PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering
Session: Jan - May 2022

UE19CS353 – Object Oriented Analysis and Design with Java

Theory ISA (Mini Project)

Report on

UNIVERSITY MANAGEMENT SYSTEM

By:

Sneha Adhikary – PES2UG19CS392

Shreesh Devi – PES2UG19CS382

Shishir Menon – PES2UG19CS380

6th Semester F

Table of Contents

List of tables and figures

1.Project Description

a. GITHUB REPO -https://github.com/Sneha0811/University_Management_System

The University Management System is a project written entirely in Java. It is a fully functional Java project which includes information about colleges, universities, and schools in general. It was created for universities and its affiliated institutions to conduct, monitor, and analyze complicated activities such as centralized admission, centralized examinations, attendance management and much more.

2.Analysis and Design Models

Use Case diagram

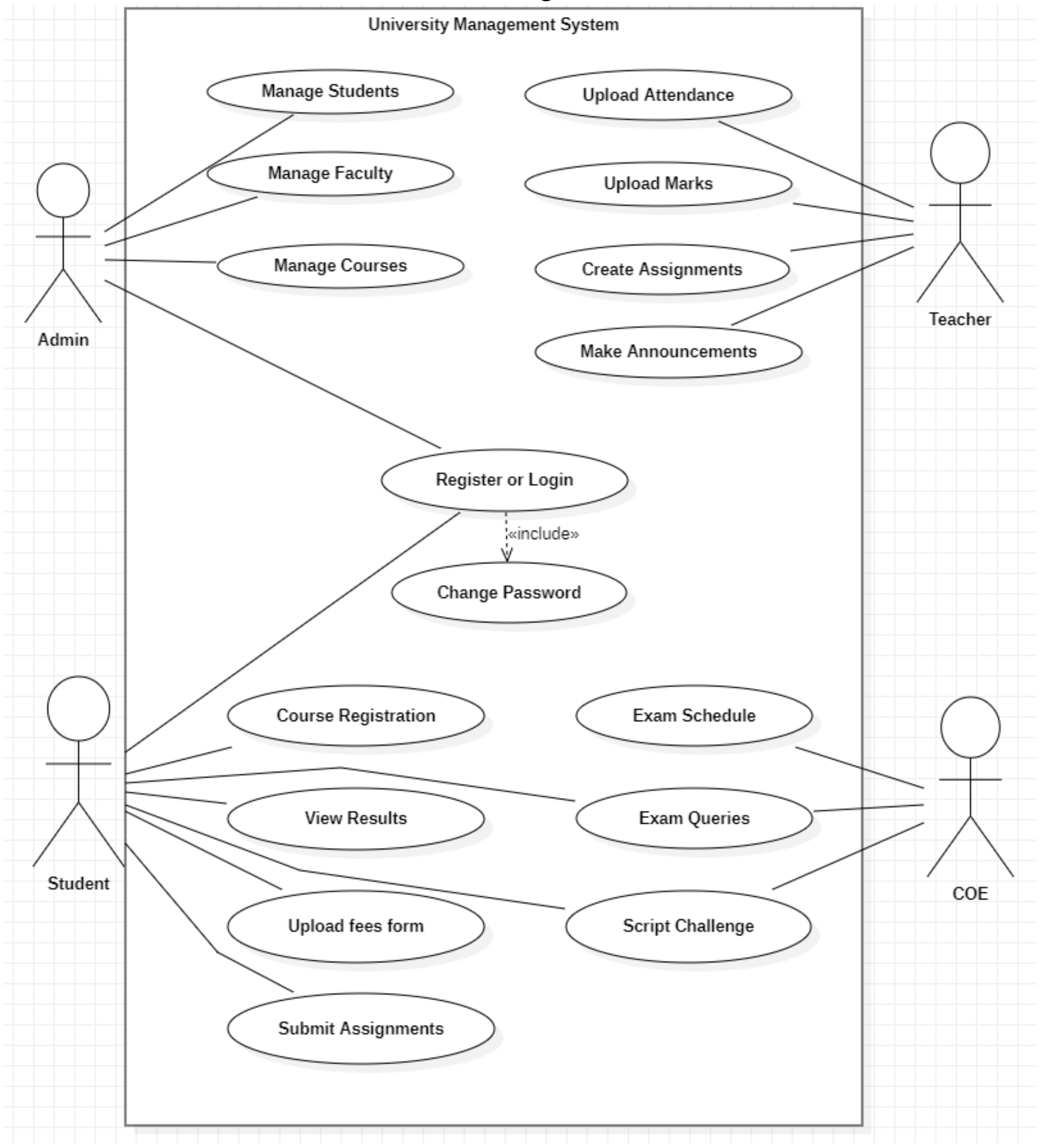


Fig. 2.1

Class diagram

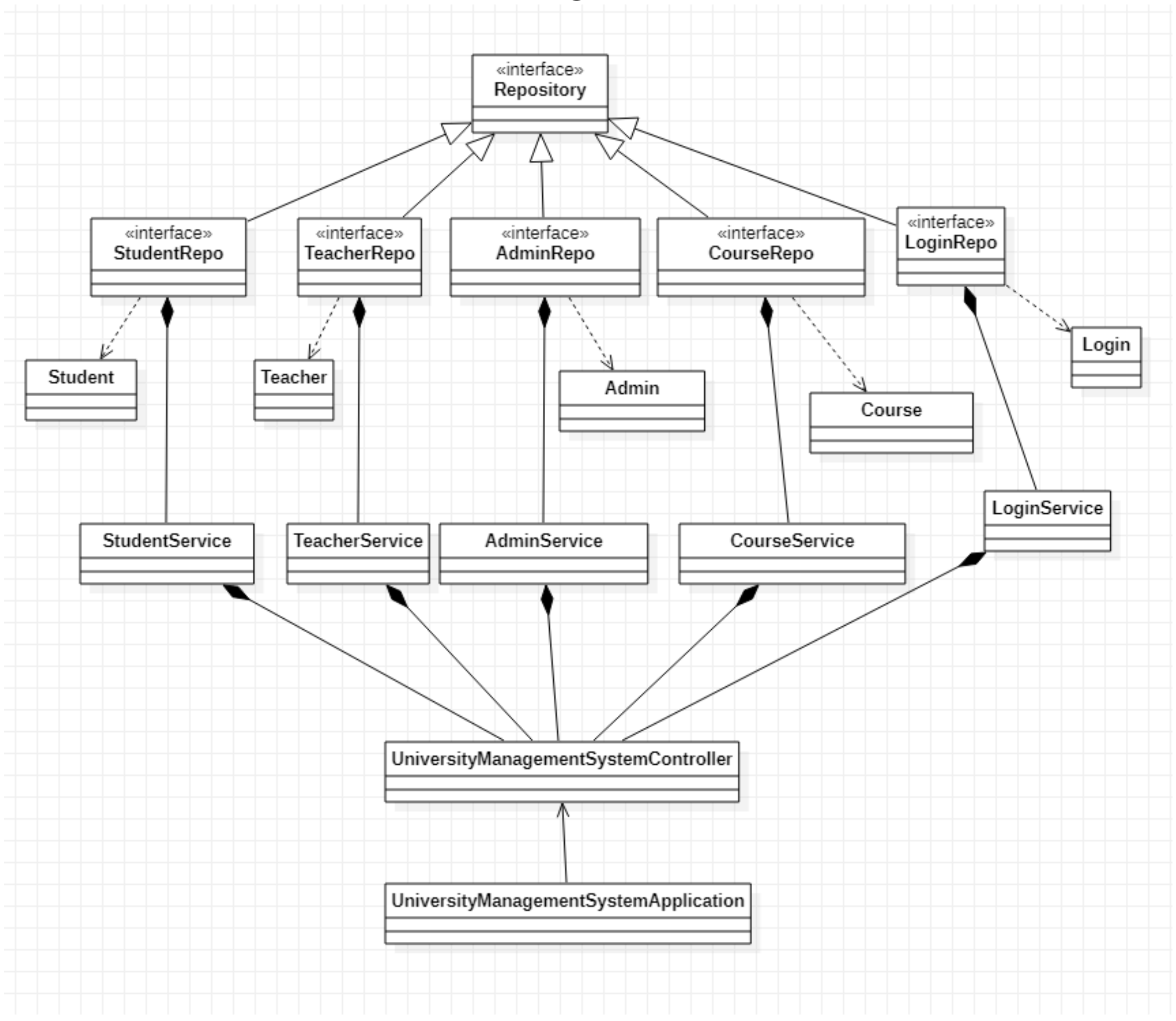


Fig. 2.2

3.Tools and frameworks

- MySQL
- Eclipse IDE
- JDBC
- Swing

4. Design Principles and Design Patterns

a. MVC

   i. Model: Handles data logic.

   ii. View: It displays the information from the model to the user.

   iii. Controller: It controls the data flow into a model object and updates the view whenever data changes.

   In our project-

   Data Access Objects(Model) – The Instructor DAO, Course DAO and Student DAO are responsible for interfacing with the Instructor Table, Course Table and the Student Table respectively.

   Services(View) – This is the intermediary layer between the controller and the database. The main job of services is to integrate data from multiple repositories/DAOs. For example, the Instructor Service uses both Instructor DAO and Course DAO to get its data. Moreover, the Student Service uses both Student DAO and Course DAO to get its data.

   Controllers – The controllers handle requests. Store data in the model and send it to the appropriate view template.

b. Design principles

   i. Single-responsibility Principle-

   A class should have one and only one reason to change, meaning that a class should have only one job.

   In our project - we have made the classes as single responsibility classes so that each of them only have one job.

c. Design pattern

   i. Facade -

   Facade is a part of the Gang of Four design pattern and it is categorized under Structural design patterns. It acts as an interface that hides the complexities.

In our project - the JDBC interface can be called a facade because, we as users or clients create a connection using the "java.sql.Connection" interface, the implementation of which is dealt by the provider of the driver.

Facade design pattern

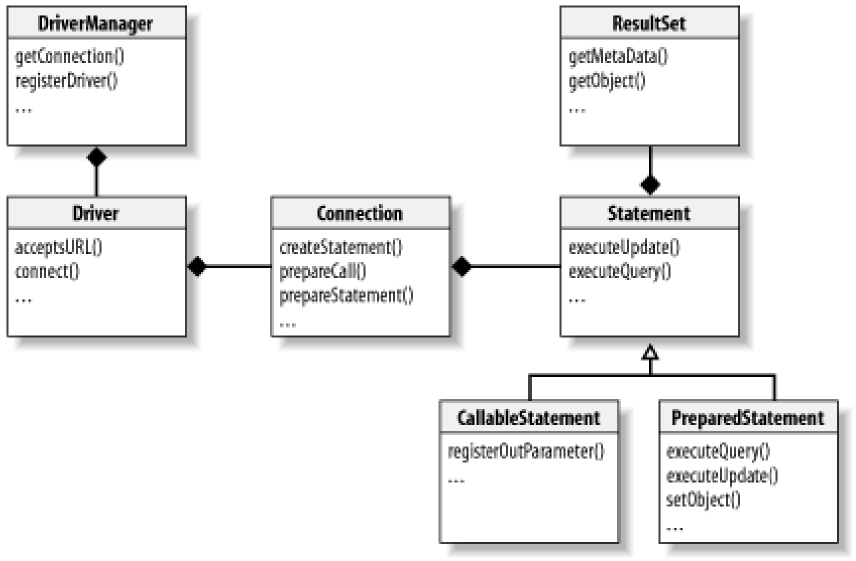


Fig. 4.1

ii. Factory-

Factory design pattern -We defined a factory method inside an interface. The subclass implements the above factory method and decide which object to create. Since this design patterns talk about instantiation of an object and so it comes under the category of creational design pattern.

In our project - The adding Teacher and the adding Student classes is making a use of factory design pattern.

Factory design pattern

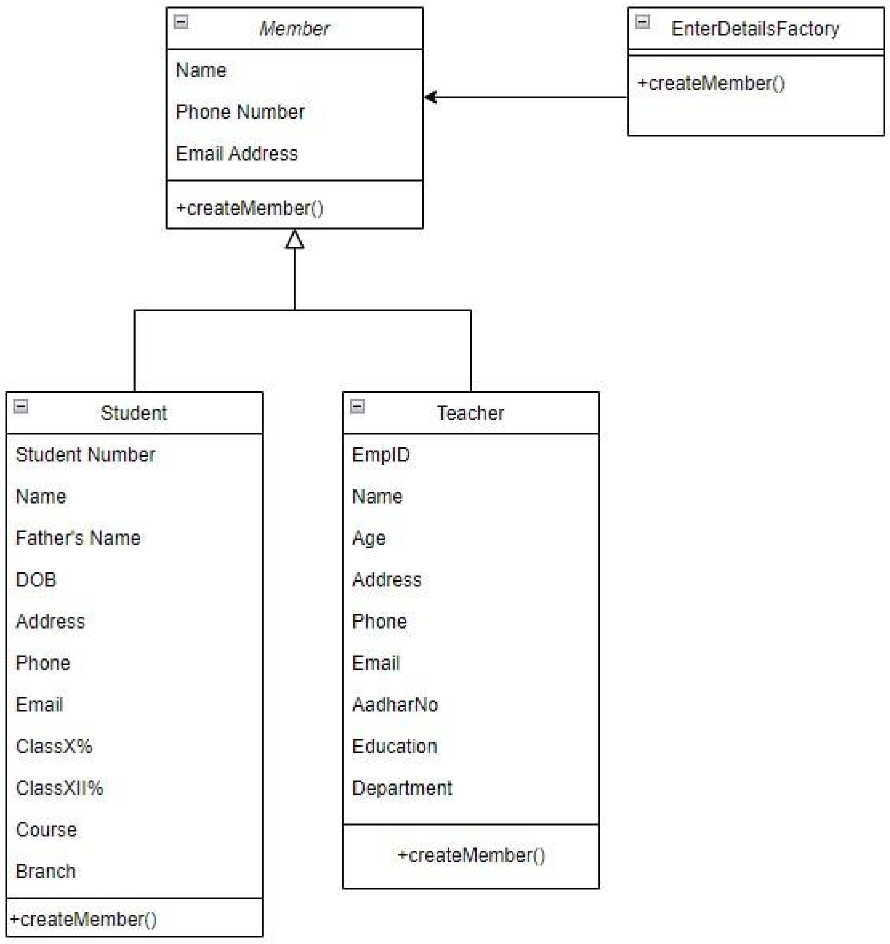


Fig. 4.2

iii. Singleton-

The singleton pattern is a software design pattern that restricts the instantiation of a class to one "single" instance. This is useful when exactly one object is needed to coordinate actions across the system.

In our system - each class creates only a single instance hence singleton design pattern is used.

5.Application screenshots

Login page

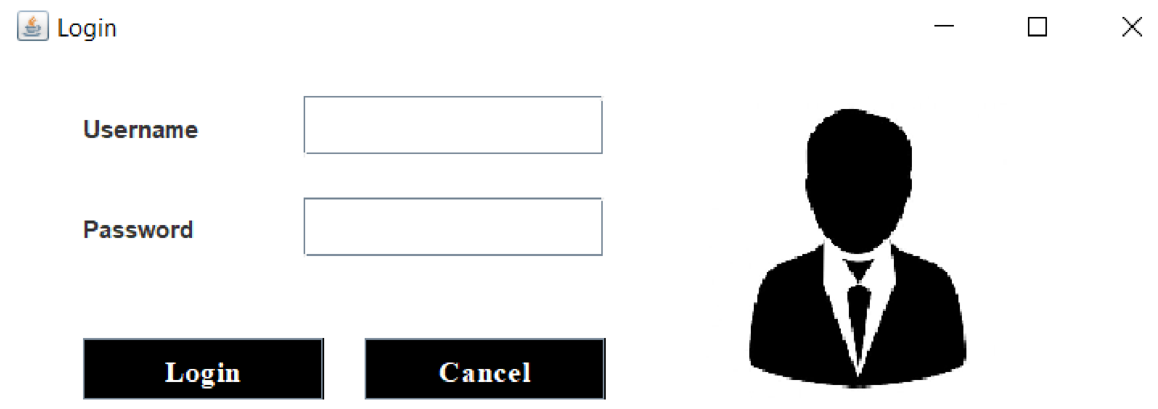


Fig. 5.1

Main page

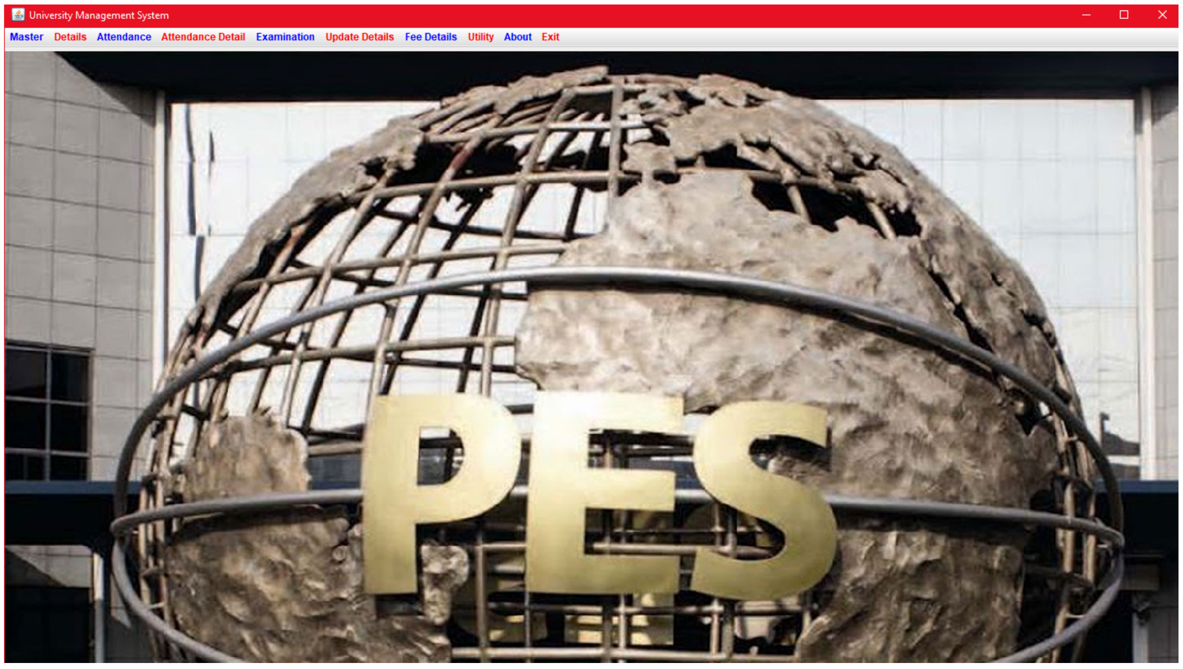
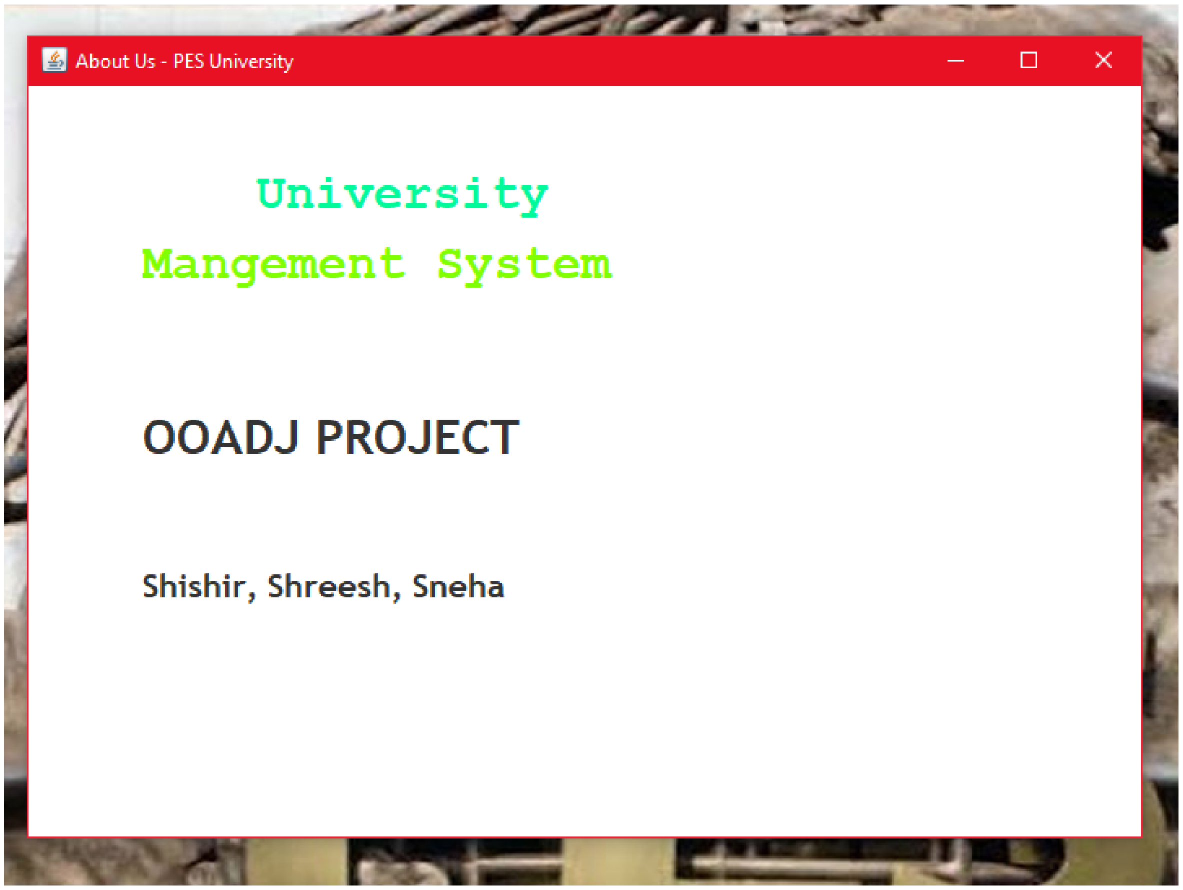


Fig. 5.2

About us page



About Us - PES University

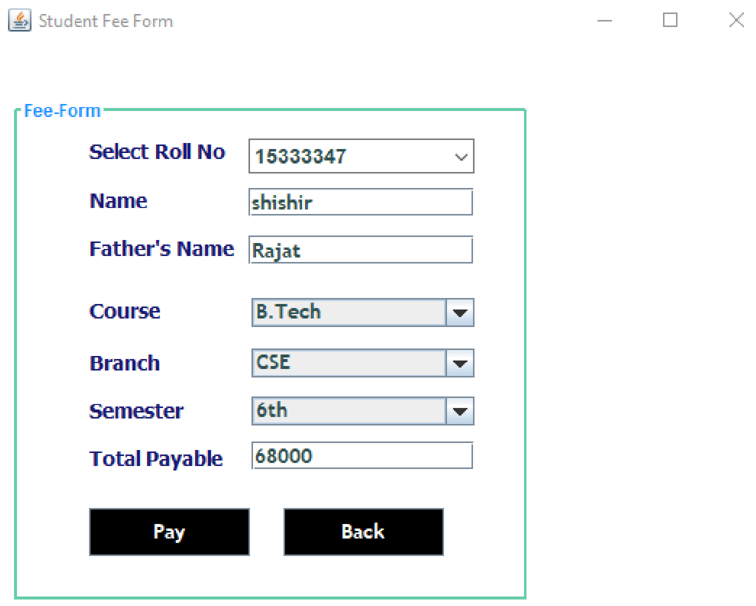
University
Mangement System

OOADJ PROJECT

Shishir, Shreesh, Sneha

Fig. 5.3

Student fee form page



Student Fee Form

Fee-Form

Select Roll No	15333347
Name	shishir
Father's Name	Rajat
Course	B.Tech
Branch	CSE
Semester	6th
Total Payable	68000

Fig. 5.4
Add student page

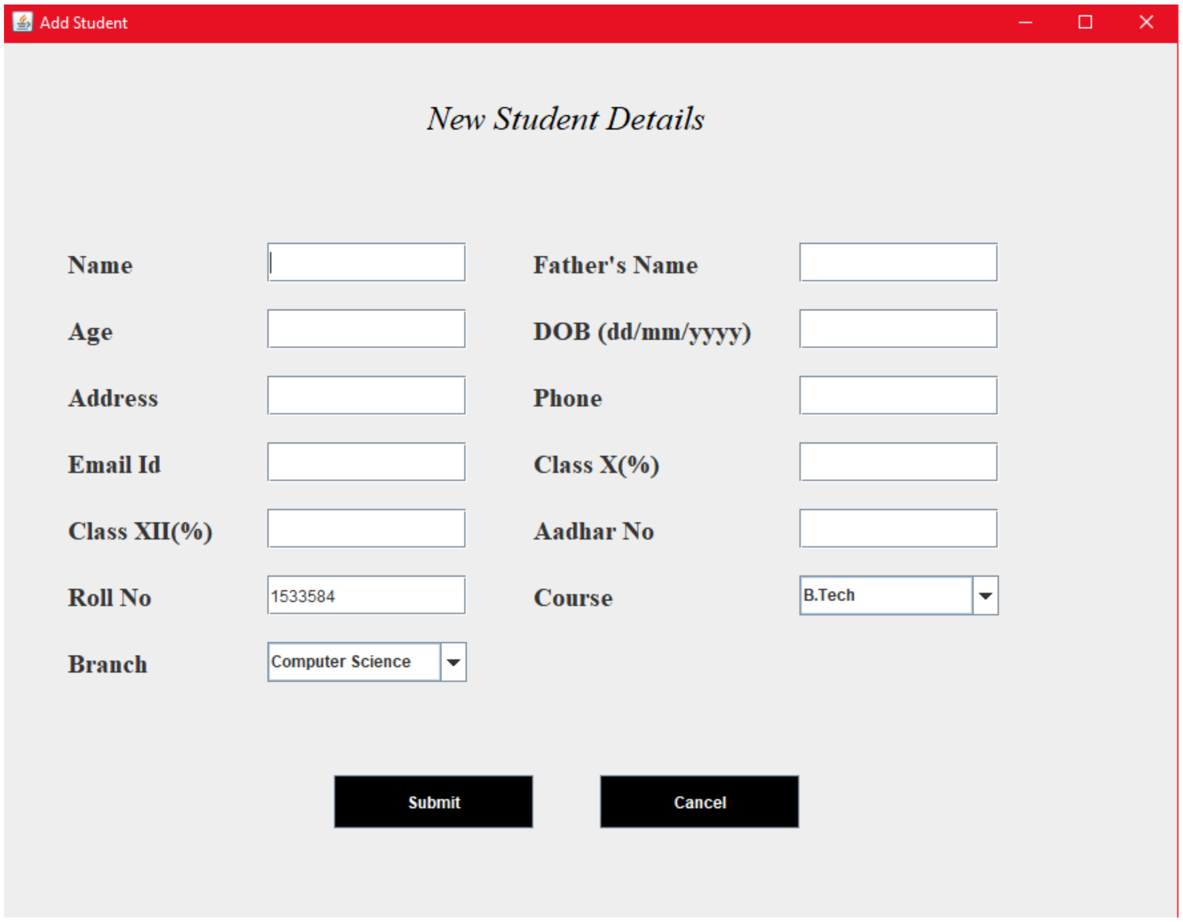


Fig. 5.5

6.Team member contributions

NAME	PART CONTRIBUTED
Shishir Menon	Major Task-Faculty Minor Task-Server / Database
Shreesh Devi	Major Task-Courses Minor Task-Admin
Sneha Adhikary	Major task - Student module Minor task- Register /login

7.Conclusion

A university management system using swing which was done with the help of MVC architectural pattern. We have used factory and facade design patterns. We have used the Single-responsibility Principle as a design principle.

8.References

https://www.javatpoint.com/java-swing
https://docs.oracle.com/javase/tutorial/jdbc/overview/architecture.html