Gyalpozhing College of Information Technology
Royal University of Bhutan
Kabjisa, Chamjekha, Thimphu

Fullstack

CSF304 Design Patterns

Final Report On Attendance Management System

Group members:

Kuenga Tshering, 12210018

Nim Dorji, 12210021

Sonam Rigphel, 12210031

Attendance Management System

Attendance Management System

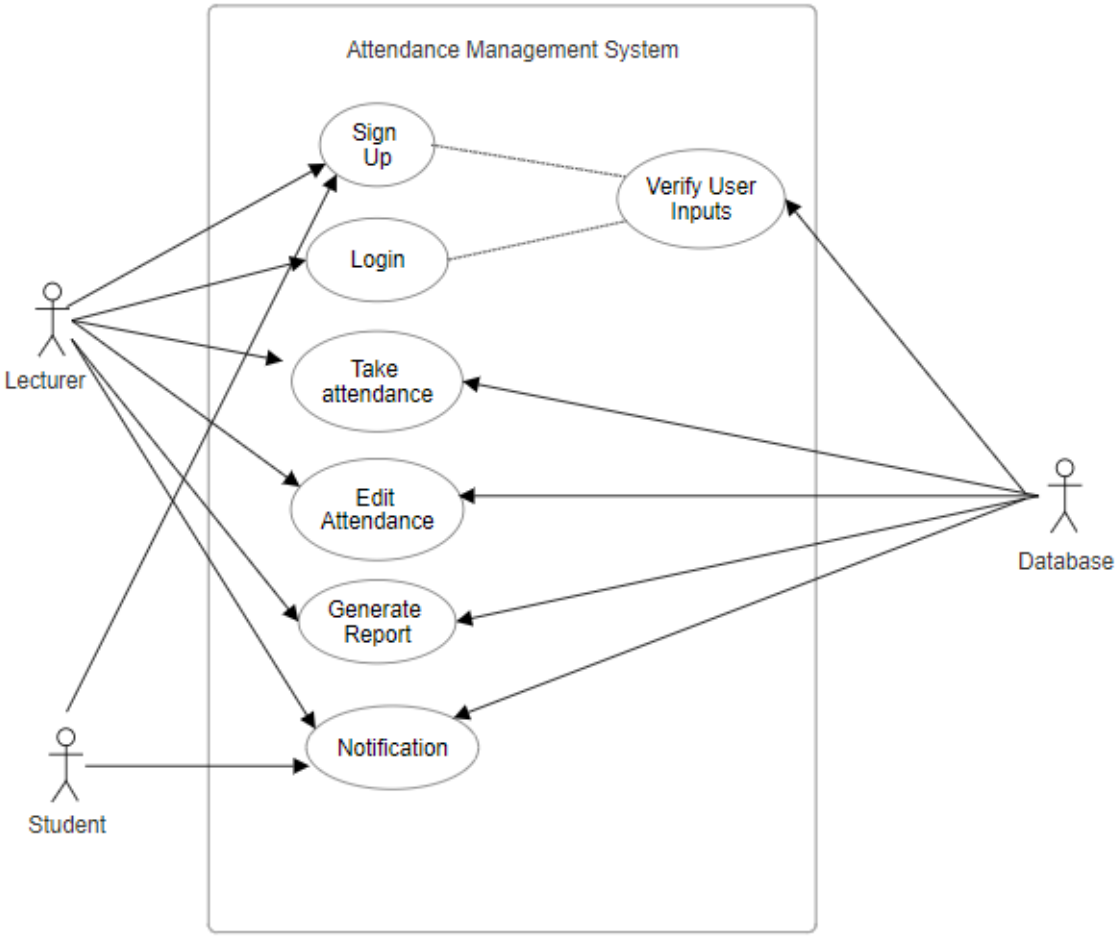
1. **Project Title:**

   Attendance Management System

2. **Brief Project Description:**

   The Student Attendance Management System is a project designed to automate and streamline the process of recording and managing student attendance. The system will provide functionalities for students and lecturers to track attendance on the particular date for each module (Students), generate reports and take attendance (lecturer).

3. **Use Cases(Functionality):**



Attendance Management System

I. Signup/Registration:

Actors: Lecturers

Description: New users can register by submitting their personal and authentication information.

II. Login/Authentication:

Actors: Students, Teachers

Description: Users can access the system by logging in with their credentials.

III. Mark Attendance:

Actors: Lecturers

Description: Lecturers enrolled in a specific module can record attendance for their classes.

IV. View Attendance:

Actors: Students

Description: Users can check attendance records for specific dates for all the modules they are enrolled in.

V. Generate Reports:

Actors: Students

Description: The system can generate detailed attendance reports for individual students for weekly and monthly basis.

VI. Edit Attendance

Actors: Lecturers

Description: Lecturers can modify attendance records for their classes, updating students' status as present or absent. They choose a date to mark attendance, and the interface allows them to mark students accordingly.

VII. Notification

Actors: Student

Description: A message is displayed on the students' page if their attendance falls below 90 percent.

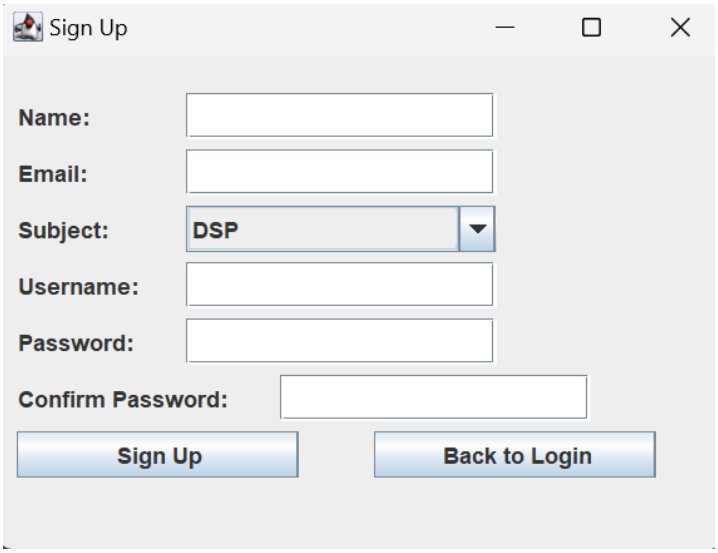
4. **Source Code:**

https://github.com/Rickphell/Group4_Attendance_Management_System/tree/master

5. **User Interface: (How to use it):**

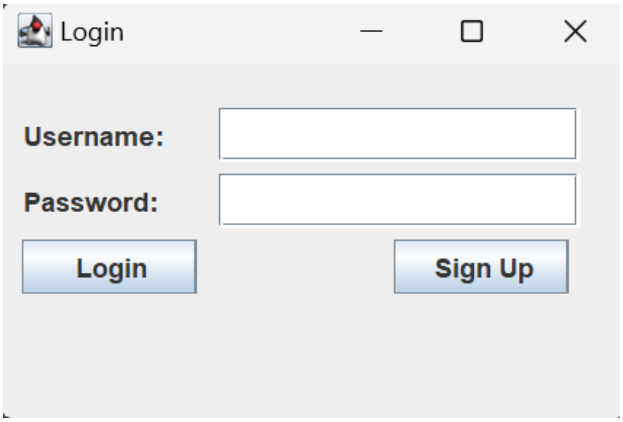
I. Signup Page:

Lecturers enter their personal information and set a password.



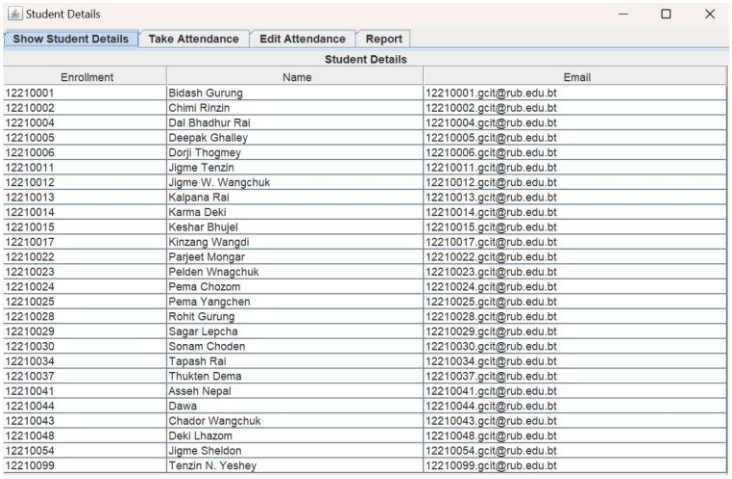
II. Login Page:

Users (Student or Lecturers) enter their username and password to log in.



III. View Student details

If the user is a lecturer, they have access to view detailed student information, including their enrollment number, name, and email address, for students enrolled in a specific module.

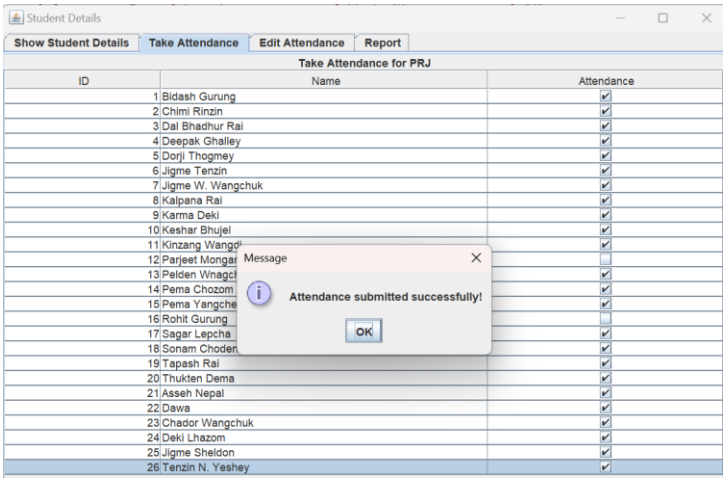


Enrollment	Name	Email
12210001	Bidash Gurung	12210001.gcit@rub.edu.bt
12210002	Chimi Rinzin	12210002.gcit@rub.edu.bt
12210004	Dal Bhadhur Rai	12210004.gcit@rub.edu.bt
12210005	Deepak Ghalley	12210005.gcit@rub.edu.bt
12210006	Dorji Thogmey	12210006.gcit@rub.edu.bt
12210011	Jigme Tenzin	12210011.gcit@rub.edu.bt
12210012	Jigme W. Wangchuk	12210012.gcit@rub.edu.bt
12210013	Kalpana Rai	12210013.gcit@rub.edu.bt
12210014	Karma Deki	12210014.gcit@rub.edu.bt
12210015	Keshar Bhujel	12210015.gcit@rub.edu.bt
12210017	Kinzang Wangdi	12210017.gcit@rub.edu.bt
12210022	Parjeet Mongar	12210022.gcit@rub.edu.bt
12210023	Pelden Wnagchuk	12210023.gcit@rub.edu.bt
12210024	Pema Chozom	12210024.gcit@rub.edu.bt
12210025	Pema Yangchen	12210025.gcit@rub.edu.bt
12210028	Rohit Gurung	12210028.gcit@rub.edu.bt
12210029	Sagar Lepcha	12210029.gcit@rub.edu.bt
12210030	Sonam Choden	12210030.gcit@rub.edu.bt
12210034	Tapash Rai	12210034.gcit@rub.edu.bt
12210037	Thukten Dema	12210037.gcit@rub.edu.bt
12210041	Asseh Nepal	12210041.gcit@rub.edu.bt
12210044	Dawa	12210044.gcit@rub.edu.bt
12210043	Chador Wangchuk	12210043.gcit@rub.edu.bt
12210048	Deki Lhazom	12210048.gcit@rub.edu.bt
12210054	Jigme Sheldon	12210054.gcit@rub.edu.bt
12210099	Tenzin N. Yeshey	12210099.gcit@rub.edu.bt

IV. Take Attendance

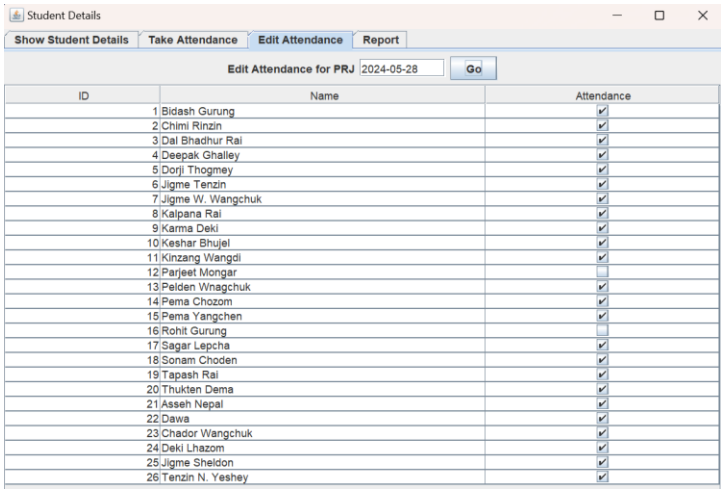
As a lecturer, they have the capability to mark attendance for students, indicating their presence or absence, and subsequently save this information in the database.

ID	Name	Attendance
1	Bidash Gurung	☑
2	Chimi Rinzin	☑
3	Dal Bhadhur Rai	☑
4	Deepak Ghalley	☑
5	Dorji Thogmey	☑
6	Jigme Tenzin	☑
7	Jigme W. Wangchuk	☑
8	Kalpana Rai	☑
9	Karma Deki	☑
10	Keshar Bhujel	☑
11	Kinzang Wangdi	☑
12	Parjeet Mongar	☐
13	Pelden Wnagchuk	☑
14	Pema Chozom	☑
15	Pema Yangchen	☑
16	Rohit Gurung	☐
17	Sagar Lepcha	☑
18	Sonam Choden	☑
19	Tapash Rai	☑
20	Thukten Dema	☑
21	Asseh Nepal	☑
22	Dawa	☑
23	Chador Wangchuk	☑
24	Deki Lhazom	☑
25	Jigme Sheldon	☑
26	Tenzin N. Yeshey	☑

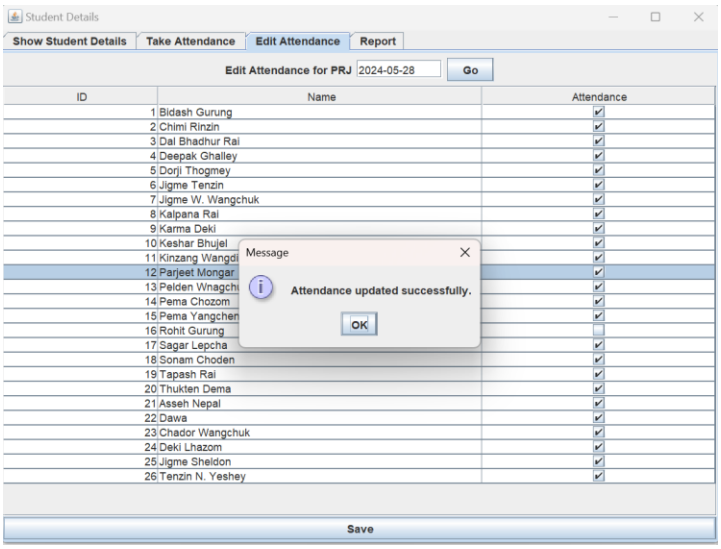


V. Edit Attendance

Lecturers are responsible for updating the attendance records for their classes, with the ability to adjust the attendance or absence status of their students. Instructors specify a designated day for recording attendance and have the capability to edit it accordingly. Students can be marked as present or absent through the interface provided.

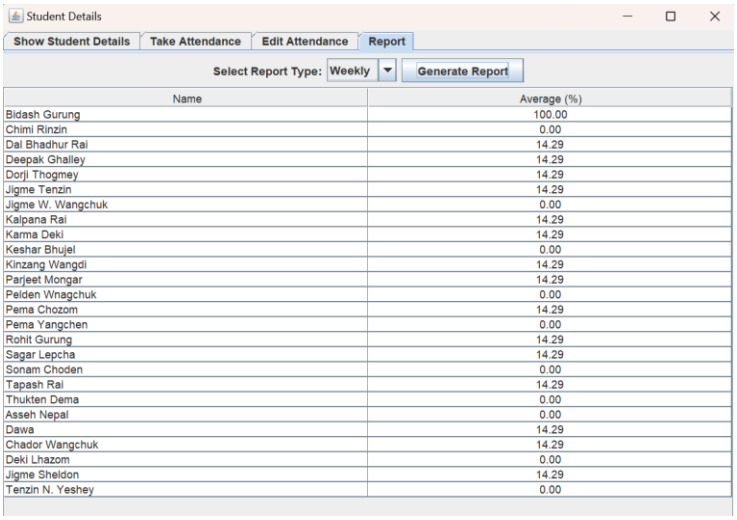


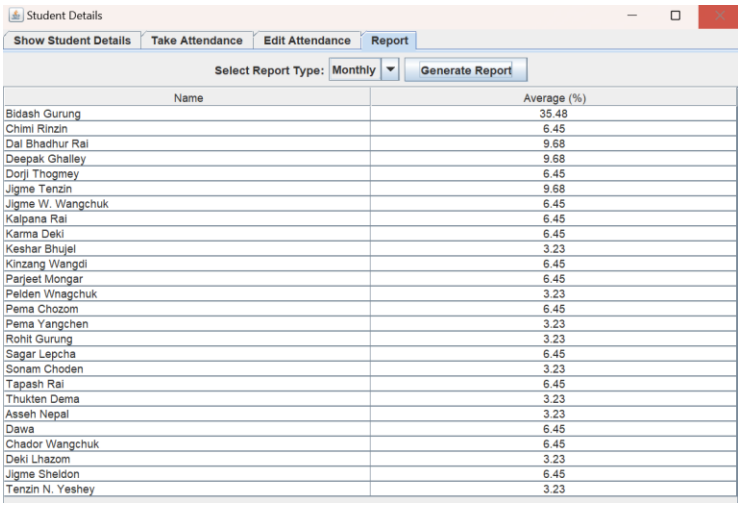
Attendance Management System



VI. Generate Attendance Report

Every student can receive comprehensive weekly and monthly attendance reports from the system.

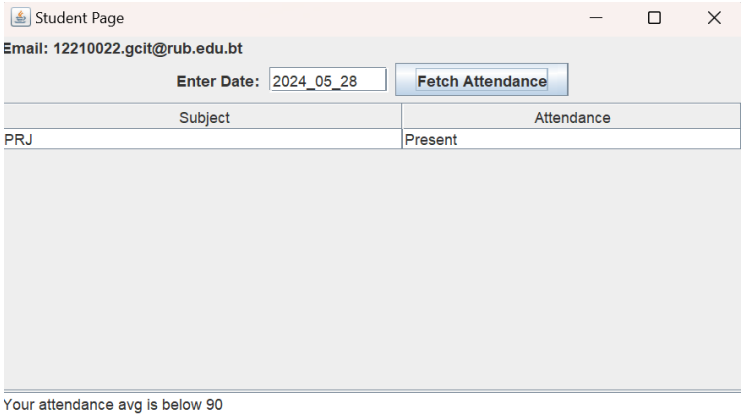




Name	Average (%)
Bidash Gurung	35.48
Chimi Rinzin	6.45
Dal Bhadhur Rai	9.68
Deepak Ghalley	9.68
Dorji Thogmey	6.45
Jigme Tenzin	9.68
Jigme W. Wangchuk	6.45
Kalpana Rai	6.45
Karma Deki	6.45
Keshar Bhujel	3.23
Kinzang Wangdi	6.45
Parjeet Mongar	6.45
Pelden Wnagchuk	3.23
Pema Chozom	6.45
Pema Yangchen	3.23
Rohit Gurung	3.23
Sagar Lepcha	6.45
Sonam Choden	3.23
Tapash Rai	6.45
Thukten Dema	3.23
Asseh Nepal	3.23
Dawa	6.45
Chador Wangchuk	6.45
Deki Lhazom	3.23
Jigme Sheldon	6.45
Tenzin N. Yeshey	3.23

VII. Notification

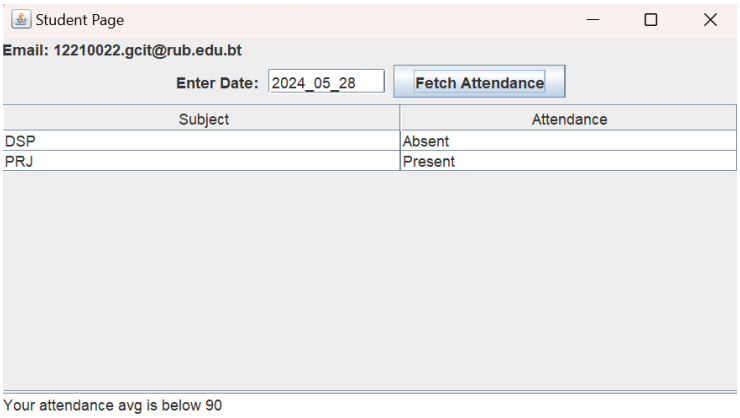
A message appears on the student's page if their attendance is below 90%.



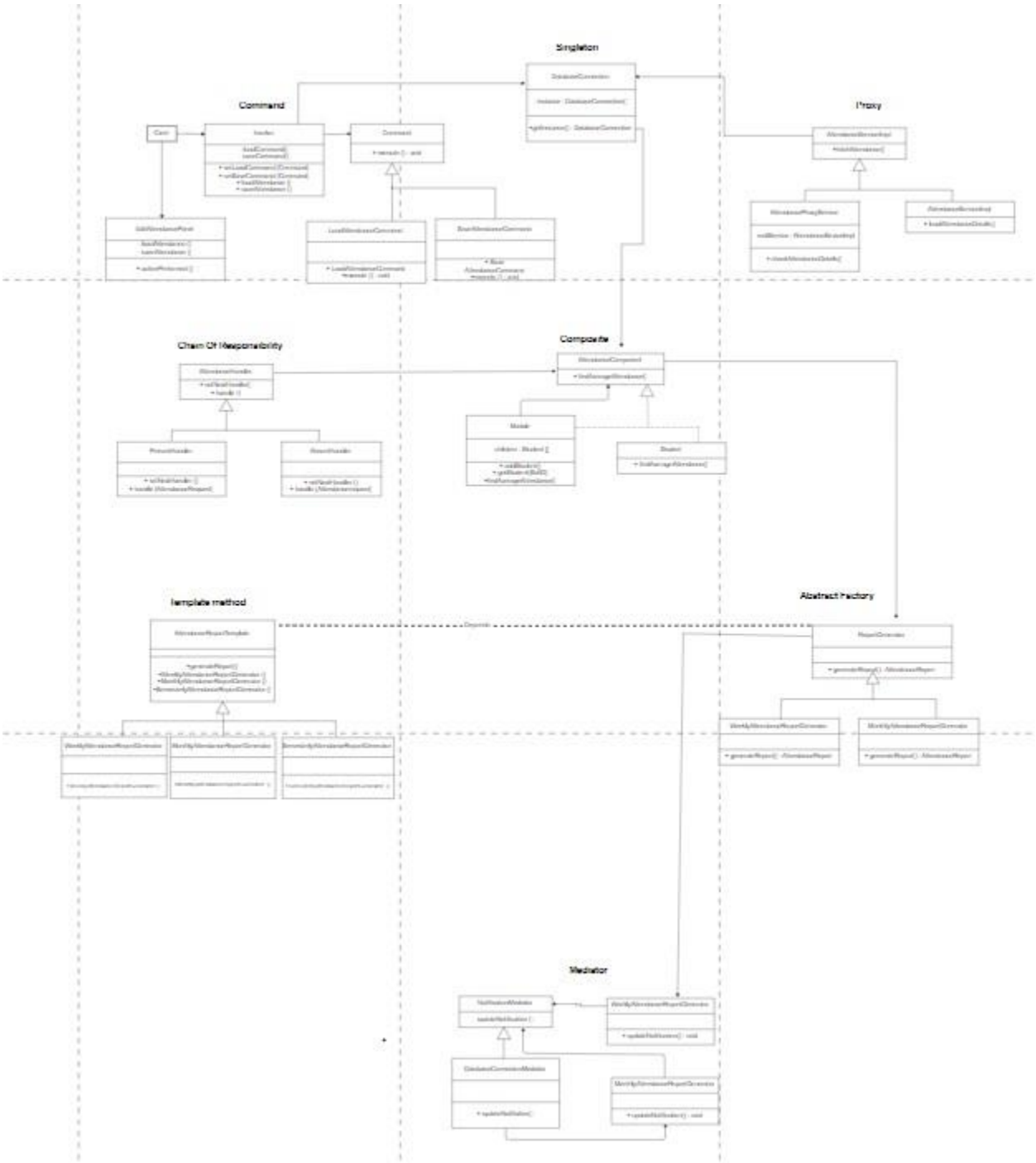
VIII. Viewing ~~Attendance for a S~~pecific Date Across All Modules

Users have the capability to access attendance records for particular dates across all modules in which they are enrolled.

Attendance Management System



6. All Class Diagrams:



Class diagram link:
https://app.diagrams.net/#G1jJ73FrrSCWiCYSZL4cO_M1nOxEQMRsmM#%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D

7. **Justification for all the design pattern used:**

Singleton: Managing multiple connections to the database can be resource-intensive and lead to inconsistencies. The Singleton pattern ensures that only one instance of the database connection is used throughout the application. This centralizes access to the database and avoids the overhead associated with establishing multiple connections.

Abstract factory: Different types of attendance reports (e.g., weekly, monthly) require different calculations and data handling. The Abstract Factory pattern provides a way to encapsulate a group of individual factories that have a common theme. This allows the creation of related objects (attendance report generators).

Proxy: Direct access to sensitive parts of the database by students could lead to security breaches and unauthorized modifications. Proxy acts as a control point, ensuring that any request to load attendance details is logged before being passed to the actual implementation, providing an additional layer of functionality and control.

CoR: Different handlers (e.g., marking attendance as present or absent, handling login) require distinct processing steps that should be processed in a sequence. The Chain of Responsibility pattern allows a request to be passed along a chain of handlers, where each handler decides either to process the request or to pass it to the next handler in the chain.

Command: Command pattern encapsulates editing operations for attendance records as discrete objects, enabling flexible, maintainable, and extendable handling of various edit actions.

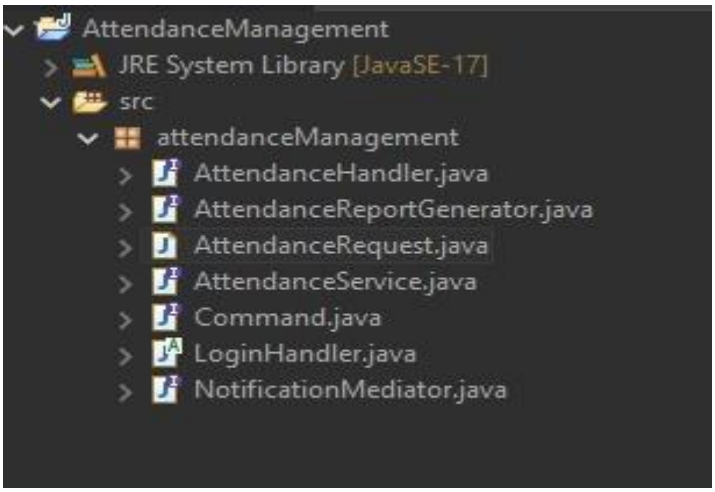
Mediator: Mediator pattern manages notifications for students with attendance below 90% by centralizing interaction logic, promoting loose coupling, enhancing maintainability, and simplifying complex communication between components.\

8. Framework

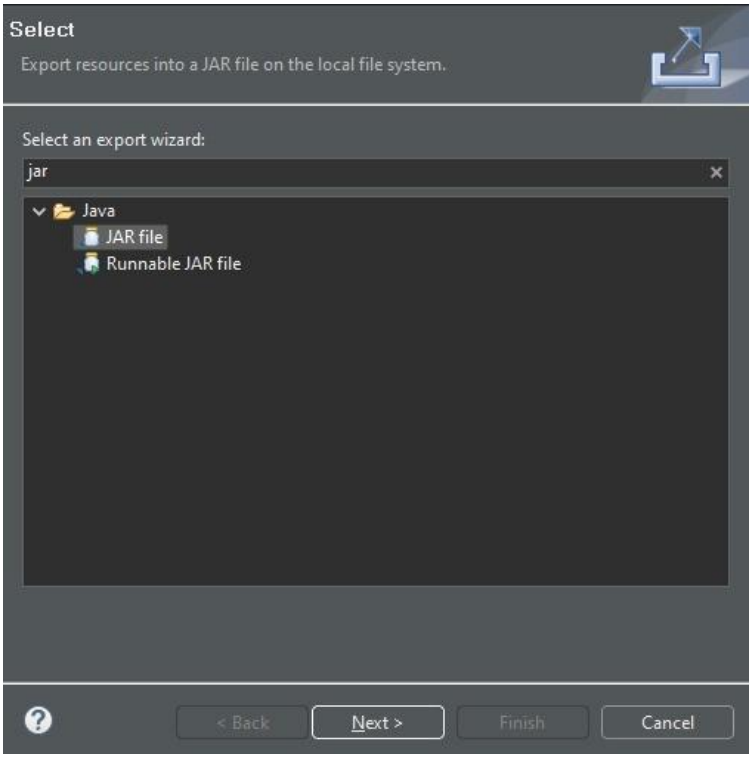
The framework remains constant and cannot be changed, whereas the implementation is flexible and can be adjusted to meet the client's requirements or the application developer's preferences. The framework offers a foundational structure and guidelines, ensuring uniformity and dependability across various applications.

9. How the framework is converted into jar files

I. We ensured our project structure was organized. Our source files were in the src folder, and all resources were placed in their respective folders. In the Package Explorer, we right-clicked on the project we wanted to export i.e, atttendanceMangement.



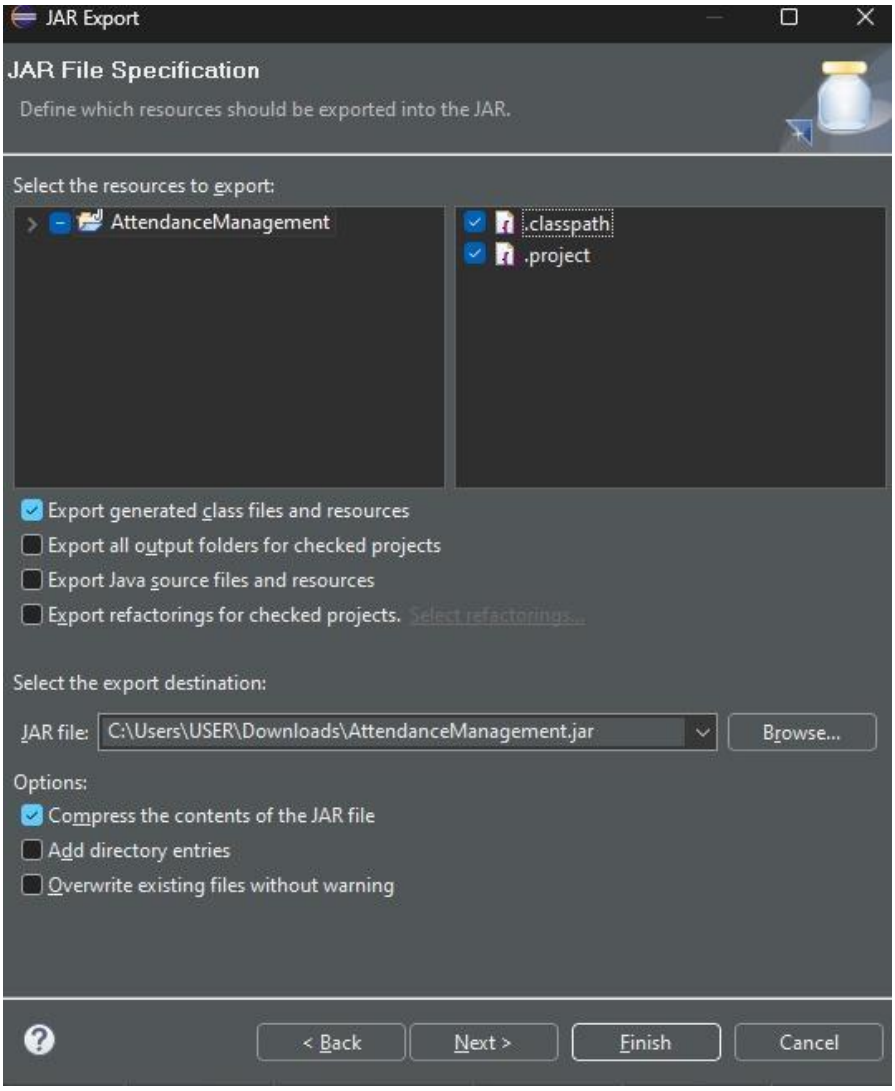
II. From the context menu, we selected Export. In the Export dialog, we expanded the Java folder and selected the JAR file. Then, we clicked Next.



III. We selected ‘export generated class files and resources’ to ensure our compiled .class files were included.

We saved the JAR file as AttendanceManagement.jar in our desired location.

Clicked Finish to create the JAR file.



IV. And in our downloads a new JAR file was created.

10. Challenges:

Patterns: Implementing design patterns in the project was challenging and took a lot of effort.

Scalability: Designing the system to handle a large number of users and records efficiently.

Security: Protecting sensitive user information and preventing unauthorized access.

User Interface: Creating a user-friendly interface that is easy for all user types to navigate.

11. Conclusion:

The Student Attendance Management System effectively addresses the need for a streamlined and automated process of managing student attendance. By utilizing appropriate design patterns and providing a robust set of features, the system enhances the efficiency and accuracy of attendance tracking while offering valuable insights through report generation and notifications. Despite challenges in implementing design patterns, scalability, security, and user interface design, the system provides a comprehensive solution to attendance management in educational institutions.