

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

1. Project Title:	3
2. Brief Project Description:	3
4. Source Code:	5
5. User Interface: (How to use it):	5
6. All Class Diagrams:	11
7. Justification for all the design pattern used:	12
8. How the framework is converted into jar files	13
9. Challenges:	16
10. Conclusion:	16

Attendance Management System

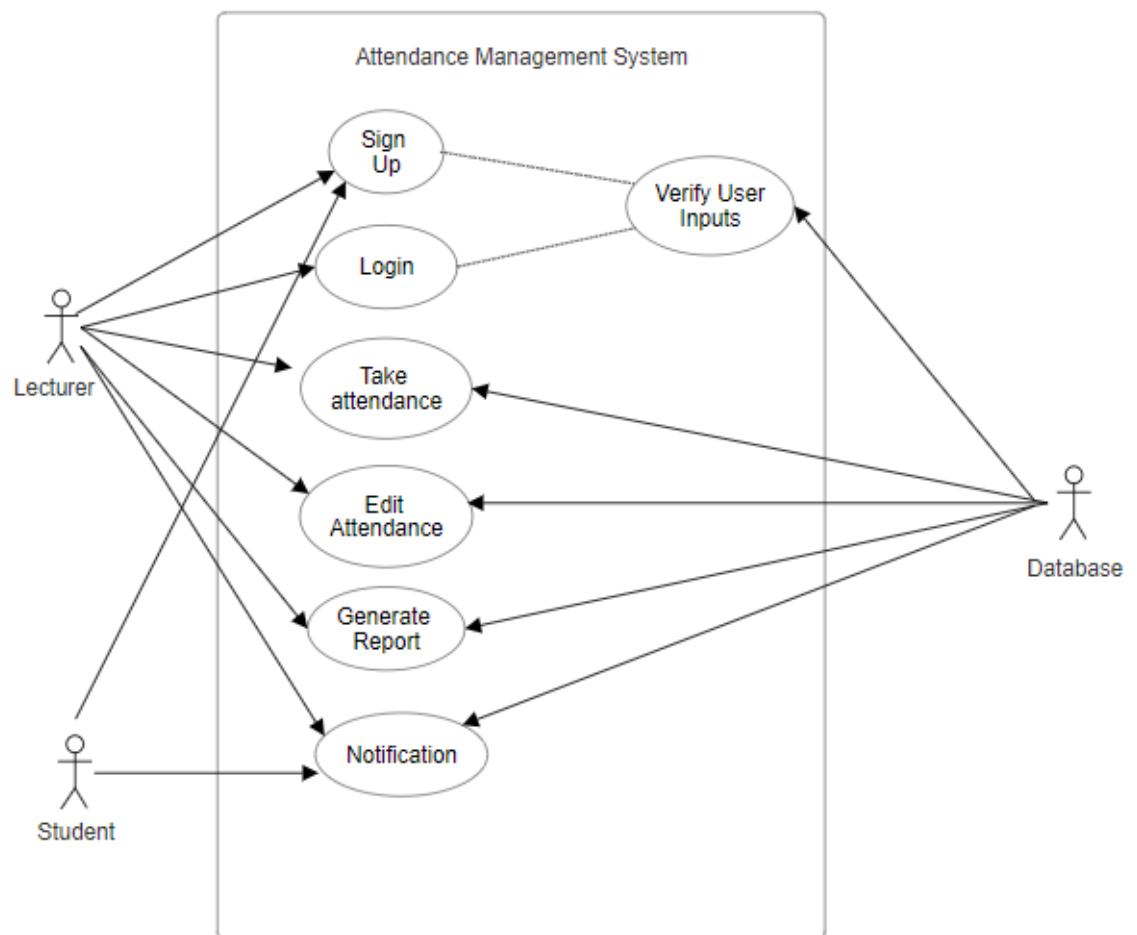
1. Project Title:

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.

Attendance Management System

VII. Notification

Actors: Student

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

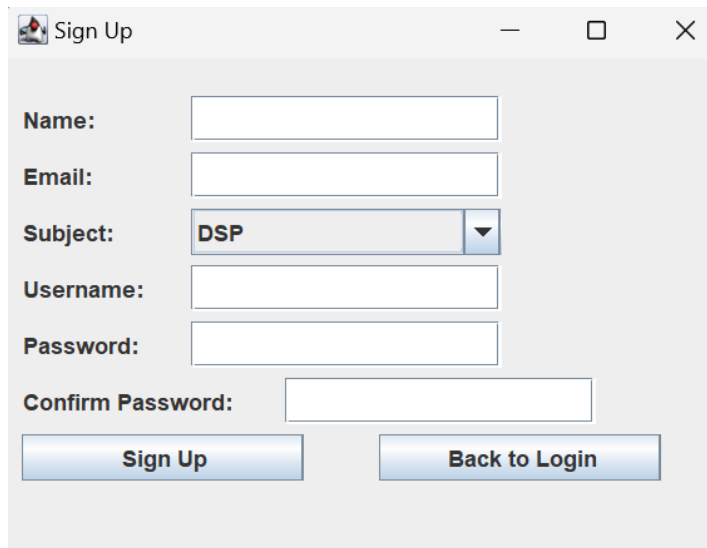
4. Source Code:

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.

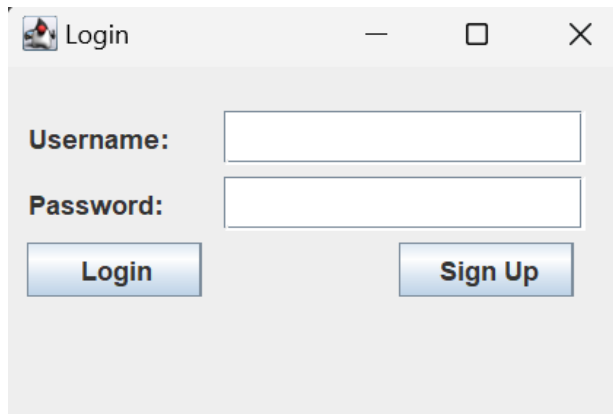


The screenshot shows a 'Sign Up' window with a title bar containing a small icon, the text 'Sign Up', and standard window controls (minimize, maximize, close). The form contains the following fields and controls:

- Name:** A text input field.
- Email:** A text input field.
- Subject:** A dropdown menu with 'DSP' selected.
- Username:** A text input field.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Buttons:** Two buttons at the bottom: 'Sign Up' and 'Back to Login'.

II. Login Page:

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



The screenshot shows a 'Login' window with a title bar containing a small icon, the text 'Login', and standard window controls (minimize, maximize, close). The form contains the following fields and controls:

- Username:** A text input field.
- Password:** A text input field.
- Buttons:** Two buttons at the bottom: 'Login' and 'Sign Up'.

III. View Student details

Attendance Management System

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.



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

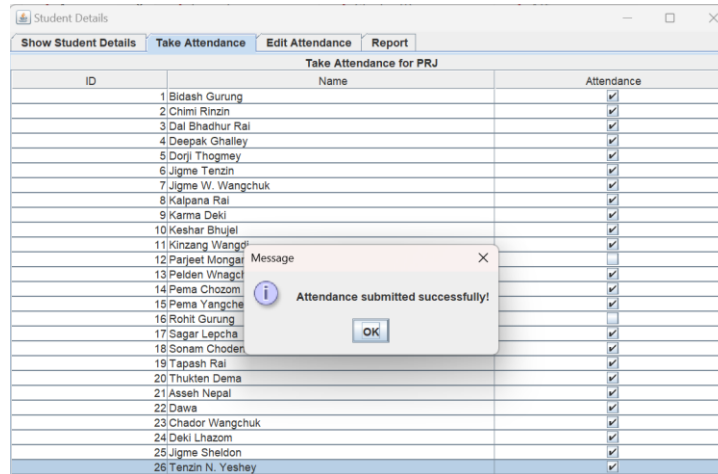
IV. Take Attendance

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.



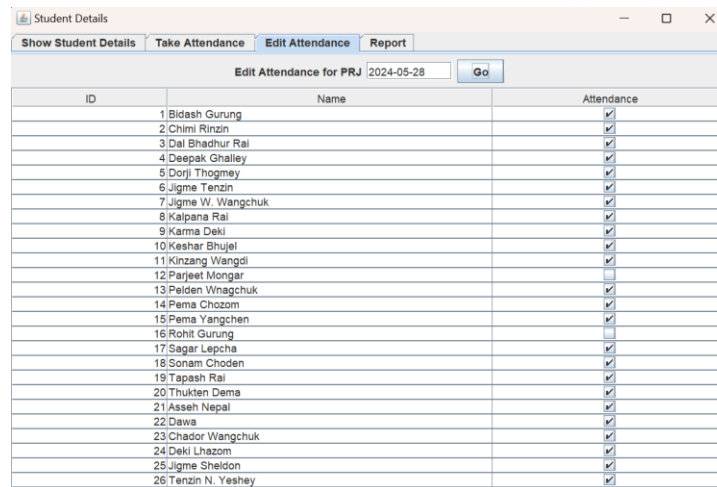
Take Attendance for PRJ		
ID	Name	Attendance
1	Bidash Gurung	<input checked="" type="checkbox"/>
2	Chimi Rinzin	<input checked="" type="checkbox"/>
3	Dal Bhadrur Rai	<input checked="" type="checkbox"/>
4	Deepak Ghalley	<input checked="" type="checkbox"/>
5	Dorji Thogmey	<input checked="" type="checkbox"/>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

Attendance Management System



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

Student Details

Show Student Details

Take Attendance

Edit Attendance

Report

Edit Attendance for PRJ

2024-05-28

Go

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

Save

Message

Attendance updated successfully.

OK

VI. Generate Attendance Report

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

Student Details

Show Student Details

Take Attendance

Edit Attendance

Report

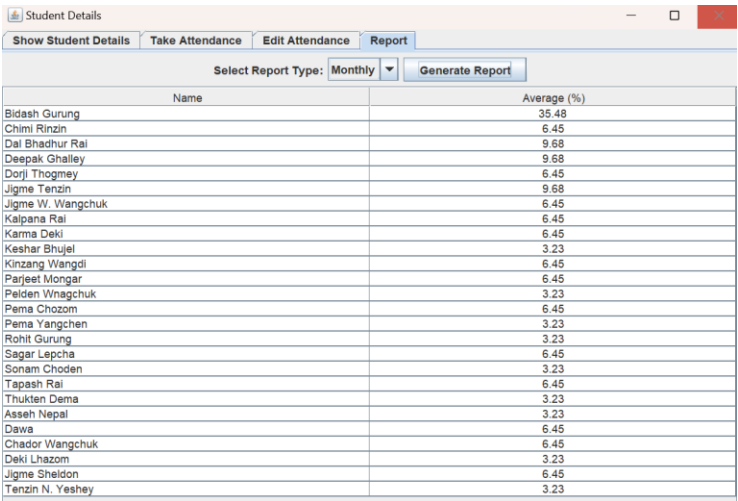
Select Report Type:

Weekly

Generate Report

Name	Average (%)
Bidash Gurung	100.00
Chimi Rinzin	0.00
Dal Bhadrur Rai	14.29
Deepak Ghalle	14.29
Dorji Thogmey	14.29
Jigme Tenzin	14.29
Jigme W. Wangchuk	0.00
Kalpana Rai	14.29
Karma Deki	14.29
Keshar Bhujel	0.00
Kinzang Wangdi	14.29
Parjeet Mongar	14.29
Pelden Wnagchuk	0.00
Pema Chozom	14.29
Pema Yangchen	0.00
Rohit Gurung	14.29
Sagar Lepcha	14.29
Sonam Choden	0.00
Tapash Rai	14.29
Thukten Dema	0.00
Asseh Nepal	0.00
Dawa	14.29
Chador Wangchuk	14.29
Deki Lhazom	0.00
Jigme Sheldon	14.29
Tenzin N. Yeshey	0.00

Attendance Management System

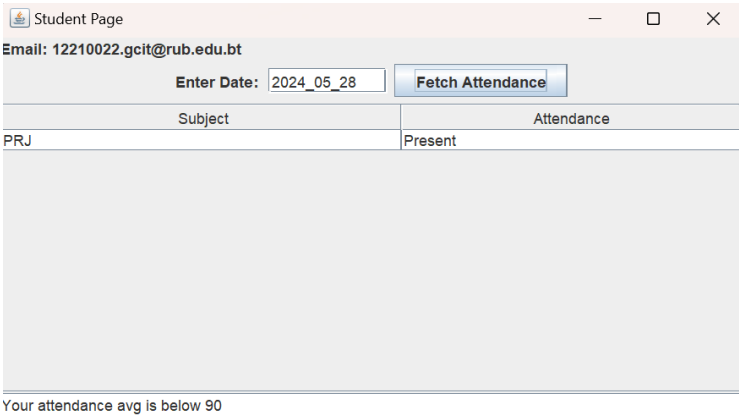


The screenshot shows a web application window titled "Student Details". It has four tabs: "Show Student Details", "Take Attendance", "Edit Attendance", and "Report". The "Report" tab is active. Below the tabs, there is a "Select Report Type:" dropdown menu set to "Monthly" and a "Generate Report" button. The main content area displays a table with two columns: "Name" and "Average (%)". The table lists 30 students and their respective attendance averages.

Name	Average (%)
Bidash Gurung	35.48
Chimi Rinzin	6.45
Dal Bhadrur 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 Wangchuk	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%.



The screenshot shows a web application window titled "Student Page". It displays the email "12210022.gcit@rub.edu.bt". Below the email, there is a form with an "Enter Date:" label, a text input field containing "2024_05_28", and a "Fetch Attendance" button. Below the form, there is a table with two columns: "Subject" and "Attendance". The table has one row with "PRJ" in the "Subject" column and "Present" in the "Attendance" column. Below the table, there is a message: "Your attendance avg is below 90".

Subject	Attendance
PRJ	Present

Your attendance avg is below 90

VIII. Viewing Attendance for a Specific 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

Student Page

Email: 12210022.gcit@rub.edu.bt

Enter Date: 2024_05_28

Fetch Attendance

Subject	Attendance
DSP	Absent
PRJ	Present

Your attendance avg is below 90

The image displays seven UML class diagrams, each representing a different design pattern. The diagrams are organized into a grid with dashed lines separating them.

- Command:** Shows a `Client` using a `Invoker` to execute a `Command`. The `Invoker` has methods `addCommand()`, `removeCommand()`, `executeCommand()`, and `undoCommand()`. The `Command` interface has `execute()` and `undo()`. Concrete classes include `AddMovieToQueueCommand`, `RemoveMovieFromQueueCommand`, `PlayMovieCommand`, and `StopMovieCommand`.
- Singleton:** Shows a `DatabaseConnection` class with a static `instance` and a static `getConnection()` method. It also shows a `DatabaseConnectionPool` class.
- Proxy:** Shows an `AbstractMoviePlayer` interface and a `MoviePlayerProxy` class that implements it. The `MoviePlayerProxy` class has a `movie` attribute and a `loadMovieFromDisk()` method.
- Chain of Responsibility:** Shows an `AbstractMoviePlayer` interface and two concrete classes, `MoviePlayer1` and `MoviePlayer2`, both implementing the `AbstractMoviePlayer` interface.
- Composite:** Shows an `AbstractMoviePlayer` interface and two concrete classes, `Movie` and `Director`. The `Movie` class has a `children` attribute and a `loadMovie()` method. The `Director` class has a `loadMovie()` method and a `loadMovieFromDisk()` method.
- Template Method:** Shows an `AbstractMoviePlayerTemplate` class with a `playMovie()` method. It has three concrete subclasses: `MoviePlayerTemplate1`, `MoviePlayerTemplate2`, and `MoviePlayerTemplate3`.
- Abstract Factory:** Shows an `AbstractMoviePlayerFactory` interface and two concrete classes, `MoviePlayerFactory1` and `MoviePlayerFactory2`, both implementing the `AbstractMoviePlayerFactory` interface.
- Mediator:** Shows a `MoviePlayer` interface and two concrete classes, `MoviePlayer1` and `MoviePlayer2`. The `MoviePlayer` interface has a `loadMovie()` method. The `MoviePlayer1` class has a `loadMovie()` method and a `loadMovieFromDisk()` method. The `MoviePlayer2` class has a `loadMovie()` method and a `loadMovieFromDisk()` method.

Attendance Management System

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.

Attendance Management System

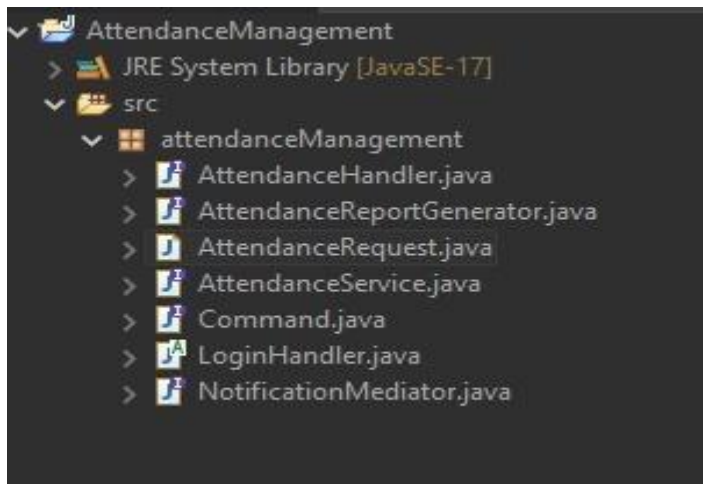
Mediator: Mediator pattern manages notifications for students with attendance below 90% by centralizing interaction logic, promoting loose coupling, enhancing maintainability, and simplifying complex communication between components.\

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.

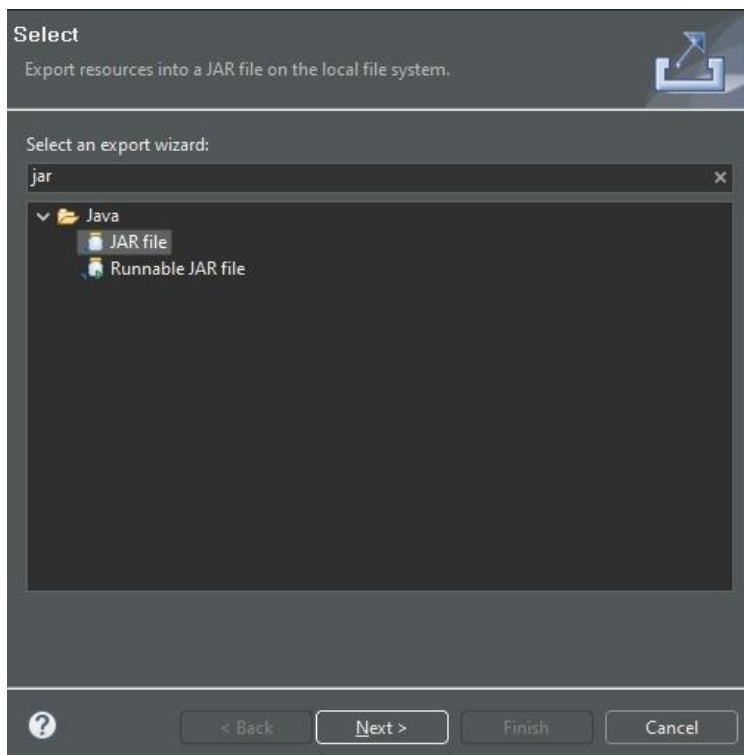
9. How the framework is converted into jar files

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, attendanceMangement.



Attendance Management System

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.

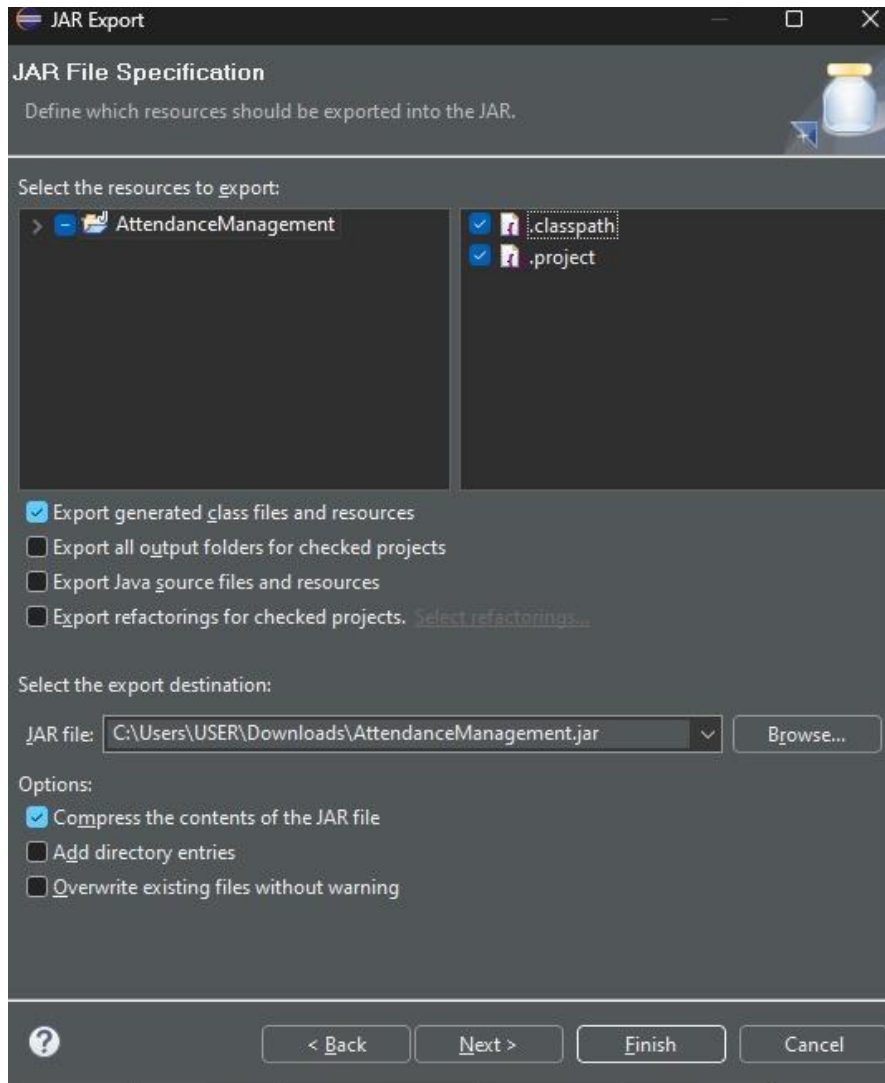


III. We selected 'export generated class files and resources' to ensure our compiled .class files were included.

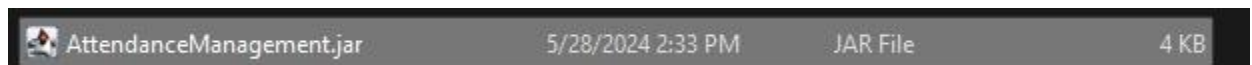
We saved the JAR file as AttendanceManagement.jar in our desired location.

Clicked Finish to create the JAR file.

Attendance Management System



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



Attendance Management System

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.