**1. Designing: Entity Relationship Diagram (ERD)**

**Here's an Entity Relationship Diagram (ERD) for a job search website:**

**Entities:**

**• Person**

ID (Primary Key)

Name

Email

Password

Education

Skills

Profile Created At

**• Company**

ID (Primary Key)

Name

Industry

Email

Password

Address

Company Description

Profile Created At

**• Job**

ID (Primary Key)

Requirements

Benefits

Salary

Address

Posted By (Foreign Key referencing Company ID)

Posted At

**• Application**

ID (Primary Key)

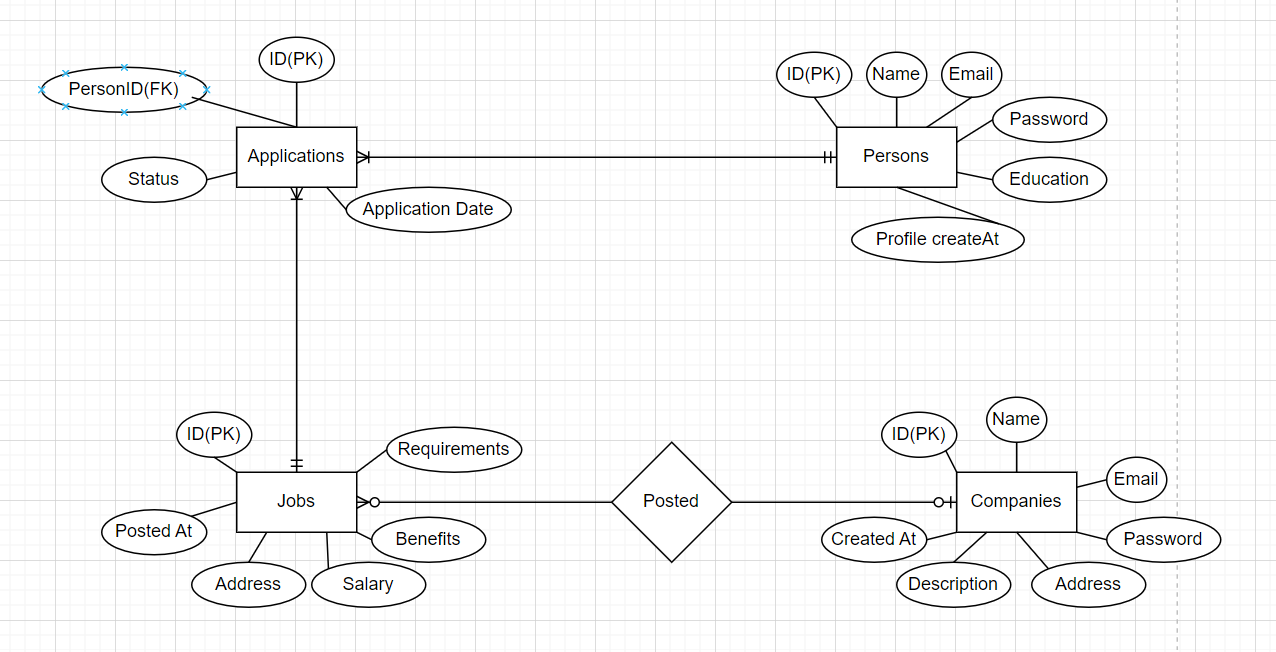
Person ID (Foreign Key referencing Person ID)

Job ID (Foreign Key referencing Job ID)

Application Date

**Status (e.g., Applied, Interviewing, Rejected, Hired)**

**Relationships:**



**2. Project Process: Git Branches and Development Steps**

**Git Branches**

1. **Main/Master Branch**: The stable version of the project that is ready for release.
2. **Develop Branch**: The integration branch for features being developed.
3. **Feature Branches**: Branches created from develop for each new feature/module.
4. **Release Branches**: Branches created from develop when preparing for a release.
5. **Hotfix Branches**: Branches created from main to quickly address bugs in production.

**Development Steps for a New Module**

1. **Requirement Analysis**: Gather and analyze requirements for the new module.
2. **Create Feature Branch**: Create a feature branch from develop.
3. **Design**: Design the architecture and components of the module.
4. **Development**: Implement the module in the feature branch.
5. **Unit Testing**: Write and run unit tests to ensure code correctness.
6. **Integration Testing**: Integrate the module with other parts of the system and perform integration testing.
7. **Code Review**: Conduct code reviews to ensure code quality and standards.
8. **Merge to Develop**: Merge the feature branch back into develop.
9. **System Testing**: Perform system testing on the develop branch.
10. **Create Release Branch**: If preparing for a release, create a release branch.
11. **User Acceptance Testing (UAT)**: Perform UAT to get feedback from end-users.
12. **Bug Fixing**: Address any bugs found during testing.
13. **Merge to Main**: Merge the release branch into main and deploy.

**3. Testing Strategies and Steps**

**Types of Testing Strategies**

1. **Unit Testing**: Testing individual components or units of code in isolation.
2. **Integration Testing**: Testing combined parts of an application to ensure they work together.
3. **System Testing**: Testing the complete and integrated application as a whole.
4. **Acceptance Testing**: Testing to ensure the system meets business requirements and is acceptable to the end-user.
5. **Regression Testing**: Testing to ensure that new changes have not adversely affected existing functionality.
6. **Performance Testing**: Testing to determine the system’s performance under load.
7. **Security Testing**: Testing to identify vulnerabilities and ensure the system is secure.

**Purpose of Unit Testing**

Unit testing is used to verify that individual units or components of a software application perform as intended. This helps catch bugs early in the development process and ensures that each part of the codebase is reliable and functions correctly.

**Purpose of Integration Testing**

Integration testing is used to test the interfaces between components or systems to ensure they work together correctly. It helps identify issues in the interaction between integrated units, ensuring that different parts of the application function together as a cohesive system.

**Testing Steps for a New Module**

1. **Understand Requirements**: Review the requirements and specifications for the new module.
2. **Create Test Plan**: Develop a comprehensive test plan outlining the testing strategy, scope, resources, schedule, and deliverables.
3. **Write Test Cases**: Create detailed test cases based on the requirements and design documents.
4. **Prepare Test Environment**: Set up the testing environment, including test data, hardware, software, and network configurations.
5. **Unit Testing**: Execute unit tests to ensure individual components work as expected.
6. **Integration Testing**: Test the interactions between integrated components.
7. **System Testing**: Perform end-to-end testing on the complete system.
8. **Regression Testing**: Re-run previously passed tests to ensure new changes have not introduced new bugs.
9. **User Acceptance Testing (UAT)**: Conduct UAT to ensure the module meets business requirements and is user-friendly.
10. **Log and Fix Defects**: Document any defects found, prioritize them, and work with the development team to resolve them.
11. **Retesting and Regression Testing**: Re-test fixed defects and perform regression testing to ensure no new issues are introduced.
12. **Final Review and Sign-off**: Conduct a final review of the test results, obtain necessary sign-offs, and prepare the module for release.
13. **Deployment**: Deploy the module to the production environment.
14. **Post-Deployment Testing**: Perform smoke testing in the production environment to ensure everything works correctly.