**Assignment Cover Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | | **Module Number and Title** | |
| HD in Computing/HD in Software Engineering | | **Object Oriented Programming- CSE4006** | |
| **Student Name & No.** | | **Assessor** | |
| KG/HDCSE/CMU/24/41  A.H.Divya Nethmi Amarasinghe | | Mr. Tharindu Bandara | |
| **Hand out date** | | | **Submission Date** |
| 2023.12.02 | | | 2024.01.02 |
| **Assessment type**  WRIT1-Coursework | **Duration/Length of**  **Assessment Type**  3000 wordsequivalent | | **Weighting of Assessment**  100% |

|  |  |
| --- | --- |
| **Learner declaration** | |
| I A.H.D.N.Amarasinghe KG/HDCSE/CMU/24/41 certify that the work submitted for this assignment is my own and research sources are fully acknowledged. | |
| |  |  |  |  | | --- | --- | --- | --- | | **Marks Awarded** | | | | | First assessor | |  | | | IV marks | |  | | | Agreed grade | |  | | | Signature of the assessor |  | Date |  | |

**Feedback Form**

**International College of Business & Technology**

**Module:**

**Student:**

**Assessor:**

**Assignment:**

**Strong features of your work:**

**Areas for improvement:**

**Marks Awarded:**

**Acknowledgment**

**I am very grateful to MR.Tharindu Bandara who show me the path and helped me do this assignment, and to my parents and friends who supported me to go on my path.**

**Introduction**

The Colombo Institute of Studies has made the decision to switch from manually handling staff records to an automated system in recognition of the requirement for accuracy and efficiency. By streamlining the personnel information management process, this shift seeks to empower HR managers and assistants. The proposed employee management system supports multiple user tiers, each with specialized features to improve the management process as a whole.

User Levels and Features:

Different user levels are defined by the system, with HR managers and administrators (admins) being the main categories. HR managers now have the power to easily onboard new hires, establish new departments and designations, and quickly search for employee information using a variety of parameters, including name, department, and EPF number.

Task 01.

Object Oriented Programming

* 1. what is OOP

Classes and objects are used in the programming method known as object-oriented programming, or OOP. It breaks down software program into reusable code blueprints, or classes, that may be used to produce specific instances of objects later on. The programming languages PHP, Ruby, Perl, JavaScript, C++, Java, and Python are all object-oriented.

A class provides the building blocks for the production of more tangible, specialized items. Classes are big groupings of people with similar traits, like dog or bird. These classes specify the characteristics of a type instance, like color, but not the attribute values for an individual object.

Classes also have procedures, which are functions that are specific to that type of object. These functions, which are defined in the class, provide the intended purpose for the particular object type.

* + 1. OOP Concepts.



* **Object**

In Java, classes and objects, together with their characteristics and functions, are connected to everything. For instance, an automobile is an object in the actual world. The car contains features like colour and weight, as well as functions like brake and drive.

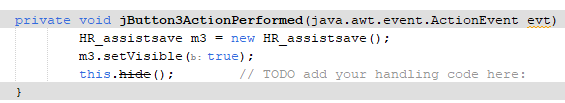
A class is a "blueprint" for constructing things, similar to an object constructor..

Figure . example code for object

* Class

A class is an object collection. It makes sense as an entity.

Another way to think of a class is as a blueprint from which you can make a unique item. There is no space needed for class.

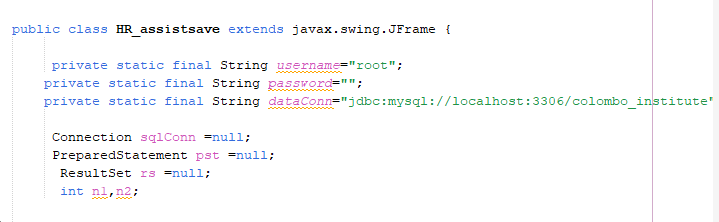


Figure ex. for class

* Inheritance

It is feasible for classes to inherit properties and methods from one another in Java. The "inheritance concept" is divided into two groups:

1. subclass (child) - the class that inherits from another class
2. superclass (parent) - the class being inherited from

To inherit from a class, use the extends keyword.

* Polymorphism

One activity carried out in multiple ways is referred to as polymorphism. To persuade the buyer in a different way, for instance, sketch a shape—a triangle, a rectangle, etc.

Polymorphism in Java is accomplished through the usage of overloading and overriding methods.

* Abstraction

Data abstraction is the process of presenting the user only the information that is absolutely necessary and hiding all other details.

Interfaces or abstract classes can be used to accomplish abstraction. An abstract class must be inherited from another class in order to access it; it is a limited class that cannot be used to construct objects.

Abstract methods don't have bodies and can only be used in abstract classes. The subclass (inherited from) provides the body.

* Encapsulation

Encapsulation is the binding, or wrapping, of data and code into a single unit.A Java class serves as an illustration of encapsulation. Since every member of the data is private in this case, the Java bean is the fully encapsulated class.

**Advantages and disadvantages in OOP**

Table

Description automatically generated

Figure advantages and disadvantages

**Task 02.**

2.1 UML diagrams

Software system artefacts can be specified, visualised, built, and documented using the UML standard language.

The Object Management Group (OMG) developed UML, and in January 1997, the OMG received a draft of the UML 1.0 definition.

OMG is working tirelessly to establish a true industry standard.

Unified Modelling Language is known as UML.

Compared to other popular programming languages like C++, Java, COBOL, etc., UML is distinct.

A visual language called UML is used to create software blueprints.

A general-purpose visual modelling language for software system specification, construction, and documentation is called UML.

UML is not constrained by the fact that software systems are typically modelled using it. It is also employed in the modelling of non-software systems. Consider the manufacturing unit's process flow, etc.

It is possible to think of UML as the evolution of object-oriented (OO) analysis and design.

Data and methods for controlling the data are both contained in an object. The object's state is reflected in the data. In addition to describing an object, classes create a hierarchy that mimics the structure of real-world systems. Inheritance is used to illustrate the hierarchy, and classes can also be connected in many ways depending on the situation.

Real-world items are called objects, and UML may be used to represent fundamental ideas like inheritance, polymorphism, abstraction, and encapsulation.

All of the concepts used in object-oriented analysis and design can be represented using UML thanks to its strength. UML diagrams alone depict concepts that are object-oriented. Thus, it's crucial to understand this before studying UML.

**2.1.1 types of UML diagrams**

UML diagrams are divided into two subcategories: behavioral and structural diagrams.

Structural diagrams The elements that comprise a system and their interrelationships are shown in structural diagrams. These schematics display a system's static components.

Behavioral diagrams show the actions that take place inside a system. They display the interactions between all the parts and other users or systems.

**Structural Diagram**

A essential component of any object-oriented solution is a UML class diagram. It shows an object-oriented, static system with projects defined by classes, attributes, and functions. Stated differently, it illustrates the various classes present in a system and how they function. This interaction diagram is used by business managers and software engineers to represent the various relationships that are present in a process.

A rectangle in the diagram represents the class. Every rectangle has three vertically divided pieces. The class name appears in the top area, and information on the class's behaviors, operations, and characteristics may be found in the second and third sections.

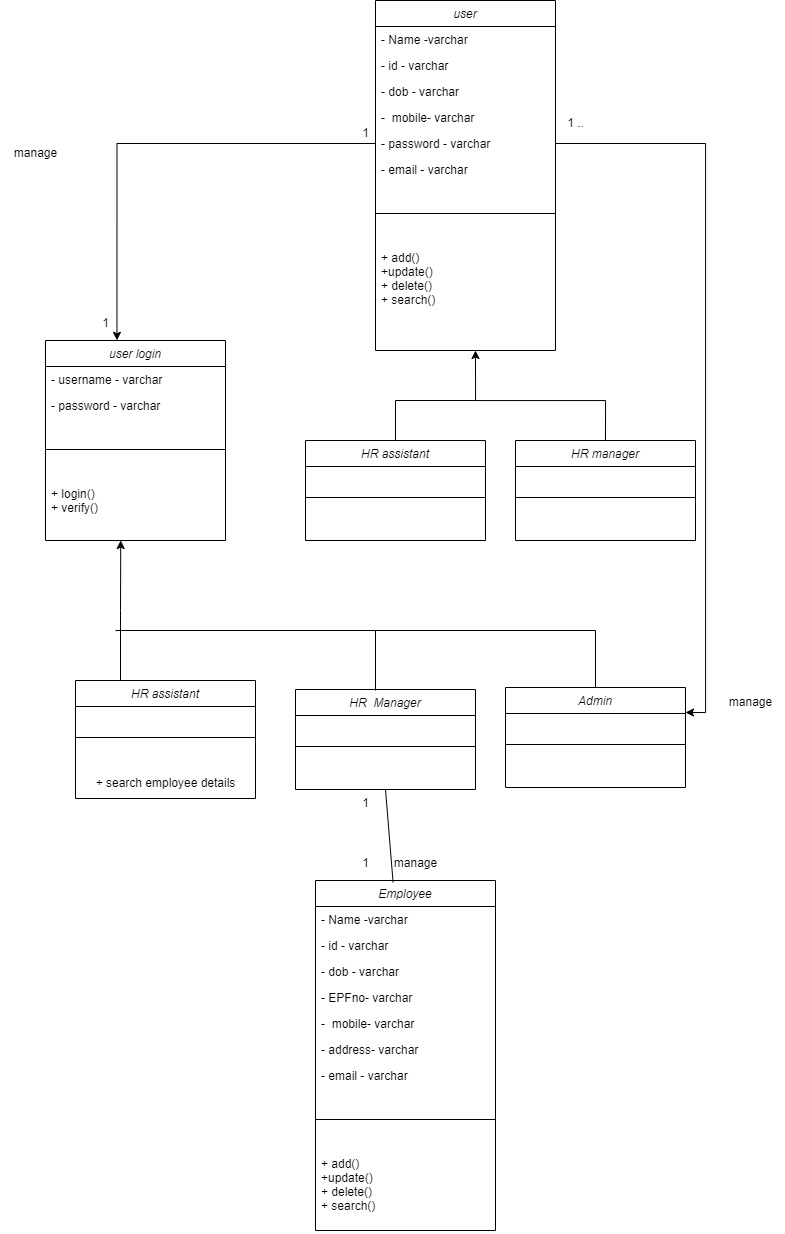


Figure . class diagram

* Explanation

**User:**

Represents a generic user of the system.

Subclasses include HR Manager, HR Assistant, and potentially other user types.

**Admin:**

Use case: createAccount() - Permits the administrator to establish an account for the HR Manager and HR Assistant.

**HR Manager:**

Use cases:

addDepartments() - Enables the HR Manager to add new departments.

addDesignations() - Allows the HR Manager to add new designations.

addEmployees() - Lets the HR Manager add new employees and allocate them to available departments and designations.

searchEmployees() - gives the HR Manager the ability to look for employee information using a variety of parameters, including name, department, EPF number, and designation.

**HR Assistant:**

Use cases ;

addEmployees() - Allows the HR Assistant to add new employees and allocate them to available departments and designations.

searchEmployees() - Enables the HR Assistant to search for employee details based on various criteria such as department, designation, name, EPF number, etc.

**Employee:**

Represents the details of an employee.

* Associations:

The User class has associations with Admin and HR Manager, representing the relationships between different user types.

HR Manager is associated with Employee, representing the management of employee-related functionalities.

**Behavioral diagrams**

1. Use case diagram

Use case diagrams give a visual summary of the key players in a software system. They assist developers in analyzing the connections between use cases and personas by providing an overview of a system's intended behavior and displaying its capabilities.

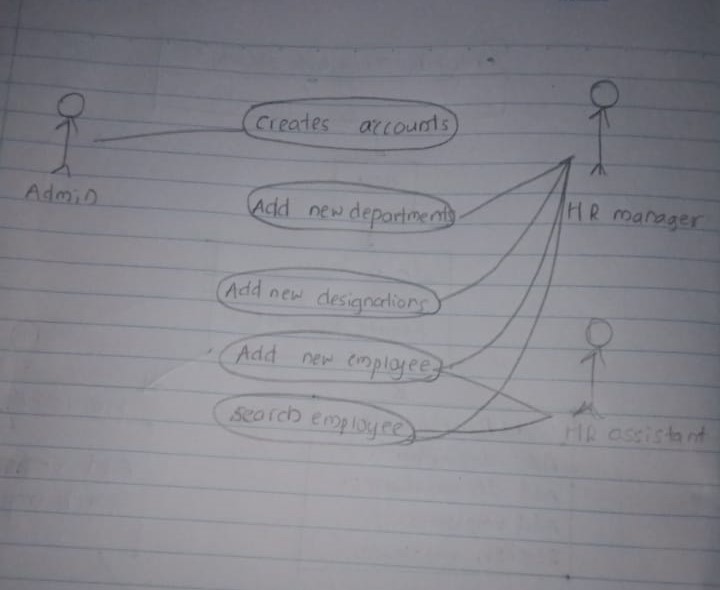


Figure 5. use case diagram

Classes:

user: Represents common attributes like name, employee ID, and contact details.

Employee: Inherits from Person, includes attributes like department, designation, and methods for searching.

HRManager: Inherits from Employee, includes methods for managing departments and designations.

Admin: Manages user accounts, inherits from Person.

System: Manages the overall system and file operations.

Use Case 1: Add New Department

**Actors:**

HR Manager

**Description:**

HRManager logs into the system.

HRManager navigates to the "Manage Departments" section.

HR Manager selects the option to "Add New Department."

The system prompts HR Manager to enter the department details (e.g., name, description).

HR Manager submits the information.

The system validates and adds the new department to the database.

The system confirms the successful addition of the department.

**Alternate Flow:**

If the entered department information is incomplete or invalid, the system prompts HR Manager to provide valid details.

Use Case 3: Add New Employee

**Actors:**

HR Manager

**Description:**

HR Manager logs into the system.

HR Manager navigates to the "Add New Employee" section.

HR Manager fills in the employee details (e.g., name, contact information, department, designation).

HR Manager submits the information.

The system validates the entered details.

If valid, the system adds the new employee to the database.

The system generates an employee ID and provides it to HR Manager.

The system confirms the successful addition of the employee.

**Alternate Flow:**

If the entered employee information is incomplete or invalid, the system prompts HR Manager to provide valid details.

Use Case 4: Search Employee Details

**Actors:**

HR Manager

**Description:**

HR Manager logs into the system.

HR Manager navigates to the "Search Employee" section.

HR Manager selects the search criteria (e.g., department, designation, name, EPF number).

HR Manager enters the search parameters.

The system retrieves and displays the matching employee details.

HR Manager can view, edit, or take further actions based on the displayed information.

**Alternate Flow:**

If there are no matching records, the system notifies HR Manager accordingly.

1. Sequence diagram

Sequence diagrams (sometimes called event diagrams) in UML illustrate the sequence in which your items interact. This comprises the messages that are sent back and forth between your objects to carry out certain tasks, as well as the lifelines of your objects and the processes that engage with them.

These diagrams are frequently used by developers and business experts to comprehend how to organize a new system or enhance an existing procedure.

**Employee**

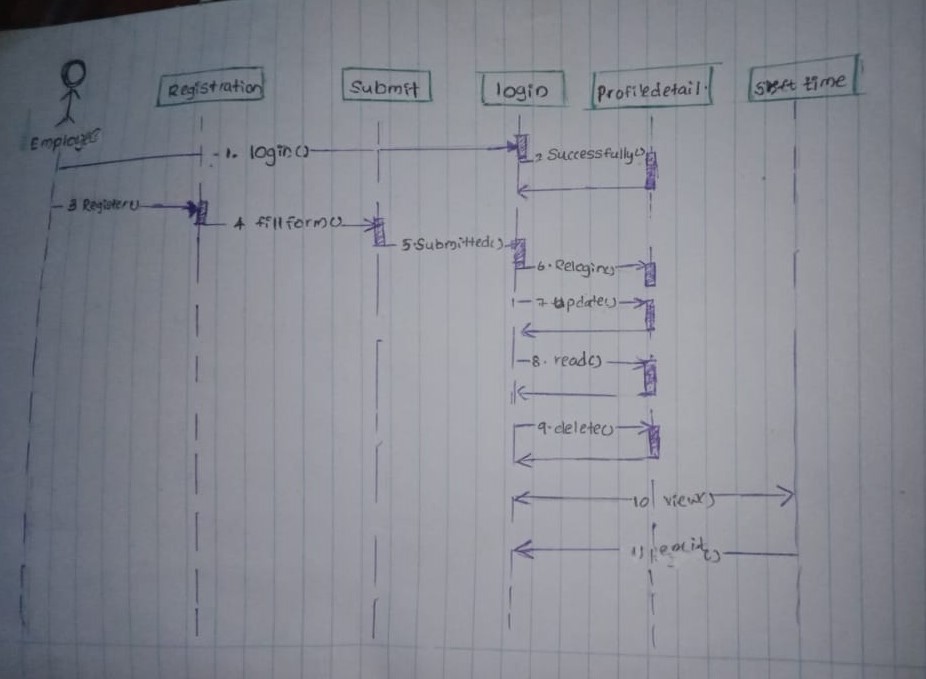


Figure sequence diagram employee

**HR Manager**

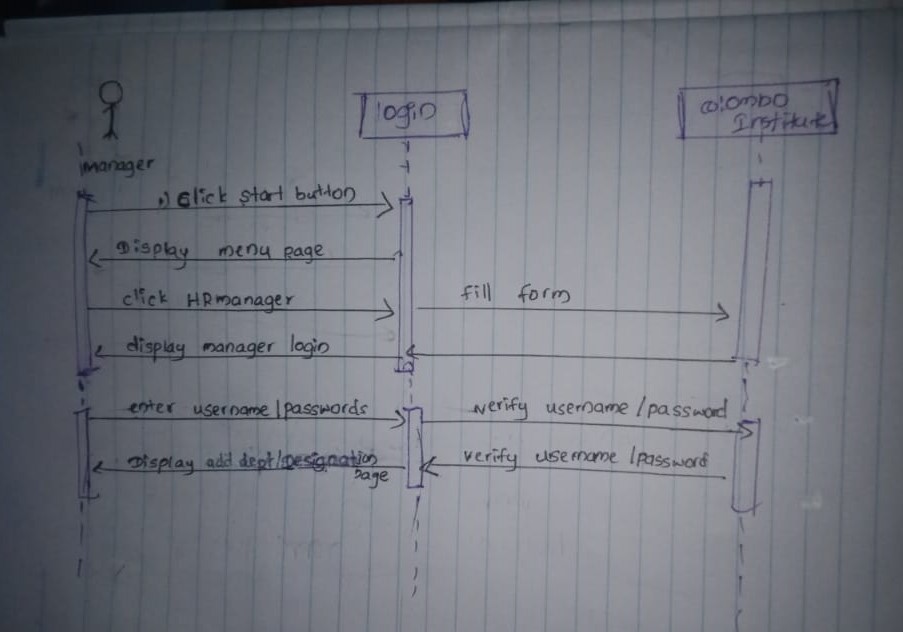


Figure . sequence diagram manager

**HR Assistant**

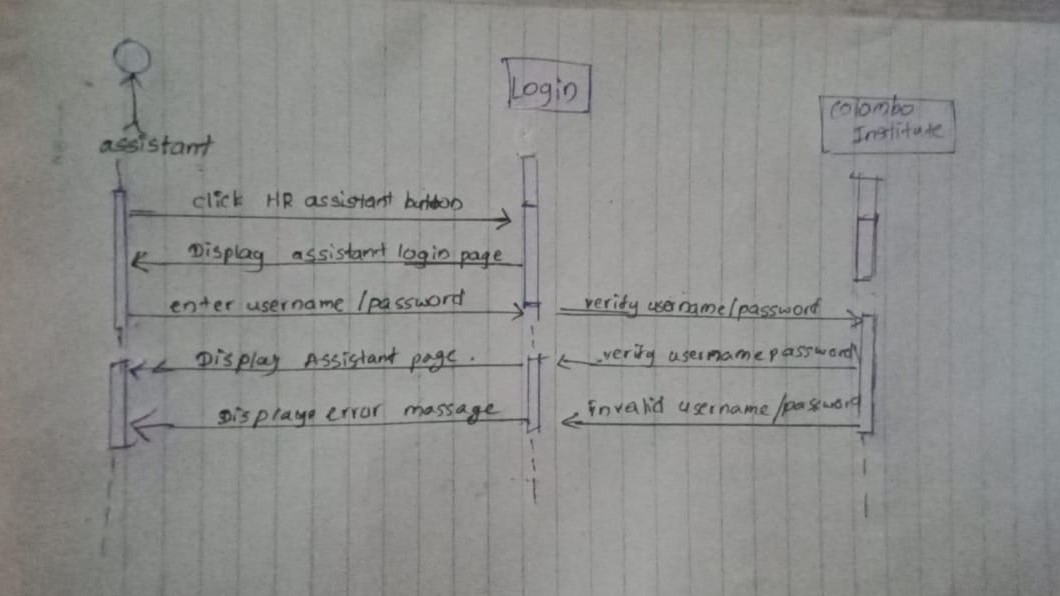


Figure sequence diagram assistant

Task 03.

**User guide of Colombo Institute of Studies**

**© Colombo Institute Of Studies 2023**

**Introduction**

Welcome to the "Colombo Institute of Studies Employee Management System" User Guide. The purpose of this tutorial is to help HR managers and administrators use the newly built system's capabilities and navigate it. The system's goal is to automate and expedite the process of maintaining employee data, allowing HR staff to work more effectively on a variety of duties.

**About the System**

The "Colombo Institute of Studies Employee Management System" is a user-friendly and easy-to-use program designed to make the duties of the HR department easier. HR managers may effortlessly add and manage departments, designations, workers, and user accounts with the system. To provide safe and restricted access to the system, administrators can make new accounts for HR Managers and HR Assistants.

UI of start page



UI of 2nd page



1. Adding a New Department:

Log in as HR Manager.

Select the "Add Department" option.

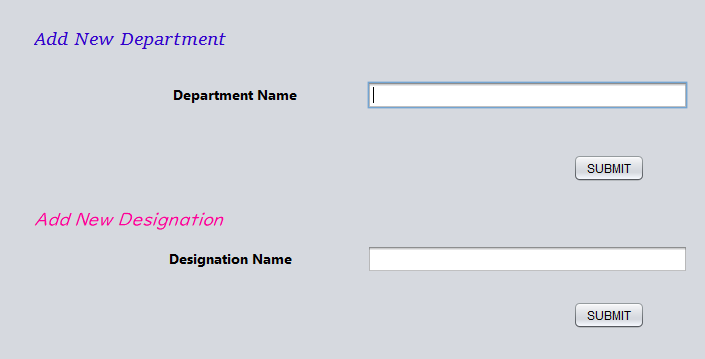
Enter the department name and submit.

1. Adding a New Designation:

Log in as HR Manager.

Select the "Add Designation" option.

Enter the designation name and submit.

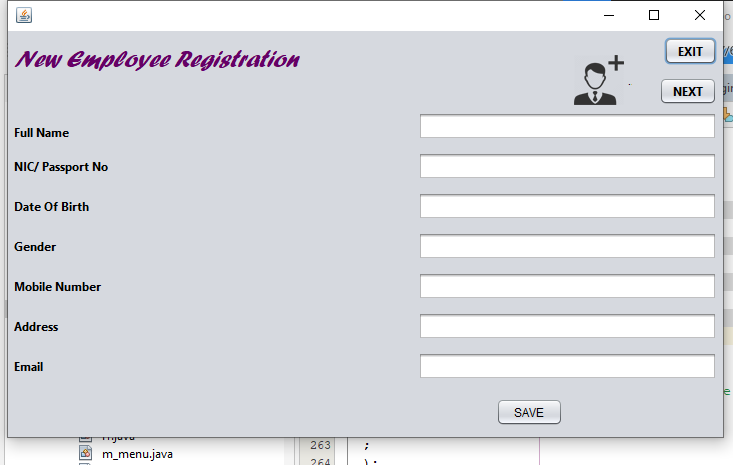


1. Adding a New Employee:

Log in as HR Manager.

Select the "Add Employee" option.

Enter employee details (name, ID, contact, department, designation) and submit.

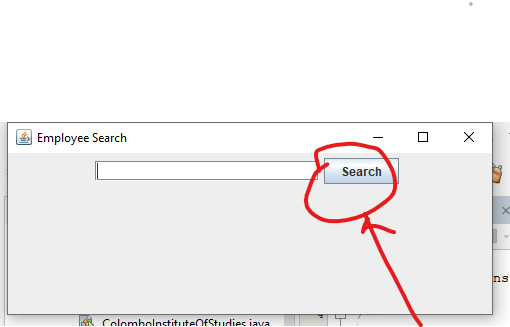


1. Searching for Employee Details:

Log in as HR Manager.

Select the "Search Employee" option.

Enter search criteria (department, designation, name, employee ID, etc.) and submit.



1. Creating a New Account:

Log in as Admin.

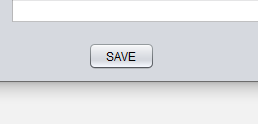
Select the "Create Account" option.

Choose the user type (HR Manager or HR Assistant).

Enter user details and submit.



Same UI for HR assistant



*Note: Ensure to save data regularly using the "Save" option to prevent data loss. Data save in a file .*

Thank you .

# References

website

Copyright © 1995, 2022 Oracle and/or its affiliates. *Purpose, java variables*[online] available at < <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html> > [accessed 21st December 2023]

© Copyright 2011-2021 www.javatpoint.com . *purpose, oop concept in java*[online] available at < <https://www.javatpoint.com/java-oops-concepts> > [accessed 22nd December 2023]

© Copyright 2023.tutorials point. *Purpose , java tutorials*[online] available at <<https://www.tutorialspoint.com/java/> > [ accessed 16th December 2023]

**Conclusion**

The provided tasks involved designing and implementing an Employee Management System for the "Colombo Institute of Studies" using Object-Oriented Programming (OOP) concepts. The tasks were divided into three main parts,

**Part A: Design Solution**

**Task 1: UML Diagrams**

a) Use Case Diagram:

Illustrated the interactions between users (HR Manager, Admin) and the system.

Use cases included adding new departments, designations, employees, searching for employee details, and creating new accounts.

b) Class Diagram:

Captured the static structure of the system, including classes, attributes, and relationships.

Represented HR Manager, Admin, Department, Designation, and Employee classes with their associations.

c) Sequence Diagram:

Depicted the dynamic interactions between objects for specific scenarios, such as adding a new employee.

Showed the flow of messages between HR Manager, Admin, and the system.

**Part B: System Development**

**Task 2: Object-Oriented System Implementation**

Developed a system using Java and Object-Oriented concepts.

Implemented functionalities such as adding departments, designations, employees, and searching for employee details.

Utilized concepts like Object, Class, Abstraction, Inheritance, Encapsulation, and Polymorphism.

Used JFrame for creating a graphical user interface.

**Task 3: User Manual**

Provided a user manual for the developed solution, including instructions for using the system.

Covered functionalities such as adding new departments, designations, employees, and searching for employee details.

Offered guidance on creating new accounts for HR Manager and HR Assistant.

A thorough approach to system design, implementation, and user documentation was required to complete these tasks. An understanding of the requirements and interactions of the system was made easier by the use of UML diagrams, and the implementation followed the guidelines of object-oriented programming. The user manual guarantees that end users can use the system efficiently, and in the event that runtime problems arise, troubleshooting support is offered.

This procedure, which includes both design and coding components, shows how software development can be done in an organized and methodical manner. UML diagrams and real code implementation work together to provide a comprehensive solution for the given situation.

