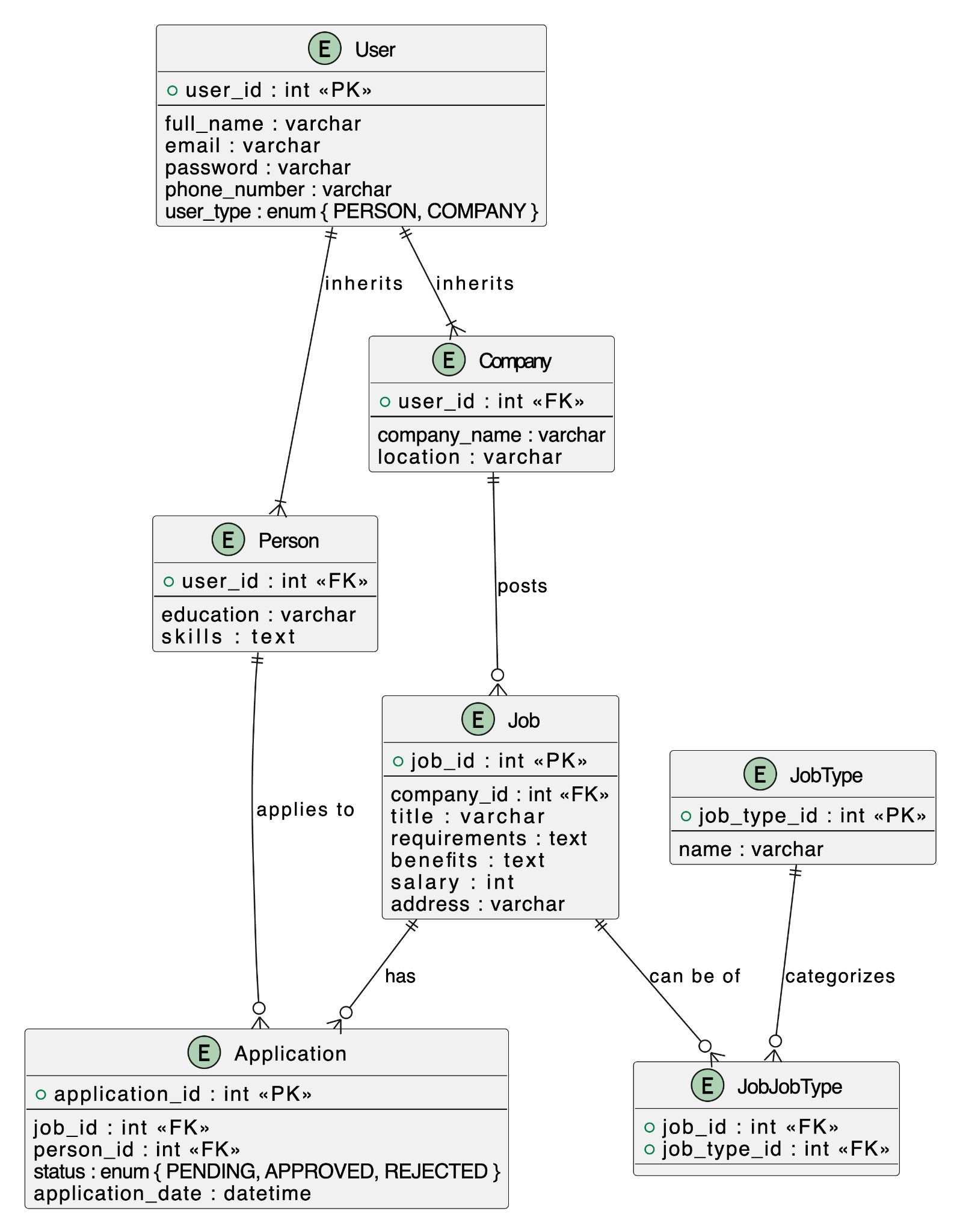
**Bài thi môn** **Computing Project**

1. Designing



2. Project process

#### **a. Branches in GIT**

Trong một dự án phần mềm, Technical Manager thường sử dụng các nhánh GIT sau:

1. **Main (hoặc Master)**:
   * Đây là nhánh chính, chứa mã nguồn hoàn chỉnh và ổn định của dự án, thường được sử dụng để phát hành phiên bản sản phẩm chính thức.
2. **Develop**:
   * Nhánh phát triển, nơi các tính năng mới và các bản sửa lỗi được hợp nhất. Đây là nhánh chính mà các lập trình viên làm việc và kiểm thử trước khi được hợp nhất vào nhánh main.
3. **Feature**:
   * Nhánh tính năng, được tạo cho từng tính năng hoặc module mới. Mỗi lập trình viên làm việc trên nhánh này để phát triển một tính năng cụ thể mà không ảnh hưởng đến mã nguồn chính.
4. **Release**:
   * Nhánh phát hành, được tạo từ nhánh develop khi chuẩn bị phát hành phiên bản mới. Nhánh này chủ yếu dùng để sửa lỗi và tối ưu hóa trước khi hợp nhất vào main.
5. **Hotfix**:
   * Nhánh sửa lỗi khẩn cấp, được tạo từ nhánh main khi cần sửa lỗi nghiêm trọng trong phiên bản đang chạy.

#### **b. Các bước phát triển một module mới từ khi bắt đầu đến khi phát hành**

1. **Xác định yêu cầu**:
   * Tìm hiểu kỹ yêu cầu của module mới, xác định tính năng và cách module sẽ tương tác với các thành phần khác.
2. **Thiết kế kiến trúc**:
   * Xây dựng kiến trúc và thiết kế chi tiết cho module mới, bao gồm sơ đồ ERD, API (nếu cần), và các thành phần UI.
3. **Tạo nhánh Feature**:
   * Tạo nhánh feature/new-module từ nhánh develop để phát triển module.
4. **Lập trình và kiểm thử đơn vị (Unit Testing)**:
   * Phát triển từng phần của module theo yêu cầu đã xác định và viết các bài kiểm thử đơn vị để đảm bảo mã nguồn hoạt động chính xác.
5. **Kiểm thử tích hợp (Integration Testing)**:
   * Tích hợp module vào hệ thống và kiểm thử để đảm bảo tương thích với các module khác.
6. **Kiểm thử hệ thống và kiểm thử người dùng (System & User Acceptance Testing)**:
   * Đảm bảo module hoạt động đúng theo yêu cầu của người dùng và không có lỗi khi tích hợp vào hệ thống lớn.
7. **Tạo nhánh Release và sửa lỗi**:
   * Sau khi kiểm thử thành công, hợp nhất nhánh feature/new-module vào nhánh release và sửa các lỗi nhỏ nếu có.
8. **Phát hành**:
   * Hợp nhất release vào main để phát hành chính thức. Nhánh release cũng cần được hợp nhất ngược vào develop để cập nhật mã nguồn mới nhất.

3. Testing

#### **a. Các loại chiến lược kiểm thử**

1. **Unit Testing (Kiểm thử đơn vị)**: Kiểm thử từng đơn vị nhỏ nhất của mã nguồn (như hàm, lớp) để đảm bảo chúng hoạt động đúng.
2. **Integration Testing (Kiểm thử tích hợp)**: Kiểm tra tích hợp giữa các module để đảm bảo chúng tương tác tốt và đúng cách.
3. **System Testing (Kiểm thử hệ thống)**: Kiểm thử toàn bộ hệ thống để đảm bảo tất cả thành phần hoạt động chính xác và tuân theo yêu cầu.
4. **Acceptance Testing (Kiểm thử chấp nhận)**: Kiểm thử từ phía người dùng cuối để đảm bảo hệ thống đáp ứng yêu cầu và mong đợi.

#### **b. Unit Testing dùng để làm gì?**

* Được sử dụng để kiểm tra tính chính xác của từng đơn vị mã nguồn nhỏ nhất (thường là hàm hoặc phương thức) trong quá trình phát triển. Mục tiêu của kiểm thử đơn vị là đảm bảo từng đơn vị mã nguồn hoạt động đúng và không có lỗi logic.

#### **c. Integration Testing dùng để làm gì?**

* Được sử dụng để kiểm tra tích hợp giữa các module, giúp xác định xem các module này có thể hoạt động cùng nhau một cách liền mạch không. Kiểm thử tích hợp đặc biệt quan trọng trong việc phát hiện lỗi xảy ra khi các module có giao diện không tương thích hoặc lỗi khi truyền dữ liệu.

#### **d. Các bước kiểm thử module mới từ khi bắt đầu đến khi phát hành**

1. **Hiểu yêu cầu kiểm thử**:
   * Đọc tài liệu yêu cầu, tìm hiểu cách thức hoạt động và tương tác của module mới với hệ thống.
2. **Lập kế hoạch kiểm thử và viết test case**:
   * Xác định các trường hợp kiểm thử dựa trên yêu cầu, bao gồm kiểm thử đơn vị, tích hợp, và kiểm thử chức năng.
3. **Thiết lập môi trường kiểm thử**:
   * Cài đặt phiên bản mới nhất của hệ thống và cấu hình để chạy kiểm thử trên module mới.
4. **Chạy Unit Testing**:
   * Kiểm thử từng đơn vị mã nguồn trong module để đảm bảo mọi chức năng hoạt động chính xác.
5. **Chạy Integration Testing**:
   * Kiểm thử tích hợp module mới với các module khác trong hệ thống để đảm bảo tương thích.
6. **System Testing**:
   * Kiểm tra toàn bộ hệ thống với module mới để đảm bảo không có lỗi nghiêm trọng hoặc lỗi tương tác.
7. **Acceptance Testing**:
   * Đảm bảo module đáp ứng yêu cầu của người dùng thông qua kiểm thử chức năng và trải nghiệm người dùng.
8. **Sửa lỗi và phát hành**:
   * Nếu phát hiện lỗi, báo cáo lỗi và phối hợp với các lập trình viên để sửa lỗi. Sau khi kiểm thử lại, chuẩn bị cho phát hành module.