## **Phần 1: Lý thuyết**

### **1. Phân biệt các kỹ thuật kiểm thử**

**a. Sự khác biệt giữa black-box testing, white-box testing và kiểm thử dựa trên kinh nghiệm**

* **Black-box testing**: Là kỹ thuật kiểm thử mà người kiểm thử không biết cấu trúc hoặc mã nguồn bên trong của hệ thống. Người kiểm thử chỉ tập trung vào đầu vào và đầu ra, dựa trên các yêu cầu chức năng hoặc đặc tả hệ thống. Ví dụ: Kiểm thử giao diện người dùng của một ứng dụng mà không cần biết cách lập trình phía sau.
* **White-box testing**: Người kiểm thử có quyền truy cập vào mã nguồn và cấu trúc bên trong của hệ thống. Kỹ thuật này tập trung vào kiểm tra logic, luồng điều khiển, và các nhánh mã. Ví dụ: Kiểm thử một hàm tính toán trong mã nguồn để đảm bảo tất cả các nhánh điều kiện được thực thi.
* **Kiểm thử dựa trên kinh nghiệm**: Dựa trên kinh nghiệm, trực giác, hoặc kiến thức của người kiểm thử về hệ thống hoặc các lỗi phổ biến, không dựa vào quy trình chính thức hay tài liệu đặc tả. Ví dụ: Một người kiểm thử kỳ cựu có thể kiểm tra các tính năng dễ xảy ra lỗi dựa trên kinh nghiệm từ các dự án trước.

**b. Ví dụ về black-box testing trong kiểm thử tính năng đăng nhập của website**

* **Tình huống**: Kiểm thử tính năng đăng nhập của một website thương mại điện tử.
* **Cách thực hiện**: Người kiểm thử nhập các tổ hợp tên người dùng và mật khẩu khác nhau (đúng, sai, trống, ký tự đặc biệt) mà không cần biết cách hệ thống xử lý dữ liệu. Ví dụ:
  + **Đầu vào**: Tên người dùng đúng, mật khẩu đúng → **Kết quả mong đợi**: Đăng nhập thành công.
  + **Đầu vào**: Tên người dùng đúng, mật khẩu sai → **Kết quả mong đợi**: Hiển thị thông báo lỗi “Mật khẩu không đúng”.
  + **Đầu vào**: Tên người dùng trống → **Kết quả mong đợi**: Hiển thị thông báo lỗi “Vui lòng nhập tên người dùng”.

### **2. Equivalence Partitioning**

**a. Equivalence partitioning là gì?**

* **Equivalence Partitioning (EP)** là kỹ thuật chia dữ liệu đầu vào thành các nhóm (phân vùng) mà hệ thống được kỳ vọng sẽ xử lý giống nhau. Mỗi nhóm đại diện cho một tập hợp dữ liệu có cùng hành vi, do đó chỉ cần kiểm thử một giá trị đại diện từ mỗi nhóm để giảm số lượng test case mà vẫn đảm bảo độ bao phủ.
* **Cách chia dữ liệu**: Dữ liệu được chia thành các phân vùng hợp lệ (valid) và không hợp lệ (invalid) dựa trên yêu cầu đặc tả. Ví dụ, nếu một trường nhập tuổi chỉ chấp nhận từ 18 đến 60, thì các phân vùng sẽ là:
  + Hợp lệ: 18–60.
  + Không hợp lệ: < 18, > 60.

**b. Ví dụ áp dụng equivalence partitioning**

* **Trường hợp**: Kiểm thử trường nhập tuổi của người dùng (yêu cầu: tuổi từ 18 đến 60).
* **Phân vùng**:
  + **Hợp lệ**: 18 ≤ Tuổi ≤ 60 (ví dụ: 25).
  + **Không hợp lệ (dưới giới hạn)**: Tuổi < 18 (ví dụ: 17).
  + **Không hợp lệ (vượt giới hạn)**: Tuổi > 60 (ví dụ: 61).
  + **Không hợp lệ (dữ liệu không phải số)**: Chuỗi ký tự, ví dụ: “abc”.
* **Test case**:
  + Nhập tuổi = 25 → Hệ thống chấp nhận.
  + Nhập tuổi = 17 → Hệ thống báo lỗi “Tuổi phải từ 18 trở lên”.
  + Nhập tuổi = 61 → Hệ thống báo lỗi “Tuổi không được vượt quá 60”.
  + Nhập tuổi = “abc” → Hệ thống báo lỗi “Vui lòng nhập số hợp lệ”.

### **3. Boundary Value Analysis**

**a. Boundary Value Analysis (BVA) là gì?**

* **BVA** là kỹ thuật kiểm thử tập trung vào các giá trị biên hoặc ranh giới của các phân vùng dữ liệu đầu vào. Các lỗi thường xảy ra tại các giá trị biên (ví dụ: giá trị tối thiểu, tối đa, hoặc ngay ngoài giới hạn).
* **Tại sao hiệu quả?** Lỗi thường xuất hiện tại các ranh giới do sai sót trong lập trình logic (ví dụ: sử dụng “<” thay vì “<=”). Kiểm thử các giá trị biên giúp phát hiện những lỗi này nhanh chóng.

**b. Ví dụ sử dụng BVA**

* **Trường hợp**: Kiểm thử trường nhập số lượng vé đặt trong ứng dụng đặt vé xem phim (yêu cầu: số lượng từ 1 đến 10 vé).
* **Giá trị biên**:
  + Biên dưới: 1 (hợp lệ), 0 (không hợp lệ), 2 (hợp lệ).
  + Biên trên: 10 (hợp lệ), 11 (không hợp lệ), 9 (hợp lệ).
* **Test case**:
  + **Mô tả**: Kiểm tra giá trị biên dưới.
    - **Dữ liệu đầu vào**: Số lượng = 0.
    - **Kết quả mong đợi**: Hệ thống báo lỗi “Số lượng vé phải từ 1 đến 10”.
  + **Mô tả**: Kiểm tra giá trị biên dưới hợp lệ.
    - **Dữ liệu đầu vào**: Số lượng = 1.
    - **Kết quả mong đợi**: Hệ thống chấp nhận.
  + **Mô tả**: Kiểm tra giá trị biên trên.
    - **Dữ liệu đầu vào**: Số lượng = 11.
    - **Kết quả mong đợi**: Hệ thống báo lỗi “Số lượng vé phải từ 1 đến 10”.
  + **Mô tả**: Kiểm tra giá trị biên trên hợp lệ.
    - **Dữ liệu đầu vào**: Số lượng = 10.
    - **Kết quả mong đợi**: Hệ thống chấp nhận.

### **4. Thiết kế Test Case cho Black-Box Testing**

**a. Tại sao thiết kế test case hiệu quả giúp tăng độ bao phủ kiểm thử?**

* Thiết kế test case hiệu quả đảm bảo kiểm tra tất cả các trường hợp sử dụng chính, bao gồm cả trường hợp hợp lệ và không hợp lệ, giúp phát hiện lỗi ở nhiều khía cạnh của hệ thống.
* Sử dụng các kỹ thuật như EP và BVA giúp giảm số lượng test case nhưng vẫn bao phủ toàn bộ yêu cầu chức năng, tập trung vào các khu vực dễ xảy ra lỗi (như giá trị biên hoặc dữ liệu không hợp lệ).
* Test case được thiết kế tốt giúp kiểm thử có hệ thống, tránh bỏ sót các kịch bản quan trọng và tăng khả năng phát hiện lỗi.

**b. Các yếu tố cần có trong một test case**

* **ID của test case**: Mã định danh duy nhất.
* **Mô tả test case**: Mô tả ngắn gọn mục đích của test case.
* **Điều kiện tiên quyết**: Các điều kiện cần thiết trước khi thực hiện kiểm thử.
* **Dữ liệu đầu vào**: Giá trị hoặc hành động được nhập vào hệ thống.
* **Kết quả mong đợi**: Kết quả mà hệ thống phải trả về nếu hoạt động đúng.
* **Trạng thái**: Kết quả thực tế (pass/fail) sau khi kiểm thử.

## **Phần 2: Bài tập thực hành**

### **1. Áp dụng Equivalence Partitioning**

**Trường hợp**: Form nhập liệu số lượng sản phẩm (từ 1 đến 100) trong ứng dụng đặt hàng trực tuyến.

**a. Chia dữ liệu đầu vào thành các nhóm hợp lý**

* **Phân vùng hợp lệ**:
  + Số lượng từ 1 đến 100 (ví dụ: 50).
* **Phân vùng không hợp lệ**:
  + Số lượng < 1 (ví dụ: 0, -1).
  + Số lượng > 100 (ví dụ: 101).
  + Dữ liệu không phải số (ví dụ: “abc”, ký tự đặc biệt).

**b. Mô tả các nhóm và lý do**

* **Nhóm hợp lệ (1–100)**: Đại diện cho các giá trị mà hệ thống được thiết kế để chấp nhận. Kiểm thử một giá trị trong nhóm này (ví dụ: 50) đảm bảo hệ thống xử lý đúng các đầu vào hợp lệ.
* **Nhóm không hợp lệ (< 1)**: Kiểm tra khả năng xử lý lỗi khi người dùng nhập số lượng nhỏ hơn giá trị tối thiểu. Ví dụ, nhập 0 hoặc -1 để xác minh thông báo lỗi.
* **Nhóm không hợp lệ (> 100)**: Kiểm tra khả năng xử lý lỗi khi người dùng nhập số lượng vượt quá giới hạn tối đa. Ví dụ, nhập 101 để xác minh thông báo lỗi.
* **Nhóm không hợp lệ (dữ liệu không phải số)**: Kiểm tra khả năng xử lý lỗi khi người dùng nhập dữ liệu không hợp lệ, như chuỗi ký tự hoặc ký tự đặc biệt, để đảm bảo hệ thống từ chối đúng cách.

### **2. Áp dụng Boundary Value Analysis**

**Trường hợp**: Form nhập liệu số lượng sản phẩm (từ 1 đến 100).

**a. Xác định các giá trị biên cần kiểm thử**

* **Biên dưới**: 0 (không hợp lệ), 1 (hợp lệ), 2 (hợp lệ).
* **Biên trên**: 99 (hợp lệ), 100 (hợp lệ), 101 (không hợp lệ).

**b. Thiết kế 4 test case dựa trên giá trị biên**

1. **Mô tả**: Kiểm tra giá trị biên dưới không hợp lệ.
   * **Dữ liệu đầu vào**: Số lượng = 0.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Số lượng phải từ 1 đến 100”.
2. **Mô tả**: Kiểm tra giá trị biên dưới hợp lệ.
   * **Dữ liệu đầu vào**: Số lượng = 1.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xử lý đơn hàng.
3. **Mô tả**: Kiểm tra giá trị biên trên hợp lệ.
   * **Dữ liệu đầu vào**: Số lượng = 100.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xử lý đơn hàng.
4. **Mô tả**: Kiểm tra giá trị biên trên không hợp lệ.
   * **Dữ liệu đầu vào**: Số lượng = 101.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Số lượng phải từ 1 đến 100”.

### **3. Thiết kế Test Case cho Form nhập liệu**

**Trường hợp**: Form nhập liệu đặt lịch hẹn bác sĩ với các trường:

* Tên bệnh nhân (chuỗi, tối đa 50 ký tự).
* Ngày hẹn (phải là ngày trong tương lai so với ngày hiện tại, giả sử ngày hiện tại là 18/07/2025).
* Giờ hẹn (từ 08:00 đến 17:00).

**a. Thiết kế 6 test case sử dụng kỹ thuật black-box testing**

1. **Test Case 1**: Kiểm tra trường hợp hợp lệ với dữ liệu tiêu chuẩn.
   * **Mô tả**: Kiểm tra nhập liệu hợp lệ cho tất cả các trường.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A” (10 ký tự).
     + Ngày hẹn: 20/07/2025.
     + Giờ hẹn: 10:00.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xác nhận lịch hẹn thành công.
2. **Test Case 2**: Kiểm tra trường tên bệnh nhân vượt quá giới hạn ký tự.
   * **Mô tả**: Kiểm tra giá trị biên trên của trường tên bệnh nhân.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: Chuỗi 51 ký tự (ví dụ: “A” lặp lại 51 lần).
     + Ngày hẹn: 20/07/2025.
     + Giờ hẹn: 10:00.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Tên bệnh nhân không được vượt quá 50 ký tự”.
3. **Test Case 3**: Kiểm tra trường tên bệnh nhân trống.
   * **Mô tả**: Kiểm tra trường hợp không hợp lệ khi tên bệnh nhân để trống.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “” (trống).
     + Ngày hẹn: 20/07/2025.
     + Giờ hẹn: 10:00.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Vui lòng nhập tên bệnh nhân”.
4. **Test Case 4**: Kiểm tra ngày hẹn trong quá khứ.
   * **Mô tả**: Kiểm tra trường hợp không hợp lệ khi ngày hẹn không phải trong tương lai.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A”.
     + Ngày hẹn: 17/07/2025.
     + Giờ hẹn: 10:00.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Ngày hẹn phải là ngày trong tương lai”.
5. **Test Case 5**: Kiểm tra giá trị biên dưới của giờ hẹn.
   * **Mô tả**: Kiểm tra giờ hẹn tại ranh giới dưới hợp lệ.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A”.
     + Ngày hẹn: 20/07/2025.
     + Giờ hẹn: 08:00.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xác nhận lịch hẹn thành công.
6. **Test Case 6**: Kiểm tra giờ hẹn ngoài phạm vi cho phép.
   * **Mô tả**: Kiểm tra trường hợp không hợp lệ khi giờ hẹn nằm ngoài khoảng 08:00–17:00.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A”.
     + Ngày hẹn: 20/07/2025.
     + Giờ hẹn: 17:01.
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Giờ hẹn phải từ 08:00 đến 17:00”.

**b. Giải thích cách các test case đảm bảo độ bao phủ kiểm thử cao**

* **Kết hợp EP và BVA**: Các test case trên sử dụng EP để kiểm tra các phân vùng hợp lệ và không hợp lệ (ví dụ: tên hợp lệ, tên trống, ngày trong quá khứ, giờ ngoài phạm vi) và BVA để kiểm tra các giá trị biên (tên 51 ký tự, giờ 08:00, giờ 17:01). Điều này đảm bảo kiểm tra cả trường hợp thông thường và các trường hợp ranh giới dễ xảy ra lỗi.
* **Bao phủ yêu cầu**: Các test case kiểm tra tất cả các trường nhập liệu (tên, ngày, giờ) với các điều kiện hợp lệ và không hợp lệ, đảm bảo mọi khía cạnh của đặc tả được kiểm tra.
* **Phát hiện lỗi tiềm ẩn**: Việc kiểm tra các giá trị biên và dữ liệu không hợp lệ giúp phát hiện các lỗi logic hoặc xử lý ngoại lệ trong hệ thống, như kiểm tra ranh giới sai hoặc không xử lý được đầu vào không hợp lệ.