



A Project Report on

School Management System

Department of Computer Science and Engineering
Object-Oriented Programming Lab
CSE 2112
University of Dhaka

Submitted To:
Dr. Chowdhury Farhan Ahmed
Professor
Md. Redwan Ahmed Rizvee
Lecturer

Submitted By:
Group ID: 9

H.M. Mehedi Hasan(13)
MD.Abu Bakar Siddique(47)

2nd Year 1st Semester
Session: 2022-2023

Project Overview

The **School Management System** is a comprehensive software solution designed to facilitate the management of school operations for administrators, teachers, and students. This system streamlines various tasks such as **user authentication**, **student and teacher management**, **attendance tracking**, **fee payment processing**, **study material sharing**, and **real-time chat**. Built with **Java Swing** for UI, **MySQL** for database management, and **Apache Ant** for project automation, this system ensures modularity, scalability, and efficiency.

Objectives

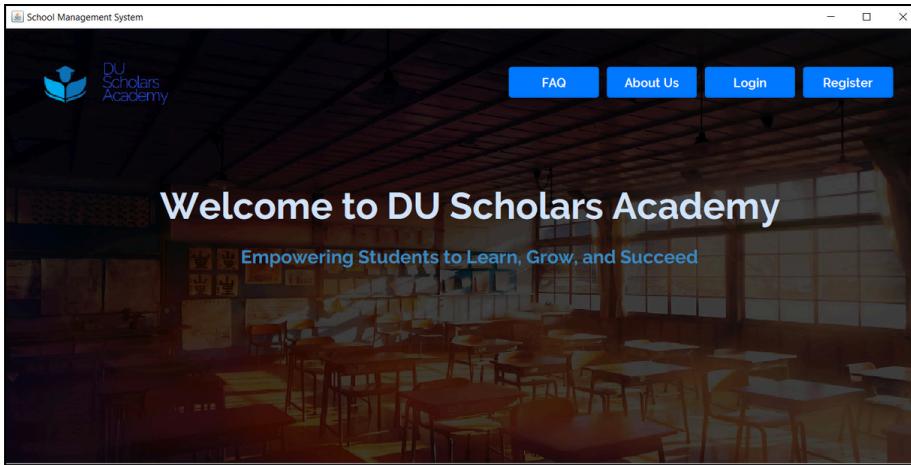
- **Automation:** Reduce manual efforts in school administration by digitizing attendance, payment tracking, and report generation.
 - **User Accessibility:** Provide a secure and intuitive platform for admins, teachers, and students.
 - **Scalability:** Implement a modular design to allow easy extension with additional features.
 - **Real-Time Communication:** Enable seamless interaction between students and teachers through chat functionality.
 - **Data Management:** Maintain structured student and teacher records with secure database storage.
-

Platform and Tools

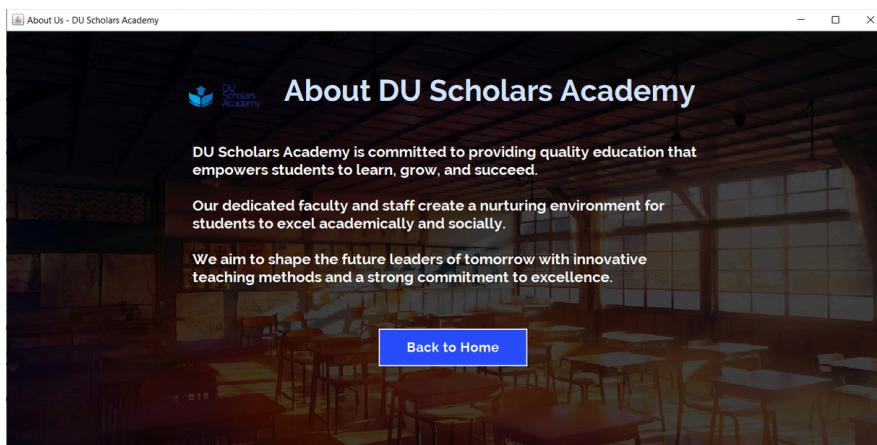
- **Programming Language:** Java (JDK 22)
 - **IDE:** NetBeans
 - **Build Tool:** Apache Ant
 - **Database:** MySQL
 - **UI Libraries & Tools:**
 - **Java Swing** (Graphical User Interface)
 - **JCalendar** (Date Selection)
 - **iText** (PDF Generation)
 - **Networking:**
 - **Socket Programming** (Real-Time Chat)
 - **Multithreading** (Concurrent Message Handling)
-

Pages & Sections(UI):

Landing Page:

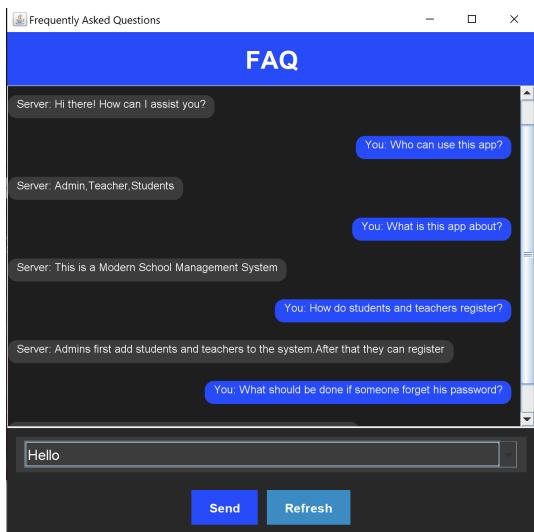


- About Us:

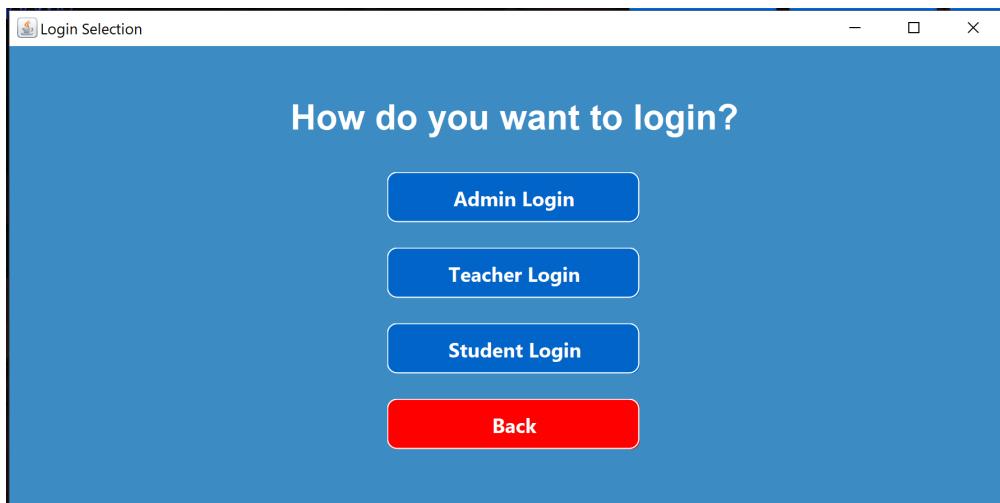


- FAQ Chatbot:

Built using **socket programming** for **real-time question handling**. Provides **instant responses to common questions** related to application, registration, login etc.

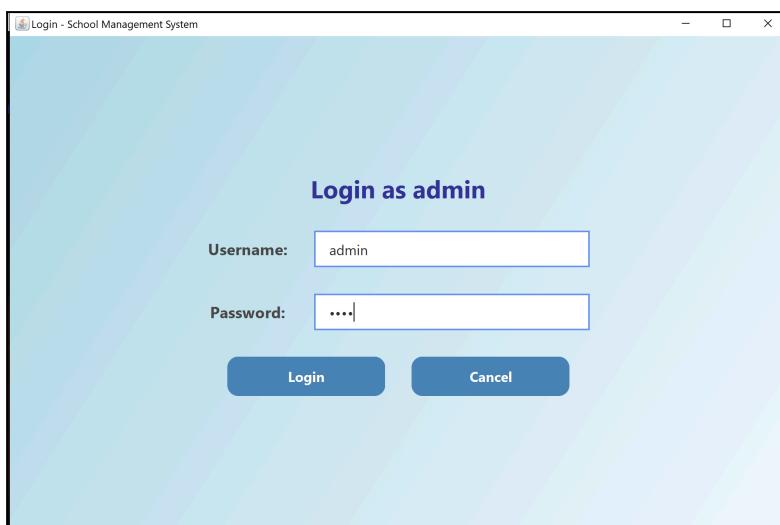


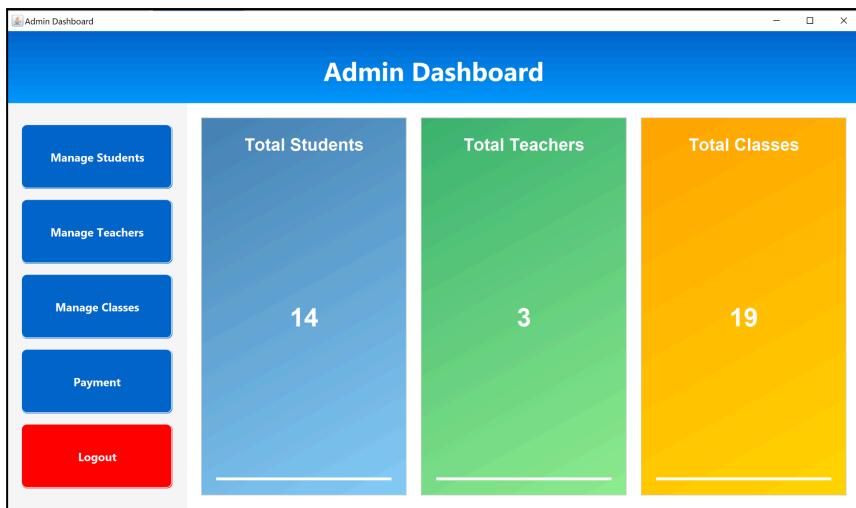
- **Features:**



Admin Section:

- **Login Authentication** – Secure admin access.

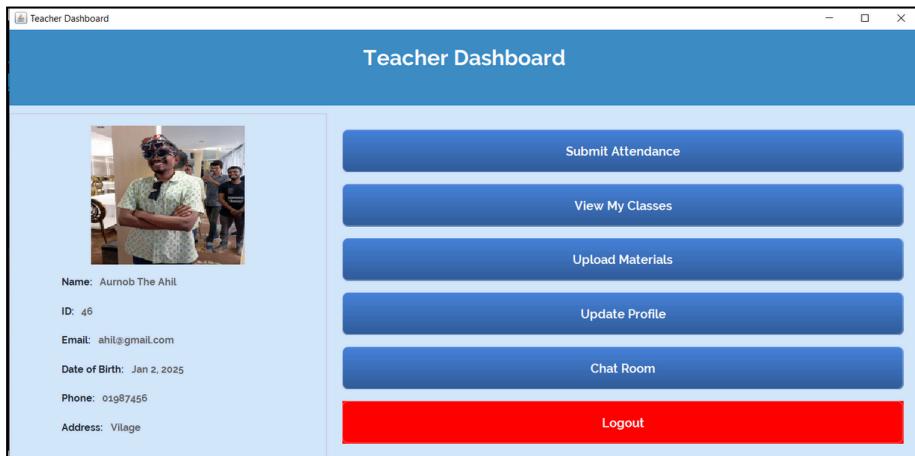




- **Manage Students & Teachers** – Add, update, delete records.
- **Class Management** – Assign teachers to specific classes.
- **Fee Management** – Set and manage payments.

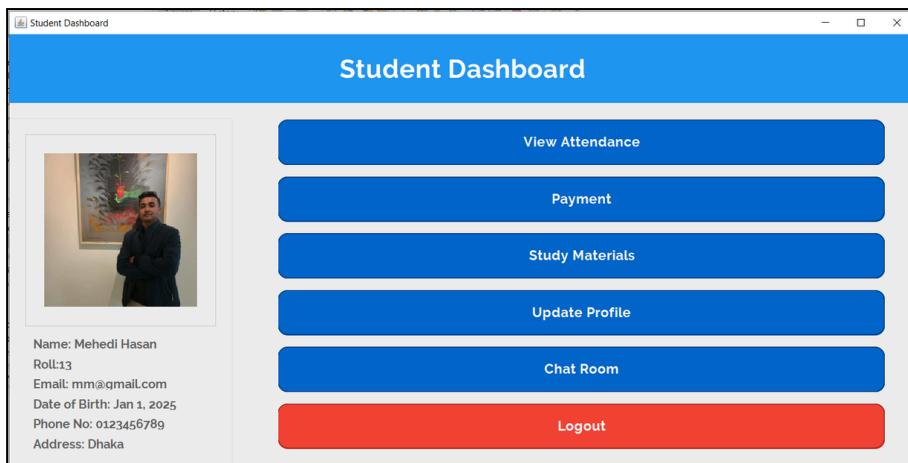
Teacher Section:

- **Attendance Management** – Record and track student attendance.
- **Study Material Upload** – Share PDFs, images, and notes.
- **Real-Time Chat** – Communicate with students.



Student Section:

- **View Attendance & Grades** – Check academic progress.
- **Download Study Materials** – Access shared resources.
- **Fee Payment** – Calculate fees and download receipts.
- **Chat with Teachers** – Get assistance in real time.

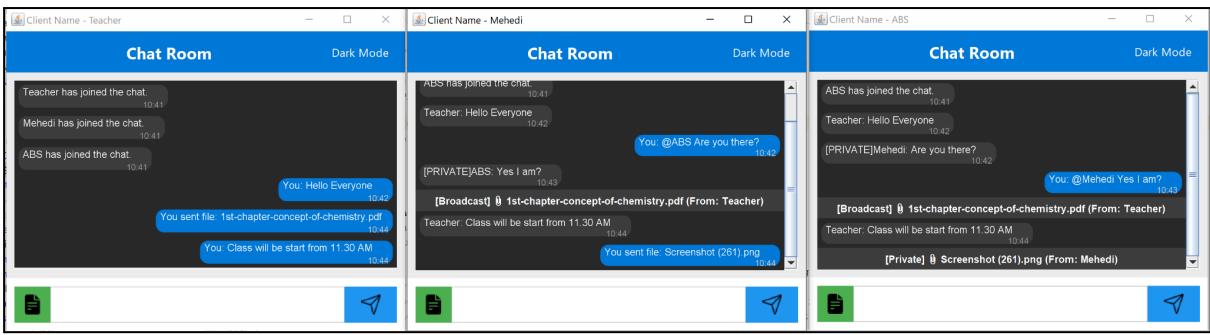


The screenshot shows the 'Student Fee Payment' window. It displays student details and fee-related information. The student's name is Mehedi Hasan, roll number is 13, and class is 10A. The total due amount is 80.00, and the given amount is 7700. The remaining amount is 80.00. At the bottom, there are three blue buttons: 'Calculate', 'Pay', and 'Print Receipt'.

Student Name:	Mehedi Hasan
Roll Number:	13
Class:	10A
Total Due Amount:	80.00
Given Amount:	7700
Remaining Amount:	80.00

Realtime Chat:

- Uses **Socket Programming** to handle real-time queries.
- Provides answers on **registration, login, payments, attendance, and study materials**.
- Reduces manual workload for admins.



UML Diagram

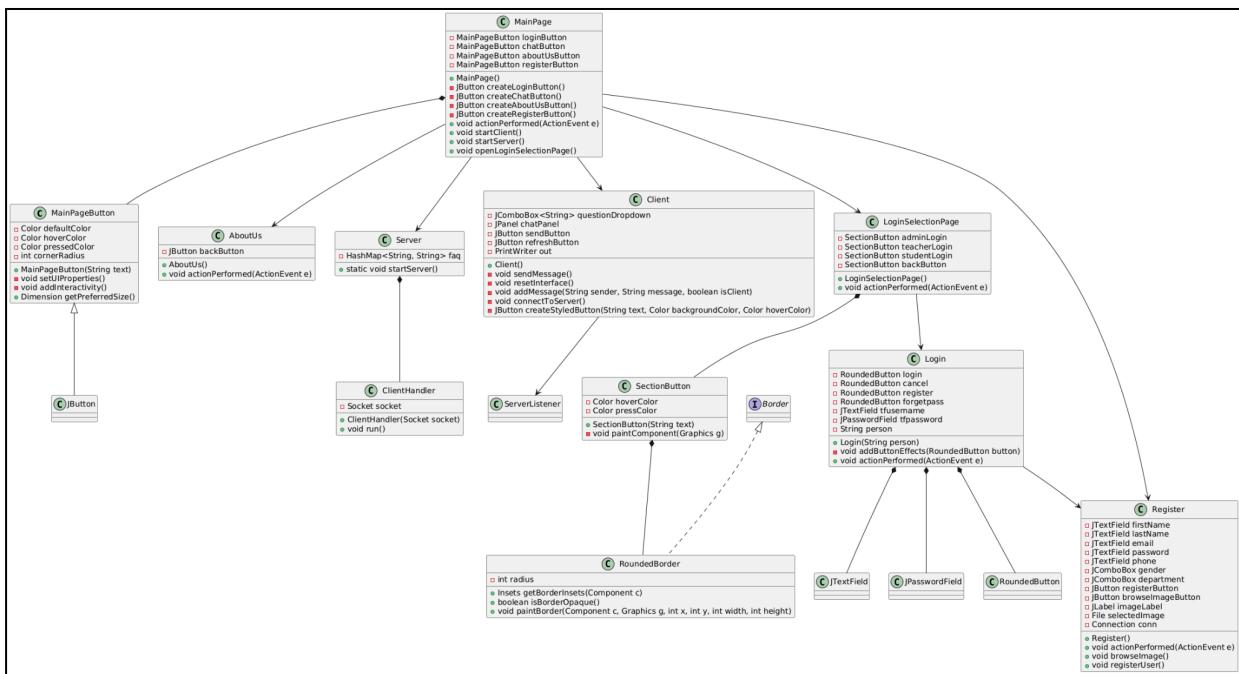


Figure: UML Diagram - MainPage

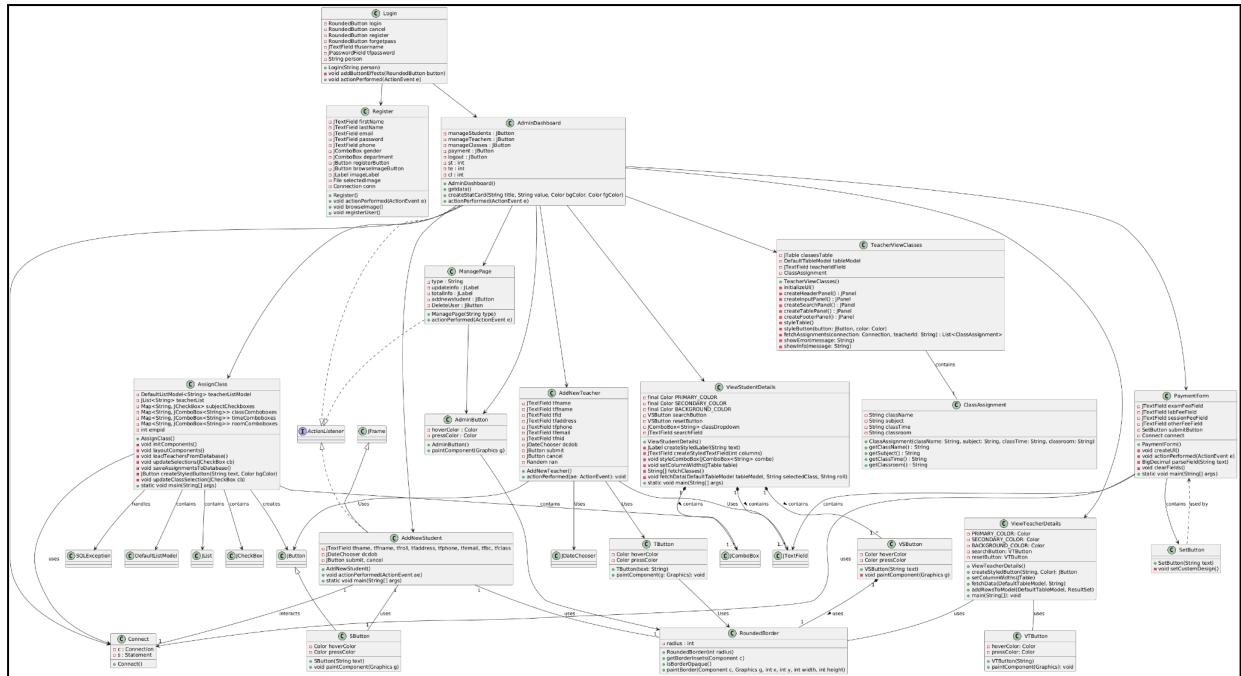


Figure: UML Diagram - Admin

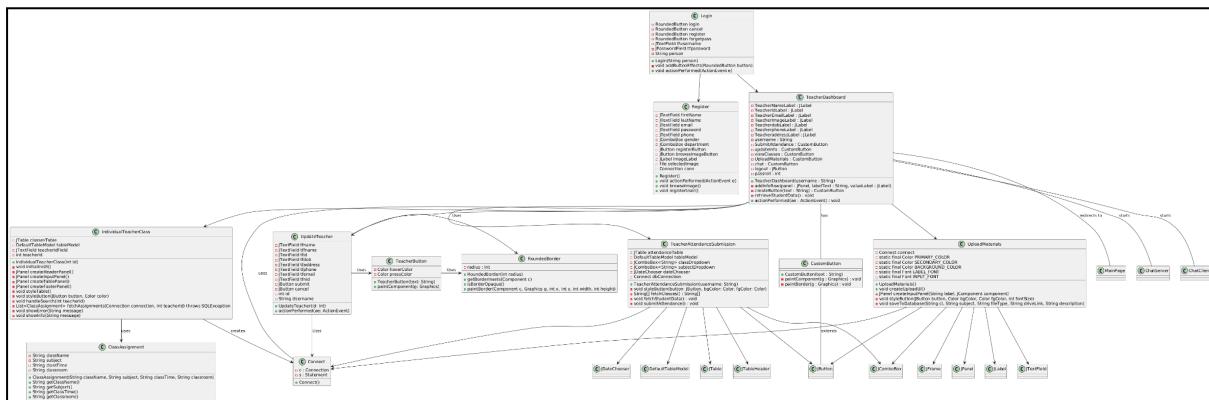


Figure: UML Diagram - Teacher

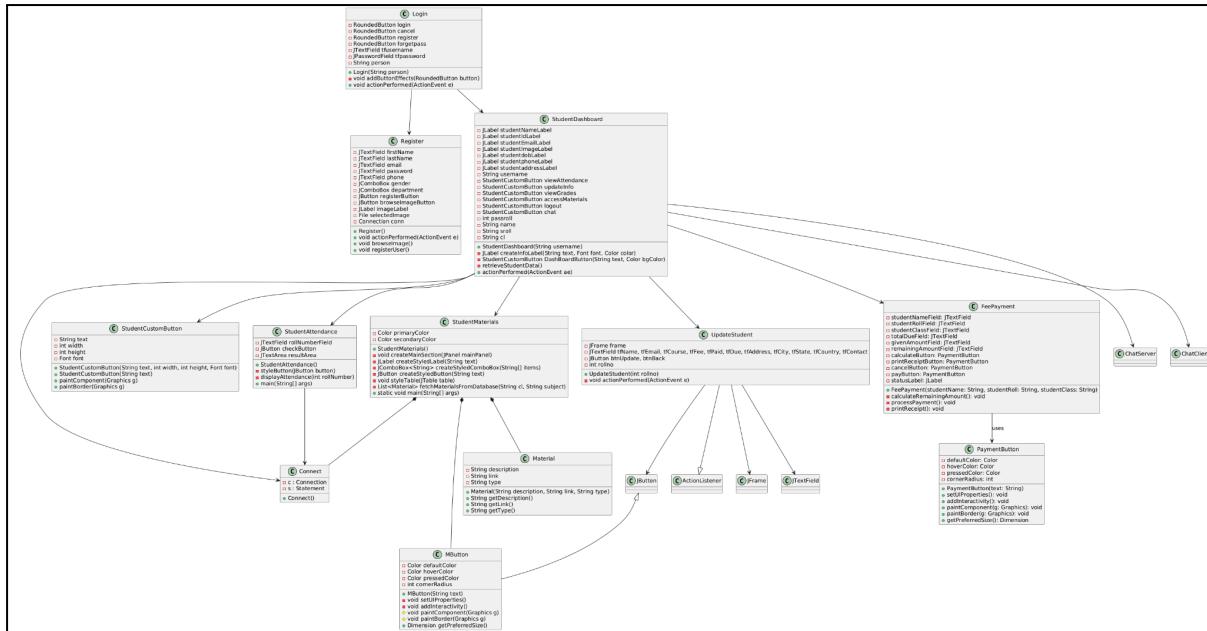


Figure: UML Diagram - Student

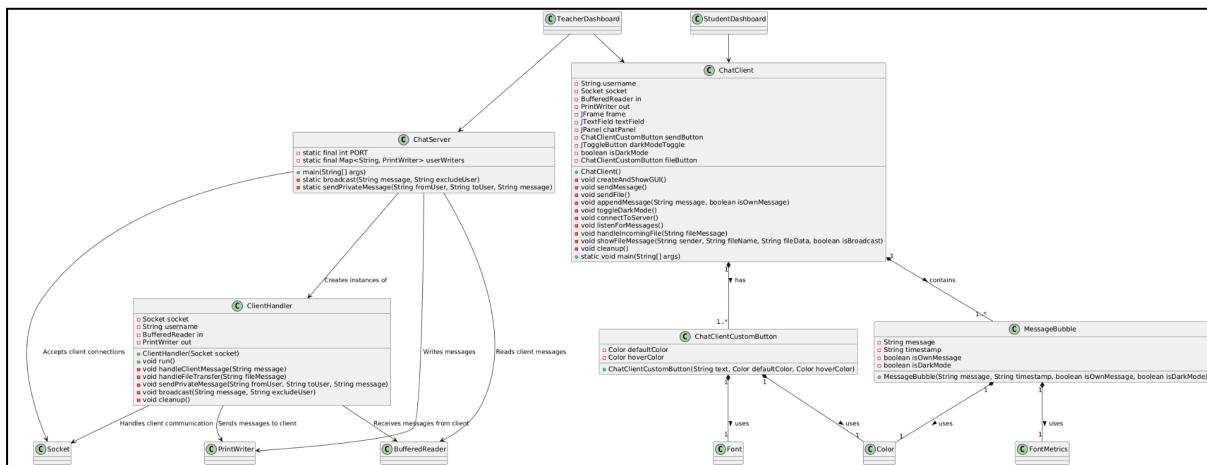


Figure: UML Diagram - Realtime Chatting

Implementation of OOP and Design Principles

- **Encapsulation:**
 - Securely manages **user credentials, attendance, grades, and payment details**.
 - Used private attributes with getter and setter methods.
- **Abstraction:**

- Implemented **Runnable** for threads and **ActionListener** for event handling to simplify logic.
 - **Inheritance:**
 - Extended **JFrame**, **JPanel**, and **JButton** for reusable UI components.
 - **Polymorphism:**
 - Overrode **actionPerformed()** for dynamic event handling.
 - **Open/Closed Principle:**
 - Designed modular components to allow easy addition of new features without modifying existing code.
 - **Thread & Socket Usage:**
 - **Threads** handle asynchronous message reception.
 - **Sockets** enable real-time communication for chat and FAQ automation.
-

Conclusion: Challenges, Discussion, and Future Plan

Challenges:

- Implementing **real-time chat** using sockets.
- Maintaining **data consistency** across users.
- Ensuring **modular yet scalable design**.

Discussion:

- The system successfully integrates **OOP principles, database management, and real-time communication**.
- Automates administrative tasks, improving **efficiency and user experience**.

Future Plan:

- **Multi-Semester Support** – Track long-term student progress.
 - **Online Payment Integration** – Enable secure online transactions.
 - **Mobile Application** – Develop an Android/iOS version for better accessibility.
-

Code Repository

The project's source code is managed using Git and hosted on GitHub. This facilitates **version control, collaboration, and easy deployment**. The repository includes: [GitHub Repository Link](#)

Video Demonstration

For a complete walkthrough of the project, refer to our **video demonstration**: Project Demo Video Link [Video Demo Link](#)

Contribution

1. H.M. Mehedi Hasan(13)

- Login authentication and registration.
- Database handling.
- Payment system, fee setup and receipt generation.
- User management (create, update, delete).
- Student Functionalities.
- Real-time chat system.

2. Md. Abu Bakar Siddique(47)

- Landing page
 - FAQ Chatbot section.
 - Admin functionalities.
 - Teacher functionalities.
 - Frontend(UI/UX, logo)
-

