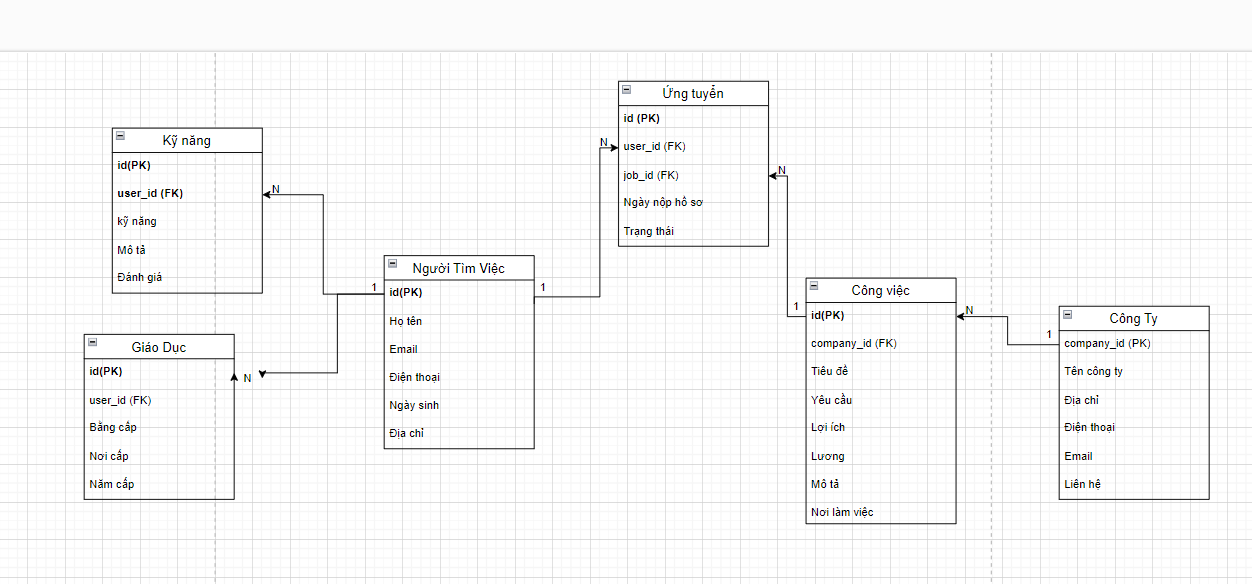
**Câu 1: Sơ đồ mối quan hệ thực thể**



* Một **Người Tìm Việc** có thể có nhiều **Giáo Dục** và **Kỹ Năng**.
* Một **Công Ty** có thể đăng nhiều **Công Việc**.
* Một **Người Tìm Việc** có thể ứng tuyển vào nhiều **Công Việc** thông qua **Ứng Tuyển**.
* Một **Công Việc** có thể nhận được nhiều **Ứng Tuyển** từ các **Người Tìm Việc** khác nhau.

**Câu 2:**

Trong một dự án phát triển phần mềm, người Quản lý Kỹ thuật thường sử dụng các nhánh sau trong GIT:

* **main/master**: Đây là nhánh chính chứa mã nguồn ổn định và đã được kiểm thử kỹ lưỡng. Nó là phiên bản sản phẩm cuối cùng đang hoạt động.
* **Develop**: Nhánh này chứa mã nguồn đang được phát triển và kiểm thử. Các tính năng mới và sửa lỗi được tích hợp vào đây trước khi được đưa vào nhánh main/master.
* **feature**: Mỗi nhánh feature được tạo ra để phát triển một tính năng mới. Sau khi hoàn thành và kiểm thử, nhánh này sẽ được hợp nhất (merge) vào nhánh develop.
* **release**: Nhánh này được tạo ra từ nhánh develop khi dự án đã sẵn sàng để phát hành. Nó được sử dụng để kiểm thử và sửa lỗi cuối cùng trước khi hợp nhất vào nhánh main/master.

Các Bước Chuẩn Bị Phát Triển Một Module Mới

1. **Phân Tích Yêu Cầu**: Thu thập và phân tích các yêu cầu chức năng và phi chức năng của module mới.
2. **Lập Kế Hoạch**: Xác định các công việc cần làm, phân công nhiệm vụ, và thiết lập mốc thời gian hoàn thành.
3. **Tạo Nhánh Feature**: Tạo một nhánh feature mới từ nhánh develop để bắt đầu phát triển module.
4. **Thiết Kế**: Thiết kế kiến trúc, cơ sở dữ liệu, và giao diện người dùng cho module.
5. **Phát Triển**: Tiến hành lập trình và thực hiện các đơn vị kiểm thử (unit test) cho từng phần của module.
6. **Kiểm Thử**: Thực hiện kiểm thử tích hợp và kiểm thử hệ thống để đảm bảo module hoạt động đúng như mong đợi.
7. **Kiểm Tra Code**: Thực hiện các buổi kiểm tra code (code review) để đảm bảo chất lượng mã nguồn.
8. **Tích Hợp**: Hợp nhất nhánh feature vào nhánh develop sau khi hoàn tất kiểm thử và kiểm tra code.
9. **Chuẩn Bị Phát Hành**: Tạo một nhánh release từ nhánh develop để kiểm thử cuối cùng và sửa lỗi (nếu cần).
10. **Phát Hành**: Hợp nhất nhánh release vào nhánh main/master và triển khai module mới vào môi trường sản xuất.
11. **Bảo Trì**: Theo dõi và sửa lỗi nếu có trong quá trình sử dụng module mới.

**Câu 3: Các Chiến Lược Kiểm Thử**

1. **Unit Testing**: Được sử dụng để kiểm thử các phần nhỏ nhất của mã nguồn, thường là các hàm hoặc phương thức riêng lẻ, để đảm bảo rằng chúng hoạt động đúng như mong đợi. Giúp phát hiện lỗi sớm trong quá trình phát triển, dễ dàng xác định và sửa lỗi trong các đơn vị nhỏ.
2. **Integration Testing** : Được sử dụng để kiểm thử các mô-đun hoặc đơn vị đã được tích hợp với nhau để đảm bảo rằng chúng hoạt động cùng nhau một cách chính xác.Giúp phát hiện lỗi liên quan đến sự tương tác giữa các mô-đun hoặc thành phần của hệ thống.
3. **System Testing**: Kiểm thử hệ thống tổng thể để đảm bảo rằng nó đáp ứng đầy đủ các yêu cầu đã đề ra. Đảm bảo toàn bộ hệ thống hoạt động chính xác và tích hợp tốt các phần khác nhau.
4. **Acceptance Testing**: Được thực hiện bởi người dùng cuối hoặc khách hàng để đảm bảo rằng hệ thống đáp ứng các yêu cầu nghiệp vụ và sẵn sàng để triển khai. Xác nhận rằng hệ thống phù hợp với nhu cầu thực tế của người sử dụng.
5. **Regression Testing**: Được sử dụng để đảm bảo rằng các chức năng cũ vẫn hoạt động đúng sau khi có sự thay đổi hoặc cập nhật mã nguồn. Giúp phát hiện và ngăn chặn các lỗi mới được giới thiệu vào hệ thống do các thay đổi gần đây.
6. **Performance Testing**: Kiểm thử hệ thống dưới các điều kiện tải cao để đảm bảo rằng nó hoạt động ổn định và hiệu quả. Đảm bảo hệ thống đáp ứng yêu cầu về tốc độ và hiệu suất khi có nhiều người dùng hoặc tải cao.

**Các Bước Kiểm Thử Một Module Mới**

1. **Hiểu Yêu Cầu**: Thu thập và phân tích các yêu cầu của module để hiểu rõ các chức năng và yêu cầu cần kiểm thử.
2. **Lập Kế Hoạch Kiểm Thử**: Tạo kế hoạch kiểm thử chi tiết, bao gồm xác định các chiến lược kiểm thử, phạm vi kiểm thử, tài nguyên cần thiết, và lịch trình kiểm thử.
3. **Viết Test Case**:Tạo các kịch bản kiểm thử (test case) chi tiết dựa trên các yêu cầu và chức năng của module.
4. **Thiết Lập Môi Trường Kiểm Thử**: Chuẩn bị môi trường kiểm thử, bao gồm cài đặt phần mềm, cấu hình hệ thống, và chuẩn bị dữ liệu kiểm thử.
5. **Thực Hiện Unit Testing**: Kiểm thử từng đơn vị của module để đảm bảo rằng chúng hoạt động đúng như mong đợi.
6. **Thực Hiện Integration Testing**: Kiểm thử sự tích hợp giữa các đơn vị và mô-đun khác nhau để đảm bảo rằng chúng hoạt động cùng nhau một cách chính xác.
7. **Thực Hiện System Testing**: Kiểm thử toàn bộ module trong ngữ cảnh của hệ thống tổng thể để đảm bảo rằng nó hoạt động đúng và tương thích với các phần khác của hệ thống.
8. **Thực Hiện Regression Testing**: Kiểm thử lại các chức năng cũ của hệ thống để đảm bảo rằng module mới không gây ra lỗi cho các chức năng hiện có.
9. **Thực Hiện Acceptance Testing**: Thực hiện kiểm thử chấp nhận để đảm bảo rằng module đáp ứng yêu cầu nghiệp vụ và sẵn sàng để triển khai.
10. **Báo Cáo và Sửa Lỗi**: Ghi nhận và báo cáo các lỗi phát hiện được trong quá trình kiểm thử, sau đó phối hợp với đội phát triển để sửa lỗi.
11. **Kiểm Thử Lại**: Kiểm thử lại module sau khi các lỗi đã được sửa để đảm bảo rằng chúng đã được khắc phục hoàn toàn.
12. **Chuẩn Bị Phát Hành**: Thực hiện các kiểm thử cuối cùng và chuẩn bị tài liệu để phát hành module mới.
13. **Phát Hành**: Triển khai module mới vào môi trường sản xuất và theo dõi để đảm bảo rằng nó hoạt động ổn định.