Assignment Cover Sheet

Qualification		Module Number and Title
HD in Computing/HD in Software Engineering		**Object Oriented Programming- CSE4006**
Student Name & No.		**Assessor**
M.N.M.Dilshad CL/HDCSE/CMU/112/32		Mrs.Upekha
Hand out date		**Submission Date**
Assessment type WRIT1-Coursework	**Duration/Length of Assessment Type** 3000 words equivalent	**Weighting of Assessment** 100%

Learner declaration
I 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: OOP CSE 4006

Student: M.N.M.Dilshad

Assessor: Mr.Upekha

Assignment:

Strong features of your work:

Areas for improvement:

st20282023Dilshad.docx

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes	Off	Exclude matches	< 10%
Exclude bibliography	Off		

Contents

Acknowledgement

I am appreciative of what object-oriented programming, or OOP, has brought to the field of software development. Its fundamental ideas of polymorphism, inheritance, and encapsulation have enabled programmers to write more structured, reusable, and flexible code. OOP will undoubtedly continue to be significant in the future, in my opinion. I am also appreciative of the contributions that class diagrams, sequence diagrams, use case diagrams, and OOP have made to the field of software development. Everyone who has contributed to the creation, dissemination, or use of OOP is appreciated.

Introduction

At the moment, "Colombo Institute of Studies" handles personnel details by hand. Installing plaining will automate the procedure, making it simple for the assistant and HR manager to handle personnel details in a system.

What is UML ?

UML is a standardized modeling language used in software engineering for describing, designing, and documenting software systems. Created by the Object Management Group, it offers graphical notation tools for creating visual models of software-intensive systems. UML diagrams illustrate system features, behavior, structure, and interrelationships. Some most popular UML diagrams are such as : (Sheldon, 2023).

1) Class diagram - Show the classes in a system along with their relationships with each other.
2) Use Case diagram - Show the exchanges that take place between users and the system.
3) Sequence diagram - Show how different system components interact over time.
4) Activity diagrams.
5) State machine diagrams.
6) Component diagrams.
7) Deployment diagrams.

Advantages of UML diagrams

There are many uses in UML diagram here are some of them.

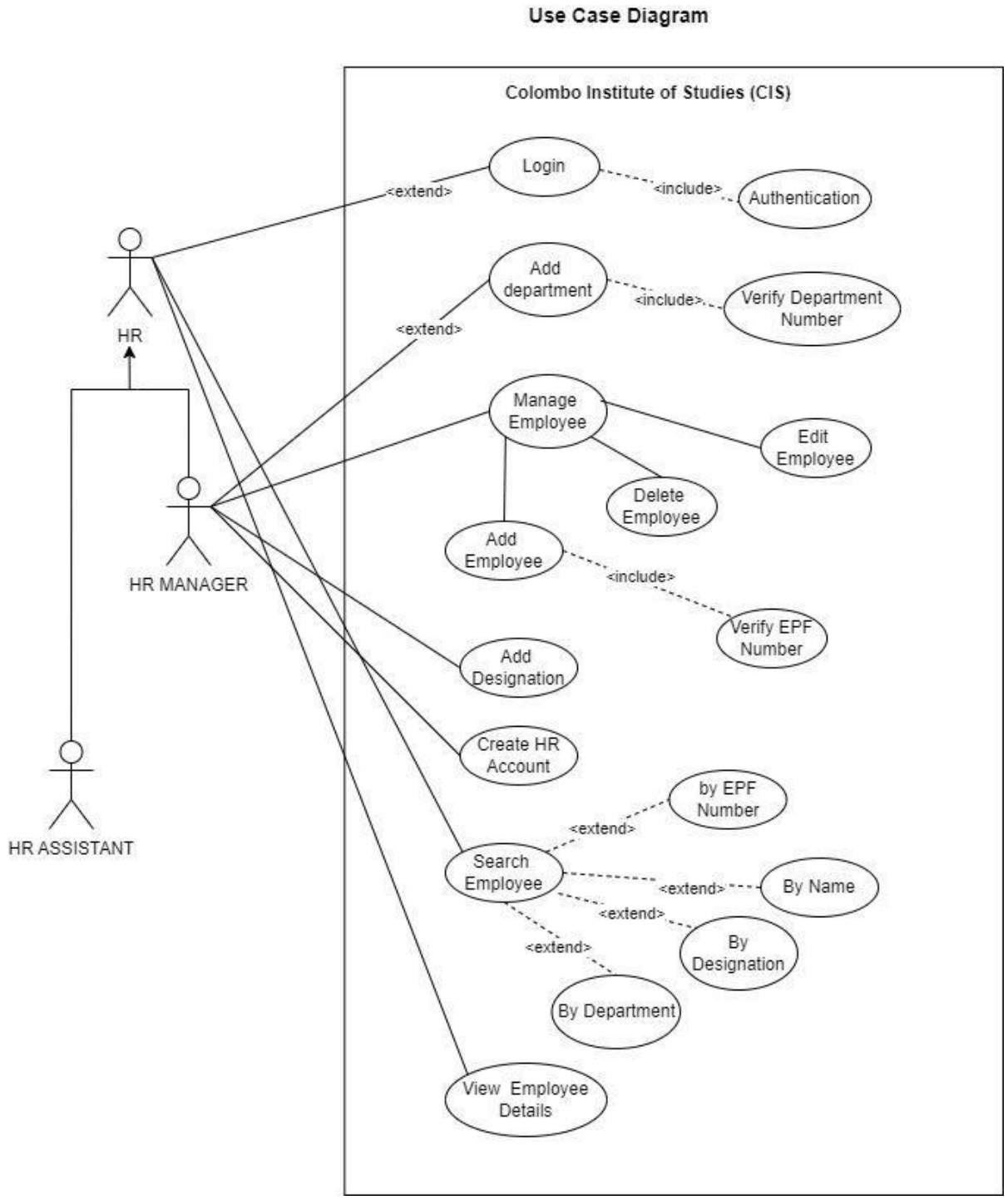
- Readable And Flexible – Easy to readable to any programmer that understands.
- Proper Communication for the Software Architecture – Helps to assessing performance of the users with tracking, security with relevant guidelines.
- Planning Tool Before Making a Program – The tool can help with generating codes to set up the model UML.

Disadvantages of UML diagrams

- Time consuming – Takes a lot of time to manage the diagrams.
- Just A Language – Only the person who studies can get the diagrams easier.
- Ascent Complexity – Some think UML diagrams can grow with complexity.

(Executive, *4 advantages and disadvantages of UML diagrams for companies* 2022).

USE CASE DIAGRAM (Colombo Institute of Studies)



In the context of automating employee management in “Colombo Institute of Studies (CIS)”, this above is the Utilize a use case diagram to illustrate how the actors—such as the HR manager and the HR assistant—interact with the CIS system. Now we will explain the actors and the use cases furthermore.

Actors in CIS:

- **HR Manager (Primary Actor):** The main user with advanced rights in charge of overseeing HR operations is the HR Manager.
- **HR Assistant (Secondary Actor):** By helping with numerous HR tasks, the HR Assistant provides support to the HR Manager.

Use Cases in CIS:

Add Departments and Designations (HR Manager):

The HR Manager can expand the present system by adding new departments and designations.

Add New Employees (HR Manager):

The HR Manager can add new employees to the system by going through the procedure of adding them with the appropriate departments and designations.

Search Employee Details (HR Manager and HR Assistant):

The HR Manager and HR Assistant look for employee information according to different needs, including name, department, and EPF number. This feature facilitates the retrieval of particular employee records ("Employee.txt").

Create New Account for HR Assistant and HR Manager:

HR Managers can work together and get assistance with HR-related duties by allowing HR Assistants to register for new accounts.

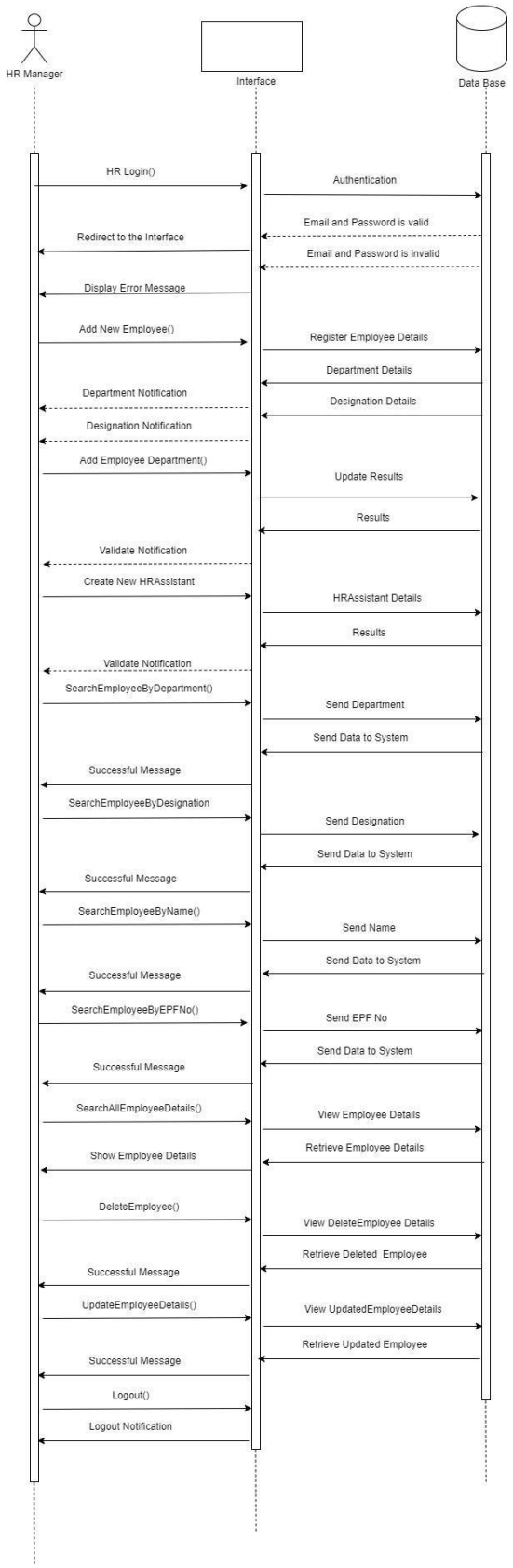
Search Employee Details (HR Assistant):

HR Assistant can search Employee details depends on department, name, designation, EPF Number and so on. This allows HR Assistant to assist employee and manage them in a clever manner.

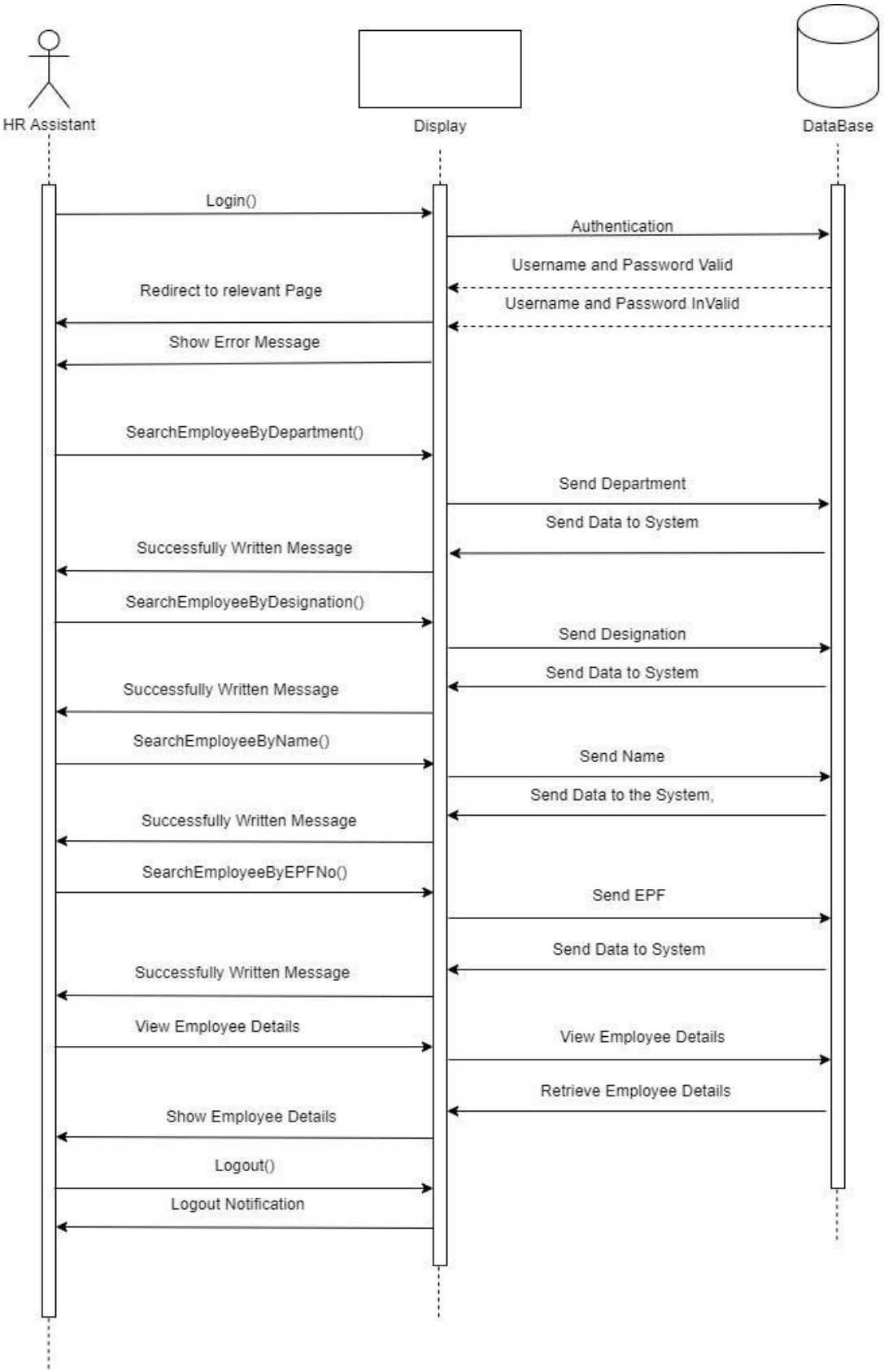
The HR Manager and HR Assistant may connect more easily thanks to automated personnel management, which gives them the tools they need to handle employee data in a smooth manner, as this use case diagram illustrates.

The solution streamlines HR processes at "Colombo Institute of Studies," guaranteeing accuracy and effectiveness in personnel management across the board for all departments and titles.

Sequence Diagram - HR Manager



Sequence Diagram – HR Assistant



In the current scenario “Colombo Institute of Studies”, where the aim is to automate their employee management process into a system. The above Sequence Diagram shows the interaction between the primary user (HR Manager) and secondary user (HR Assistant) in the database/File System.

Sequence Diagram – HR manager interacts with File System:

- The HR Manager sets off the pattern by initiating the employee management system login procedure.
- The HR Manager submits a login request to the Authentication object by entering the login credentials, which include the username and password.
- After receiving the request, the Authentication object checks the HR Manager's credentials.
- Next, in order to validate the supplied login credentials, the Authentication object sends a legitimate request to the Database/File System.
- The Database then conducts authentication by cross-referencing the provided information with stored data after receiving the validation request.
- In response to the authentication, the File System provides information, such as the validity of the HR Manager's login credentials.
- After receiving a validated response, the HR Manager continues working on the employee object. This could entail creating new divisions, new titles, or additional staff.
- While adding new raw data such as designation or department, the HR Manager request the Database/File System to store the entered information.
- The File System receives the request to store new data and update a successful message.
- Confirmation is sent from file system to HR manager, including the successful created option of the requested information.
- The HR Manager might keep working with the File System and carry out further duties inside the build system.

- The procedure comes to a conclusion when HR Manager sends a logout request to File System, starting the logout process.
- File System processes the request and updates HR Manager's status in the system accordingly.
- The HR Manager receives a confirm answer from the File System verifying that the logout process was completed successfully.

The sequence illustrates the dialogue that takes place step-by-step between the HR Manager and the Database/File System when the "Colombo Institute of Studies" automates staff management. The diagram demonstrates how the manager can log out, add a new employee, department, or designation, and access the system all while guaranteeing data security and integrity with the confirm message.

Sequence Diagram – HR Assistant Interacting with Database/File System:

- The HR Assistant starts the sequence by starting the employee management system and triggering the login procedure.
- The HR Assistant logs in to the system using their login information, which consists of a password and username.
- The HR Assistant initiates a login request to the system's object Authentication upon inputting the credentials.
- After receiving the login request, the Authentication object verifies the HR Assistant's registered credentials.
- The authentication object establishes a connection with the database or file system and sends a request to validate the entered password and username.
- After receiving the validation request, the File System verifies identity by comparing the supplied credentials with the information that has been stored.
- If the HR Assistant's login credentials are valid for the confirmation, a dialog box will be generated by the database and delivered back to the authentication object.
- The HR Assistants make a request for a search with the desired search parameters to the Employee Information object.
- The search request is received by the Employee Information object, which prepares the database/file system query.
- The search formula containing the designated criteria is transmitted by the Employee object via communication with the database.
- The Database receives the search formula and initiates a search operation, looking for the employee records to match the provided search criteria (such as department, name, EPF Number).
- Employee details that fulfill the following criteria are included in the search results that are compiled by the database.
- The Employee stored Information object receives a response from the File System that contains the employee details that meet the search criteria.

- The response from the database containing the employee data that was fetched is received by the Employee Information object.

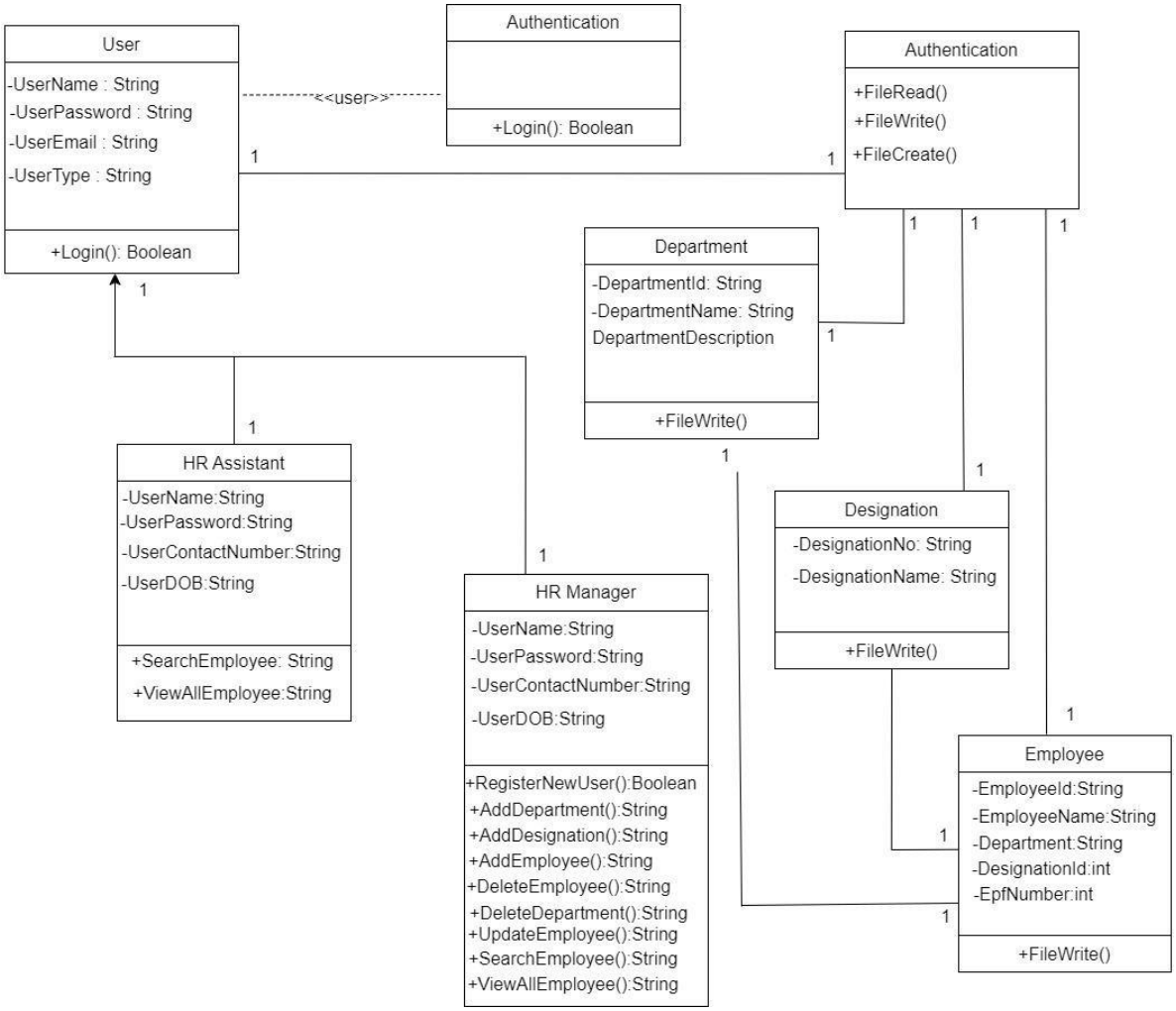
- The HR Assistant chooses the logout option from the user interface of the (CIS) system to start the logout procedure.

- The HR Assistant notifies the object Authentication of their plan to log out by sending a logout request.

- The object Authentication notifies the HR Assistant of the successful logout after obtaining the confirmation.

- Upon finishing their work and logging out, the HR Assistant terminates their interaction with the personnel management system.

Class Diagram



User Class

Description:

The User Class represents individuals who interact with the current system, including both primary user (HR Manager) and secondary user (HR Assistant).

Attributes:

userId (String): A unique identity assigned to each user.

userEmail(String): The email address associated with the user.

userPassword(String): The password used for login to the system.

Methods:

1) AuthenticatorUser():

This method that checks the user's login information to confirm their authorization to access the system.

2) CreateUserAccount():

This method is to create new user accounts which includes roles such as HR Manager and HR Assistants.

HR Manager Class

Description:

This class represents, HR Manager who oversee employee management operations.

Attributes:

ManagerId (String): A unique identify number assigned to each different HR Manager.

ManagerName (String): The name of the HR Manager.

ManagerBirthOfDate (Int): The DOB of each manager.

ManagercontactNumber (Int): The Contact Number of each manager.

Methods:

1) CreateAccount(): Allows HR Managers to create new user accounts with the current system, including accounts for HR Assistants or any other roles in the system.

2) AddNewEmployee():permits the HR Manager to enter new workers into the system and provide them departmental and title assignments.

3) SearchEmployeeDetails(): Allows HR Managers to search for employee details based on various criteria such as EPF Number, designations, department , name etc.

HR Assistant Class

Description:

The HR Assistant class represents HR Assistants who support HR Managers in employee management tasks.

Attributes:

HRAssisatntID (String): A unique identification number assigned to uniquely identify each HR Assisatnts.

HRAssisatantName (String): The name of the HR Assistant.

HRAssistantDOB (Integer): The Birth Of Date of the HR Assistant.

HRAssistantContactNO. (Integer): The Contact Number of HR Assistant.

Methods:

SearchEmployeeDetails(): Employee information can be found by HR Assistant using parameters such as name, department, classification, EPF Number, etc.

Department Class

Description:

The Department class represents the various Department available in the Colombo Institute of Studies.

Attributes:

DepartmentId (String): A unique identifier for each Department.

DepartmentName(String): The name of the department.

Category (String): The Category to which department includes.

Methods:

None specified in the diagram, but additional methods could be include to handle Department related operations.

Designation Class

Description:

The Product Category class represents different categories or types of product available in the "Colombo Institute of Studies".

Attributes:

DesignationId (String): A unique identifier for each Designation.

DesignationName (String): The name of the Designation.

DesignationDescription(String): A description of the Designation.

Methods:

None specified in the diagram, but additional methods could be added for Designation class if needed.

Authentication Class

Description:

The Authentication class manages the authentication process for system users.

Methods:

AuthenticateUser(): Validates the user's login information to confirms their authorization access to the system.

CreateUserAccount(): Created new user accounts with specified roles (such as : HR Assistant, HR Manager).

File System Class

Description:

The File system class manages data storage and retrieval for the system.

Methods:

SaveData(): Saves the data, such as user information or product information, to a file (Notepad).

RetrieveData(): Retrieves stored data from the file, such as product or user information.

The above class diagram provides a comprehensive overview of the structural elements and relationships within the automated employee management system at "Colombo Institute of Studies". It shows how different classes interacts to support HR tasks, user authentication, and data management effectively.

TASK 02

Main Functionalities

Now let us shows the Main functions in the Colombo Institute of Studies. Such as:

1) Login.
2) Registration of users.
3) Add department.
4) Add Employee.
5) Update Employee.
6) Delete Employee.
7) Search Employee.
8) View All Employee.
9) Email validation. (Username)

Login.

```
public void setUserType(String usertype) {
    this.userType = userType;
}

public boolean login() {
    try (BufferedReader reader = folderUser.readfile()) {
        String line;
        while((line = reader.readLine()) != null){
            String[] userInfo = line.split(",");
            String storedEmail = userInfo[0];
            String storedPassword = userInfo[1];
            String storedUserType = userInfo[2];

            if(getEmail().equals(storedEmail) && getPassword().equals(storedPassword))
            {
                setUserType(storedUserType);
                return true;
            }
        }
    } catch (IOException e)
    {
     System.out.println("Error Validating User " + e.getMessage());
    }
    return false;
}
}
```

To login to the application, users enter their email address and password into the login Jframe form. The application then checks the entered information against the user data stored in the database. If the credentials match, the user is granted access to the application with a successful message.

Register of New Users

```
public boolean registerNewUser()
{
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("User.txt", true))) {
        String line = manager.getEmail()+ "," + getUserName() + "," + manager.getUserType() + "," +
                manager.getPassword();
        writer.write(line);
        writer.newLine();
        return true;
    } catch (IOException e) {
        System.out.println("Error inserting data: " + e.getMessage());
        return false;
    }
}
```

- The HR Manager need to enter the valid information to the system. This entered information should includes the username, email address, password, contact number, Birth of Date and the user type.
- Then the system validates the information. It means that system confirm that information entered is correct and the user type is valid.
- If the given information is valid, a new user record is created and saved in the Fie System in the system. That record contains the information that the HR Manager entered, as well as the date and time that the user was created in a detail manner.
- The HR Manager can then view the user record in the system.
- The HR Manager can also update the information in the record or delete the record together.

Add Department

```
public int addDepartment() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("Department.txt", true))) {
        String line = getDepartmentId() + "," + getDepartmentName() + "," + getDepartmentInformation();
        writer.write(line);
        writer.newLine();
        return 1;
    } catch (IOException e) {
        System.out.println("Error inserting data: " + e.getMessage());
        return 0;
    }
}
```

HR Manager can add new Department as referred to the system using this Add New Department function.

The DepartmentId, Name, Description are all needed to add to the function. To confirm the accuracy and the consistency, The data input is verified. Following validation, a new Department record is produced in the File System. (Department.txt)

Add Employee

```
// Add New Employee

public int addNewEmployee() {
    int noRows = 0;
    String employeeData = getEmployeeId()+ "," + getEmployeeName()+ "," + getEmployeeDepartment()+ ","
            + getDesignations()+ ","
            + getEpfNumber()+ "\n";

    try (FileWriter fileWriter = new FileWriter("Employee.txt", true)) {
        fileWriter.write(employeeData);
        noRows = 1;
    } catch (IOException ex) {
        System.out.println("Error inserting employee data " + ex);
    }
    return noRows;
}
```

HR Managers can utilize the Add Employee function to add new staff. This function gathers employee data, including EPF number, department, employee ID, and designation ID. The entered data is reviewed to ensure that it meets the necessary criteria. A new employee record is created and saved in the file system if the data is valid.

Update Employee

```
    public int deleteEmployee() {
        int noRows = 0;
        String employeeId = getEmployeeId();
        try {
            File inputFile = new File("Employee.txt");
            File tempFile = new File("temp.txt");
            BufferedReader reader = new BufferedReader(new FileReader(inputFile));
            BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));

            String line;
            while ((line = reader.readLine()) != null) {
                String[] employeeData = line.split(",");
                if (employeeData[0].equals(employeeId)) {
                    // Skip deleting the line with matching bookId
                    continue;
                }
                writer.write(line);
                writer.newLine();
                noRows++;
            }
            writer.close();
            reader.close();

            boolean successful = tempFile.renameTo(inputFile);
            if (!successful) {
                // Unable to rename temporary file, reset noRows to 0
                noRows = 0;
            }
        } catch (Exception ex) {
            System.out.print("Error deleting employee: " + ex);
            noRows = 0;
        }
        return noRows;
    }
}
```

The Update employee function allows HR Managers to change the details of stored employee. The file System containing information is retrieved based on the provided employee Id. A form with pre-filled information for the selected employee is displayed, allowing manager to change the details. The relevant employee record in the filesystem is updated after the confirmation.

Delete Employee

```
public int deleteEmployee() {
    int noRows = 0;
    String employeeId = getEmployeeId();
    try {
        File inputFile = new File("Employee.txt");
        File tempFile = new File("temp.txt");
        BufferedReader reader = new BufferedReader(new FileReader(inputFile));
        BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));

        String line;
        while ((line = reader.readLine()) != null) {
            String[] employeeData = line.split(",");
            if (employeeData[0].equals(employeeId)) {
                // Skip deleting the line with matching bookId
                continue;
            }
            writer.write(line);
            writer.newLine();
            noRows++;
        }
        writer.close();
        reader.close();

        boolean successful = tempFile.renameTo(inputFile);
        if (!successful) {
            // Unable to rename temporary file, reset noRows to 0
            noRows = 0;
        }
    } catch (Exception ex) {
        System.out.print("Error deleting employee: " + ex);
        noRows = 0;
    }
    return noRows;
}
```

Managers can delete an existing employee from the system using Delete Employee function. The system first retrieves the employee information (Employee.txt) from the file system based on the registered employee Id. The system then prompts the manager to confirm the delete option. If the manager confirms the deletion the Employee record us deleted from data base permanently.

Search Employee

```
public String[] searchEmployee() {
    String[] employeeData = null;
    try {
        String employeeId = getEmployeeId();
        String line;
        BufferedReader reader = new BufferedReader(new FileReader("Employee.txt"));
        while ((line = reader.readLine()) != null) {
            employeeData = line.split(",");
            if (employeeData[0].equals(employeeId)) {
                // Found matching book
                break;
            }
        }
        reader.close();
    } catch (Exception ex) {
        System.out.print("Error searching employee: " + ex);
    }
    return employeeData;
}
```

In the Search Employee option it allows the HR Manager or HR Assistant to find a specific employee by their name, employee Id, or EPF Number. Then the system retrieves the employee information (Employee.txt) from the file system based on the given detail. If there is a matching employee found the details will be display if the details are invalid it shows a error message.

View All Employee

```
public String viewAllEmployee()
{

    String employees = "";
    String employeeDetails = "";
    String[] words = null;

    try ( BufferedReader emp = new BufferedReader(new FileReader("Employee.txt"))) {
        while ((employees = emp.readLine()) != null) {
            words = employees.split(",");
            employeeDetails += words[0]+"\t"+words[1]+"\t"+ words[2]+"\t"+words[3]+"\t"+words[4]+"\n";
        }
    }
    catch (IOException ex) {
        System.out.println("Error viewing employee data " + ex);
    }

    return employeeDetails;
}
```

A list of every employee in the system is shown by using the View All Employees function. It accesses the file system to obtain the personnel data. Details including Employee ID, Designation, Name, Department, and EPF Number are visible to users.

Email Validation.(Username)

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    boolean valid=true; // Check user input fields are empty
    if (txtEmail.getText().toString().isEmpty() || txtPassword.getText().toString().isEmpty()) {
        JOptionPane.showMessageDialog(null,
            "Text Fields Shouldn't Be Blank",
            "The Teach Company", JOptionPane.ERROR_MESSAGE);
        valid = false;
    }

    if(valid)
    {
        String email = txtEmail.getText().trim();
        String password = new String(txtPassword.getText()).trim();
        boolean userValid = validateUser(email, password);

        if (userValid) {
            JOptionPane.showMessageDialog(rootPane,
                "Hey, Welcome Back " + getUserType(email));
            String userType = getUserType(email);

            if (userType.equals("HR")) {
                new ManagersUI().setVisible(true);
                this.setVisible(false);
            }
            else if (userType.equals("HR Assisten")) {
                new SystemUsersUI().setVisible(true);
                this.setVisible(false);
            }
        }
        else {
            JOptionPane.showMessageDialog(rootPane,
                "Inserted details are not valid, " +
                "please try again with a valid email address or password",
                "Colombo Institue of Education", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

This feature verifies the validity of the email. It matches a given character pattern using a regular expression. Both lowercase and uppercase letters, digits, and special characters such as '_', '#', '$', '%', '&', '”', '*', '+', '/', '=', '?', '^', '.', and '-' are all allowed in this pattern. Additionally, by looking for a combination of letters, numbers, and hyphens, followed by a dot and a two- to seven-letter top-level domain (TLD), it verifies that the email has a legitimate domain name. The method returns true if the email string fits the pattern; if not, it returns false.

\

Object Oriented Concepts

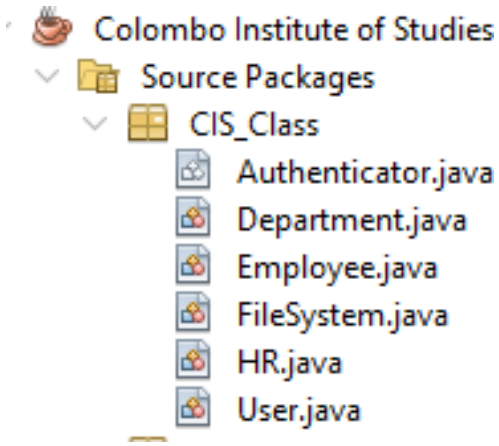
There are many types of OOP concepts. Here are some of them.

1) Class.
2) Objects.
3) Abstraction.
4) Polymorphism.
5) Inheritance.
6) Encapsulation.

Class Concepts

```java
package CIS_Class;

import java.util.ArrayList;
import java.io.*;

public class Employee {

    FileSystem createfolder = new FileSystem ("Employee.txt");

    private String employeeId;
    private String employeeName;
    private String employeeDepartment;
    private String designations;
    private int epfNumber;
```

```java
package CIS_Class;
import java.io.* ;
import java.util.*;

public class Department
{
 private String departmentId , departmentName , departmentInformation;

 public Department ()
 {

 }
```



Code defines instance variables like EmployeeId, Name, EPF Number, department, and designation in the Employee class. To manage access to these variables, getter and setter methods have been supplied. The class also contains methods for updating employees, adding new employees, deleting employees, and automatically generating IDs.

Code that defines instance variables, like department Id, department Name, and department description, is found in the Department class. We have supplied getter and setter methods for these variables, just like we did with the Employee class. The class has methods to enter departments, retrieve department names, and generate an ID automatically.

Objects

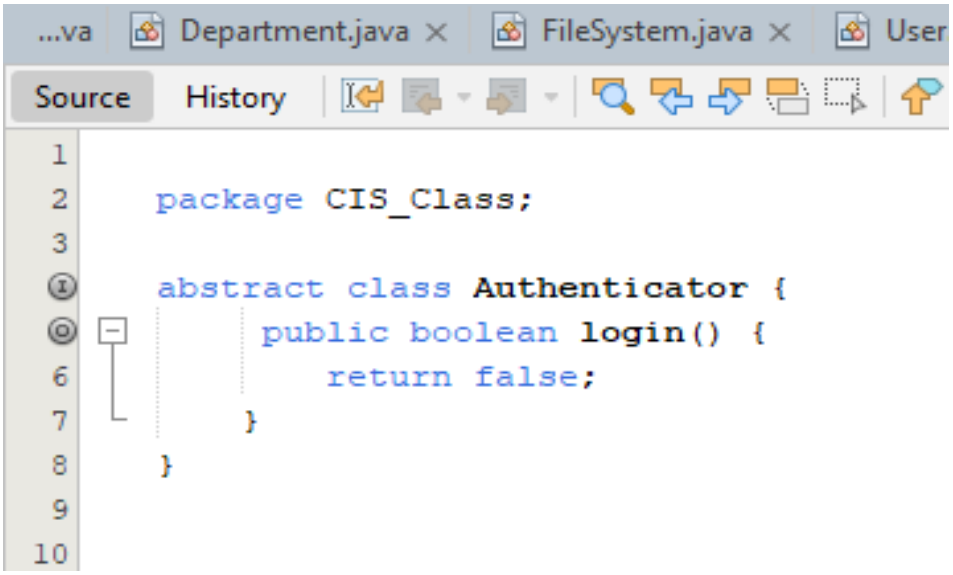
```
package CIS_Frame;

import CIS_Class.Employee;
import CIS_Class.Department;
import java.io.*;
import javax.swing.*;
import java.util.regex.Pattern;


public class ManagersUI extends javax.swing.JFrame {

   Department category = new Department();
   Employee employee = new Employee();
    public ManagersUI() {
        initComponents();
    }
```

Class names for different object types are represented by the Department classes. We can create instances of these classes, or the real objects, by using the new keyword that invokes the class constructors.

References to the recently produced objects are stored in the department object variables. The related class defines the state (data) and behavior (methods) of these objects.

\

Abstraction



We employ an abstract class named Authenticator to accomplish abstraction. This lesson outlines the standard operating procedures and authentication techniques. The Authenticator class cannot be instantiated directly because it is abstract. Rather, it is limited to usage as the basic class for subclasses, like User.

Polymorphism

Start Page | User.java

Source | History

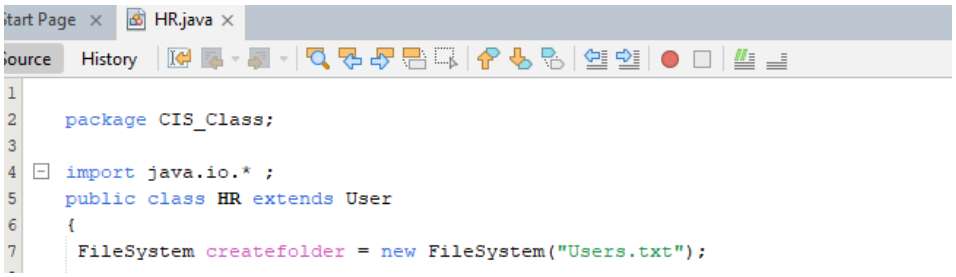
```
1
2    package CIS_Class;
3
4    import java.io.BufferedReader;
5    import java.io.IOException;
6
8    public class User extends Authenticator
8    {
9      FileSystem folderUser = new FileSystem("User.txt");
10
11   private String email , password , userType;
12
```

Source | History

```
43       this.userType = userType;
   }
44
46   public boolean login() {
46       try (BufferedReader reader = folderUser.readfile()) {
47           String line;
48           while((line = reader.readLine()) != null){
49               String[] userInfo = line.split(",");
50               String storedEmail = userInfo[0];
51               String storedPassword = userInfo[1];
52               String storedUserType = userInfo[2];
53
54               if(getEmail().equals(storedEmail) && getPassword().equals(storedPassword))
55               {
56                   setUserType(storedUserType);
57                   return true;
58               }
59           }
60       } catch (IOException e)
61       {
62        System.out.println("Error Validating User " + e.getMessage());
63       }
64       return false;
65   }
66   }
```

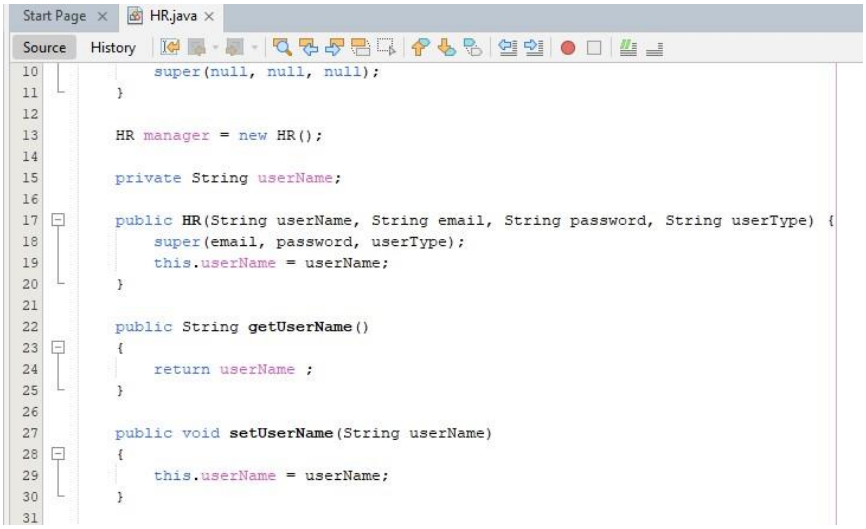
An example of polymorphism is the login() method. Any class that inherits from Authenticator must supply its own implementation of the login () method since the Authenticator class specifies it to be an abstract method. We give a customized implementation of the login () function in the User class, which is suitable for the user authentication procedure.

Inheritance



By utilizing the keyword extends to extend the User class in the HR class, the code makes use of inheritance. This enables all of the public and protected members (fields and methods) of the User class to be inherited by the HR class. The HR class can reuse and expand the functionality defined in the User class by deriving from it.

Encapsulation



Utilizing access modifiers like private, protected, and public to limit class members' visibility and access allows for encapsulation. The HR Class's state and behavior are encapsulated in the code through the right usage of private and public access modifiers.

For instance, the UserName property can only be accessible within the HR class because it is designated as private. However, in order to give restricted access to the UserName field from outside the class, the code has also included public getter and setter methods (getUserName() and setUserName()). This adheres to the encapsulation concept, which states that an object's internal state is safeguarded and only accessible via specific methods.

TASK 03

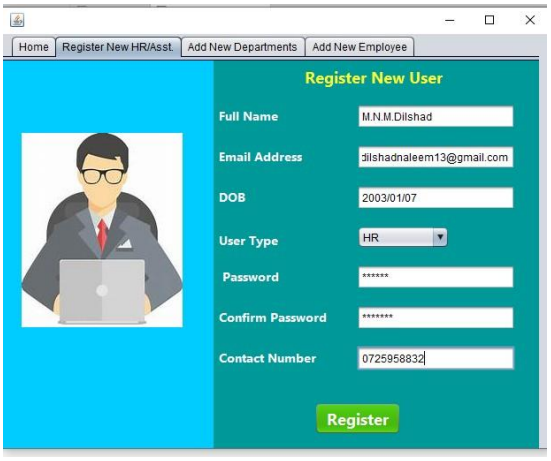
User Manual

The HR Manager and HR Assistants can register users and log in and out using the components and functionalities of the user interface (UI) form.

- To register, a user clicks the “Register” button and provided their login, email address, DOB, Contact Number, password and confirm password.
- HR Managers can login by entering their userID and the password. When logged in the access to UI forms designed specifically for HR Managers,
- HR Assistants can login by entering their userID and the password. When logged in the access to UI forms designed specially for HR Assistants.
- In the process of login, user need to select their user type (HR Manager and HR Assistant). This information is used to direct users to the appropriate UI interface.
- HR Managers and HR assistants can log out of the system by clicking the “Logout” button. This ends the current session and returns the user to the login page or a predetermined destination page.

Register HR Manager/ HR Assistant

- To access the system at the first step we need to register as HR Assistant in the Interface with the name, password, email address, contact number and date of birth.



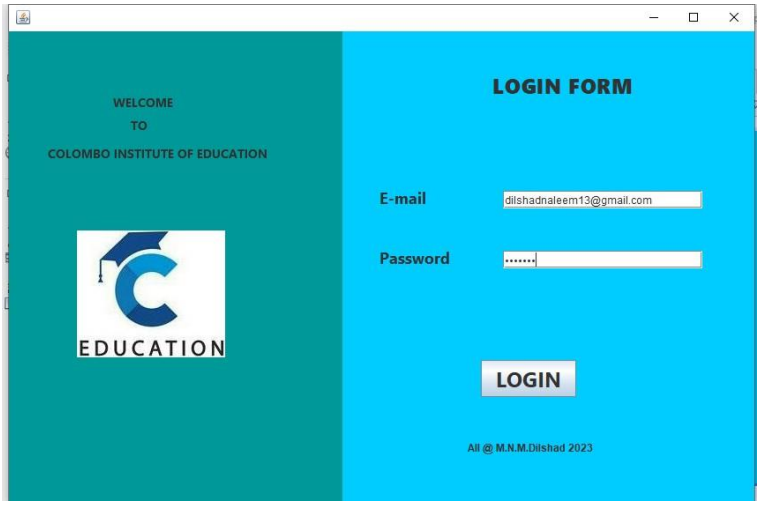
- Then click the Register button to register the user in to the system. Then a confirm dialog box appears “New HR added Successfully”.
- In the same procedure we can register the HR Assistant to by only changing the user type option. Then a confirmation dialog box appears “New HR Assistant added Successfully”.

EX: E-mail : dilshadnaleem13@gmail.com

Password: 1234567.

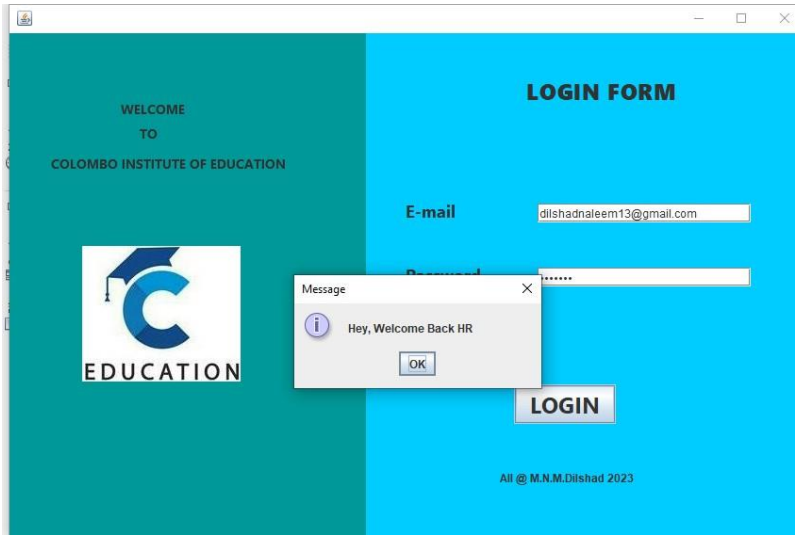
- If any form is empty it gives a error message “Text Fields cannot be Empty, Please enter valid details.”

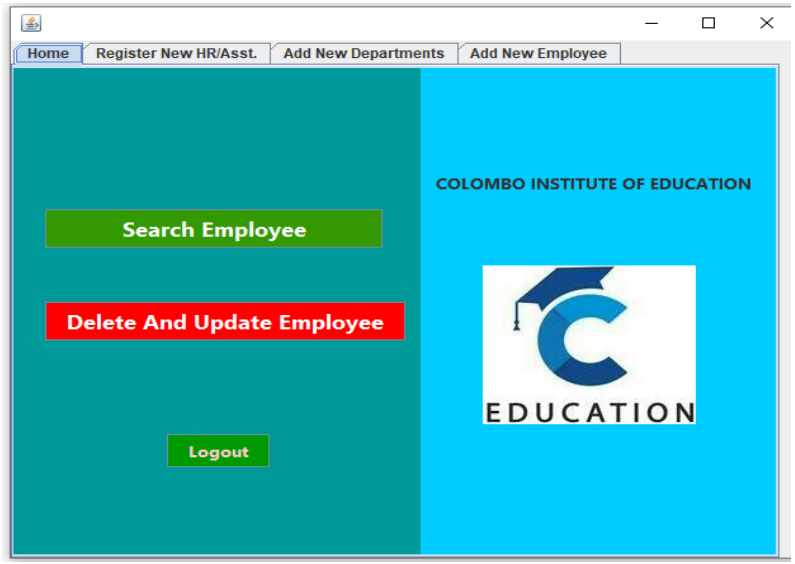
Login



An essential part of the application that enables users to log in and access their individual accounts is the login form. By guaranteeing that only authorized users may access the system using their username and password, it offers a secure entry point for the program. The system verifies a user's credentials when they submit the login form by comparing the registered email address and password to the information entered into the system. The user can access the program and carry out their assigned activities according to their role with a greeting message if the validation is successful. “Hey, welcome back HR” if the username or password is incorrect is shows an error message “Inserts details are not valid, please try again with a valid email address or password” if the both text fields are empty it shows “Text fields shouldn’t be blank” from the system for the high security.

HR Manager and their functions





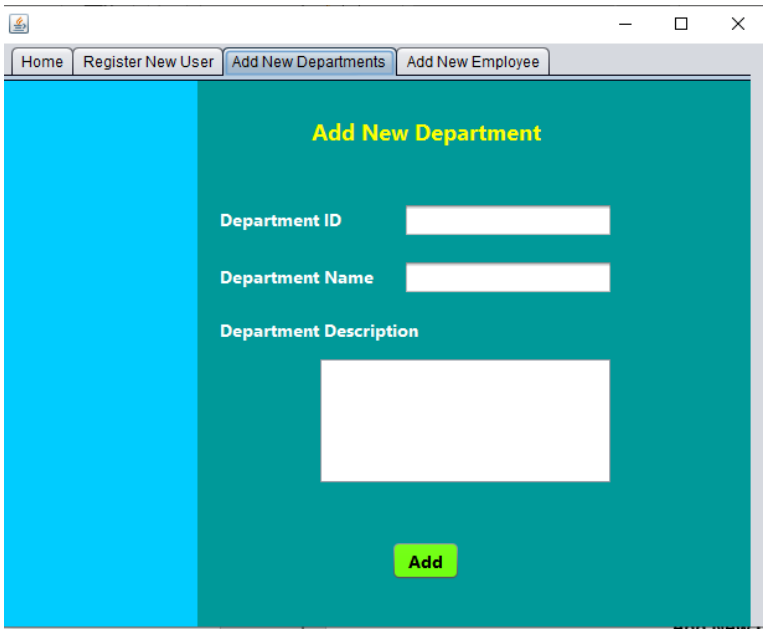
The HR Manager user interface provides all the functionalities that HR Managers need to perform their tasks. These includes registering new users, adding new department designations, and adding new employees. The HR Manager can also access the HR Assistant section. This section allows the HR Manager to check the entered data for accuracy before storing it in the system's database. The HR Manager can also delete and update departments in this section. At last HR manager can logout of the system.

Register New Users

The “Register New User” function allows HR Managers to create new user accounts in the system. Each account will have a unique username and password, which users will use to log in to the system

When the user information has been entered, the HR Manager should click the “Register” button to create a new account. The new user can access the system with their email and the password. If the text fields are empty it shows an error message when the text fields are filled with valid details it shows a dialog box that confirming of the new user.

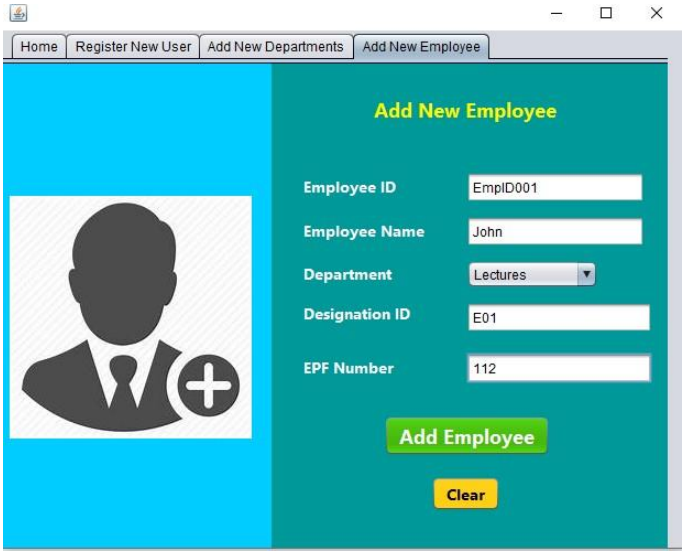
Add New Department



The “Add New Department” functionality allows HR Managers to create new departments in the system. Each department will have a unique Id, name and the description which will be used to track the manage employees in the department.

Once the department information has been entered, the HR Manager can click the “Add” option to add the department to the system with a confirmation message. If the text fields are empty it shows an error message.

Add New Employee

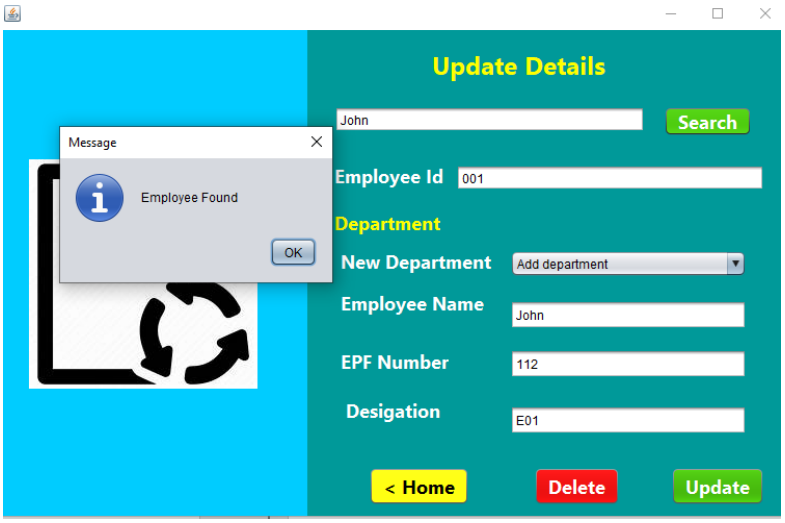


The “Add New Employee” function allows HR Managers to create new employee records in the system. Each employee record will have a unique employee ID and name which will be used to manage the employee.

After the filling the above form with the valid details the HR should click “Add Employee” button to register the employee to the system with a confirmation message if the text fields are black then it shows an error message.

The button “Clear” can delete the filled details to default.

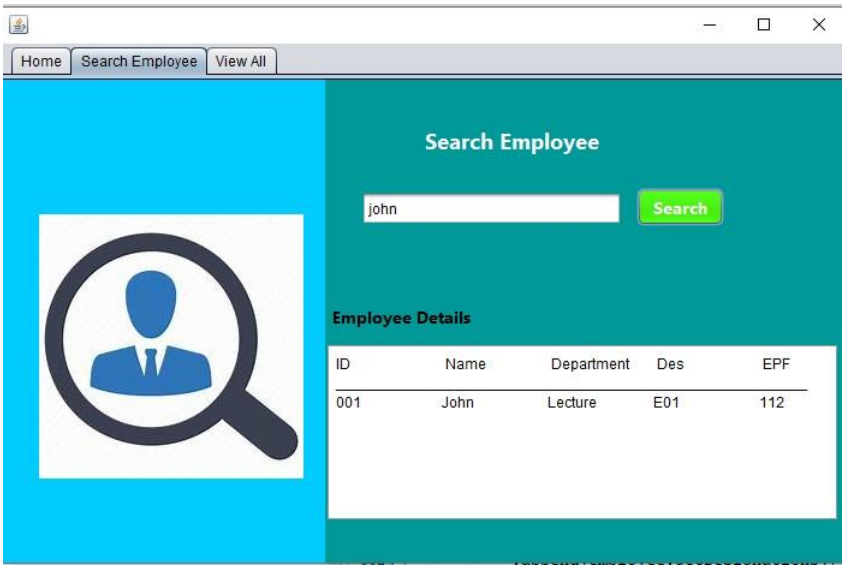
Delete and Update Employee



This “Delete and Update Employee” option allows HR Managers to update or delete employee information for an existing employee records in system After entering the name of the pre registered employee and the employee should click the button “Search” then it shows the information of the employee with a confirmation dialog box. If the employee details are invalid is shows an error message.

Then the HR can click the button “Delete” to delete the employee permanently from the system. If not he can click the button “Update” to update the details to the existing employee. In the button “Home” it take back to the “Search or delete employee details

Search and View Employee



To view an employee record, the HR Manager/Assistant would first need to desired an employee record from the list. Once the employee record has been selected, the HR Manager can view the employee's information, including their employee ID, name, department, designation and EPF Number.

If the employee is valid is shows the information if the employee is not valid it shows an error message.

Ex: We have registered a employee before as john so we can withdraw the details we registered with the employee.

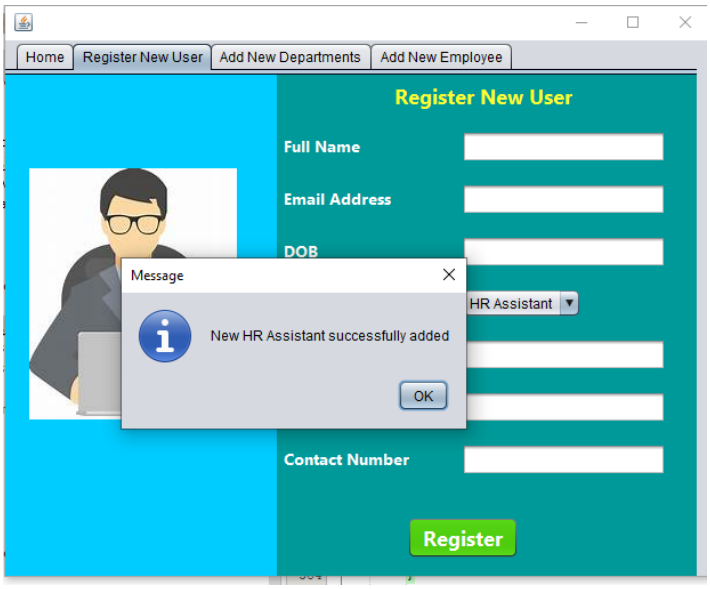
View All Employee

This option is to display all the employees who is in the system.

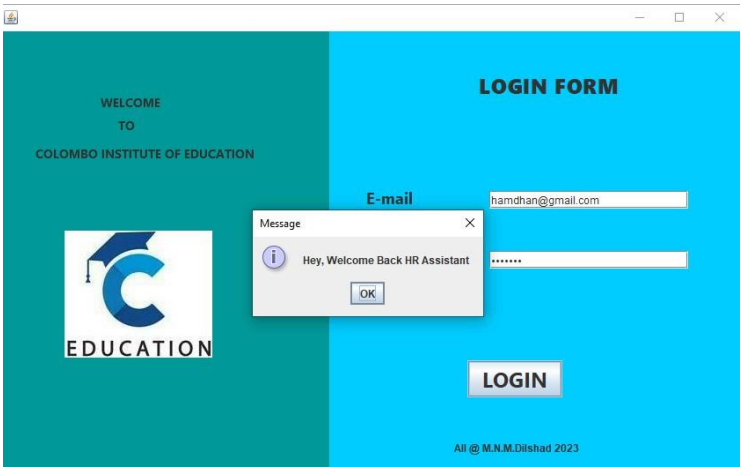
HR Assistant

Login

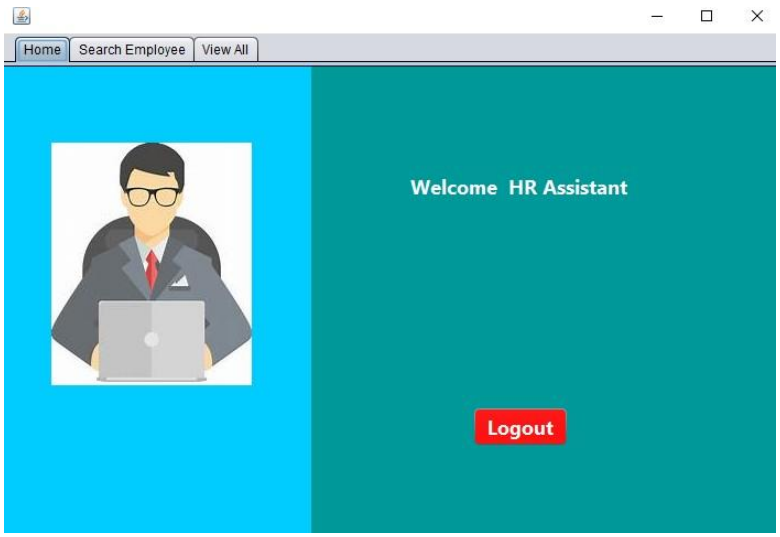
To login as the HR Assistant we need to register him in the user type HR Assistant with, name, email, DOB, Contact Number and password.



Then it creates a new user as HR Assistant. HR Assistant can search employee with name and id, and view all the employees in the system.

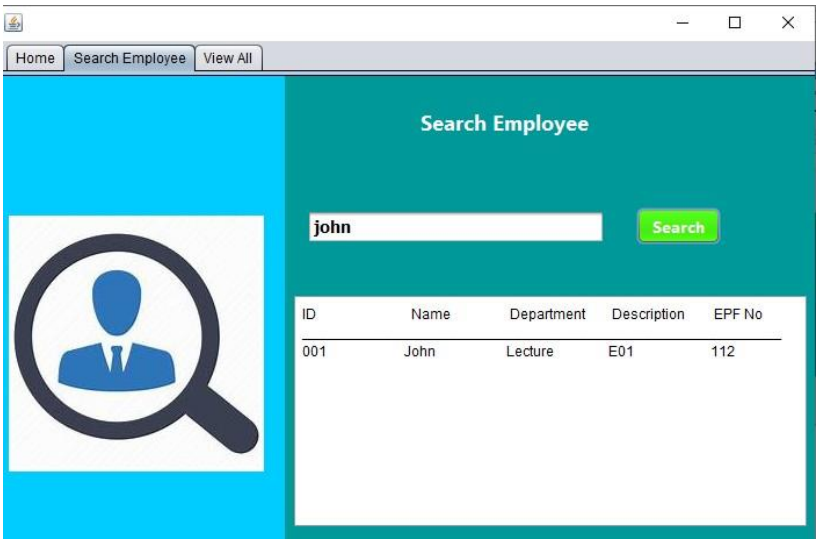


When the username and email is valid it gives a confirmation message when its not valid it shows an error message.



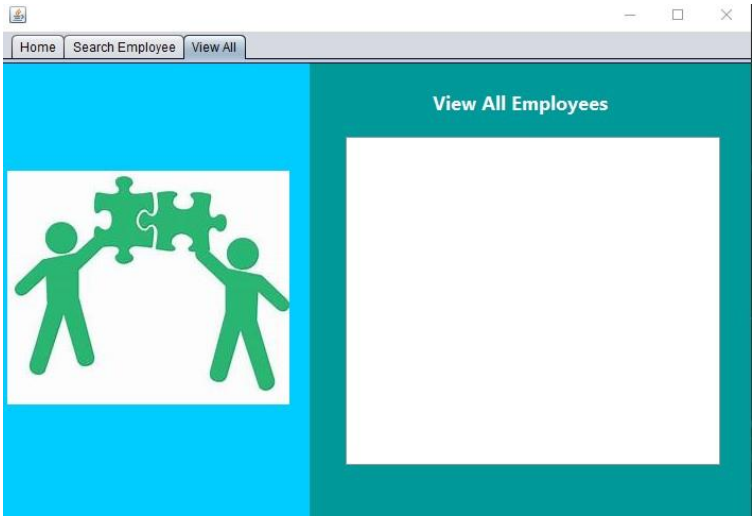
This is the interface for the HR Assistant with the Search Employee and view all employee option.

Search and View All Employees



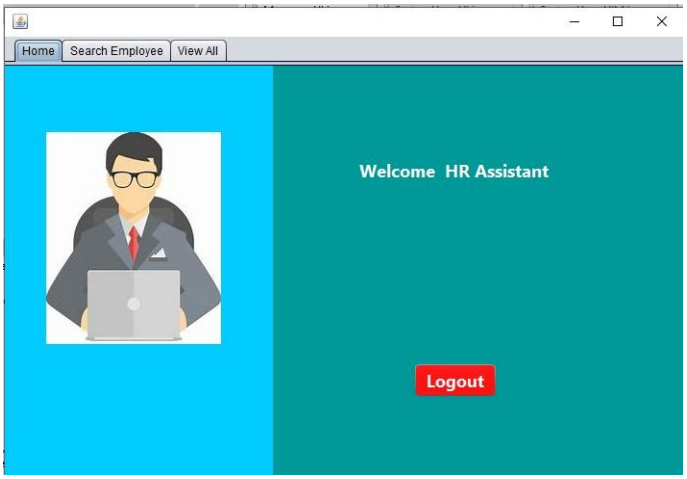
To view an employee record, the HR Assistant would first need to select the desired employee record from the list. Once the employee record has been selected, the HR Assistant can view the employee's information, including their employee ID, name, department, designation and EPF Number.

View All Employee



This shows all the registered Employee to the HR Assistant login.

Logout



by clicking the button “Logout” the HR Assistant can logout from his login then it comes to the home page with the username and password.

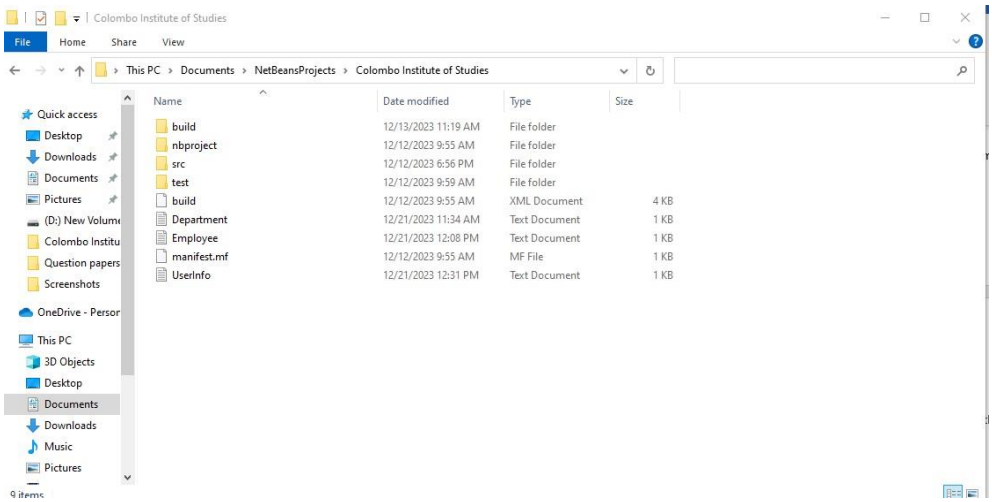
File System/ File Handling

Every data is been saved and can be retrieved from the File System. Which is inside the inside the Colombo Institute of Studies folder.

Employee = Emplyee.txt

Department = Department.txt

Users (HR/HR Assistant) = UserInfo.txt



Conclusion

To sum up, object-oriented programming, or OOP, is a popular and effective programming paradigm that makes use of the ideas of classes, objects, attributes, and methods to build software that is well-organized, modular, and reusable. OOP has many benefits, including polymorphism, abstraction, encapsulation, and modularity, which help developers create complex systems more effectively and sustainably. But it also comes with drawbacks, such as coupling, complexity, and the possibility of spaghetti code if not handled correctly. Despite these difficulties, object-oriented programming (OOP) is still a cornerstone of many programming languages, including Python, Java, and C++. By modeling real-world items and their interactions, OOP helps programmers create scalable and reliable programs. In the end, knowing OOP concepts can significantly improve a programmer's capacity to write organized and maintainable software systems.

References.

- 24, wildaJune et al. (2020) All you need to know about UML diagrams: Types and 5+ examples, Tallyfy. Available at: https://tallyfy.com/uml-diagram/ (Accessed: 26 December 2023).
- Author Amanda Athuraliya Amanda Athuraliya is the communication specialist/content wr et al. (2022) Sequence diagram tutorial – complete guide with examples, Creately. Available at: https://creately.com/guides/sequence-diagram-tutorial/ (Accessed: 26 December 2023).
- Churchville, F. (2021) What is User Interface (UI)? definition from searchapparchitecture, App Architecture. Available at: https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI (Accessed: 26 December 2023).
- Michelle Friesen (2023) *List of advantages of UML*, *Techwalla*. Available at: https://www.techwalla.com/articles/list-of-advantages-of-uml (Accessed: 26 December 2023).
- Sheldon, R. (2023) *What is unified modeling language (UML)?: Definition from TechTarget*, *Software Quality*. Available at: https://www.techtarget.com/searchsoftwarequality/definition/Unified-Modeling-Language (Accessed: 26 December 2023).