

Capstone Project: "Employee Management System"

Domain: HR – managing employees, salaries, departments, and basic reporting.

Project Overview

A **console-based** application to manage employees of a company.

The system should allow:

- Adding new employees
 - Viewing and updating employee details
 - Assigning them to departments
 - Calculating and processing salaries
 - Viewing employee reports
-

Features

1. Employee Creation & Storage

- Use **classes** and **inheritance**:
 - Employee (base)
 - PermanentEmployee and ContractEmployee (derived)
- Store employees in a **Collection** (List<Employee> or Dictionary<int, Employee>).

2. Salary Processing

- Salary calculation logic:
 - Permanent employee: Basic + Allowances - Deductions
 - Contract employee: Hours worked × Rate
- Allow salary updates.

3. Exception Handling

- Throw and catch exceptions for:
 - Invalid salary values
 - Negative working hours
 - Employee not found

5. File Handling

- Save **employee details** and **payroll history** to a file (StreamWriter / StreamReader).
- Optionally, load existing employee data on startup.

6. SQL & RDBMS Integration

- Store employees in an Employees table.
- Store payroll history in a Payroll table.
- Use **ADO.NET** for CRUD operations.

7. Code Quality

- Follow **naming conventions**, **single responsibility principle**, and **comments**.
- Make reusable methods for salary calculation.

8. Unit Testing

- Write **unit tests** for:
 - Salary calculation
 - Adding employee
 - Updating employee

Suggested Modules for Development Flow

1. **OOP Design** – Create employee class hierarchy.
2. **Collections** – Manage employee records in memory.
3. **Exception Handling** – Wrap risky operations.
4. **File Handling** – Save employee and payroll data.
5. **SQL Integration** – Store/retrieve employee data.
6. **Unit Tests** – Test core functions.