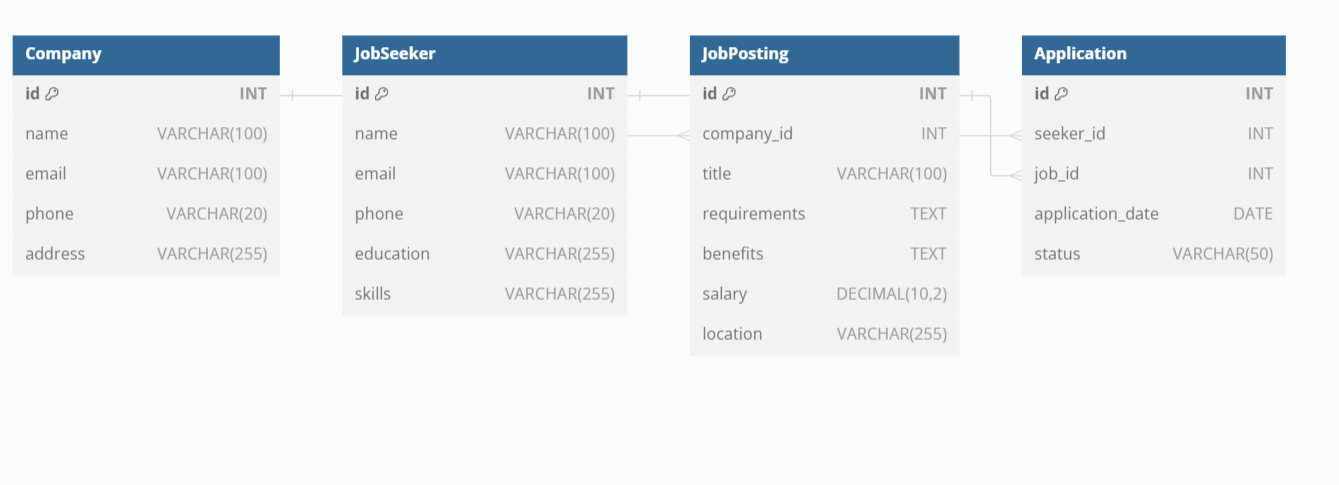
                                                              Bài kiểm tra

1.Thiết kế

Dưới đây là sơ đồ miêu tả



2.

**Nhánh master**:

* Mục đích: Nhánh chính được sử dụng để triển khai sản phẩm hoàn chỉnh.
* Thường chỉ có quyền truy cập cho các phiên bản ổn định của phần mềm.

1. **Nhánh develop**:
   * Mục đích: Nhánh phát triển, nơi tích hợp tất cả các tính năng đã hoàn thành và chuẩn bị cho phiên bản tiếp theo của sản phẩm.
   * Tất cả các tính năng được phát triển từ các nhánh tính năng (feature branches) đều được merge vào nhánh này.
2. **Nhánh tính năng (feature branches)**:
   * Mục đích: Được sử dụng để phát triển tính năng mới hoặc cải tiến vào sản phẩm.
   * Mỗi tính năng mới được phát triển trong một nhánh riêng biệt từ nhánh develop. Sau khi hoàn thành và kiểm tra, nhánh tính năng sẽ được merge vào nhánh develop.
3. **Nhánh sửa lỗi (bugfix branches)**:
   * Mục đích: Sửa các lỗi phát sinh trong quá trình phát triển hoặc sau khi sản phẩm đã được triển khai.
   * Nhánh này được tạo ra từ nhánh develop. Sau khi sửa lỗi, thay đổi sẽ được merge vào cả develop và master (nếu cần).
4. **Nhánh sửa lỗi cấp bách (hotfix branches)**:
   * Mục đích: Sửa các lỗi cấp bách hoặc khẩn cấp trên môi trường sản phẩm.
   * Sau khi sửa lỗi, thay đổi sẽ được merge vào cả develop và master ngay lập tức.

**Chuẩn bị và phát triển một mô-đun mới:**

1. **Tạo nhánh mới từ nhánh develop**:
   * Tạo một nhánh mới từ nhánh develop với tên phản ánh mô-đun bạn đang phát triển.
2. **Phát triển mô-đun mới**:
   * Phát triển mô-đun mới trong nhánh đã tạo, tuân thủ các quy trình và tiêu chuẩn mã nguồn của dự án.
3. **Kiểm thử mô-đun mới**:
   * Kiểm tra mô-đun mới để đảm bảo tính đúng đắn và hiệu suất của nó.
4. **Sửa lỗi và cải tiến (nếu cần)**:
   * Nếu phát hiện lỗi, sửa chúng và thực hiện các cải tiến cần thiết để đảm bảo tính ổn định và chất lượng của mã nguồn.
5. **Kiểm tra tích hợp**:
   * Kiểm tra tích hợp mô-đun mới vào nhánh develop để đảm bảo không có xung đột mã nguồn và tính tương thích với các tính năng khác.
6. **Triển khai mô-đun**:
   * Khi mô-đun đã hoàn thành và kiểm tra, merge nhánh mô-đun vào nhánh develop và thực hiện việc triển khai theo quy trình của dự án, có thể là merge vào nhánh master để triển khai lên môi trường sản phẩm.

3.

**1. Các loại chiến lược thử nghiệm:**

1. **Thử nghiệm hộp đen (Black-box testing)**: Kiểm tra dựa trên yêu cầu và không cần biết cấu trúc nội bộ của ứng dụng.
2. **Thử nghiệm hộp trắng (White-box testing)**: Kiểm tra dựa trên cấu trúc nội bộ của ứng dụng và mã nguồn.
3. **Thử nghiệm hộp xám (Grey-box testing)**: Kết hợp cả hai phương pháp trên, kiểm tra từ cả hai góc độ nội bộ và bề ngoài của ứng dụng.

**2. Kiểm tra đơn vị được sử dụng để làm gì?**

Kiểm thử đơn vị (Unit testing) được sử dụng để kiểm tra từng phần riêng lẻ của mã nguồn (có thể là hàm, lớp, hoặc mô-đun) để đảm bảo tính đúng đắn và hoạt động của từng đơn vị logic trong ứng dụng.

**3. Kiểm thử tích hợp được sử dụng để làm gì?**

Kiểm thử tích hợp (Integration testing) được sử dụng để kiểm tra các tính năng hoặc các thành phần của hệ thống khi được kết hợp và tích hợp lại với nhau. Mục tiêu là đảm bảo các phần của hệ thống hoạt động một cách mượt mà và tương tác đúng đắn với nhau.

**4. Bước kiểm tra mô-đun mới từ khi bắt đầu mô-đun đến khi phát hành mô-đun:**

1. **Phát triển mô-đun**: Phát triển mô-đun mới với quy trình và yêu cầu đã xác định.
2. **Kiểm thử đơn vị**: Tạo các ca kiểm thử cho từng phần riêng lẻ của mô-đun để đảm bảo tính đúng đắn của chúng.
3. **Kiểm thử tích hợp**: Kiểm tra tích hợp mô-đun mới vào hệ thống để đảm bảo tính tương thích và không gây ra lỗi cho các tính năng khác.
4. **Kiểm tra chấp nhận (Acceptance testing)**: Đưa mô-đun mới vào môi trường thử nghiệm cuối cùng và kiểm tra bởi người dùng cuối hoặc người quản lý dự án để xác nhận rằng mô-đun hoạt động đúng theo yêu cầu và mong đợi.