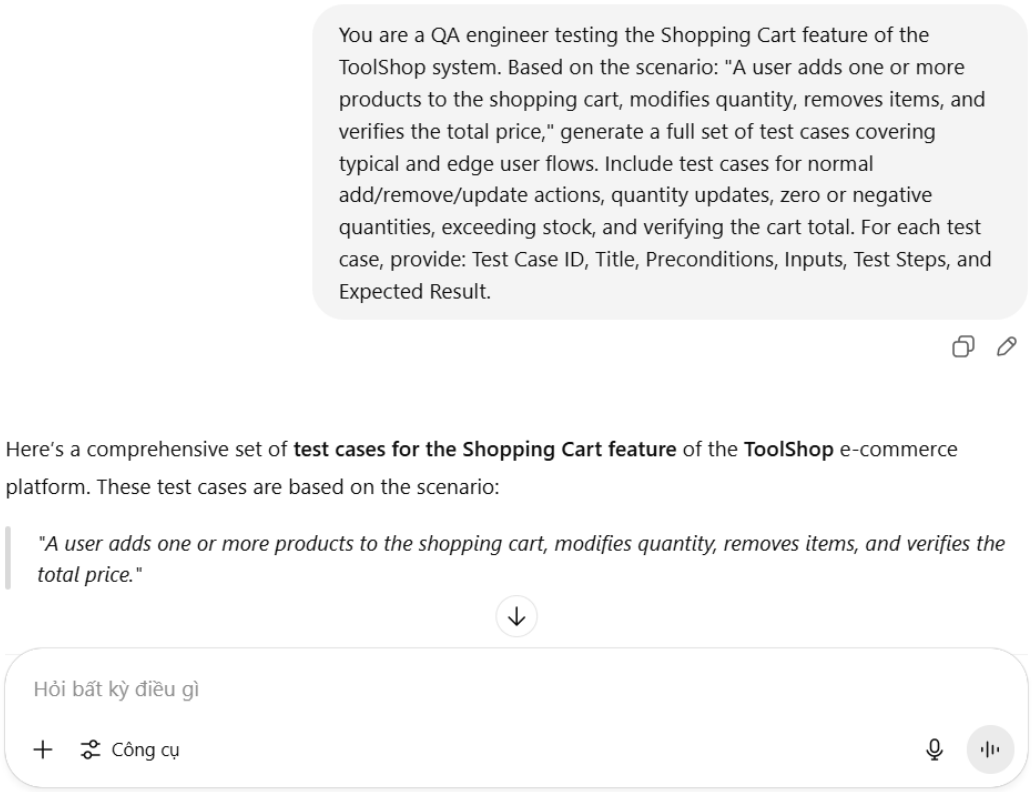

Figure 3: Prompting ChatGPT to generate testcase for Shopping Cart Operation

✅ **Shopping Cart Test Cases**

Test Case ID	Title	Preconditions	Inputs	Test Steps	Expected Result
TC001	Add single product to cart	Product A is in stock	Product A, Qty: 1	Click "Add to Cart"	Product A appears in cart with quantity 1
TC002	Add multiple different products	Product A and B are in stock	Product A (Qty: 1), Product B (Qty: 2)	Add both products	Both products appear with correct quantities
TC003	Update product quantity in cart	Product A is in cart	New Qty: 3	Update quantity in cart	Product A updates to Qty: 3
TC004	Increase quantity beyond stock	Product A has stock = 5	Set Qty: 10	Update quantity	Error shown: "Only 5 items available"
TC005	Update quantity to zero	Product B is in cart	Qty: 0	Update quantity	Product B is removed from cart
TC006	Update quantity to negative number	Product B is in cart	Qty: -1	Update quantity	Error: "Quantity must be 1 or more"

Figure 4: Detailed test cases for the Shopping Cart Operation

5.3 Validation and Refinement

All AI-generated content was critically reviewed and refined by myself before being included in the report:

- I cross-checked each test case against the actual system behavior in the ToolShop repository to ensure relevance and realism.
- I removed or revised test cases that were not applicable to the current implementation of ToolShop.
- I adjusted wording, logic, and expected results to match the real business rules and database schema.

5.4 Test Case Attribution

- **AI-Generated Test Cases:**

  Most test cases in both scenarios (Product Search & Catalog, Shopping Cart Operation) were initially generated by ChatGPT based on my prompt and then reviewed and edited.

- **Manually Created or Modified:**

  I manually adjusted edge cases and system-specific conditions based on my understanding of ToolShop's features and database (e.g., handling of invalid price ranges, cart quantity limits, and stock behavior).