System Design Document: MotorPH Payroll System

1. Introduction

1.1 Purpose

This document describes the design of the MotorPH Payroll System, a Java-based application that manages employee records, attendance tracking, payroll processing, and reporting for MotorPH company.

1.2 Scope

The system handles:

- Employee record management
- Attendance tracking and reporting
- Payroll calculation with mandatory deductions
- Payroll reporting (individual payslips, weekly/monthly summaries)

1.3 Definitions, Acronyms and Abbreviations

- **SSS**: Social Security System

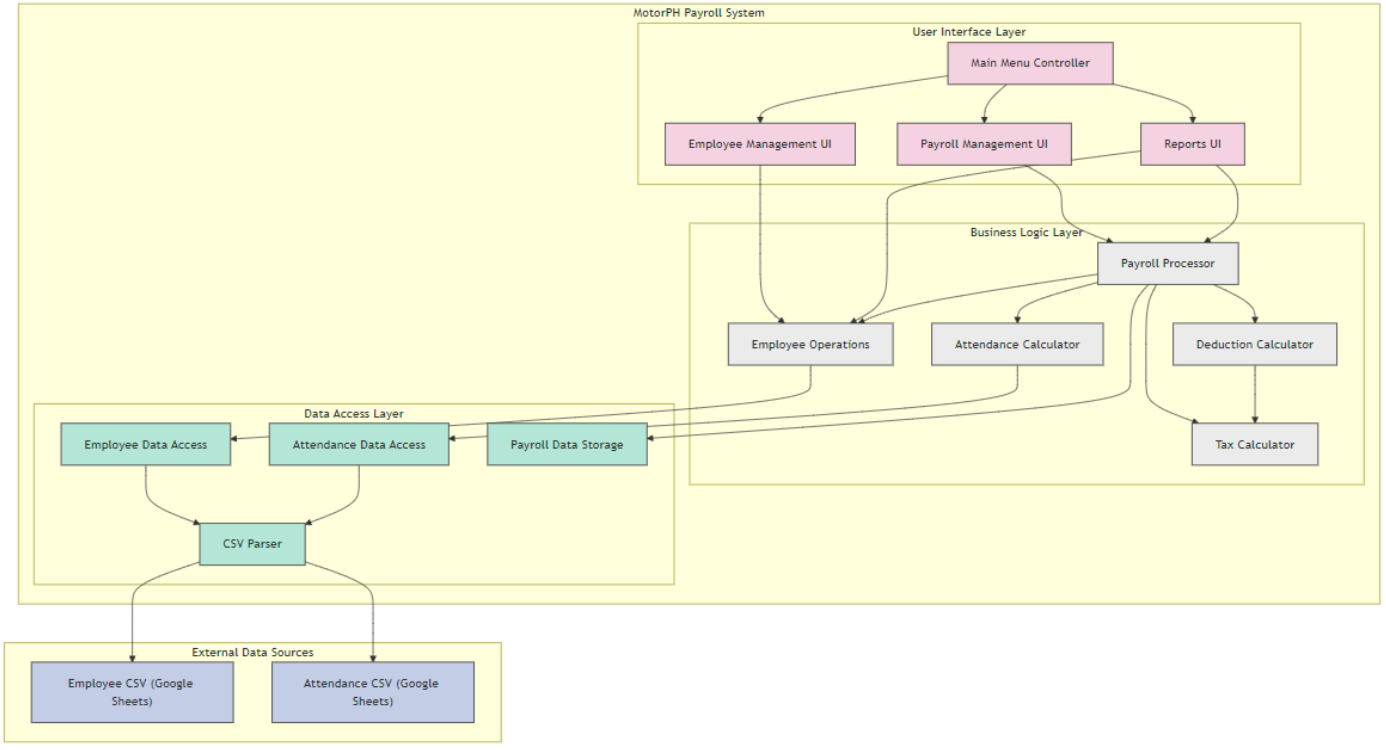
- **PhilHealth**: Philippine Health Insurance Corporation
- **Pag-IBIG**: Home Development Mutual Fund
- **CSV**: Comma-Separated Values
- **URL**: Uniform Resource Locator

2. System Architecture

2.1 Overview

The MotorPH Payroll System uses a single-tier architecture implemented in Java. The system is console-based with modular components handling different aspects of payroll processing.

2.2 Architectural Diagram



2.3.1 Main Controller (MotorPHPayroll class)

Provides the main execution flow and menu system.

2.3.2 Data Access Layer

- `loadEmployeesFromCSV()`: Fetches and parses employee data

- `loadAttendanceFromCSV()`: Fetches and parses attendance records
- `findEmployeeById()`: Retrieves specific employee data

2.3.3 Employee Management Module

- `employeeManagement()`: Entry point for employee management features
- `searchEmployee()`: Search functionality by name or employee number
- `listAllEmployees()`: Display all employee records
- `viewAttendance()`: View attendance for a specific employee

2.3.4 Payroll Processing Module

- `payrollManagement()`: Entry point for payroll features
- `generatePayroll()`: Generate payroll for all employees
- `customPayroll()`: Generate payroll for a specific employee
- `calculateHoursWorked()`: Calculate total hours worked
- `extractHourlyRate()`: Extract or calculate hourly rate
- `calculateNetPay()`: Calculate net pay with mandatory deductions

2.3.5 Reporting Module

- `reportsMenu()`: Entry point for reporting features
- `generatePayslipReport()`: Generate individual payslips
- `generateWeeklySummary()`: Generate weekly payroll summary
- `generateMonthlySummary()`: Generate monthly payroll summary

2.3.6 Deduction Calculator

- `calculateSSSContribution()`: Calculate SSS contributions
- `calculatePhilHealthContribution()`: Calculate PhilHealth contributions
- `calculatePagIbigContribution()`: Calculate Pag-IBIG contributions
- `calculateWithholdingTax()`: Calculate withholding tax based on tax brackets

3. Data Design

3.1 Data Sources

- **Employee Data CSV**: Contains employee personal information, salary details

- **Attendance Records CSV**: Contains daily time-in/time-out records

3.2 Data Models

3.2.1 Employee Record Structure

```
[0] Employee Number
[1] Last Name
[2] First Name
[3] Middle Name (optional)
...
[10] Status
[11] Position
...
[13] Basic Salary
...
[18] Hourly Rate
```

3.2.2 Attendance Record Structure

```
[0] Employee Number
...
[3] Date (M/d/yyyy format)
[4] Time In (H:mm format)
[5] Time Out (H:mm format)
```

3.2.3 Payroll Entry Structure

- Employee Number
- Name
- Total Hours Worked
- Hourly Rate
- Gross Pay
- Net Pay
- Start Date
- End Date

3.3 In-Memory Data Storage

- `employees`: List of employee records
- `attendanceRecords`: List of attendance records
- `postedPayrolls`: Map of processed payroll entries, keyed by employee ID + date range

3.4 Data Flow

1. External CSV data is fetched via URL and parsed
2. Data is stored in memory as lists of string arrays
3. Business logic processes this data when calculating payroll
4. Processed payroll can be "posted" for future reference
5. Reports use either posted payroll data or calculate on-the-fly

4. Interface Design

4.1 User Interface

The system implements a console-based text menu interface with the following structure:

```
Main Menu
├── Employee Management
│   ├── Search Employee
│   ├── List All Employees
│   └── Attendance
├── Payroll Management
│   ├── Generate Payroll
│   └── Custom Payroll
└── Reports
    ├── Payslip
    ├── Weekly Summary
    └── Monthly Summary
```

4.2 External Interfaces

4.2.1 Data Input

- Employee data fetched from Google Sheets CSV (via HTTPS)

- Attendance data fetched from Google Sheets CSV (via HTTPS)

5. Processing Design

5.1 Key Processes

5.1.1 Payroll Generation Process

1. Input date range for payroll period
2. For each employee:
    - Calculate total hours worked in period
    - Determine hourly rate
    - Calculate gross pay (hours × rate)
    - Calculate deductions:
        - SSS contribution
        - PhilHealth contribution
        - Pag-IBIG contribution
        - Withholding tax
    - Calculate net pay
3. Display payroll summary
4. Option to post/save payroll

5.1.2 CSV Parsing Process

1. Open URL connection to CSV file
2. Skip header row
3. Parse each line:
    - For employee data, handle quoted fields containing commas
    - For attendance data, split on commas
4. Store records in memory

5.2 Critical Algorithms

5.2.1 Hourly Rate Determination

The system uses a failover approach to determine hourly rate:

1. Try to extract directly from employee record (column 18)
2. If unsuccessful, calculate from basic salary: `(basicSalary / 21) / 8`
3. If still unsuccessful, use position-based defaults
4. Final fallback to a default rate (133.93)

5.2.2 Hours Calculation

Calculate worked hours by:

1. Parse time-in and time-out from attendance records
2. Calculate duration between times
3. Sum up all durations within the specified date range

5.2.3 Tax Calculation

Progressive tax calculation based on Philippine tax brackets:

- 0% for income up to 2,083
- 20% of excess over 2,083 up to 33,333
- 6,250 plus 25% of excess over 33,333 up to 66,667
- 14,583.33 plus 30% of excess over 66,667 up to 166,667
- 44,583.33 plus 32% of excess over 166,667 up to 666,667
- 204,583.33 plus 35% of excess over 666,667

6. Error Handling

6.1 Input Validation

- Date format validation using try-catch with DateTimeParseException
- Numeric input validation with try-catch for NumberFormatException
- Menu option validation with range checks

6.2 Processing Errors

- Employee search with no results handling
- Empty data set handling
- CSV parsing error handling

6.3 Graceful Degradation

- Default values for missing employee data
- Multiple fallback mechanisms for hourly rate determination
- Max caps on deductions to prevent negative net pay

7. Non-Functional Requirements

7.1 Performance

- In-memory data processing for fast calculations
- Efficient parsing of CSV data with minimal memory overhead

7.2 Maintainability

- Modular design with separate method responsibilities
- Consistent error handling patterns
- Well-commented code

7.3 Security

- No local data storage (transient in-memory only)
- Read-only access to external data sources

7.4 Usability

- Clear menu structure
- Consistent output formatting
- Helpful error messages

8. Limitations and Future Enhancements

8.1 Current Limitations

- No persistent storage for posted payrolls (lost on program exit)
- Limited to console-based interface
- No authentication or user roles
- No functionality to modify employee or attendance records

8.2 Future Enhancements

- Database integration for persistent storage
- GUI implementation
- User authentication system
- Advanced reporting capabilities
- Payroll editing functionality
- Employee and attendance record management
- Export reports to PDF or Excel

9. Conclusion

The MotorPH Payroll System provides a functional console-based solution for managing payroll operations. Its modular design allows for future enhancements while maintaining core payroll calculation functionality. The system effectively handles CSV data parsing, attendance calculation, and Philippine-specific deduction rules to provide accurate payroll processing.