CSE-2112: Object Oriented Programming Lab

University of Dhaka

Department of Computer Science & Engineering

Project Name: School Management System

Project report

Team Members:

1. Tahshina Islam Munni (Roll : 26)
2. S. M. Imran Rahman (Roll : 40)
3. Nirjhar Singha (Roll : 46)

Submitted To:

1. Dr. Muhammad Ibrahim
2. Md. Ashraful Islam

Introduction:

The School Management System software is an application that has been specifically designed for managing the administration of schools. It is a Java-based software that enables the storage of all school-related information, including academic, administrative, and other tasks. The primary objective of developing such a system was to eliminate errors that often arise in manual systems, which make it challenging to store and access records. It is an essential tool for school administrators and teachers to streamline processes, improve communication, maintain data, and enhance the learning experience. The software will be maintained by the administrator. The data of the students and teachers is stored according to their respective IDs. The system will Store data of students, teachers, other staff and information related to the results. Besides information related to employee salary, student fees, routine, notice etc can also be managed by the software.

Access to the system is restricted to the administrator, who uses a unique ID to manage all aspects of the system. The administrator has complete control over modifying or editing the information of all school members stored in the system. The School Management Software can assist educational institution administrators in organizing their structure, curriculum, syllabus, communication, and various reporting features that track all procedures.

The primary purpose of the School Management System is to manage all school activities digitally. The system handles all processes, such as adding teachers, students, and departments, as well as controlling the registration process, thus streamlining the process of organizing school data. The system was developed to provide a secure, easy-to-use, and reliable system that makes the job of the administrator much easier.

Objectives:

The objectives of the school management system software project are multifaceted and encompass the following:

1. Enhancing administrative efficiency by minimizing paperwork and automating record generation.
2. Fostering communication and collaboration between teachers, parents, and students through a centralized platform for sharing information.
3. Providing accurate and timely information about student performance and progress.
4. Providing a user-friendly and intuitive interface for all stakeholders.

5. Effectively managing school finances, including fee collections, budgeting, and expenditure tracking.
6. Managing exam schedules and results to enhance student academic performance.

Requirement analysis:

The purpose and features of the school management system software are given below:

a. Student and Staff Management: The application should manage the personal information of the student and staff, their attendance, and performance reports.

b. Financial Management: The application can manage data related to school finances, including fee collection and employee salary.

c. Curriculum Management: The application will provide tools for managing routine, curriculum, exams and results.

D. Security: The software will be more secure and reliable to manage the data.

The main objective of school management system software is to allow the administrator of the institution to edit and find out the personal details of a student,teachers and staffs and allows them to keep up date Information .It'll also facilitate keeping all the records of students, such as their id, name, class, gender, phone number etc. So all the information of a student to staff will be available in a few seconds. It will also help to generate routine for the all class, result management, financial records and also notice management.

The software will be managed by the admin. The students and other staff cannot access the software directly.

Different sections:

Student section:

In the student section the information of the students are stored (through a database), which can be added, deleted and modified easily. It can keep all the records of students, such as their id, name, class, gender, guardian's information, date of birth, contact number etc. So all the information about a student will be available within a moment.

Employee section:

In the employee section, the information of the employees are stored, which can be added, deleted and modified easily. It can keep all the records of employees, such as their id, name, profession, gender, date of join, contact number etc. So all the information about an employee will be available within a moment.

The employee section is divided into two subsections:

1. Teacher
2. Other staff

Routine section:

The school management software includes a functionality for storing class routines. This feature provides a user-friendly interface for users to create, modify, and update class schedules with ease.

The software allows for convenient access to routine data, which can be easily retrieved and reviewed as necessary. The ability to efficiently modify routine schedules ensures that the software remains relevant and up-to-date, reflecting changes in class schedules as required. This feature ultimately streamlines routine management, simplifying the process for administrators and educators alike.

Result section:

This section is designed for the purpose of calculating and generating student results. Users are required to input the marks obtained by students in various subjects, after which the software will automatically compute their total marks, grades, GPA, and other relevant metrics. By streamlining the result calculation process, this software significantly simplifies and expedites the process of generating student results.

Log in section:
This software features a secure login system that allows authorized users, including the administrator and other designated users, to access the system using their unique user-ID and password. Upon successful login and authentication, users are granted access to the system and its functionalities.

It is important to note that the creation of new accounts for other users with employee ID, user type, and password is solely the prerogative of the system administrator. This serves as an added security measure, ensuring that only authorized users are granted access to the software and its contents.

Salary section:
This section maintains a record of the salaries of the employees. The salary amount of different employees and the date of the last salary received is stored here.

Fee section:
This section maintains the record of fee submissions by the students. The fee amount of the students and the date of last fee submission is stored here.

Notice section:
In this section, different notices, their descriptions, dates, etc. will be stored and kept updated. While uploading the notice, the software automatically detects the date. Each notice will be identified uniquely by the notice id.

Co-curriculum section:
This section of the software is dedicated to managing information about the various clubs and their activities within the school. It enables the creation of new clubs and the maintenance of funds associated with each club. Additionally, it provides a platform for monitoring the activities of each club through a designated club moderator. With this feature, administrators can efficiently oversee club operations, ensure compliance with school policies, and promote student engagement in extracurricular activities.

Others:
Besides all these, information related to courses, etc will also be stored in the software.

System design:

System design is the process of creating a new system. A detailed implementation process is described here. System design has two parts:

1. Physical design
2. Logical design

During the logical design phase of the system development life cycle, the analyst identifies and describes the inputs from various sources, outputs to various destinations, databases or data sources, and procedures or data flows that meet the user requirements. The goal of this phase is to develop a logical model of the system that describes the functionality and flow of data through the system.

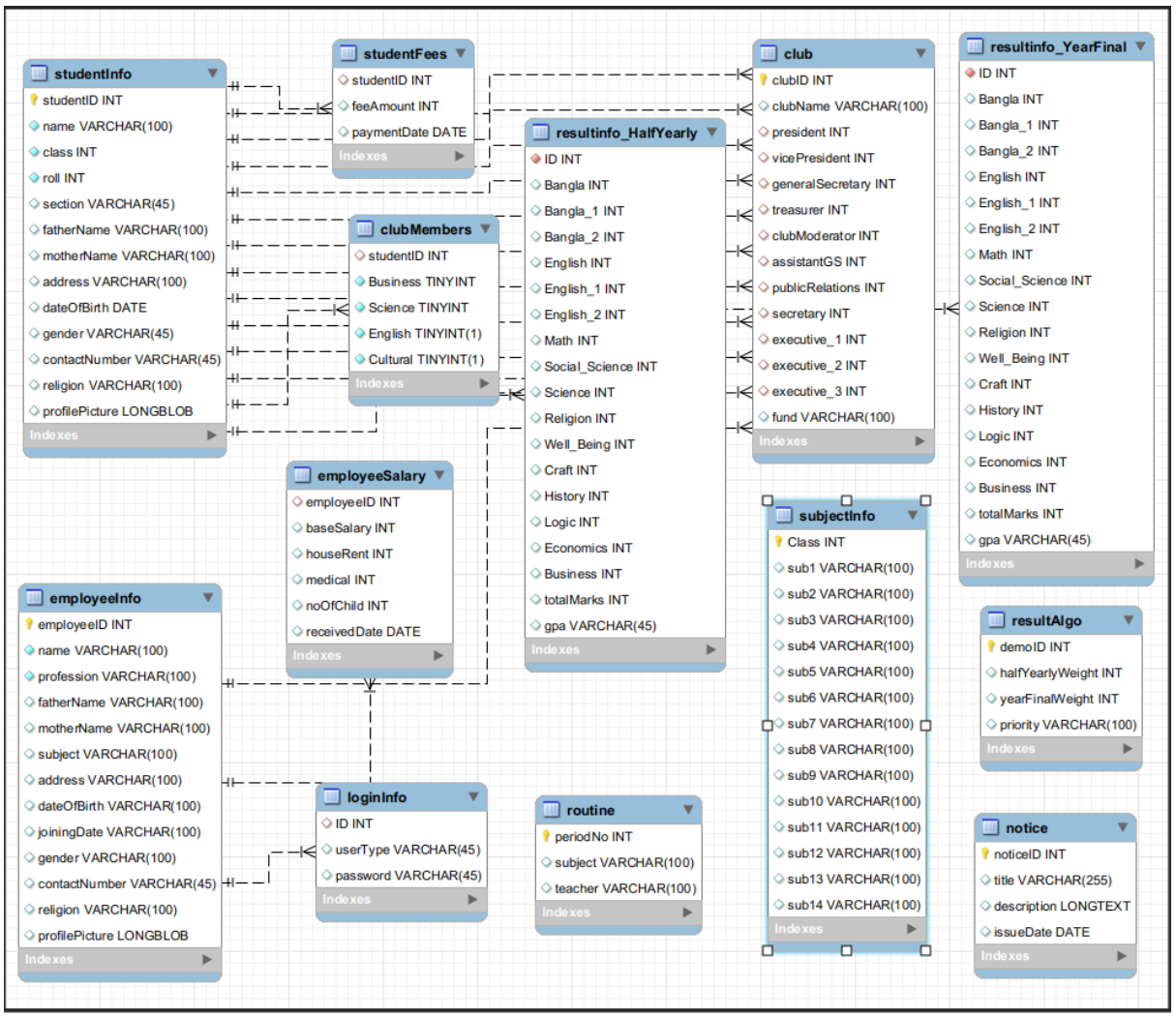
In the physical design phase, the logical design is converted into a working system. This phase defines the design specifications that detail how the system must operate,

including the hardware and software requirements, user interface design, data storage and retrieval mechanisms, and security considerations. The physical design phase involves translating the logical design into a concrete implementation that meets the requirements of the users and stakeholders. This phase is critical in ensuring that the system is robust, reliable, and scalable.

Database Design:

Databases serve as repositories of data that are utilized in software systems. These data are organized and stored in tables within the database. Various tables are created to store different types of data for the system. Relationships between the tables are established by employing foreign keys. Consequently, the data can be stored in an organized manner, eliminating redundancies and ensuring efficiency in data management.

The ER Diagram of the database:



Authentication:

In order to ensure the security of data, the login process for the software requires authentication of user identity through a unique user ID and password. The system will verify the validity of the user ID and password entered by the user, and will only grant access to authorized users. In the event that an incorrect ID or password is entered, the user will be denied access to the system. This robust security measure ensures that the software and its data remain protected from unauthorized access or tampering.

Data validation:

During the process of data entry in the software system, it is imperative to ensure that the data entered is accurate and valid. In this regard, the software has inbuilt validation rules that are executed each time a new record is entered into the database. The validation rules help to ensure that only correct and relevant data is entered into the system. In the event that the data entered is not in line with the validation rules, the system will not accept the data and will prompt the user to make the necessary corrections. This approach ensures that the integrity of the database is maintained, and that data corruption is avoided.

Tools:

The project will be made using the following tools:

Front end:

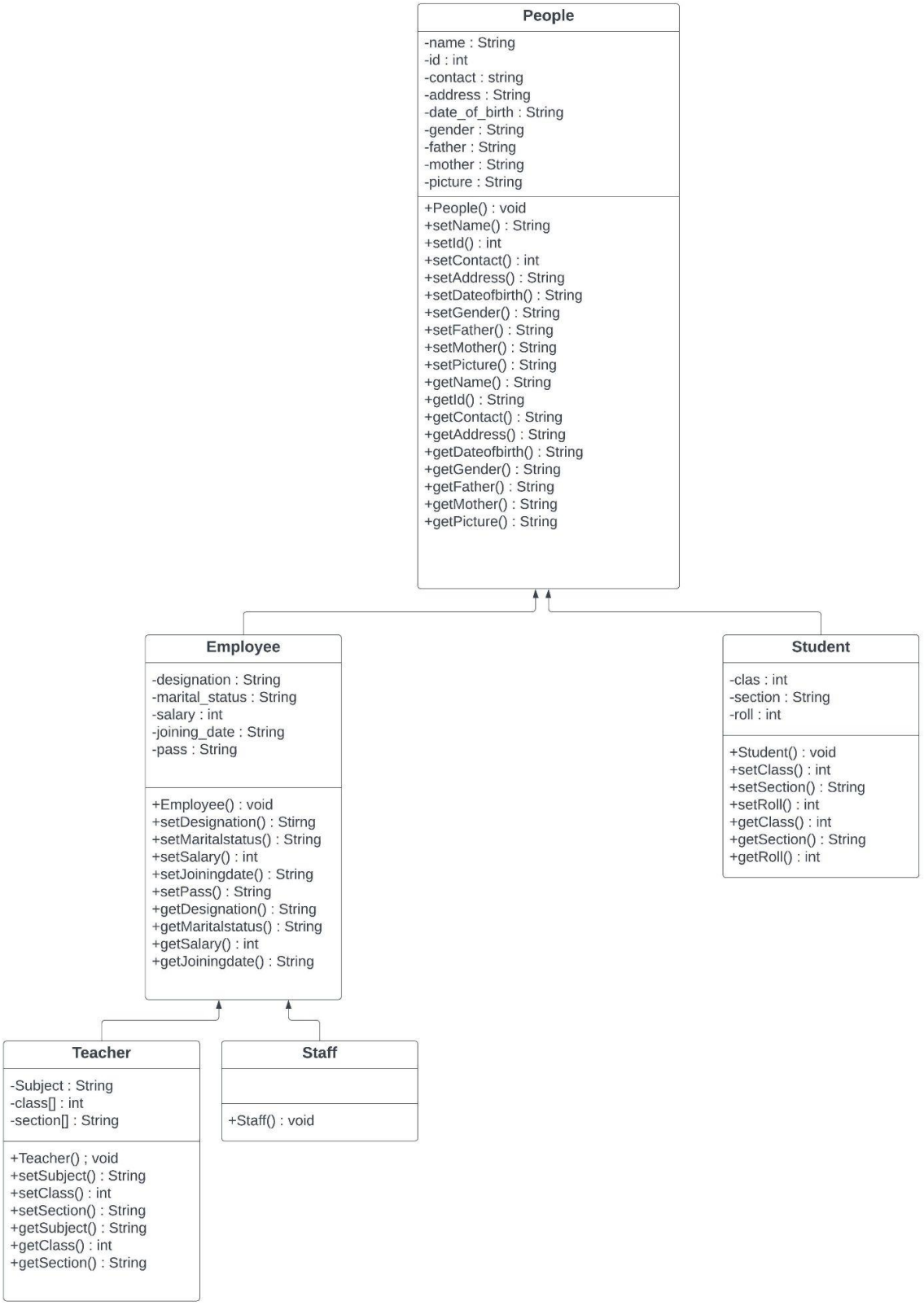
- Java,
- JavaFx,
- xml,
- css.

Back end:

- Java,
- MySQL database.

Class Diagrams:

Class hierarchy for people management:



1) People:
It's the super class of this hierarchy. As this hierarchy is basically used for storing the information, the common information such as id, name, gender, picture, mobile number and its fields are here in this class. There are also methods to access the field of the class.

1.1) Student:
This class is the subclass of the People class. As it inherits the People class, it has the field to store the information of a student's class. It has methods to access its own fields as well as the parent class fields.

1.2) Employee:
This class is a subclass of People class and super class of two other classes named Teacher and Staff.

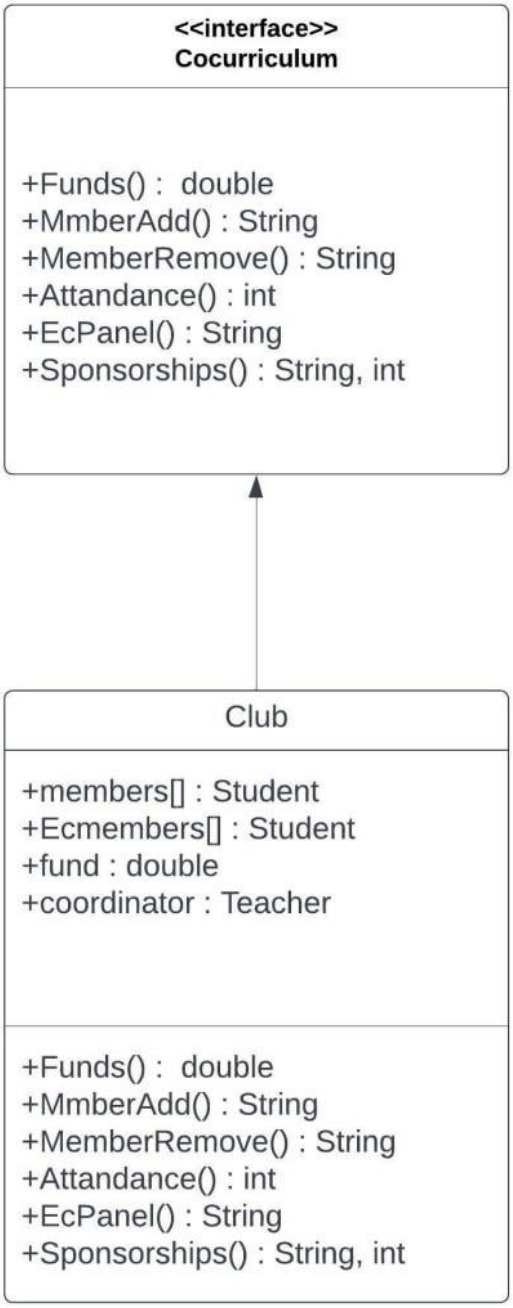
1.2.1) Teacher:
This class is the subclass of the Employee class. As it inherits the Employee class, it has the field to store the information of a teacher class. It has methods to access its fields.

1.2.2) Staff:
This class is the subclass of the People class. As it inherits the Employee class, it has the field to store the information of the employee and people class.

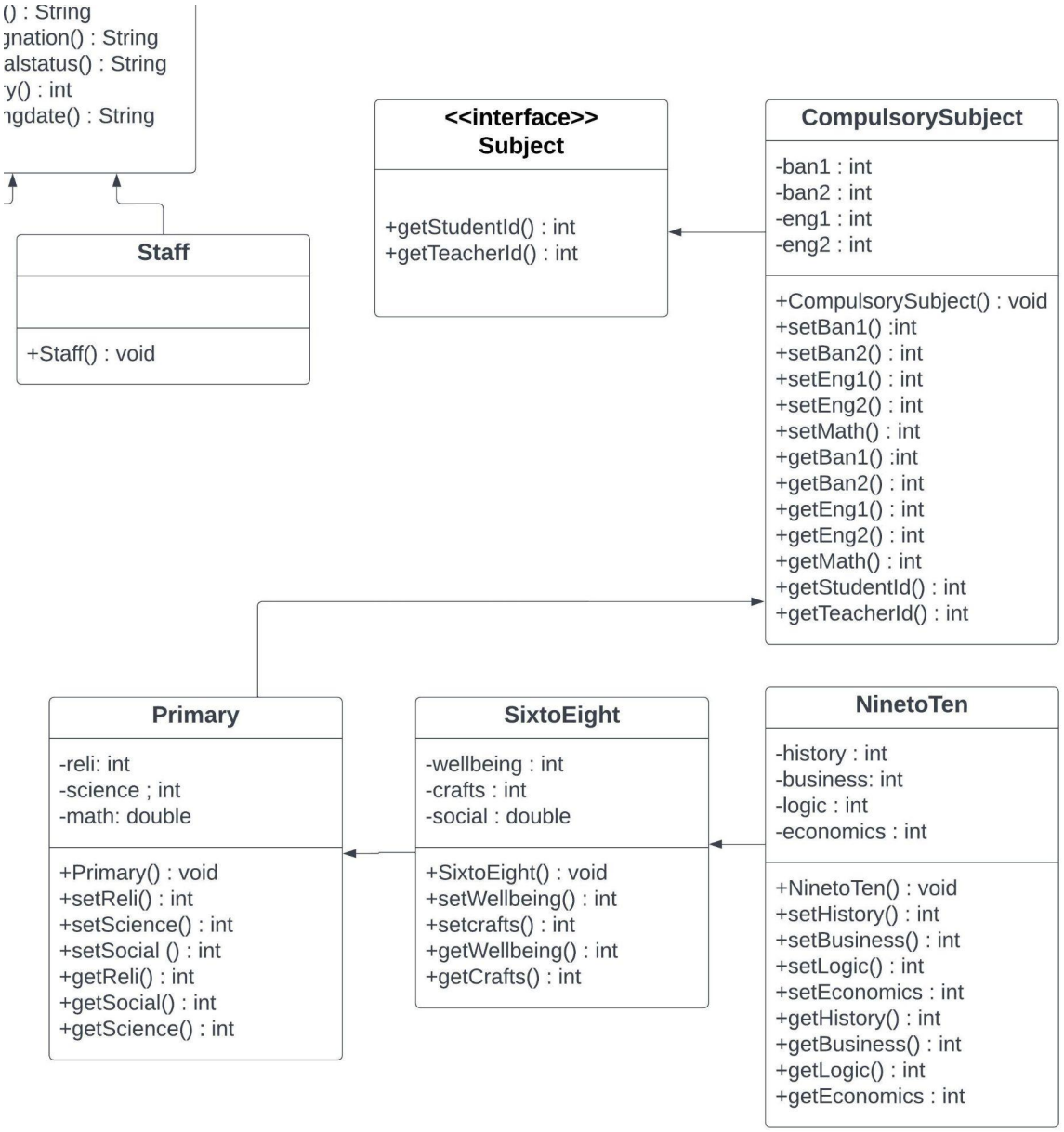
Class hierarchy for co-curriculum management:

1) Cocurriculum:
It is an interface that contains methods related to managing co curricular activities. It contains methods to manage funds and members.

1.1) Club:
This class implements Cocurriculum interface. It contains its own attributes and the overridden functions of the Cocurriculum interface.


<<interface>>
Cocurriculum
+Funds() : double
+MmberAdd() : String
+MemberRemove() : String
+Attandance() : int
+EcPanel() : String
+Sponsorships() : String, int
Club
+members[] : Student
+Ecmembers[] : Student
+fund : double
+coordinator : Teacher
+Funds() : double
+MmberAdd() : String
+MemberRemove() : String
+Attandance() : int
+EcPanel() : String
+Sponsorships() : String, int

Class hierarchy for result management:



1) Compulsory subject:

This class inherits the class Student. It has attributes and methods related to the subjects that are common and compulsory for all classes from one to ten.

2) Primary:

This class contains the data of the primary classes which are class 1 to 5.

3) SixtoEight:

This class contains the data of the High School classes which are class 6 to 8. The extra subjects of this curriculum are added in this class.

4.) NinetoTen:
This class contains the data of the Secondary School classes which are class 9 to 10.

Use of controllers:

StudentControl:

The StudentControl class is an essential component of the School Management System application, responsible for managing and organizing all information related to students. The class maintains a database of student information, including their ID, name, class, mobile number, gender, and picture.

To display this information to the user, the StudentControl class utilizes a table view, which is also controlled by the class. The table view allows the user to view all the stored data in a single, organized location, where it can be sorted by various categories. Students can be sorted alphabetically by name or ID, or in ascending or descending order by their ID number.

StudentControl includes three controller classes: Student Profile controller, student registration controller and student result controller. These classes provide a range of functionalities for managing student information. For example, it includes methods for adding new students, updating existing student information, and deleting students from the database. The class also allows users to search for specific students using various criteria, such as their name or ID number.

TeacherControl:

TeacherControl is like StudentControl, including two controller classes: teacher Profile controller and Teacher registration controller. It maintains the information related to teachers like their id ,name ,subject,gender,mobile, picture,gender etc.

StaffControl:

The StaffControl module provides a powerful and efficient tool for managing employee data within the School Management System. Its intuitive design and comprehensive functionality make it an indispensable component of the system, enabling users to efficiently manage employee records and streamline the employee management process.

LoginValidator

The School Management System has implemented a robust authentication mechanism to ensure the security and privacy of user data. The login functionality is designed to authenticate the identity of the user, whether it is the admin or any other user, by checking the entered credentials against the stored user data in the system.

When a user attempts to log in to the system, they must provide their unique user ID and password. The system then matches the entered credentials with the stored user data to authenticate the identity of the user. If the entered credentials do not match the stored data, the system will deny access and prompt the user to re-enter their credentials. This step ensures that only authorized personnel can access sensitive information within the system.

ResultControl:
The ClassResultController and ResultAlgorithmController are two important components of the School Management System application. Together, they provide the functionality to generate student results quickly and easily. The ClassResultController is responsible for handling the user interface of the result generation process. It provides a simple and intuitive interface where the user can input the marks of each student in different subjects.

Once the marks are entered, the ClassResultController calls upon the ResultAlgorithmController to calculate the total marks, grade, and GPA of each student. The ResultAlgorithmController uses a predefined algorithm to calculate the results. The algorithm takes into account the marks of each student in different subjects, as well as the weightage assigned to each subject.

NoticeControl:
The NoticeController class handles the notice section. Here we have added the features of viewing all the notices along with their issue date, title and description. We have also added the features of creating a new notice, updating an existing notice as well as deleting a notice.

RoutineControl:
The RoutineController class handles the notice section. Here we have added the features of viewing the routines of different classes including the subjects and corresponding teachers. We have also added the feature of editing the routine.

ClubControl:
The ClubController is a Java class that provides functionalities for managing the clubs in the School Management System application. It includes three more controllers, club form controller, club member entry controller and club member controller. The controller class allows users to create, update, and delete clubs, as well as view the details of existing clubs.

UML Class Diagram:

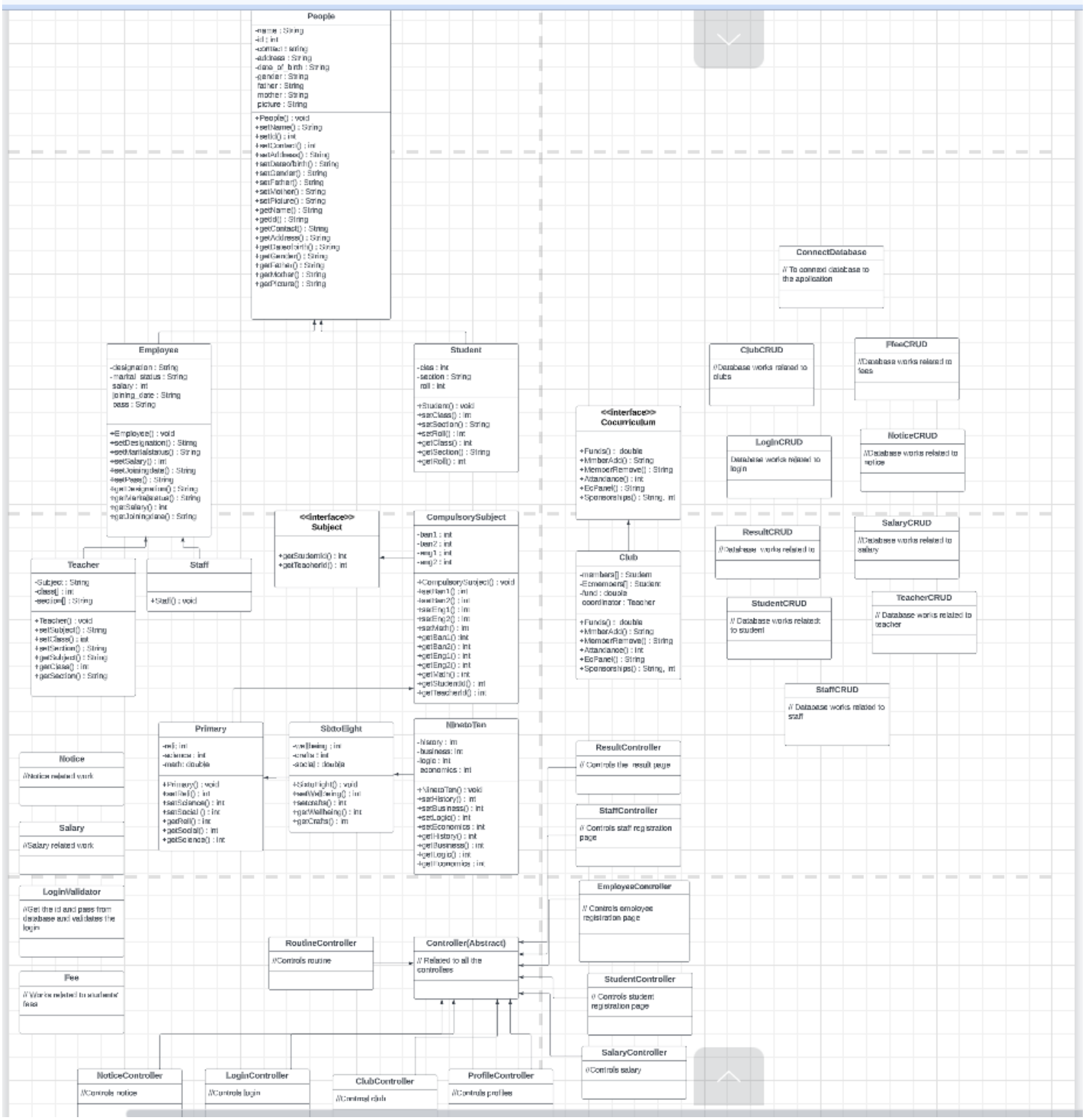


Fig:UML diagram

Use Case Diagram:

A use case diagram is the primary form of system requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

It summarizes some of the relationships between use cases, actors, and systems.

The use case diagram of the system is given below:

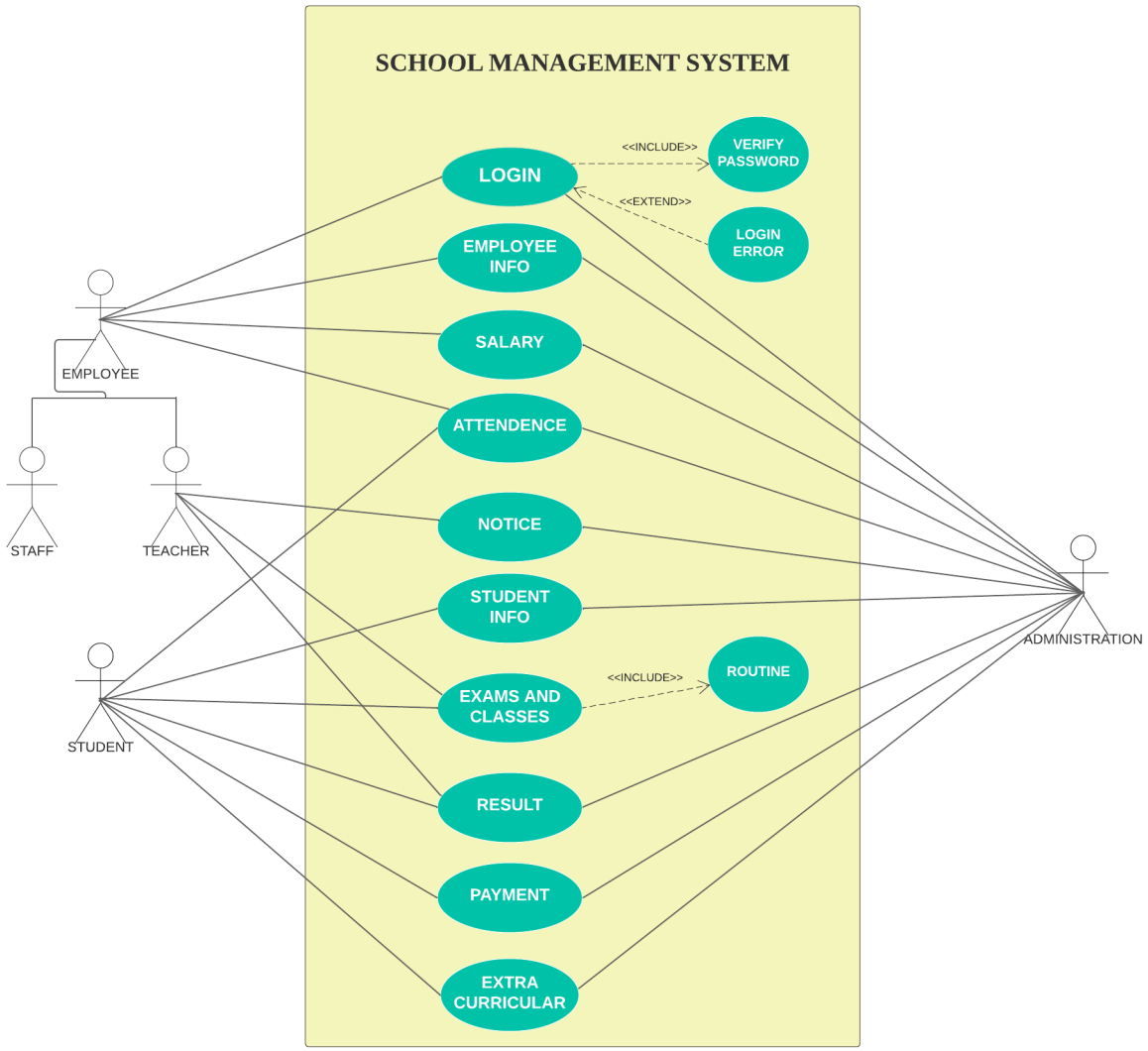


Fig: Use case diagram

Use of Java core concepts:

Inheritance:
Using the concept of inheritance the class hierarchy is made. Here all users are subclasses of person class. Again teachers and staff all are subclasses of the employee class. Thus the inheritance has provided the scope of code reusability. The Controller class is extended by all other controller classes. Another class hierarchy is used in the subject section where compulsory subject is extended by primary subject, primary subject is extended by six to eight and further extended by nine to ten.

Encapsulation:
In the design, student, teacher, staff, notice, routine, result everything is considered as an individual entity. All the attributes of an entity have been wrapped up in a single unit.

Polymorphism:
Polymorphism is a concept of object-oriented programming that allows us to perform a single action in different forms. It is an OOP design that empowers classes with various functionalities to execute or share a common interface. The helpfulness of polymorphism is that code written in various classes has no impact on which class it has a place in since they are utilized similarly.

Abstract class:
Use of abstract classes in the class hierarchy improves the code quality. Use of interface has allowed multiple inheritance. Here the Controller class is an abstract class. That means this class is just a blueprint and cannot be used to make objects.

Interface:
In the source code, the Co-curricular is a java interface which is implemented by the Club class. By the interface, the club section is associated with the co-curricular section.

Function overloading and overriding:
In the same class, functions with the same name and different parameters can be used by function overloading. In subclasses, functions overriding concepts can be used. Thus we can use the same function for multiple purposes.

Data hiding:
The concept of data hiding is applied by using private or protected data. In order to ensure security, passwords can be stored after encryption in the database.

Exception handling:
In case of fetching data from the database, authentication or validating data while inserting or updating in the database, exceptions can occur. These exceptions can be handled using try catch block or exception throwing.

Abstraction:
Abstraction is the process of hiding unnecessary details and showing only useful information to the user. Abstract class or interface can help to achieve this. In the class hierarchy and database design, the concept of data abstraction is used.

Collections Framework:
The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects. Among collections framework, we used arraylist and sort function in our project.

Lambda expression:
A lambda expression is a concise way to represent an anonymous function, which can be treated as a method argument, assigned to a variable, or used as a return value. It allows you to write more compact code by expressing functionality directly without the need for explicit class declarations. We have used lambda expressions in our project.

Snaps:

Fig: Home page

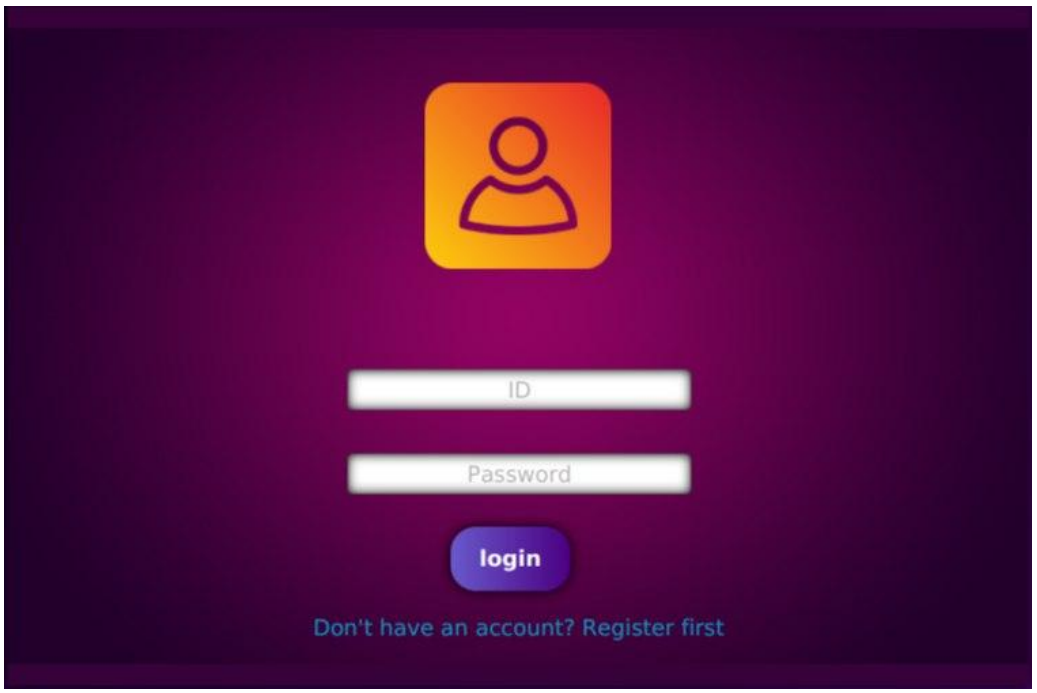


Fig: Login

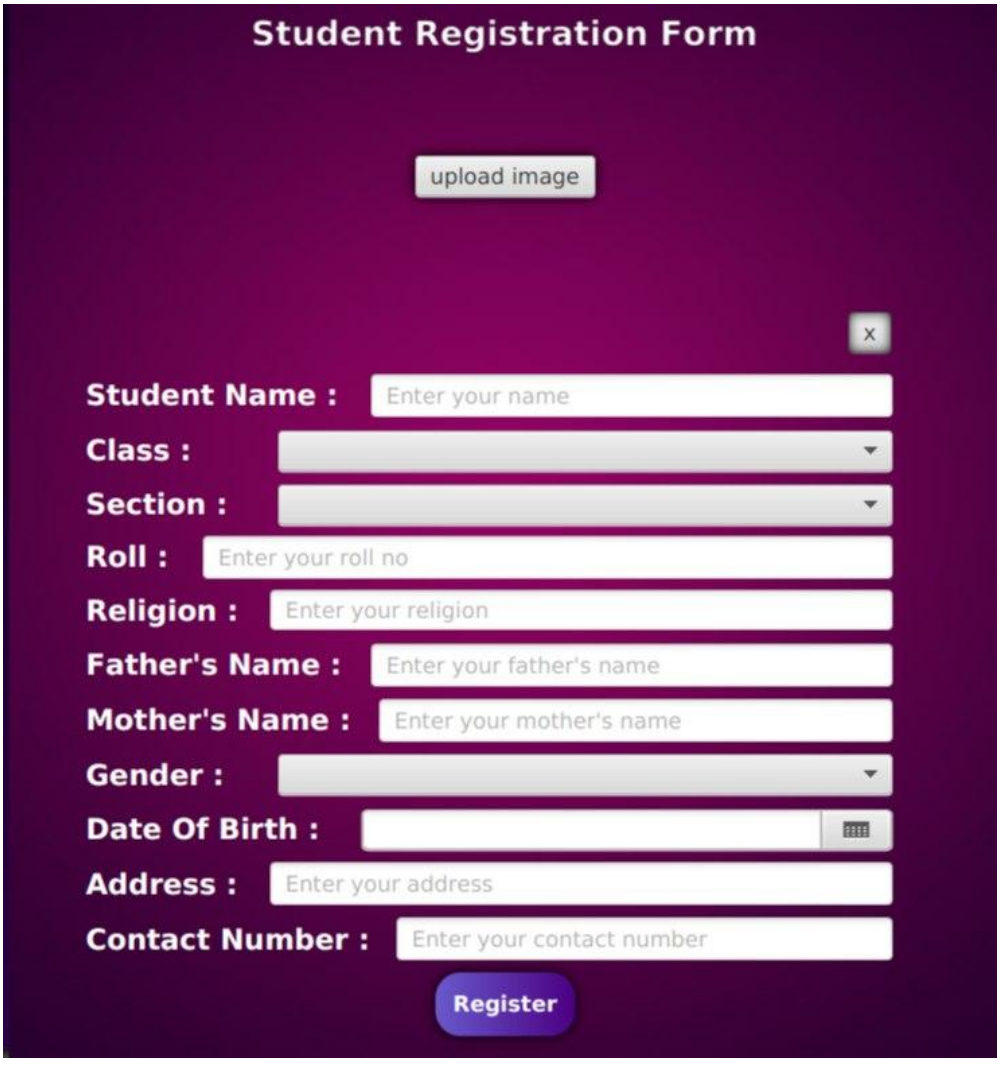


Fig: Registration Form

Fig: All User List

Fig: Student Profile

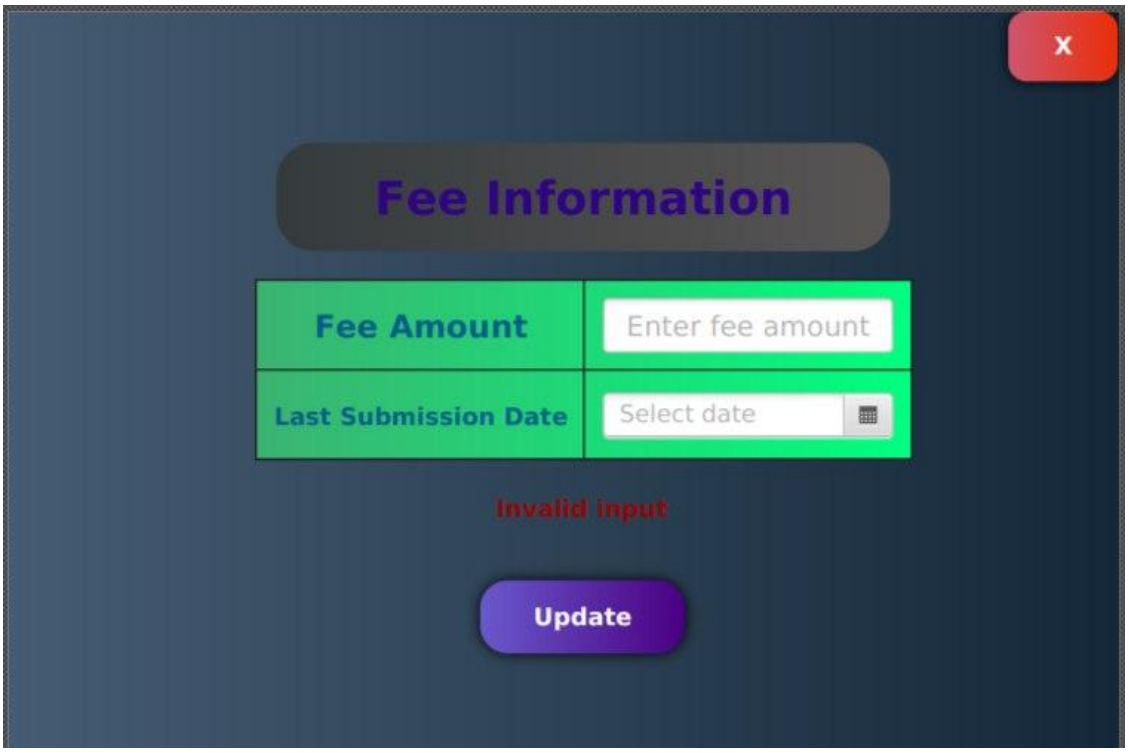


Fig: Student Fee info

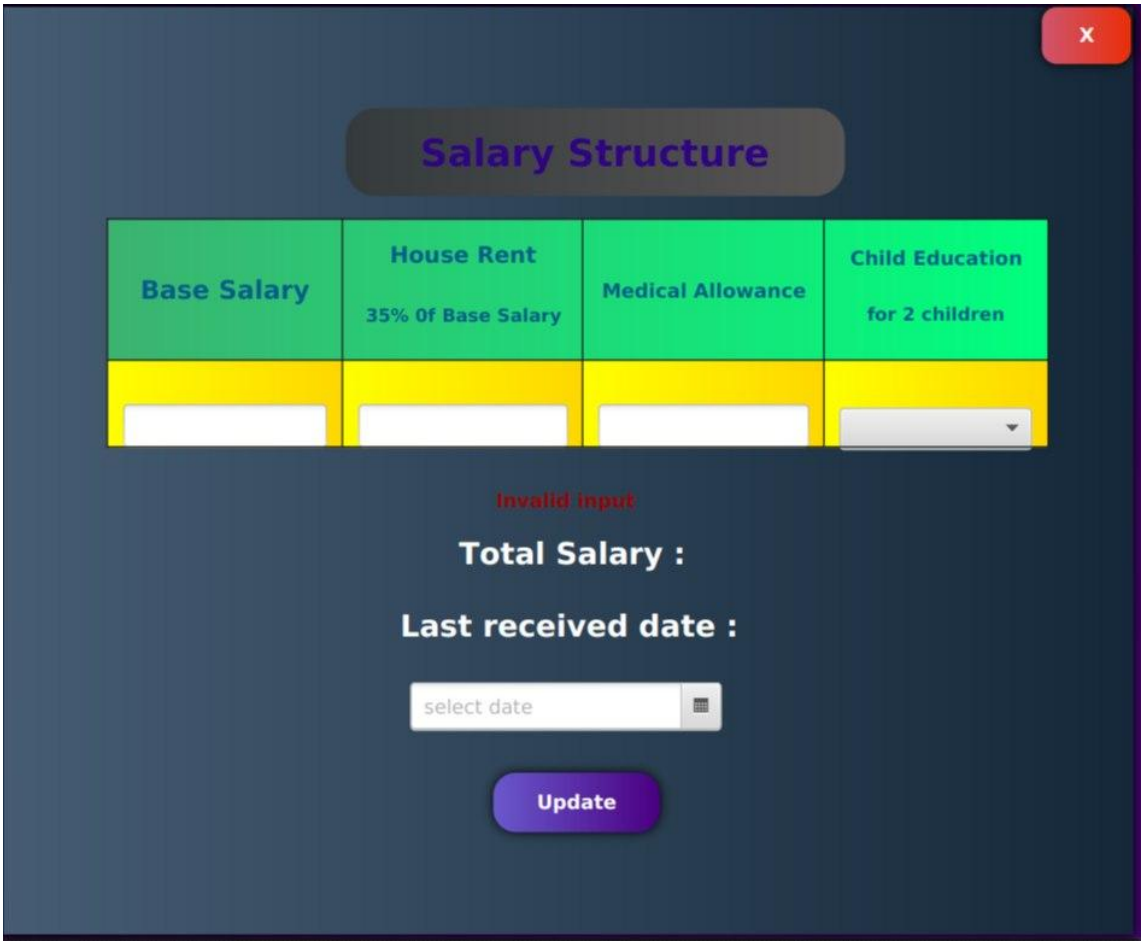


Fig: Employee Salary Structure

Fig: Personal Result

Fig:Combined Result

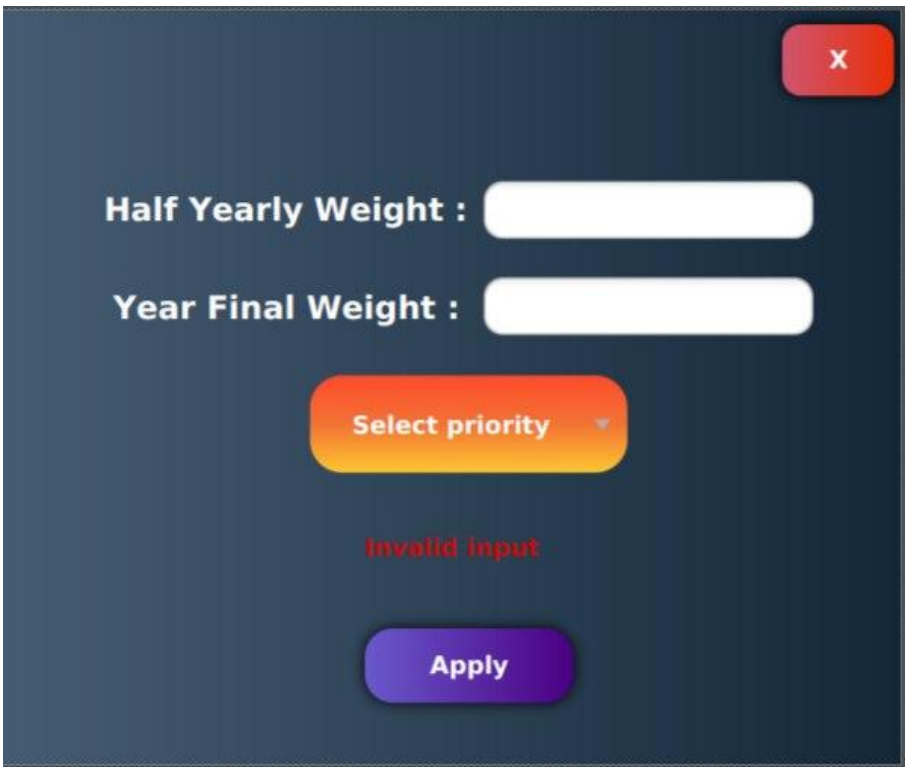


Fig: Set Algorithm for ranking

Fig: Routine

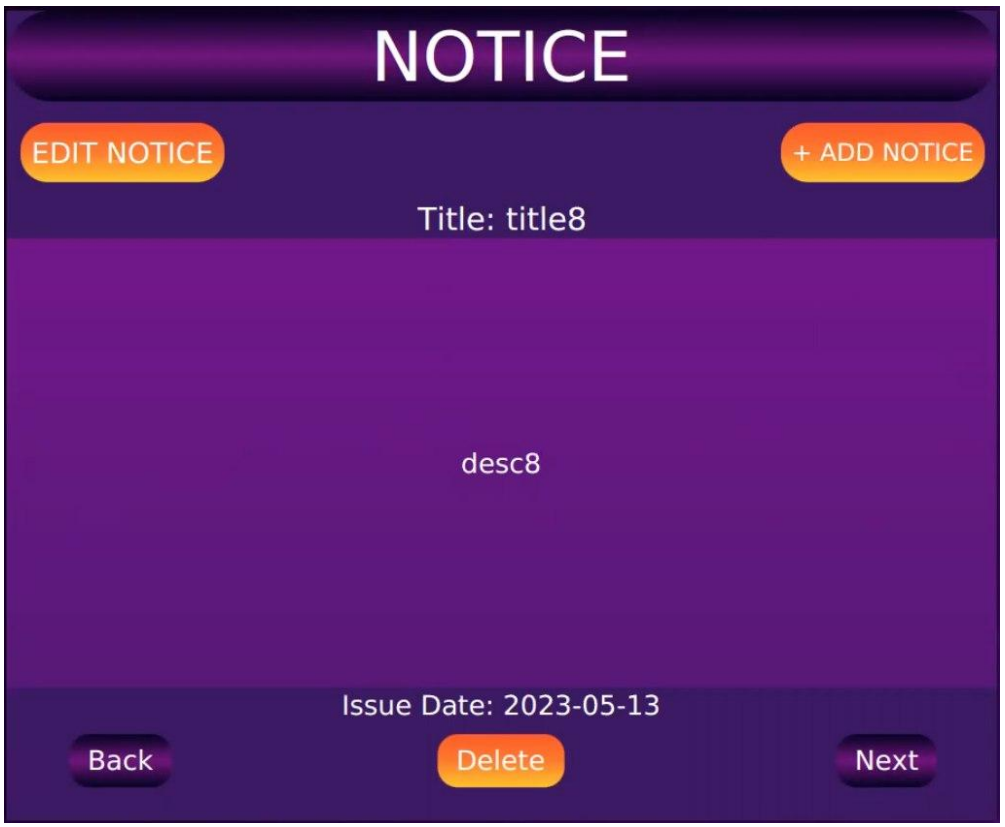


Fig: Notice

Fig: Club

Fig: Club Member List

Fig: New Club Form

Future plan of the project:

1. Improving the UI design of the project.
2. Adding more features and making it more scalable.
3. Making the UI responsive so that it can fit with different screen sizes.
4. Adding networking in the project so that we can make it online based.
5. Deploy the project, making a user base, and improving the features according to their feedback.

Conclusion:

The school management system project aims to streamline and enhance various administrative tasks within a school, including student management, staff management, attendance tracking, grade management, and communication between stakeholders. By implementing this system, the school can improve efficiency, reduce paperwork, and provide a centralized platform for accessing and managing critical information.

The school management system offers numerous benefits, such as automated student registration, easy generation of student reports and transcripts, efficient scheduling of classes and examinations, and improved communication between teachers, students, and parents. It provides a user-friendly interface that simplifies complex tasks and empowers users with quick access to relevant information.

Additionally, the system ensures data accuracy and security by centralizing information and implementing proper access controls. It allows authorized personnel to access and update data securely, while maintaining data integrity and privacy.

By implementing a school management system, the school can foster better collaboration, enhance communication, and streamline administrative processes, ultimately leading to improved productivity, better resource utilization, and a more efficient learning environment. The system can serve as a valuable tool in managing school operations, supporting decision-making, and improving overall organizational effectiveness.

In conclusion, the school management system project offers a comprehensive solution to address the administrative needs of the school, enabling efficient management of students, staff, and various academic processes. Its implementation can contribute to a more organized and effective educational institution, benefiting students, teachers, administrators, and parents alike.

Github repository link:

https://github.com/imran-rahman543/School_Management_System_with_Java