**Bài tập tổng hợp về Kỹ thuật Black-Box Testing**

**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**: Kỹ thuật kiểm thử tập trung vào chức năng của ứng dụng mà không cần biết cấu trúc hoặc mã nguồn bên trong. Tester chỉ quan tâm đến đầu vào và đầu ra, kiểm tra xem ứng dụng có hoạt động đúng như yêu cầu hay không.
* **White-box testing**: Kỹ thuật kiểm thử yêu cầu tester hiểu rõ mã nguồn và cấu trúc bên trong của ứng dụng. Tester thiết kế các test case dựa trên logic mã nguồn để đảm bảo mọi nhánh và dòng code được kiểm tra.
* **Kiểm thử dựa trên kinh nghiệm**: Dựa trên kinh nghiệm của tester, thường được sử dụng khi không có đủ tài liệu yêu cầu hoặc thời gian hạn chế. Tester dựa vào kiến thức, trực giác và kinh nghiệm để tìm lỗi mà không tuân theo quy trình kiểm thử chính thức.

**b. Ví dụ về black-box testing**

**Tình huống**: Kiểm thử tính năng đăng nhập của một website.

* **Mô tả**: Tester nhập các tổ hợp tên người dùng và mật khẩu vào form đăng nhập mà không cần biết cách hệ thống xử lý dữ liệu.
* **Ví dụ test case**:
  + Đầu vào: Tên người dùng hợp lệ, mật khẩu đúng → Kết quả mong đợi: Đăng nhập thành công.
  + Đầu vào: Tên người dùng không tồn tại, mật khẩu bất kỳ → Kết quả mong đợi: Hiển thị thông báo lỗi “Tên người dùng không tồn tại”.

**2. Equivalence Partitioning**

**a. Equivalence Partitioning là gì?**

Equivalence Partitioning (Phân vùng tương đương) là kỹ thuật chia tập hợp dữ liệu đầu vào thành các nhóm (phân vùng) mà ứng dụng được kỳ vọng sẽ xử lý giống nhau. Mỗi phân vùng đại diện cho một nhóm giá trị có cùng hành vi, giúp giảm số lượng test case mà vẫn đảm bảo kiểm tra đầy đủ các trường hợp.

**b. Ví dụ áp dụng Equivalence Partitioning**

**Trường hợp**: Kiểm thử trường nhập tuổi của người dùng (tuổi từ 18 đến 60 hợp lệ).

* **Phân vùng**:
  + Phân vùng 1: Tuổi < 18 (không hợp lệ).
  + Phân vùng 2: Tuổi từ 18 đến 60 (hợp lệ).
  + Phân vùng 3: Tuổi > 60 (không hợp lệ).
* **Test case**:
  + Tuổi = 15 (đại diện cho phân vùng 1) → Kết quả: Hệ thống báo lỗi “Tuổi phải từ 18 đến 60”.
  + Tuổi = 30 (đại diện cho phân vùng 2) → Kết quả: Hệ thống chấp nhận.
  + Tuổi = 65 (đại diện cho phân vùng 3) → Kết quả: Hệ thống báo lỗi “Tuổi phải từ 18 đến 60”.

**3. Boundary Value Analysis (BVA)**

**a. Boundary Value Analysis là gì?**

Boundary Value Analysis (Phân tích giá trị biên) là kỹ thuật kiểm thử tập trung vào các giá trị nằm ở ranh giới của các phân vùng đầu vào. Các lỗi thường xuất hiện ở biên của các phân vùng (ví dụ: giá trị tối thiểu, tối đa hoặc ngay ngoài phạm vi hợp lệ), do đó kiểm thử các giá trị biên rất hiệu quả trong việc phát hiện lỗi.

**b. Ví dụ sử dụng BVA**

**Trường hợp**: Kiểm thử trường nhập số lượng vé đặt trong ứng dụng đặt vé xem phim (từ 1 đến 10 vé).

* **Giá trị biên**: 0, 1, 2, 9, 10, 11.
* **Test case**:
  + Số lượng = 0 → Kết quả: Hệ thống báo lỗi “Số lượng vé phải từ 1 đến 10”.
  + Số lượng = 1 → Kết quả: Hệ thống chấp nhận.
  + Số lượng = 10 → Kết quả: Hệ thống chấp nhận.
  + Số lượng = 11 → Kết quả: Hệ thống báo lỗi “Số lượng vé phải từ 1 đến 10”.

**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 đầy đủ các chức năng, yêu cầu và trường hợp sử dụng của ứng dụng. Nó giúp:

* **Tăng độ bao phủ**: Đảm bảo mọi phân vùng đầu vào và giá trị biên được kiểm tra.
* **Phát hiện lỗi sớm**: Tập trung vào các trường hợp có khả năng xảy ra lỗi cao (như giá trị biên).
* **Tiết kiệm thời gian**: Giảm số lượng test case không cần thiết nhưng vẫn đảm bảo chất lượng kiểm thử.

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

* **Mô tả test case**: Mô tả ngắn gọn mục đích của test case.
* **Dữ liệu đầu vào**: Giá trị hoặc điều kiện đầu vào để thực hiện kiểm thử.
* **Kết quả mong đợi**: Kết quả dự kiến của hệ thống sau khi thực hiện test case.
* **Điều kiện tiên quyết**: Các yêu cầu hoặc trạng thái cần thiết trước khi thực hiện test case.
* **Bước thực hiện**: Các bước cụ thể để thực hiện kiểm thử.

**Phần 2: Bài tập thực hành**

**1. Áp dụng Equivalence Partitioning**

**Tình huống**: Kiểm thử form nhập liệu cho ứng dụng đặt hàng trực tuyến, trong đó người dùng nhập số lượng sản phẩm (từ 1 đến 100).

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

* **Phân vùng 1**: Số lượng < 1 (không hợp lệ).
* **Phân vùng 2**: Số lượng từ 1 đến 100 (hợp lệ).
* **Phân vùng 3**: Số lượng > 100 (không hợp lệ).

**b. Mô tả các nhóm và lý do**

* **Phân vùng 1 (Số lượng < 1)**: Đại diện cho các giá trị không hợp lệ (ví dụ: -1, 0). Hệ thống cần từ chối các giá trị này vì chúng nằm ngoài phạm vi cho phép.
* **Phân vùng 2 (Số lượng từ 1 đến 100)**: Đại diện cho các giá trị hợp lệ mà hệ thống phải chấp nhận (ví dụ: 1, 50, 100).
* **Phân vùng 3 (Số lượng > 100)**: Đại diện cho các giá trị không hợp lệ (ví dụ: 101, 1000). Hệ thống cần từ chối vì vượt quá giới hạn tối đa.

**2. Áp dụng Boundary Value Analysis**

**Tình huống**: Form nhập liệu số lượng sản phẩm từ 1 đến 100.

**a. Xác định các giá trị biên**

* **Giá trị biên**: 0, 1, 2, 99, 100, 101.
  + 0: Dưới giới hạn dưới.
  + 1: Giới hạn dưới.
  + 2: Ngay trên giới hạn dưới.
  + 99: Ngay dưới giới hạn trên.
  + 100: Giới hạn trên.
  + 101: Vượt giới hạn trên.

**b. Thiết kế test case**

1. **Test Case 1**
   * **Mô tả**: Kiểm tra giá trị biên dưới.
   * **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. **Test Case 2**
   * **Mô tả**: Kiểm tra giá trị tại giới hạn dưới.
   * **Dữ liệu đầu vào**: Số lượng = 1.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xử lý đơn hàng.
3. **Test Case 3**
   * **Mô tả**: Kiểm tra giá trị tại giới hạn trên.
   * **Dữ liệu đầu vào**: Số lượng = 100.
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xử lý đơn hàng.
4. **Test Case 4**
   * **Mô tả**: Kiểm tra giá trị vượt giới hạn trên.
   * **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**

**Tình huống**: Form nhập liệu cho ứng dụng 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).
* 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**
   * **Mô tả**: Kiểm tra tên bệnh nhân vượt quá giới hạn ký tự.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: Chuỗi 51 ký tự (ví dụ: “NguyenVanA...” với 51 ký tự).
     + Ngày hẹn: 05/07/2025 (hợp lệ).
     + Giờ hẹn: 10:00 (hợp lệ).
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Tên bệnh nhân không được vượt quá 50 ký tự”.
2. **Test Case 2**
   * **Mô tả**: Kiểm tra tên bệnh nhân hợp lệ.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A” (hợp lệ).
     + Ngày hẹn: 05/07/2025 (hợp lệ).
     + Giờ hẹn: 08:00 (giới hạn dưới).
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xác nhận lịch hẹn.
3. **Test Case 3**
   * **Mô tả**: Kiểm tra ngày hẹn trong quá khứ.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A” (hợp lệ).
     + Ngày hẹn: 03/07/2025 (quá khứ).
     + Giờ hẹn: 09:00 (hợp lệ).
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Ngày hẹn phải là ngày trong tương lai”.
4. **Test Case 4**
   * **Mô tả**: Kiểm tra giờ hẹn dưới giới hạn.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A” (hợp lệ).
     + Ngày hẹn: 05/07/2025 (hợp lệ).
     + Giờ hẹn: 07:59 (không hợp lệ).
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Giờ hẹn phải từ 08:00 đến 17:00”.
5. **Test Case 5**
   * **Mô tả**: Kiểm tra giờ hẹn tại giới hạn trên.
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “Nguyen Van A” (hợp lệ).
     + Ngày hẹn: 05/07/2025 (hợp lệ).
     + Giờ hẹn: 17:00 (hợp lệ).
   * **Kết quả mong đợi**: Hệ thống chấp nhận và xác nhận lịch hẹn.
6. **Test Case 6**
   * **Mô tả**: Kiểm tra phân vùng không hợp lệ cho tên bệnh nhân (rỗng).
   * **Dữ liệu đầu vào**:
     + Tên bệnh nhân: “” (rỗng).
     + Ngày hẹn: 05/07/2025 (hợp lệ).
     + Giờ hẹn: 10:00 (hợp lệ).
   * **Kết quả mong đợi**: Hệ thống hiển thị thông báo lỗi “Tên bệnh nhân không được để trống”.

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

* **Độ bao phủ phân vùng tương đương**: Các test case kiểm tra cả ba phân vùng (hợp lệ, không hợp lệ dưới giới hạn, không hợp lệ vượt giới hạn) cho từng trường (tên, ngày, giờ).
* **Độ bao phủ giá trị biên**: Các test case kiểm tra các giá trị biên (ví dụ: 08:00, 17:00 cho giờ hẹn; 51 ký tự cho tên bệnh nhân; ngày quá khứ và tương lai).
* **Đảm bảo lỗi thường gặp**: Các test case tập trung vào các trường hợp dễ gây lỗi như chuỗi rỗng, giá trị vượt giới hạn, hoặc ngày không hợp lệ, giúp phát hiện các vấn đề tiềm ẩn trong xử lý dữ liệu.
* **Kết hợp kỹ thuật**: Việc sử dụng cả Equivalence Partitioning và BVA đảm bảo kiểm tra toàn diện các trường hợp hợp lệ và không hợp lệ, tăng khả năng phát hiện lỗi.