



Ahsanullah University of Science and Technology (AUST)
Department of Electrical and Electronic Engineering

Course No.: EEE 1110

Course Title: Programming Language Laboratory

Employee Salary Management System

ID: 20230205176

Name: Fahim Monayem

Section: C2

For the students of the
Department of Electrical and Electronic Engineering
1st Year, 1st Semester

Overview

This project involves creating an Employee Salary Management System using C++. The system allows HR manager to add employee information, calculate payroll details, generate payroll and manage allowances and deductions. The key functionalities include adding employee records, displaying employee information, generating pay slips, and updating salary structures.

Objective

The main objective of this system is to simplify the payroll management process for a company. It allows HR managers to input employee details, automatically calculate various allowances (house, medical, transport), and generate detailed pay slips based on the basic salary and deductions. The system is designed to ensure flexibility in updating salary structures and allowances as needed.

Design and Implementation:

Employee Structure

The central data structure used in this program is the Employee struct. It encapsulates all relevant information about an employee, including:

Basic Information: id, name, department, basicSalary.

Allowances and Bills: House, medical, transport bills, and other allowances. These are calculated as percentages of the basic salary.

Deductions: This field accounts for any deductions that should be subtracted from the employee's net salary.

Functions

The program is modularized through different functions to handle various aspects of employee management. Each function is focused on a specific task, which enhances code maintainability and clarity.

addEmployee(): This function collects new employee details and calculates allowances based on predefined percentages. It validates the input to ensure data consistency.

displayEmployees(): This function displays the details of all employees in a formatted manner, which helps in reviewing the employee records. The data is displayed in a tabular format for ease of understanding.

generatePaySlip(): This function generates a detailed pay slip for each employee, including allowances, deductions, and total salary. The pay slip is calculated based on the employee's basic salary, allowances, and any deductions.

calculateNetSalary() & calculateTotalSalary(): These two functions perform calculations to determine the net salary after deductions and the total salary including allowances.

updateDeductions(): This function allows updates to the deduction amount for a specific employee. It recalculates the total salary after the update and persists the data to a file.

updateEmployeeAllowances(): This function allows HR to adjust the percentage of various allowances (house, medical, transport). The percentages are updated, and the allowance amounts are recalculated accordingly.

updateEmployeeSalary(): This function allows for updating an employee's basic salary and recalculates all the allowance values based on the new salary.

saveEmployeesToFile() & loadEmployeesFromFile(): These two functions handle file input and output. Employee details are stored in a text file, ensuring persistence across multiple runs of the program. This feature provides long-term storage and retrieval of employee data.

File Handling

The program reads and writes employee data to a file (employees.txt). This allows the system to persist employee data even after the program is closed. The file is structured in CSV format, with each employee's details saved in a single line. This design decision ensures easy retrieval and updating of employee records.

Reasoning Behind Design Decisions:

Struct for Employee Data: The use of a struct was chosen for simplicity and clarity. Since the employee data follows a straightforward format and doesn't require complex inheritance or polymorphism, a struct is an ideal solution for grouping related fields.

Modular Functions: Breaking the system into functions promotes reusability and easier debugging. Each function addresses a specific task (e.g., updating deductions, displaying employees), making the system easier to extend in the future.

Validation: Input validation was incorporated in critical parts of the system, such as when adding a new employee. This ensures that only valid data is entered, avoiding inconsistencies in payroll calculations.

Key Features

Flexible Payroll Calculation: The system can calculate various allowances as percentages of the basic salary, providing flexibility to modify the payroll structure for different employees.

Pay Slip Generation: A clear and detailed pay slip is generated, showing all salary components including allowances and deductions.

Deductions and Allowances Updates: The system allows updating both deductions and allowances, making it adaptable for companies with changing payroll structures.

Data Persistence: Employee records are stored in a text file, ensuring that data is not lost between sessions. This feature makes the system suitable for long-term use.

Problems Encountered:

Handling Complex Calculations: One challenge was ensuring that the salary calculations were accurate and consistently updated when values like basic salary or allowances changed. This was solved by breaking down the calculations into smaller, testable functions (`calculateNetSalary()` and `calculateTotalSalary()`).

File Handling: Another issue was ensuring that data is correctly written to and read from the file. CSV formatting issues were addressed by carefully managing the delimiter and ensuring that all fields are written in the correct order.

Future Improvements

Graphical User Interface (GUI): Future improvements could include a graphical interface to make the system more user-friendly. A GUI would also simplify input, making it more accessible for HR managers.

Advanced Reporting: Another possible extension would be adding reporting features, such as generating summaries of payroll expenses or tracking payroll changes over time.

Employee Search and Filtering: Implementing more advanced search and filtering options, like searching by department or salary range, could enhance the usability of the system for large companies.

Conclusion

The Employee Salary Management System successfully automates the payroll process, reducing administrative burden and improving accuracy. With a user-friendly interface and robust functionality, it simplifies salary calculations and employee management. Future enhancements could further improve the system's efficiency and expand its capabilities.