**Group members:**

**Parsa Molahosseini**

**Mehrad Bayat**

**Jerry-Lee Somera  
----------------------------------------------------------------------------------**

**Project Report**

**Title: Employee and Department Management System**

**1. Objective**

The objective of this project is to develop a Java-based Employee and Department Management System that allows users to perform the following tasks:

* Manage employee information (add, update, delete).
* Manage department details and link employees to departments.
* Generate and manage payroll for employees.
* Save and load employee and department data using serialization.
* Export payroll reports to CSV for external usage.

This system aims to streamline employee and department management processes while providing a user-friendly interface using JavaFX.

**2. System Design**

**2.1 Overview**

The system is designed with a modular structure to handle employees, departments, and payrolls efficiently. It uses JavaFX for the GUI, Java classes for the backend logic, and file-based storage for data persistence.

**2.2 Architecture**

The application consists of three main layers:

1. **Frontend**: JavaFX-based graphical user interface for interacting with users.
2. **Backend**: Java classes (Employee, Department, Payroll) handle the business logic and data representation.
3. **Data Storage**: Data is saved and loaded using serialization into .dat files, ensuring persistence between sessions.

**2.3 Key Classes**

1. **HelloController**: The main controller class that handles interactions between the GUI and the backend. It manages user inputs, updates the interface, and performs backend operations such as data validation and file handling.
2. **Employee**: Represents an employee with attributes like ID, name, department, position, email, phone, hourly rate, attendance days, and leave days.
3. **Department**: Represents a department with attributes like ID and name.
4. **Payroll**: Represents payroll details for an employee, including gross salary, tax deductions, and net salary.

**2.4 User Roles**

* **Admin Role**: Full access to employee and department management features.
* **Employee Role**: Restricted access to view and generate payroll information.

**3. Implementation Approach**

**3.1 Tools and Frameworks**

* **IDE**: IntelliJ IDEA
* **Programming Language**: Java
* **Framework**: JavaFX for GUI
* **Storage Mechanism**: File-based serialization to .dat files

**3.2 Features and Functionalities**

1. **Employee Management**
   * Add, update, delete employees.
   * Display employee details in a TableView.
   * Validate inputs such as email format, phone number, and numeric fields.
2. **Department Management**
   * Add, update, delete departments.
   * Dynamically update employees when a department name changes.
3. **Payroll Management**
   * Calculate gross salary, tax deductions, and net salary based on user inputs.
   * Link payroll records to employees for accurate reporting.
4. **Data Persistence**
   * Save and load employee and department data using serialization.
   * Maintain data integrity during file read and write operations.
5. **Export Reports**
   * Generate and export payroll reports in CSV format.
   * Include details like employee name, gross salary, tax deductions, and net salary.

**3.3 GUI Design**

* **Employee Management Tab**:
  + Fields to input employee details (ID, name, department, etc.).
  + A TableView to display employee records.
* **Department Management Tab**:
  + Fields to add or update department details.
  + A TableView for department records.
* **Payroll Management Tab**:
  + Inputs for hours worked, overtime, and bonuses.
  + A dropdown to select employees for payroll generation.
  + A TableView to display payroll reports.

**4. Challenges Encountered**

**4.1 NullPointerException**

* **Cause**: Incorrect fx:id bindings in the FXML file, leading to null values in controller fields.
* **Solution**: Ensured that all fx:id attributes in FXML matched the controller field names.

**4.2 Input Validation**

* **Issue**: Ensuring all fields are correctly filled and formatted before processing.
* **Solution**: Implemented regex validation for email and phone fields and handled NumberFormatException for numeric inputs.

**4.3 Dynamic Updates**

* **Issue**: Updating employee records when a department name changes.
* **Solution**: Added logic to iterate through employees and update their department names dynamically.

**4.4 Data Persistence**

* **Issue**: Serialization errors when saving or loading data due to changes in class structure.
* **Solution**: Maintained backward compatibility by keeping class structures consistent.

**4.5 Report Generation**

* **Issue**: Ensuring accurate calculation of payroll details.
* **Solution**: Verified the payroll calculation logic and included additional validation for inputs.

**5. Screenshots**

(Include the following screenshots in the final submission)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**6. Conclusion**

The Employee and Department Management System successfully fulfills the requirements by providing a user-friendly interface and robust backend functionalities. It streamlines employee and department management tasks, simplifies payroll generation, and ensures data persistence.

The challenges faced during development were effectively resolved, resulting in a reliable and efficient application. Future enhancements could include integrating a database for scalability and implementing authentication for secure access.

**7. References**

* Java Serialization Guide: <https://docs.oracle.com/javase/8/docs/>
* Regex for Validation: <https://regex101.com/>