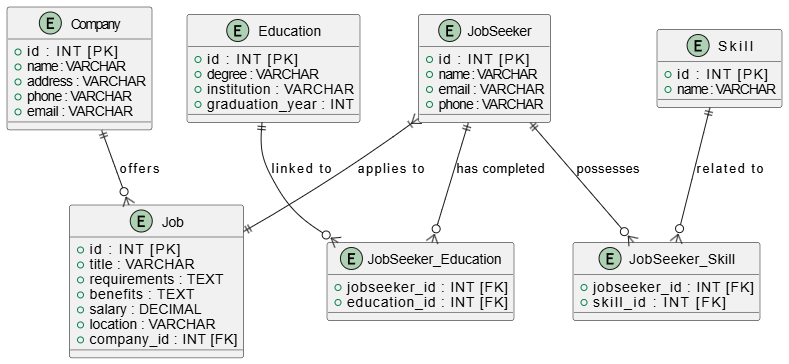
**Computing Project**

**I. Designing**



**II. Project process**

- Trong một dự án phát triển phần mềm điển hình, Quản lý Kỹ thuật có thể sử dụng một số nhánh trong Git để quản lý phát triển hiệu quả. Các nhánh phổ biến bao gồm:

1. **Nhánh Chính (Main hoặc Master)**:
   * **Mục đích**: Nhánh này chứa mã nguồn ổn định và đã sẵn sàng cho sản xuất. Tất cả các phiên bản cuối cùng đều được gán nhãn từ nhánh này.
2. **Nhánh Phát triển (Development)**:
   * **Mục đích**: Nhánh này được sử dụng cho việc phát triển đang diễn ra. Nó phục vụ như một nhánh tích hợp cho các tính năng và bản sửa lỗi đang được phát triển.
3. **Nhánh Tính năng (Feature Branches)**:
   * **Mục đích**: Các nhánh này được tạo ra cho các tính năng hoặc mô-đun mới. Chúng tách ra từ nhánh phát triển và được hợp nhất lại khi tính năng hoàn tất.
4. **Nhánh Sửa lỗi (Bugfix Branches)**:
   * **Mục đích**: Tương tự như nhánh tính năng nhưng dành riêng cho việc sửa lỗi. Chúng cho phép lập trình viên làm việc trên các bản sửa lỗi mà không làm gián đoạn quá trình phát triển đang diễn ra.
5. **Nhánh Phát hành (Release Branches)**:
   * **Mục đích**: Được sử dụng để chuẩn bị cho một phiên bản mới. Nó cho phép thực hiện các điều chỉnh cuối cùng, sửa lỗi và chuẩn bị cho việc triển khai trong khi phát triển vẫn có thể tiếp tục trong nhánh phát triển.
6. **Nhánh Sửa lỗi khẩn cấp (Hotfix Branches)**:
   * **Mục đích**: Dùng cho các bản sửa lỗi khẩn cấp đối với mã nguồn sản xuất. Chúng tách ra từ nhánh chính và được hợp nhất lại vào cả nhánh chính và nhánh phát triển.

- Các bước để phát triển một module mới :

1. **Thu thập yêu cầu**:
   * Thu thập và tài liệu hóa các yêu cầu cho module mới, bao gồm các thông số chức năng và phi chức năng.
2. **Thiết kế**:
   * Tạo một tài liệu thiết kế phác thảo kiến trúc, công nghệ và các thành phần của module.
3. **Tạo nhánh**:
   * Tạo một nhánh tính năng mới từ nhánh phát triển.
4. **Phát triển**:
   * Viết mã theo các tiêu chuẩn và quy tắc lập trình tốt nhất.
   * Thực hiện các tính năng của module theo tài liệu thiết kế.
5. **Xem xét mã**:
   * Thực hiện xem xét mã với đồng nghiệp để đảm bảo chất lượng và tuân thủ các tiêu chuẩn.
6. **Kiểm thử**:
   * Viết các bài kiểm thử đơn vị cho các thành phần riêng lẻ và kiểm thử tích hợp cho toàn bộ module.
7. **Hợp nhất vào Phát triển**:
   * Sau khi phát triển và kiểm thử hoàn tất, hợp nhất nhánh tính năng lại vào nhánh phát triển.
8. **Kiểm thử chấp nhận người dùng (UAT)**:
   * Thực hiện kiểm thử UAT để đảm bảo module đáp ứng được mong đợi của người dùng.
9. **Chuẩn bị Phát hành**:
   * Tạo một nhánh phát hành từ nhánh phát triển và thực hiện kiểm thử và sửa lỗi cuối cùng.
10. **Triển khai**:
    * Triển khai module mới vào môi trường sản xuất.
11. **Giám sát sau Phát hành**:
    * Theo dõi module để phát hiện bất kỳ vấn đề nào sau khi phát hành và giải quyết khi cần thiết.

**III. Testing**

- Các loại Chiến lược Kiểm thử:

1. **Kiểm thử Đơn vị (Unit Testing)**: Kiểm thử các thành phần hoặc chức năng riêng lẻ một cách tách biệt.
2. **Kiểm thử Tích hợp (Integration Testing)**: Kiểm thử các tương tác giữa các thành phần hoặc hệ thống tích hợp.
3. **Kiểm thử Hệ thống (System Testing)**: Kiểm thử toàn bộ hệ thống phần mềm đã được tích hợp.
4. **Kiểm thử Chấp nhận (Acceptance Testing)**: Xác thực hệ thống theo yêu cầu của người dùng.
5. **Kiểm thử Hiệu suất (Performance Testing)**: Đánh giá khả năng phản hồi, tốc độ, khả năng mở rộng và tính ổn định của ứng dụng.
6. **Kiểm thử Bảo mật (Security Testing)**: Xác định các lỗ hổng và lỗi bảo mật trong ứng dụng.
7. **Kiểm thử Định kỳ (Regression Testing)**: Đảm bảo rằng các thay đổi mã mới không làm hỏng các chức năng hiện có.
8. **Kiểm thử Khói (Smoke Testing)**: Kiểm thử sơ bộ để kiểm tra các chức năng cơ bản của ứng dụng.

- Kiểm thử đơn vị được sử dụng để xác nhận rằng các thành phần riêng lẻ của phần mềm (như hàm hoặc lớp) hoạt động như mong đợi. Nó giúp phát hiện lỗi sớm trong quá trình phát triển và đảm bảo rằng các thay đổi trong mã không làm hỏng các chức năng hiện có.

- Kiểm thử tích hợp được sử dụng để đánh giá các tương tác giữa các thành phần hoặc hệ thống khác nhau. Nó đảm bảo rằng các phần đã tích hợp của ứng dụng hoạt động cùng nhau như mong đợi. Loại kiểm thử này có thể phát hiện các vấn đề liên quan đến luồng dữ liệu và giao tiếp giữa các module.

- Các bước để Kiểm thử một module mới**:**

1. **Hiểu Yêu cầu**:
   * Xem xét các thông số và yêu cầu của mô-đun mới để hiểu những gì cần được kiểm thử.
2. **Phát triển Tình huống Kiểm thử**:
   * Tạo các tình huống kiểm thử chi tiết bao gồm tất cả các khía cạnh chức năng và phi chức năng của mô-đun.
3. **Thiết lập Môi trường Kiểm thử**:
   * Chuẩn bị môi trường kiểm thử, đảm bảo tất cả các cấu hình, công cụ và phụ thuộc cần thiết đã được thiết lập.
4. **Thực hiện Kiểm thử**:
   * Chạy các bài kiểm thử dựa trên các tình huống đã phát triển, ghi lại kết quả và bất kỳ vấn đề nào phát sinh.
5. **Báo cáo Lỗi**:
   * Báo cáo bất kỳ lỗi hoặc vấn đề nào gặp phải trong quá trình kiểm thử cho nhóm phát triển để giải quyết.
6. **Kiểm thử lại**:
   * Khi các lỗi được sửa, kiểm thử lại mô-đun để đảm bảo rằng các vấn đề đã được khắc phục và không có lỗi mới nào được giới thiệu.
7. **Kiểm thử Chấp nhận Người dùng (UAT)**:
   * Tham gia người dùng cuối để xác thực mô-đun theo yêu cầu và mong đợi của họ.
8. **Chuẩn bị Phát hành**:
   * Xác minh rằng tất cả các bài kiểm thử đã được thực hiện và chuẩn bị mô-đun để triển khai.
9. **Kiểm thử Sau Phát hành**:
   * Tiến hành theo dõi và xác minh sau khi mô-đun được phát hành để đảm bảo rằng nó hoạt động như mong đợi trong môi trường sản xuất.