**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.

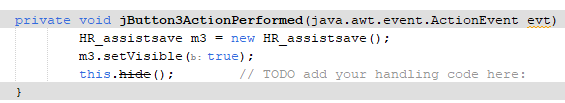
A class is a "blueprint" for constructing things, similar to an object constructor..

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.

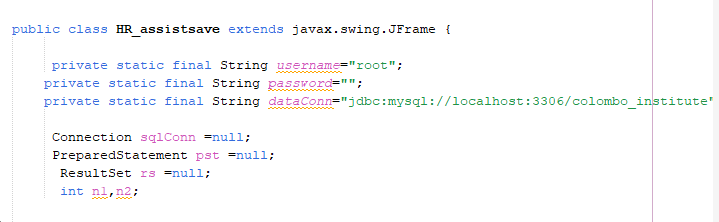


Figure  ex. for class

* Inheritance

When one object acquires all the properties and behaviors of a parent object, it is known as Inheritance.  It allows for the reuse of code. It is employed to accomplish polymorphism at runtime.

* 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 just displaying pertinent information to the user and hiding other details.

Abstract classes or interfaces can be used to accomplish abstraction.

* Encapsulation

Binding (or wrapping) code and data together into a single unit are known as encapsulation.

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

abbreviation for Unified Modelling Language, a standardized modelling language made up of an integrated collection of diagrams that was created to assist system and software developers in business modelling and other non-software systems as well as in Determining, illustrating, creating, and recording software system artefacts. The UML is a collection of cutting-edge engineering methods that have been effectively used to represent intricate and substantial systems. The UML is essential to the software development process and to the creation of object-oriented software. Software project design is mainly expressed by graphical notations in the UML. When project teams use the UML, they can explore different ideas, improve communication, and verify the software's architectural design. (what is uml diagram- guide uml. visual paradigm 2023)

**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.

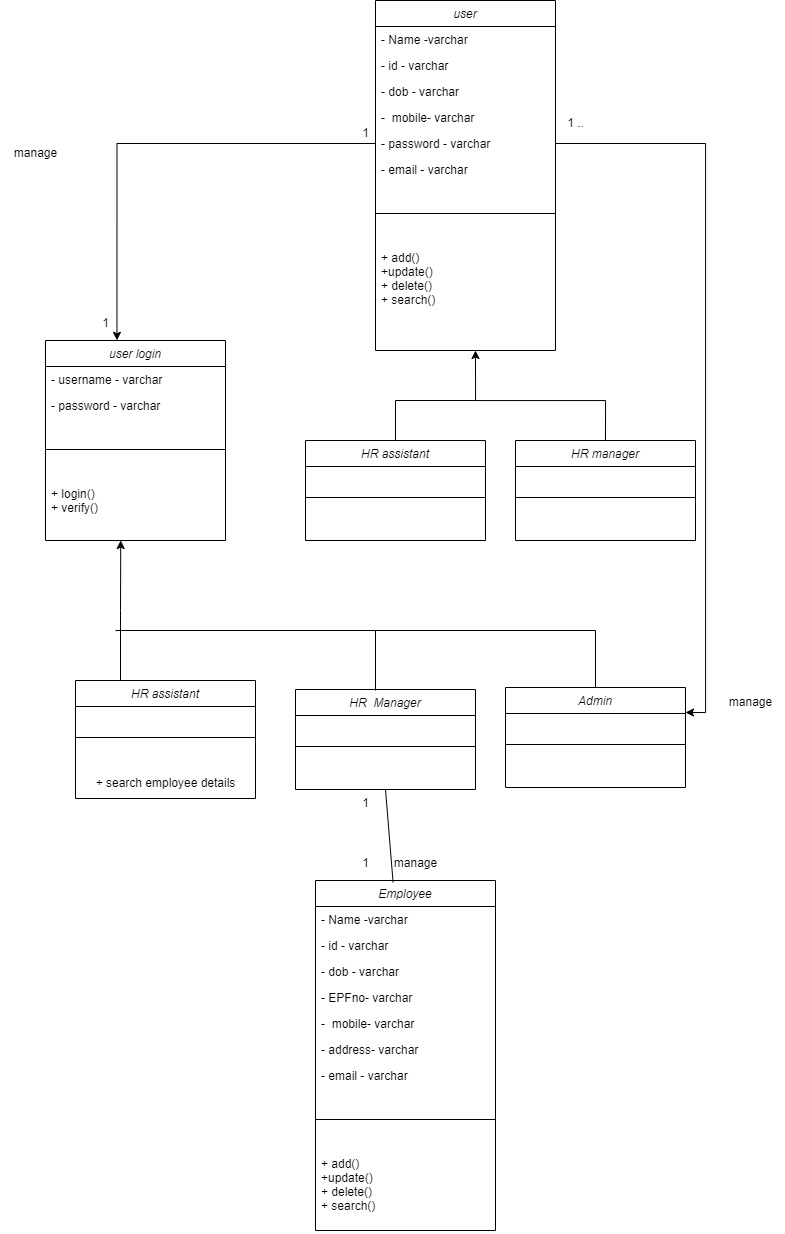


Figure  . class diagram

* Explanation

**User:**

Represents a generic user of the system.

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

**Admin:**

Use case: createAccount() - Allows the admin to create a new account for 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() - Enables the HR Manager to search for employee details based on various criteria such as department, designation, name, EPF number, etc.

**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.

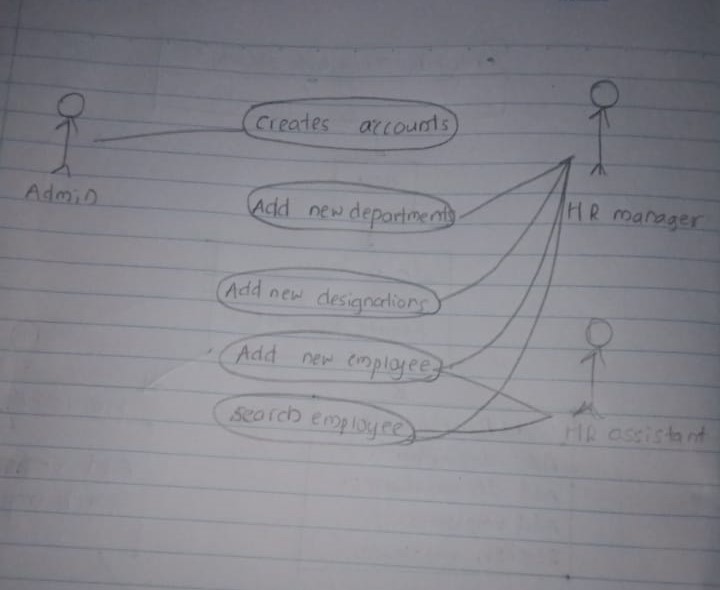


Figure 5. use case diagram

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:**

HR Manager logs into the system.

HR Manager 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

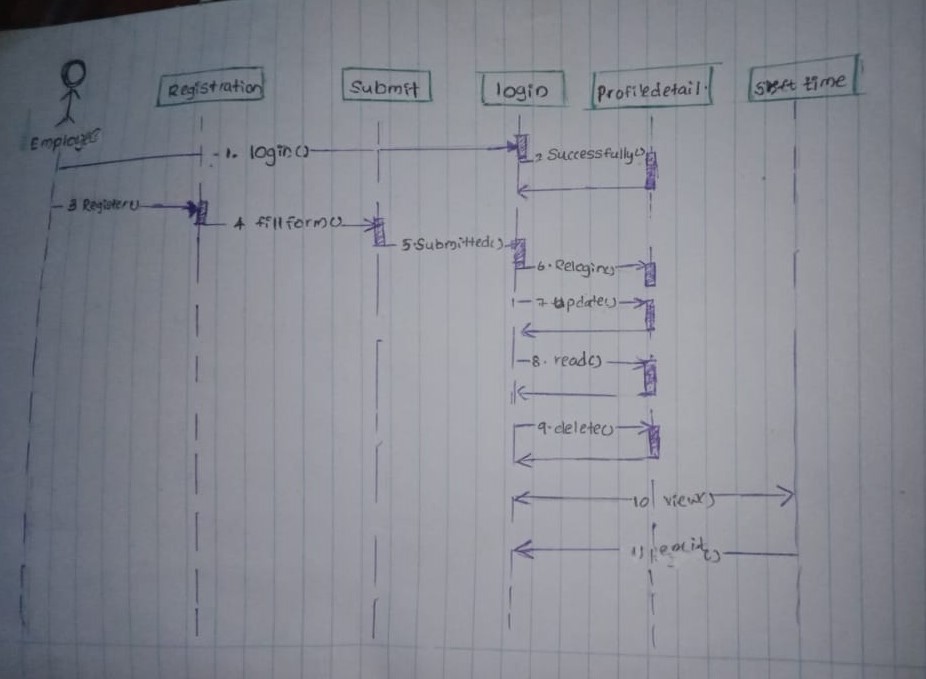


Figure  sequence diagram employee

 HR Manager

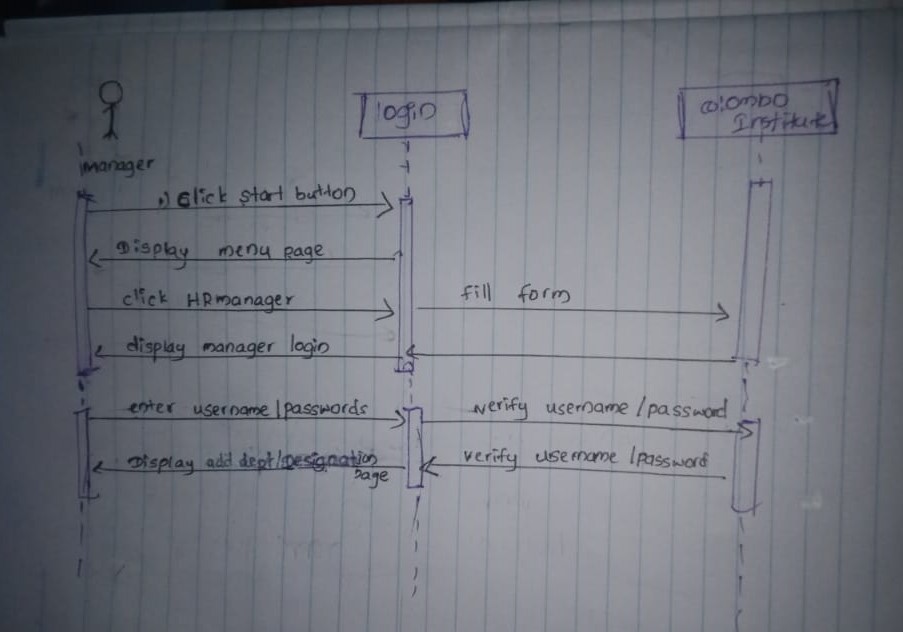


Figure . sequence diagram  manager

HR Assistant

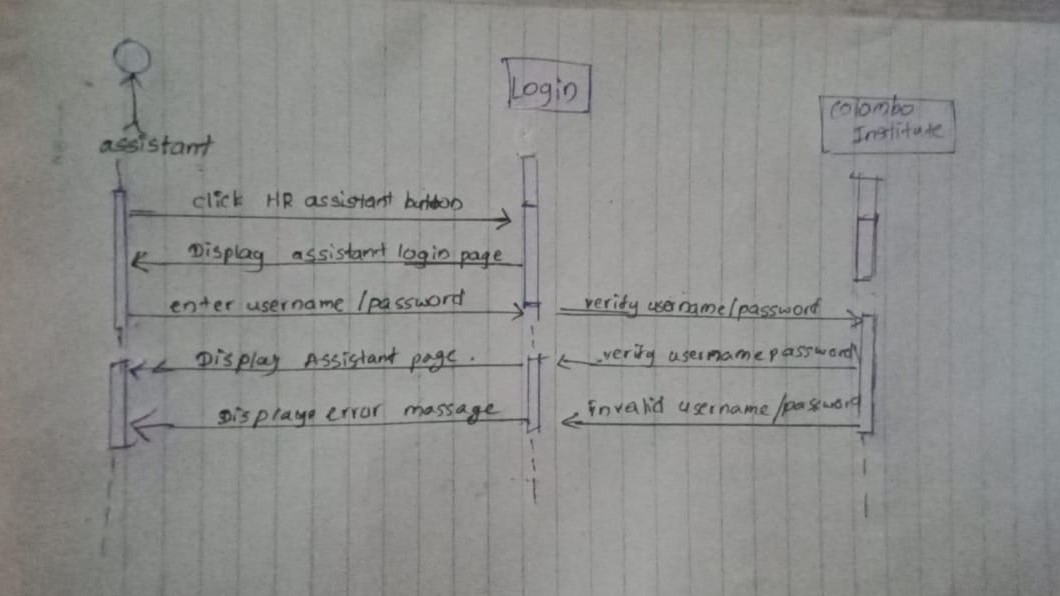


Figure  sequence diagram assistant

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.

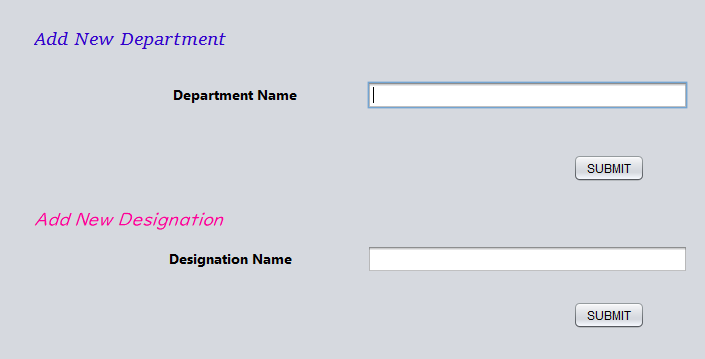
Enter the department name and submit.

1. Adding a New Designation:

Log in as HR Manager.

Select the "Add Designation" option.

Enter the designation name and submit.

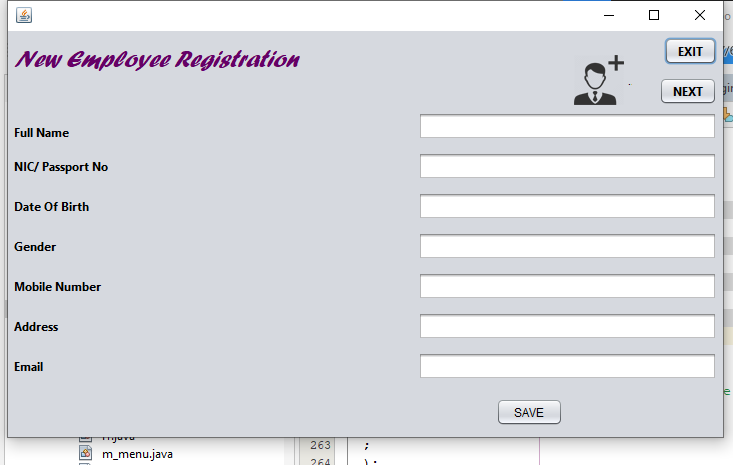


1. Adding a New Employee:

Log in as HR Manager.

Select the "Add Employee" option.

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

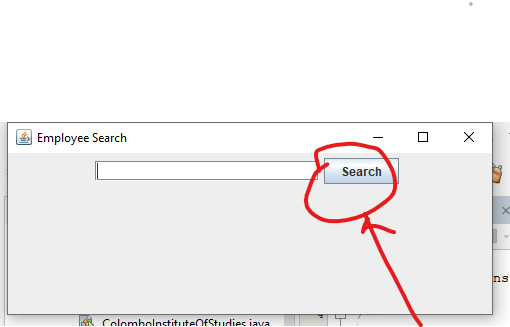


1. Searching for Employee Details:

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.

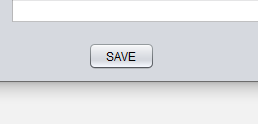
Select the "Create Account" option.

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 .