Employee Payroll Application OOP Analysis

Smart Solutions Now (SSN) requires a new system to replace its outdated payroll application. The new system will manage departments, employees, and payroll with persistent data storage. It supports CRUD operations, payroll calculations, and sorted views based on business rules and user roles. Each employee is assigned to a department with defined pay rates. Regular hours up to 40 per week are paid at the department's standard rate, and hours beyond 40 qualify for overtime pay at 1.5 times the regular rate.

Department Rates File (Sample Data)

Dept. Code	Dept. Name	Regular Hourly Rate	Overtime Hourly Rate
1001	Human Resource Management (HR)	$85.50	$10.00

Employee File (Sample Data)

ID No	First Name	Last Name	Dept. Code	Position
2201	Perry	Techno	1001	Manager

Employee Payroll File (Sample Data)

ID No	Dept. Code	Hours Worked
2201	1001	28.50

Processed Payroll File (Sample Data)

ID No	First Name	Last Name	Dept. Code	Position	Hours Worked	Regular Pay	Overtime Pay	Gross Pay
2201	Perry	Techno	1001	Manager	28.50	$2436.75	$0.00	$2436.75

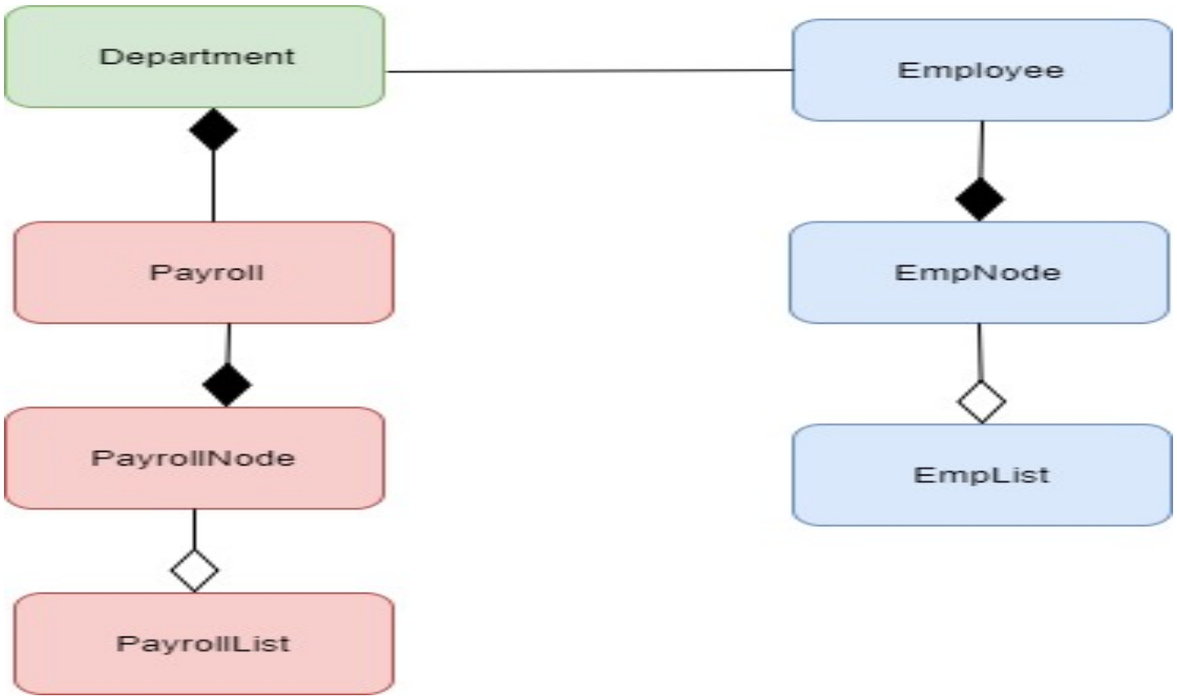
User Requirements (Use Case-Based)

1. **Use Case: Department Management**
   - **Actors**: Admin/HR
   - **Operations**: Add, View (All or By Code), Update and Delete,
   - **Persistence**: Data saved on exit
2. **Use Case: Employee Management**
   - **Actors**: Admin/HR
   - **Operations**: Add, View (By ID or sorted lists by ID, name, etc.), Update, Delete
   - **Persistence**: Data saved on exit
3. **Use Case: Payroll Processing**
   - **Actors**: Accounts Officer
   - **Operations**: Generate payroll file, View (By ID or sorted lists by ID, name, etc.), and Delete
   - **Business Rule**:
   - **Regular pay**: Employees who worked 40 hours or less per week, gets regular pay rate
   - **Overtime pay**: Employees working over 40 hours per week receive overtime pay at 1.5 times their regular hourly rate.
   - **Gross Pay** = Regular + Overtime
4. **Use Case: Application Management**
   - **Exit Functionality**: Ensures all data is saved upon exit
   - **Data Files**: Used for persistent storage for Department, Employee, and Payroll

Class Definition

Class	Attribute	Method
Department	• code • name • hourlyRate • overtimeRate	• displayHeader • display • getValidatedRate • createDepartment • checkFileAccess (ifstream version) • checkFileAccess (ofstream version) • writeRecord • addRecord • updateRecord • viewRecord • viewAllRecordss • deleteRecord
Employee	• id • firstName • lastName • departmentCode • position	• displayHeader • display • getValidatedString • getValidatedFloat • getValidatedInt • writeRecord • createEmployee • checkFileAccess (ifstream version) • checkFileAccess (ofstream version) • addRecord • updateRecord • viewRecord • viewAllRecords • deleteRecord
EmpList	• Ref head	• mergeSortById • mergeSortByLastName • mergeSortByDepartmentCode • mergeSortByPosition • mergeSortByHoursWorked
EmpNode	• Employee data • EmpNode Ref next	• display
Payroll	• id • firstName • lastName • departmentCode • position • hoursWorked • basicPay • overtimePay • grossPay	• display
PayrollNode	• data • Ref next	• display
PayList	• PayrollNode Ref head	• isEmpty • isFull • insertAtFront • countNodes • traverse • mergeSortById • mergeSortByLastName • mergeSortByDepartmentCode • mergeSortByPosition

Relations Diagram



Relationships Overview

Relationship	Type	Description
Department → Employee	One-to-Many	One department can have multiple employees
Employee → Payroll	One-to-Many	Each employee can generate multiple payroll records over time
EmpList → Employee	Aggregation	Manages multiple Employee objects through a linked list
PayList → Payroll	Aggregation	Manages multiple Payroll records
EmpNode, PayrollNode	Composition	Used internally by EmpList and PayList to manage node-based structures

Key Classes and Responsibilities

Class	Responsibility Summary
Department	Encapsulates department details and pay rates; manages department data persistence and validation.
Employee	Encapsulates employee personal and job details; manages employee data persistence and validation.
Payroll	Represents payroll calculations for an employee for a given period; computes regular, overtime, and gross pay.
EmpList	Aggregates **Employee** objects; provides sorting and traversal capabilities using linked list structure.
PayList	Aggregates **Payroll** objects; supports payroll record management, sorting, and traversal.
EmpNode	Node in linked list for employees; contains employee data and pointer to next node.
PayrollNode	Node in linked list for payroll records; contains payroll data and pointer to next node.