**1. Designing**

**Task:** Design a website for job seekers and companies looking for employees.

* **Job Seekers:** Must provide information on their education and skills.
* **Companies:** Must provide details about the job positions they are offering, including title, requirements, benefits, salary, and address.

**Deliverable:** Entity Relationship Diagram (ERD).

**Entity Relationship Diagram (ERD)**

* **Entities:**
  + **Person**
    - Attributes: ID, Name, Education, Skills, Contact Information
  + **Company**
    - Attributes: ID, Name, Address, Contact Information
  + **Job**
    - Attributes: ID, Title, Requirements, Benefits, Salary, Address, CompanyID (foreign key referencing Company)
  + **Application**
    - Attributes: ID, JobID (foreign key referencing Job), PersonID (foreign key referencing Person), ApplicationDate, Status
* **Relationships:**
  + **Person -< Application >- Job** (Many-to-Many)
  + **Company -< Job** (One-to-Many)

A diagram of a company

Description automatically generated

**2. Project Process**

**Task 1:** Number and purpose of branches in GIT used by the Technical Manager.

* **Branches in GIT:**
  + **Main/Master:** The stable version of the project.
  + **Develop:** For integrating features and preparing for release.
  + **Feature Branches:** For developing new features (one branch per feature).
  + **Release Branches:** For preparing a release, allowing for last-minute fixes.
  + **Hotfix Branches:** For quick patches to the production version.

**Task 2:** Steps to develop a new module.

* **Steps:**
  1. **Requirement Analysis:** Understand and document the module requirements.
  2. **Design:** Create design specifications and diagrams.
  3. **Development:** Code the module in a feature branch.
  4. **Testing:** Perform unit and integration testing.
  5. **Code Review:** Peer review of the code.
  6. **Merge:** Integrate the feature branch into the develop branch.
  7. **Release Preparation:** Move the develop branch changes to the release branch.
  8. **Deployment:** Deploy the module from the release branch to production.
  9. **Maintenance:** Monitor and fix any issues post-release.

**3. Testing**

**Task 1:** Types of testing strategies.

* **Testing Strategies:**
  + **Unit Testing:** Testing individual components or functions for correctness.
  + **Integration Testing:** Testing the interaction between integrated components to ensure they work together.
  + **System Testing:** Testing the complete system to validate it against requirements.
  + **Acceptance Testing:** Testing to ensure the system meets user needs and requirements.

**Task 2:** Purpose of Unit Testing.

* **Unit Testing:** Ensures that individual units of code (functions, methods) work as intended.

**Task 3:** Purpose of Integration Testing.

* **Integration Testing:** Verifies that different modules or services used by your application interact correctly.

**Task 4:** Steps to test a new module.

* **Steps:**
  1. **Test Planning:** Define the scope and approach for testing.
  2. **Test Case Design:** Create test cases based on requirements.
  3. **Test Environment Setup:** Prepare the testing environment.
  4. **Unit Testing:** Perform unit tests on individual components.
  5. **Integration Testing:** Test interactions between integrated components.
  6. **System Testing:** Validate the entire system's functionality.
  7. **Acceptance Testing:** Ensure the module meets the acceptance criteria.
  8. **Regression Testing:** Check if new changes have affected existing functionality.
  9. **Bug Reporting:** Document any defects or issues found.
  10. **Retesting:** Verify fixes for reported issues.
  11. **Final Sign-off:** Confirm that the module is ready for release.