Project Proposal
Employee Management System
35

EC/2022/053	K.S.B.Galkotuwa
EC/2022/077	C.Niroshan
EC/2022/002	R.H.M.P.Rathmalage
EC/2022/078	G.Varushiny
EC/2022/023	A.Dhilukshy

Introduction

```
IDEA.app/Contents/lib/idea_rt.jar=64011:/Applications/IntelliJ IDEA.ap
.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/kavindu
EmployManagementSystem

                           .           .
8 8888888888            ,8.         ,8.             d888888o.
8 8888                 ,888.       ,888.          .`8888:' `88.
8 8888                .`8888.     .`8888.         8.`8888.   Y8
8 8888               ,8.`8888.   ,8.`8888.        `8.`8888.
8 888888888888      ,8'8.`8888, ,8'8.`8888.        `8.`8888.
8 8888             ,8' `8.`8888' `8.`8888.          `8.`8888.
8 8888            ,8'   `8.`88'   `8.`8888.          `8.`8888.
8 8888           ,8'     `8.`'     `8.`8888.  8b      `8.`8888.
8 8888          ,8'       `8        `8.`8888. `8b.    ;8.`8888
8 888888888888 ,8'         `        `8.`8888. `Y8888P ,88P'


===============================================================
              Welcome to Employee Management System

Press 1 : To Add an Employee Details
```

This project aims at finding a simple and cost-effective approach through which it can be accomplished for small to medium-sized businesses to keep records of their employees in a way that gives easy access and a professional look. Whereas traditional plain-text storage is favored, HTML files can become much more readable and integrated with a greater availability through web applications with further flexibility.

This would make a pretty good system for HR officers, small business, or employees needing an employee management tool that focuses on neat data organization and pretty display. It wouldn't be simple, portable, or easy-to-learn, though it was designed to incorporate today's standards for record storage.

Key design aspects include ,

• Web-based Record Storage - Employee details are saved as structured HTML files, enhancing readability and enabling quick access through web browsers.

• Object-Oriented Structure - This system uses OOP principles for modularity and scalability.

• Menu-driven Console Interface - The users perform all the operations through a simple console menu.

The EMS project shows how practical applications of Java programming concepts can be paired with activities, such as file handling and string operations, to create modules for modern, user-friendly data storage. With the time that allocate to the project, we'll wish to implement Project with GUI with time.

Requirements Analysis and Design

CLASS 01 - EmployManagementSystem

Data Requirements

- *Acts as the entry point of the application and coordinates the overall system operations.*

- *No additional data fields are directly associated with this class.*

Functional Requirements

1. *Display the main menu of the Employee Management System.*

2. *Handle user input to navigate through different system operations (add, view, update, remove, exit).*

3. *Ensure program flow control using loops and conditional statements.*

CLASS 02 - MainMenu

Data Requirements

- *No direct data fields; acts as a utility class.*

Functional Requirements

1. *Display a formatted menu to guide the user through available system operations.*

2. *Serve as a reusable component for menu presentation.*

CLASS 03 - EmployDetail

Data Requirements

- *String name - Employee name.*

- *String father_name - Employee's father's name.*

- *String email - Employee's email address.*

- *String position - Employee's job title.*

- *String employ_id - Unique identifier for the employee.*

- *String employ_salary - Employee's salary.*

- *String employ_contact - Employee's contact information.*

Functional Requirements

1. *Collect employee details using user input.*

2. *Provide encapsulated data for other classes to create or manipulate records.*

CLASS 04 - Employee_Add

Data Requirements

- *Utilizes data collected from EmployDetail to create HTML,CSS and JS files.*

Functional Requirements

1. *Create an HTML file named after the employee ID.*

2. *Write employee details into the HTML file in a structured format using HTML tags.*

3. *Check for duplicate entries and notify the user if the employee already exists.*

CLASS 05 - Employee_Show

Data Requirements

- *Requires the employee ID to locate the corresponding HTML file.*

Functional Requirements

1. *Read and display the contents of the specified employee's HTML file in the console.*

2. *Handle exceptions if the file does not exist.*

CLASS 06 - Employee_Remove

Data Requirements

- *Requires the employee ID to locate the corresponding HTML file.*

Functional Requirements

1. *Locate and delete the specified employee's HTML file.*

2. *Notify the user about the success or failure of the removal operation.*

CLASS 07 - Employee_Update

Data Requirements

- *Requires the employee ID to locate the HTML file.*

- *Old and new values for updating specific employee details.*

Functional Requirements

1. *Search for the specific field in the employee's HTML file and replace the old value with the new one.*

2. *Ensure file integrity after updates and notify the user about the success or failure.*

CLASS 08 - CodeExit

Data Requirements

- *No direct data requirements.*

Functional Requirements

1. *Provide a graceful exit message and terminate the program.*

NOTE
While Implementing the project we wish to add login menu, Register menu and few utility Functions for specify that system.

Teamwork

EC/2022/53(https://github.com/kavindus0) - Main Menu and System Coordination

Responsibilities
- Implement the MainMenu class for displaying the menu.
- Implement the EmployManagementSystem class to handle program flow.
- Integrate all other classes into a seamless working system.
- Conduct unit, Intergration and System testing
- resolve inter-class dependencies.

Classes
- MainMenu
- EmployManagementSystem

EC/2022/23 - Adding Employee Details

Responsibilities
- Implement the EmployDetail class to collect employee details.
- Implement the Employee_Add class to create HTML files for employee records.
- Ensure proper validation and handle duplicate entries.
- Conduct unit testing

Classes
- EmployDetail
- Employee_Add

EC/2022/78 (https://github.com/Varushiny) - Viewing Employee Details

Responsibilities
- Implement the Employee_Show class to read and display employee details from HTML files.
- Handle exceptions for missing or corrupted files.
- Conduct unit testing

Classes

- Employee_Show

EC/2022/02 (https://github.com/HeshanMayuranga) - Removing Employee Records

Responsibilities

- *Implement the Employee_Remove class to locate and delete employee records (HTML files).*
- *Ensure error handling for non-existing files.*
- *Conduct unit testing*

Classes

- *Employee_Remove*

EC/2022/77 (https://github.com/Niroshan26) - Updating Employee Records

Responsibilities
- *Implement the Employee_Update class to modify specific details in existing employee records.*
- *Ensure the integrity of HTML formatting during updates.*
- *Conduct unit testing*

Classes

- *Employee_Update*
- *CodeExit*