**PC TEST**

**Full name: Ngô Minh Khôi**

**1. Designing**

**Entities:**

* **Person**
  + ID (Primary Key)
  + Name
  + Email
  + Password
  + Education
  + Skills
  + Resume (optional)
  + Profile Created At
* **Company**
  + ID (Primary Key)
  + Name
  + Industry
  + Email
  + Password
  + Address
  + Company Description
  + Profile Created At
* **Job**
  + ID (Primary Key)
  + Title
  + Requirements
  + Benefits
  + Salary
  + Address
  + Posted By (Foreign Key referencing Company ID)
  + Posted At
* **Application**
  + ID (Primary Key)
  + Person ID (Foreign Key referencing Person ID)
  + Job ID (Foreign Key referencing Job ID)
  + Application Date
  + Status (e.g., Applied, Interviewing, Rejected, Hired)

**Entity Relationship Diagram:**

A black background with white circles and text

Description automatically generated

**2. Project Process**

**GIT Branches**

A Technical Manager should typically use the following branches in GIT:

* **Main**: The stable version of the project, always ready for release.
* **Develop**: The integration branch for features completed and ready for the next release.
* **Feature**: Each new feature has its own branch, named feature/feature-name.
* **Release**: A branch created from develop to prepare for a release, allowing for final testing.
* **Hotfix**: A branch created from main to quickly address bugs in the production version.

**Steps for Developing a New Module**

1. **Requirement Gathering**: Understand the requirements and scope of the new module.
2. **Design**: Create design documents and architecture for the new module.
3. **Branch Creation**: Create a new feature branch (feature/module-name) in GIT.
4. **Development**: Write the code for the module in the feature branch.
5. **Unit Testing**: Write and run unit tests for the new code.
6. **Code Review**: Submit the code for review by peers.
7. **Integration**: Merge the feature branch into the develop branch after review and testing.
8. **Integration Testing**: Test the integration of the new module with existing modules.
9. **Release Preparation**: Merge develop into a release branch and conduct final testing.
10. **Deployment**: Merge release into main and deploy the module to production.
11. **Monitoring**: Monitor the module in production for any issues.

**3. Testing**

**Types of Testing Strategies**

* **Unit Testing**: Testing individual units or components of the software.
* **Integration Testing**: Testing the interaction between integrated units/components.
* **System Testing**: Testing the complete and integrated software system.
* **Acceptance Testing**: Testing the software against user requirements and business processes.

**Unit Testing**

* **Purpose**: To validate that individual units of code (functions, methods, classes) work as expected.
* **Scope**: Tests are isolated to specific sections of code, ensuring that each piece works independently.

**Integration Testing**

* **Purpose**: To validate that different modules or services used by your application work well together.
* **Scope**: Tests focus on the interfaces and interaction between integrated units/modules.

**Steps for Testing a New Module**

1. **Requirement Analysis**: Understand the requirements and functionality of the new module.
2. **Test Planning**: Create a test plan outlining the scope, approach, resources, and schedule.
3. **Test Case Design**: Write test cases and scripts for different scenarios.
4. **Environment Setup**: Prepare the test environment with necessary configurations.
5. **Unit Testing**: Ensure individual components of the module work correctly.
6. **Integration Testing**: Test interactions between the new module and existing modules.
7. **System Testing**: Test the complete system with the new module integrated.
8. **Regression Testing**: Ensure that new changes haven't adversely affected existing functionality.
9. **User Acceptance Testing (UAT)**: Conduct testing with end-users to validate the module meets their needs.
10. **Bug Fixing and Retesting**: Address any defects found and retest as necessary.
11. **Final Sign-Off**: Get final approval for release from stakeholders.
12. **Deployment**: Move the module to the production environment.
13. **Post-Deployment Testing**: Verify the module works correctly in the live environment.