Table Of Content

LIST OF FIGURES

LIST OF TABLES

Declaration

We, the undersigned students of Master of Computer Applications (MCA), hereby declare that the project titled "GLA's echo (**A Social Media Platform for GLA University)**", submitted to Mr. Shishupal Singh, Institute of Engineering & Technology, GLA University Uttar Pradesh, Mathura, in partial fulfilment of the requirements for the award of the degree of MCA, is an original record of our own work.

We further declare that this project has not been previously submitted by us or by anyone else for the award of any degree, diploma, or similar academic title or recognition.

We also affirm that permission has been obtained wherever required for the use of copyrighted material included in this project, other than brief excerpts permitted with proper academic acknowledgement. All such references have been duly cited in the report.

Place:

Date:

Signatures of Students:

1. Name: _____

   Enrollment No.: _____
2. Name: _____

   Enrollment No.: _____
3. Name: _____

   Enrollment No.: _____
4. Name: _____

   Enrollment No.: _____

Acknowledgement

We would like to express our sincere gratitude to Mr. Shishupal Singh for his valuable guidance, continuous encouragement, and support throughout the development of our project titled “G’echo-GLA’s Echo”. His insights and feedback have greatly contributed to the successful completion of this work.

We also thank the head of the project Dr. Sachendra Singh Chauhan and staff of the Institute of Engineering & Technology, GLA University Uttar Pradesh, Mathura, for providing us with the necessary academic environment, resources, and assistance during the preparation of this project.

We are deeply grateful to our families, friends, and classmates for their moral support, cooperation, and motivation at every stage of this work. Finally, we extend our thanks to everyone who contributed directly or indirectly to the successful completion of this project.

Certificate

This is to certify that the project titled "G'echo-GLA's Echo". submitted by the following students of MCA, Institute of Engineering & Technology, GLA University, Mathura, U.P. is a bonafide record of their original work carried out under my supervision:

1. Rohan Prajapati (2484200166)

2. Amit Giri (2484200022)

3. Rajat Singhal (2484200154)

4. Lalit Kumar (2484200105)

The project has been completed in partial fulfilment of the requirements for the ward of the degree of Master of Computer Applications (MCA).

_____

Project Supervisor Date:

(**Mr. Shishupal Singh**)

_____ _____

Project In-charge Program Coordinator

(**Dr. Sachendra Singh Chauhan**) (**Mr. Siddharth Singh**)

Abstract

This project introduces *G'echo – GLA's Echo*, a campus-exclusive networking platform designed to streamline communication within GLA University. The system provides a controlled environment where only authenticated university members can access and participate, ensuring privacy and reliability. Instead of functioning like open social media, G'echo focuses on structured information sharing, allowing clubs and administrators to publish official announcements and event-related content. Students and faculty members engage through predefined interactions, promoting disciplined participation without unrestricted posting. The platform aims to overcome the inconsistency of scattered communication methods by offering a unified space for academic updates, cultural activities, and campus events. With its simple interface, secure authentication, and role-based permissions, G'echo contributes to a more connected and informed university community. Overall, the project enhances digital communication by providing a trustworthy medium built specifically for institutional needs.

Chapter 1

Introduction

G'echo is a secure and private social media platform exclusively designed for GLA University to connect students, faculty, administrators, and clubs. Unlike open social platforms, it restricts access to verified university members via GLA email authentication, ensuring genuine communications. The platform allows administrators and clubs to post event updates, photos, and blogs, while students and faculties can engage by liking and commenting. This creates a centralized digital ecosystem to promote campus events, enhance engagement, and strengthen community interaction.

G'echo – GLA's Echo is a secure and university-exclusive social media platform designed to enhance communication, collaboration, and engagement within the GLA University community. Unlike public social networks, G'echo provides a closed and authenticated environment where only verified students, faculty members, administrators, and university clubs can interact. The platform aims to centralize all campus-related updates, event announcements, and activity posts, reducing dependency on scattered communication channels. It also promotes a structured interaction system, allowing students and faculties to engage through likes and comments, while clubs and administrators can share posts, photos, and blogs related to academic and non-academic events. With integrated authentication through university email credentials, G'echo ensures identity verification and eliminates the risks of fake accounts. Overall, the project strengthens digital connectivity across the campus by providing a reliable, organized, and interactive medium for exchanging information and fostering community involvement.

1.1 Motivation and Overview

The development of *G'echo – GLA's Echo* is driven by the need for a unified and secure communication medium within GLA University. Currently, information about events, notices, and campus activities is spread across multiple unofficial channels, often leading to confusion, missed updates, and inconsistent student engagement. This project aims to solve these issues by offering a dedicated, university-exclusive digital platform that centralizes all campus communication in one place. G'echo provides a structured environment where administrators and clubs can post official updates, while students and faculty interact through controlled engagement features such as likes and comments. The platform operates through secure university email authentication, ensuring that only verified members can access or participate. Built using Flutter and Firebase, the system delivers a reliable, scalable, and user-friendly experience. By integrating all communication needs into one application, G'echo strengthens campus connectivity, encourages participation, and supports the university's ongoing digital transformation.

1.2 Objectives

The main objectives of the GLA's echo are:

- **To Create a Secure Campus Communication Platform**: Develop a private social networking application accessible only to verified GLA University members using institutional email authentication.
- **To Centralize Event and Announcement Information**: Provide a single digital platform where administrators and clubs can post official updates, reducing dependency on scattered communication channels.

- **To Enhance Student Engagement and Participation**: Encourage students and faculties to stay involved in campus activities through interactive features like liking and commenting.
- **To Establish Role-Based Access and Controlled Interaction**: Assign specific permissions to users—students, faculties, clubs, and administrators—to maintain a structured and safe communication environment.
- **To Support the University's Digital Transformation Efforts**: Introduce a modern digital tool that improves transparency, accessibility, and effectiveness of internal communication across the campus.

The main objective of this project is to develop a secure and university-exclusive social media platform that centralizes all communication within GLA University. It aims to streamline event announcements, official updates, and activity posts by providing a single, reliable digital space. The project also focuses on enhancing student engagement through interactive features while maintaining strict role-based access for safety and authenticity

1.3 Summary of Similar Systems

Several digital platforms exist that support communication and event sharing within academic institutions, but most of them are either general-purpose social media or commercial learning management systems. Popular platforms like Facebook, Instagram, and WhatsApp are often used informally by students and clubs, yet they lack privacy, controlled access, and an official verification mechanism. This results in scattered information, fake accounts, and an unorganized flow of announcements. LMS tools such as Google Classroom, Moodle, and Microsoft Teams offer structured academic communication but are not designed for campus-wide social engagement or event promotion. They primarily focus on coursework, assignments, and classroom activities rather than interactive community participation. Some universities use custom portals for notices, but these systems usually do not support multimedia posts, comments, or interactive engagement features. Compared to these existing solutions, G'echo provides a unique approach by combining the security of institutional login with the social interaction style of modern platforms. It is built specifically for the needs of GLA University, ensuring authenticity, centralized updates, and a more

engaging digital ecosystem. This makes G'echo a more relevant and focused platform than the general or academic tools currently in use.

Many institutions rely on different digital platforms to share information, manage activities, and connect their campus communities. However, most of these platforms are not specifically designed to serve as an exclusive social environment for a single university. Common social media platforms like Facebook, Instagram, and WhatsApp are frequently used by students and clubs to announce events or share updates, but they lack official verification, privacy controls, and structured communication. Anyone can create or join groups, which often leads to misinformation, fake accounts, and scattered updates.

1.4 Report Organization

The remainder of this report is organized as follows.

- Chapter 1 introduces the G'echo platform, its objectives, and motivation.
- Chapter 2 describes the system requirements, feasibility study, modules, and use cases.
- Chapter 3 explains the system design through DFDs, UML diagrams, and the ER diagram.
- Chapter 4 covers the technical implementation, including technologies used and module development.
- Chapter 5 presents the testing methods, test cases, and results.
- Chapter 6 provides the conclusion of the project.
- Chapter 7 summarizes the overall system and its scope.
- Chapter 8 includes the details of meetings and guidance received from the project mentor.

Chapter 2

Software Requirement Analysis

The software requirements for *G'echo – GLA's Echo* are focused on ensuring smooth development, secure authentication, and efficient data management. The application is built using Flutter for cross-platform UI development and Firebase services for backend operations such as authentication, Firestore database, and cloud storage. Android Studio or Visual Studio Code is required as the primary development environment, along with Git for version control.

2.1 Requirement Analysis

Requirement analysis identifies what the system must do and how it should perform. It includes:

2.1.1 Functional Requirement

- **User Authentication**: The system must allow users to log in using their official GLA University email credentials to ensure only verified students, faculties, clubs, and administrators can access the platform.
- **Role-Based Access Control**: The platform must assign different permissions based on user roles. Students and faculties can like and comment on posts, while clubs and administrators can create posts, upload photos, videos, and write blogs.
- **Post Creation and Multimedia Upload**: Clubs and administrators must be able to create posts related to events or announcements and upload images, videos, and descriptions through a user-friendly interface.

- **Interactive Features**: All users should be able to interact with posts through likes and comments, supporting engagement and feedback across the university community.
- **Profile Management**: Clubs must be able to update their profile pictures and display names, while student and faculty details must be automatically fetched from the university database (static identity).
- **Secure Data Storage**: All posts, media files, and user data must be securely stored in Firebase Firestore and Firebase Storage with controlled read/write rules.
- **Real-Time Updates**: Any new post, comment, or update should reflect instantly across the application using Firebase's real-time syncing capabilities.
- **Search and Content Discovery**: Users should be able to view posts and updates from clubs, faculties, and administrators, organized in a clean and accessible feed.

2.1.2 Non-Functional Requirements

1. **Performance**: The application should load posts, images, and updates quickly, ensuring smooth navigation and minimal latency, even during peak usage by multiple university members.
2. **Security**: All user data, media files, and login credentials must remain protected using Firebase Authentication, secure database rules, and encrypted communication channels to prevent unauthorized access.
3. **Usability**: The interface must be simple, intuitive, and easy to use for students, faculties, clubs, and administrators, requiring minimal training or technical knowledge.
4. **Scalability**: The system should support future growth, including an increasing number of users, posts, and events without affecting performance, leveraging Firebase's scalable cloud infrastructure.

5. **Reliability and Availability**: The platform must function consistently with minimal downtime, ensuring that students and faculty can always access event updates and university announcements.
6. **Maintainability**: The system should be easy to update, debug, and enhance, with clean code practices and modular architecture to support future improvements.
7. **Compatibility**: The application should perform well on Android devices of different versions and screen sizes, ensuring accessibility across the university.

2.2 Feasibility Analysis

The project is feasible as it can be developed efficiently using Flutter and Firebase, requiring minimal cost, manageable effort, and resources easily available to the development team.

2.2.1 Technical Feasibility

The project is technically feasible as it uses widely supported technologies such as Flutter for cross-platform mobile development and Firebase for backend services, including authentication, database management, and cloud storage. These tools are easy to integrate, well-documented, and suitable for building a secure and scalable university communication platform. The development environment—Android Studio or Visual Studio Code—runs smoothly on standard hardware, and the required libraries for multimedia handling, networking, and UI design are readily available. Overall, the technical requirements are fully achievable with the skills and resources accessible to the project team.

2.2.2 Economy Feasibility

The project is economically feasible because it relies on cost-effective tools such as Flutter and Firebase, which offer free or low-cost usage for development and testing. No additional paid servers, software licenses, or high-cost resources are required. The development can be completed using

existing university systems and student-owned devices, making the overall financial investment minimal and manageable.

2.2.3 Operational Feasibility

The system is operationally feasible as it aligns with the daily communication needs of students, faculties, clubs, and administrators. The platform is easy to use, requiring minimal training, and integrates smoothly into existing university workflows. Its secure authentication, user-friendly interface, and real-time updates ensure that all users can adopt and operate the system effectively without disrupting current processes.

2.2.4 Time Feasibility

The project is time feasible because its development phases—planning, design, coding, testing, and deployment—fit well within the allotted academic schedule. Using Flutter and Firebase accelerates development due to pre-built tools, reducing implementation time.

2.3 Module Description

The project consists of several integrated modules that work together to ensure smooth communication within GLA University. The Authentication Module verifies users through official university email IDs, while the User Role Management Module assigns permissions to students, faculties, clubs, and administrators. A Post Creation and Media Upload Module enable authorized users to share event posts, photos, and videos. The Interaction Module supports likes and comments to enhance engagement, and the Feed Management Module displays all updates in a real-time, organized format. Together, these modules create a secure, structured, and interactive digital platform for campus-wide communication.

2.3.1 Student Module

The Student Module is the primary interface for students to access campus updates and participate in university activities on the G'echo platform. It provides the following functionalities:

1. **Secure Login:** Students log in using their official GLA University email credentials. The system verifies the identity of each student and grants access only to authenticated users.
2. **Dashboard:** After login, students can view a personalized dashboard that displays the latest posts, events, announcements, and updates shared by clubs and administrators. The dashboard provides a clean and organized feed for easy navigation.
3. **View Posts & Events:** Students can open posts to view event details, photos, and videos uploaded by clubs and administrators. This ensures that students stay updated with all campus activities in one centralized platform.
4. **Interaction (Like & Comment):** Students can engage with posts by liking and commenting, allowing them to participate in discussions and share feedback on events or announcements while maintaining controlled interaction rights.
5. **Notifications:** Students receive notifications for new posts, important updates, or event announcements. This ensures they never miss important campus information.
6. **Profile View:** Students can view their basic profile information, such as name, enrollment details, and email — all fetched from the university database to prevent impersonation or fake identities.

The goal of this module is to make campus communication simple, authentic, and accessible. It allows students to stay informed, interact responsibly, and participate actively in the university's digital ecosystem.

2.3.2 Faculty Module

The Faculty Module provides a simplified interface for faculty members to stay informed about campus activities and interact with official posts. It restricts content creation but allows meaningful participation across the platform. It offers the following functionalities:

1. **Secure Login:** Faculty members log in using their official GLA University email credentials. The system verifies their identity and grants access according to their assigned role.
2. **Dashboard:** After logging in, faculty can view their dashboard displaying a real-time feed of university announcements, club posts, event updates, and activity highlights.
3. **View Posts & Events:** Faculty members can open posts to view complete details, including event descriptions, photos, videos, and official updates shared by administrators or clubs.
4. **Interaction (Like & Comment):** Faculty can interact with posts by liking and commenting, allowing them to share feedback, appreciation, or guidance related to events and activities.
5. **Notifications:** Faculty receive notifications for major university announcements, important event posts, and updates relevant to academic or non-academic activities.

The goal of this module is to keep faculty informed, connected, and engaged with the campus community while maintaining secure and well-defined interaction boundaries.

2.3.3 Club Module

The Club Module is designed for university clubs and their mentors to share event updates, manage posts, and interact with the campus digitally. It gives them the highest posting privileges after administrators. It includes the following functionalities:

1. **Secure Login:** Club heads and mentors access the platform using official university emails, ensuring that only authorized club representatives can manage posts.
2. **Dashboard:** The dashboard displays event-related posts created by the club, analytics such as likes and comments, and recent campus activities shared by other clubs and administrators.

3. **Create Posts:** Clubs can create event announcements by uploading photos, videos, and detailed descriptions. They can also update their posts when required.
4. **Media Upload:** This feature allows clubs to upload multimedia files directly from their device, enabling richer and more engaging event promotion.
5. **Profile Management:** Clubs can update their display name and profile photo to reflect their identity, upcoming activities, or branding changes.
6. **Interaction Monitoring:** Clubs can view all likes, comments, and engagement on their posts, helping them understand student interest and event reach.

The goal of this module is to simplify event promotion, improve visibility of club activities, and create a centralized space for campus engagement.

2.3.4 Administrator Module

The Administrator Module provides full control over platform-wide communication. Administrators manage official GLA updates, oversee content shared by clubs, and ensure smooth operation of the digital ecosystem. The module offers these functionalities:

1. **Secure Login:** Administrators, including VC, Directors, HODs, and other officials, log in using their institutional credentials for high-level secured access.
2. **Dashboard:**
   Administrators view a comprehensive feed of all posts, events, announcements, and campus updates, along with engagement metrics across the platform.
3. **Create Official Posts:** Administrators can publish important notices, event invitations, academic updates, and policy-related information along with photos, videos, or documents.
4. **Content Oversight:** Administrators can review posts created by clubs to ensure accuracy, quality, and compliance with university guidelines.

5. **Post Management:** They can edit, update, or remove posts if required, ensuring the platform remains organized, credible, and relevant.
6. **System Monitoring:** Administrators oversee the smooth functioning of the application, track user engagement, and ensure that communication remains consistent and structured.

The goal of this module is to maintain authenticity, regulate communication, and provide a reliable official medium for campus-wide updates.

2.4 Use Cases

The overall use case diagram of G'echo illustrates how different user roles—Admin, Administrator, Student, Faculty, and Club—interact with the system through a variety of shared and role-specific functionalities. All users can perform essential actions such as logging in, viewing posts, liking posts, commenting on posts, receiving notifications, deleting content they own, and editing their profiles, which represents the core interaction flow of the platform. Additional privileges are granted to Admin and Administrator roles, who can review and approve users, manage content across the system, and maintain security to ensure smooth and controlled platform operations. Club users and administrators have the ability to create posts, allowing them to share events, announcements, and updates with the university community. Overall, the use case diagram visually presents the role-based access structure of G'echo, showing how each actor interacts with the system to support communication and engagement across the campus.

2.4.1 Student Use Cases

Use Case 1: Login

- **Actor:** Student
- **Description:** The student logs into the G'echo platform using their official GLA University email credentials. The system verifies the identity before granting access.
- **Outcome:** Student is authenticated and redirected to their dashboard.

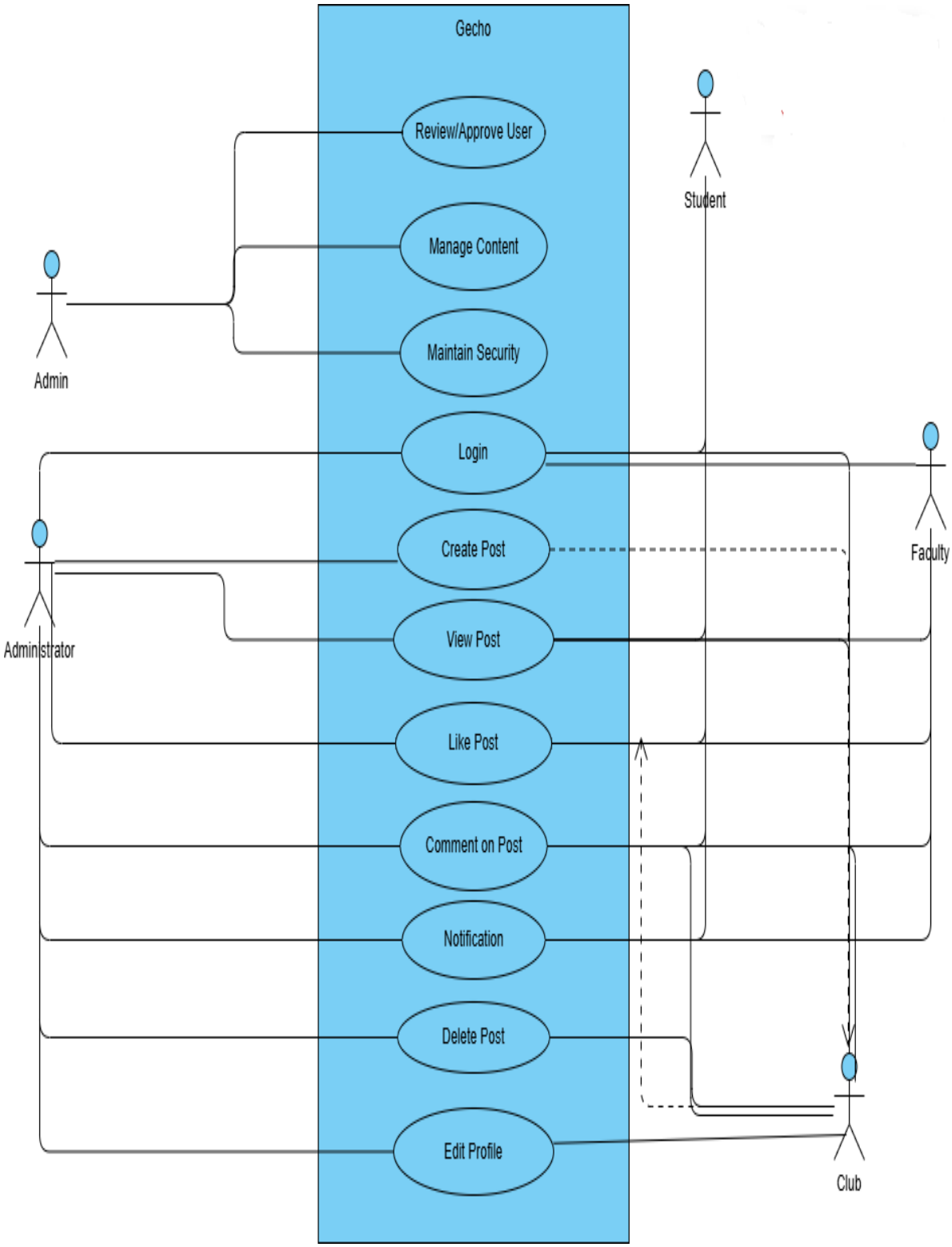

Figure 2.4 Use Case Diagram of all actors

Use Case 2: View Posts & Events

- **Actor:** Student
- **Description:** The student views event posts, announcements, photos, and videos uploaded by clubs and administrators.
- **Outcome:** Student is updated with the latest campus information in real time.

Use Case 3: Like a Post

- **Actor:** Student
- **Description:** The student likes a post to engage with events or announcements shared on the platform.
- **Outcome:** The like is recorded, increasing engagement and visibility of the post.

Use Case 4: Comment on a Post

- **Actor:** Student
- **Description:** The student adds a comment to a post to share feedback or participate in discussions related to events.
- **Outcome:** The comment is saved and displayed, contributing to student interaction.

Use Case 5: Receive Notifications

- **Actor:** Student
- **Description:** The student receives notifications about new posts, updates, or important university announcements.
- **Outcome:** Student stays informed and does not miss important campus activities.

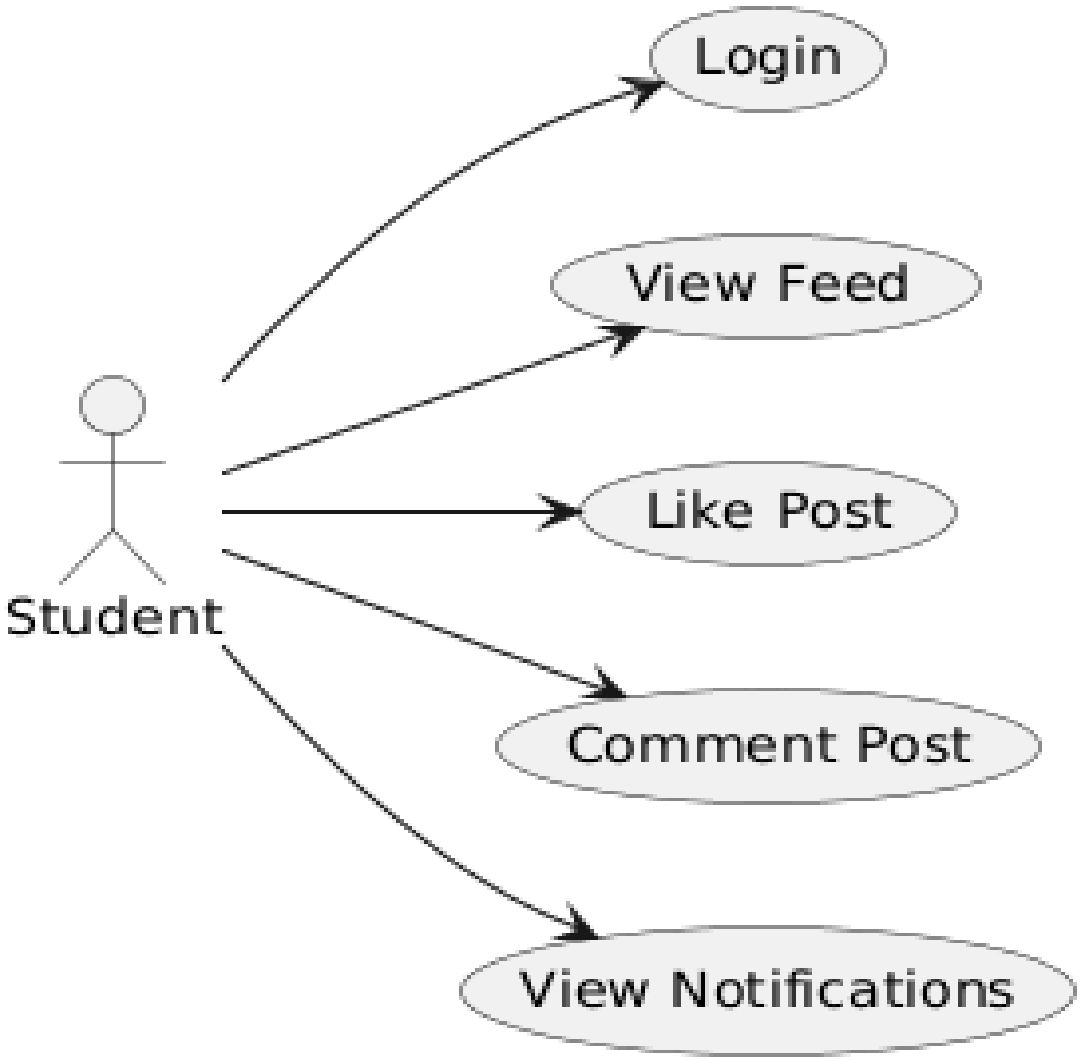

Figure 2.4.1 Use Case Diagram of student

2.4.2 Faculty Use Cases

Use Case 1: Login

- **Actor:** Faculty
- **Description:** The faculty member logs into the G'echo platform using their official GLA University email credentials. The system verifies their identity and grants role-based access.
- **Outcome:** Faculty is authenticated and redirected to their dashboard.

Use Case 2: View Posts & Events

- **Actor:** Faculty
- **Description:** The faculty member views event posts, announcements, and updates uploaded by clubs and administrators in the main feed.
- **Outcome:** Faculty stays updated with all campus activities and official information.

Use Case 3: Like a Post

- **Actor:** Faculty
- **Description:** The faculty member likes a post to show support or appreciation for events, activities, or announcements.
- **Outcome:** The like is recorded and reflected in the post's engagement count.

Use Case 4: Comment on a Post

- **Actor:** Faculty
- **Description:** The faculty member comments on posts to provide feedback, share thoughts, or guide students and clubs.
- **Outcome:** The comment is saved and displayed under the post, enhancing interaction.

Use Case 5: Receive Notifications

- **Actor:** Faculty
- **Description:** The faculty member receives notifications about new posts, important announcements, or event updates.
- **Outcome:** Faculty is instantly informed and does not miss any relevant information.

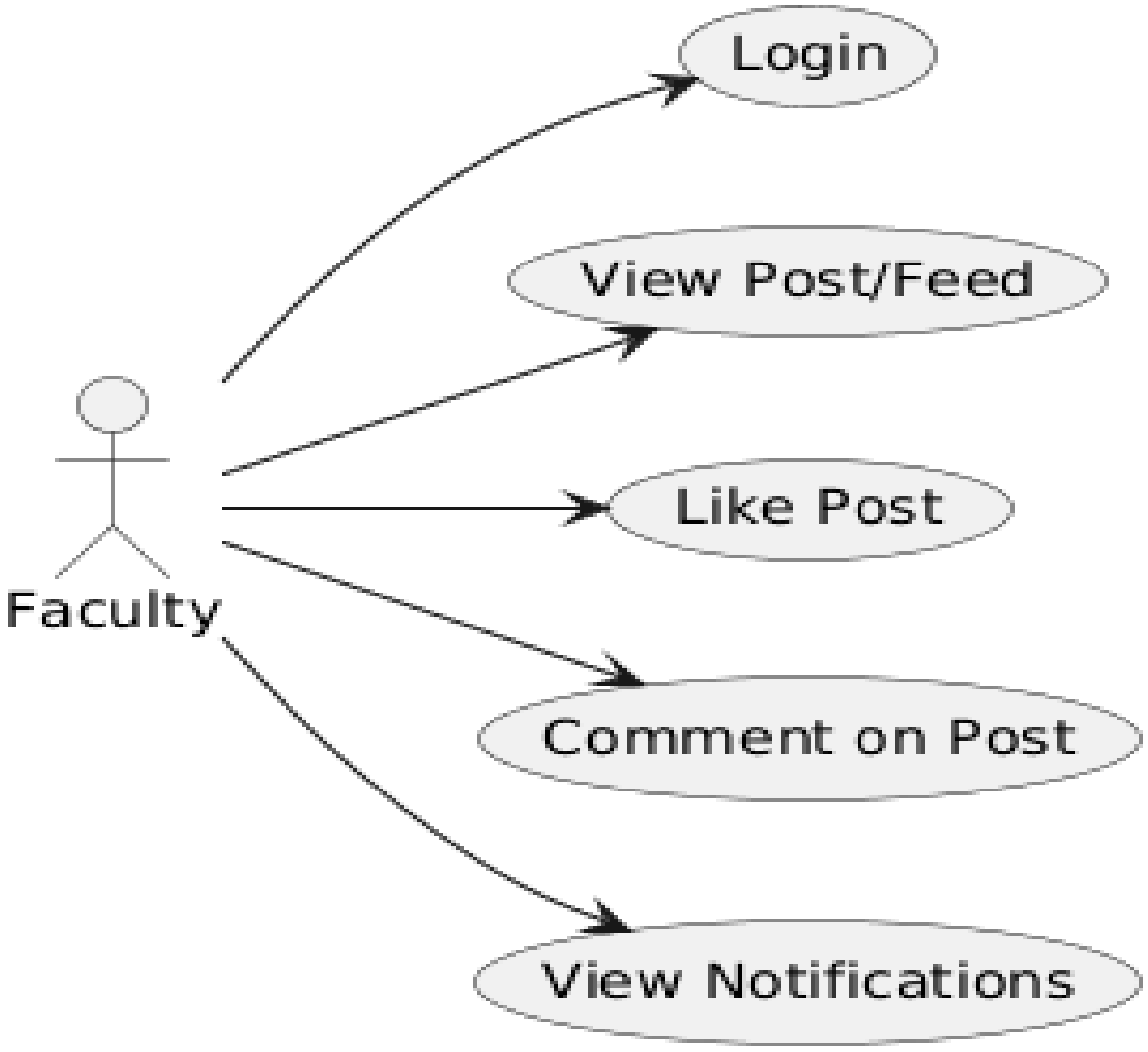

Figure 2.4.2 Use Case Diagram of Faculty

2.4.3 Administrators Use Cases

Use Case 1: Login

- **Actor:** Administrator (VC, Director, Pro-VC, HODs, Registrar)
- **Description:** The administrator logs into the G'echo platform using their official GLA University email credentials. The system verifies their identity and provides full access based on their role.
- **Outcome:** Administrator is authenticated and redirected to the dashboard.

Use Case 2: Create Post

- **Actor: Administrator**
- **Description:** The administrator creates an official post by adding text, photos, or videos about events, notices, or university-level announcements.
- **Outcome:** The post is successfully published and becomes visible to all users on the platform.

Use Case 3: Comment on Post

- **Actor: Administrator**
- **Description:** The administrator comments on posts to provide instructions, feedback, or clarification regarding events and announcements.
- **Outcome:** The comment is recorded and displayed under the post, helping users receive guidance or additional information.

Use Case 4: Edit Profile

- **Actor: Administrator**
- **Description:** The administrator updates their profile details, such as display name or profile picture, to maintain accurate identity and representation on the platform.
- **Outcome:** The updated profile information is saved and reflected across the system.

Use Case 5: Notification

- **Actor: Administrator**
- **Description:** The administrator receives notifications about new posts, comments, or important updates taking place on the platform.
- **Outcome:** Administrator stays informed and does not miss any critical activity or interaction.

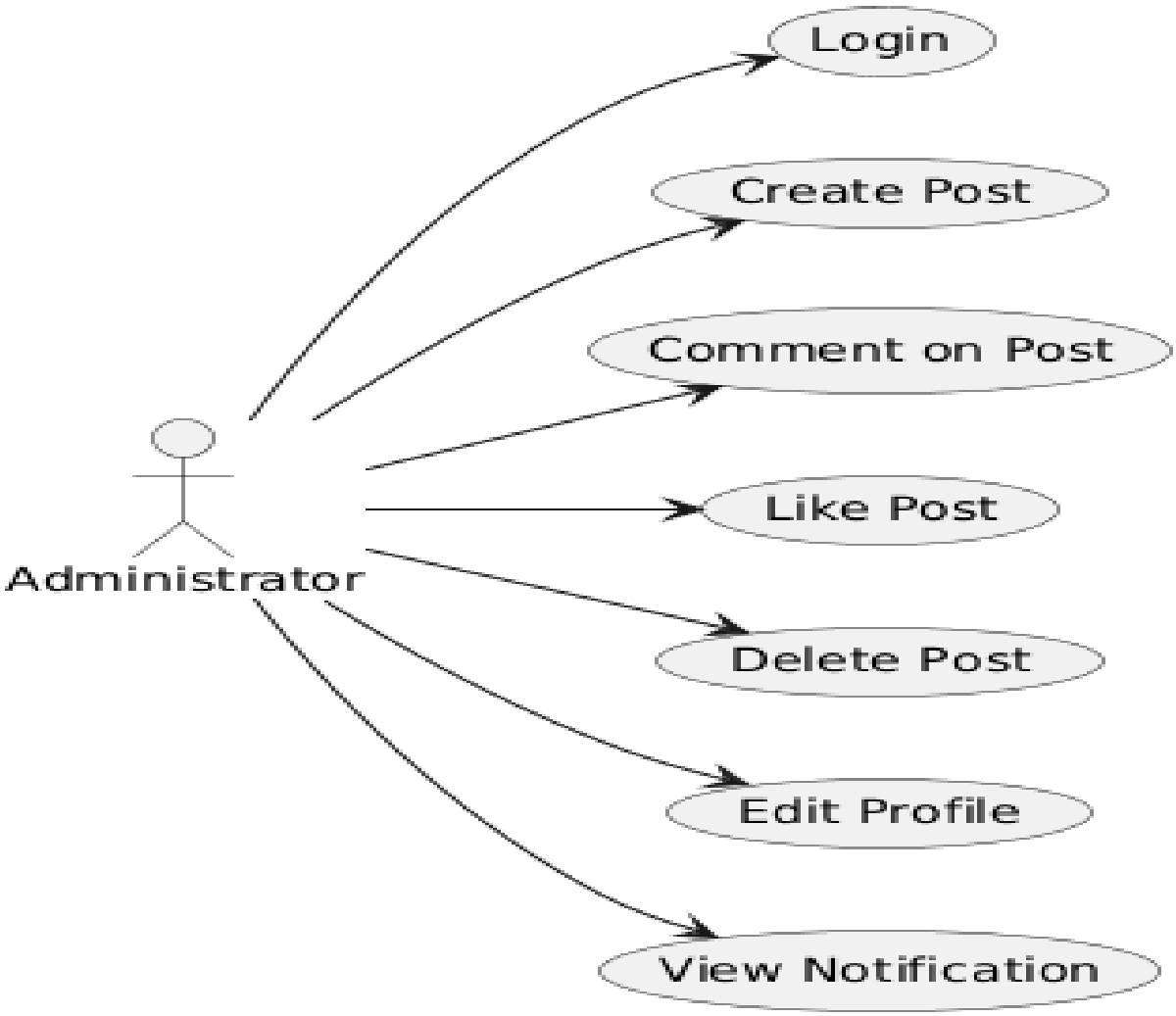

Figure 2.4.3 Use Case Diagram of Administrators

2.4.4 Club/ Club Mentor Use Cases

Use Case 1: Login

- **Actor: Club / Club Mentor**
- **Description:** The club representative logs into the G'echo platform using their official GLA University email credentials. The system verifies their identity and grants content-creation permissions.
- **Outcome:** Club user is authenticated and redirected to their dashboard.

Use Case 2: Create Post

- **Actor: Club / Club Mentor**
- **Description:** The club creates a new event post by adding descriptions, photos, or videos related to upcoming or ongoing activities.
- **Outcome:** The post is successfully published and becomes visible to students, faculty, and administrators.

Use Case 3: Comment on Post

- **Actor: Club / Club Mentor**
- **Description**: The club mentor comments on posts to provide updates, answer queries, or interact with students and faculty.
- **Outcome:** The comment is saved and displayed under the post, adding clarity and engagement.

Use Case 4: Edit Profile

- **Actor: Club / Club Mentor**
- **Description:** The club updates its profile picture or display name to maintain its identity and reflect ongoing activities or branding.
- **Outcome**: The profile is successfully updated and shown across the platform.

Use Case 5: Notification

- **Actor: Club / Club Mentor**
- **Description:** The club receives notifications about comments, interactions, or platform updates related to their posts or events.
- **Outcome:** Club stays updated and responds promptly to interactions or required actions

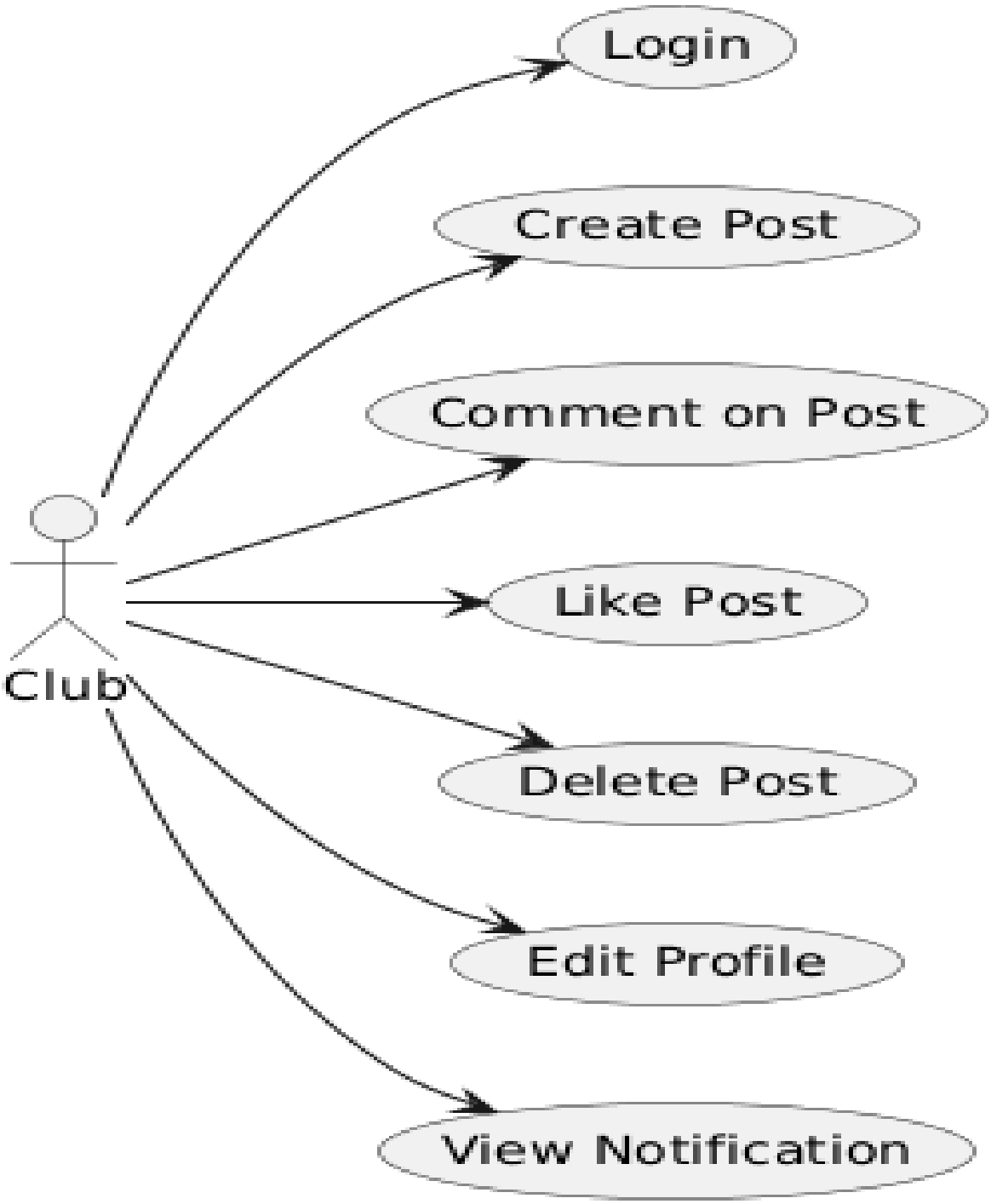

Figure 2.4.2 Use Case Diagram of Club/ Club Mentor

2.5 System Architecture

The system architecture of G'echo is designed using a client–server model that ensures secure data flow, real-time communication, and smooth interaction among all user roles—students, faculty, clubs, and administrators. The mobile application, built using Flutter, functions as the **frontend layer**, providing an intuitive interface where users can log in, view posts, interact, and create content based on their role privileges. All user authentication requests, post data, comments, notifications, and media uploads are handled through **Firebase Authentication**, **Firestore Database**, and **Firebase Storage**, which collectively form the **backend layer**.

When a user logs in with their official GLA University email, the request is processed by Firebase Auth, which validates identity and assigns the correct role. Posts, images, comments, and notifications are stored in Firestore and updated in real time, allowing every change to instantly reflect on all connected devices. Media files such as event photos and videos are securely stored in Firebase Storage and retrieved through optimized access links. This architecture ensures scalability, low latency, and safe communication across the platform. Overall, the system integrates role-based access control, secure cloud storage, and real-time data sync to deliver a unified, reliable, and interactive social platform for GLA University.

Chapter 3

System Design

The system design of G'echo follows a modular, role-based architecture that ensures smooth communication and interaction among students, faculty, clubs, and administrators. The frontend is designed using Flutter to provide a clean, responsive, and user-friendly interface, while the backend is powered by Firebase to support secure authentication, real-time database operations, and efficient media storage. Each module—such as Student, Faculty, Club, and Administrative—has dedicated access controls defined through Firebase Authentication, ensuring that users only perform actions permitted by their roles.

3.1 Data Flow Diagrams

Data Flow Diagrams (DFDs) illustrate how data moves through the system. We provide diagrams at multiple levels of detail.

3.1.1 Level 0 (Context Diagram):

The Level 0 Data Flow Diagram of G'echo represents the system as a single centralized process that interacts with four primary external entities—Students, Faculty, Clubs, and Administrators. Each user sends requests such as login credentials, post interactions, event details, and profile updates to the main system process. The system then communicates with Firebase services, where authentication verifies user identity and Firestore stores or retrieves data like posts, comments, likes, and notifications. The processed information is then delivered back to users in real time through the mobile application interface.

This top-level view shows view shows how data flows smoothly between users and the G'echo system without exposing internal components.

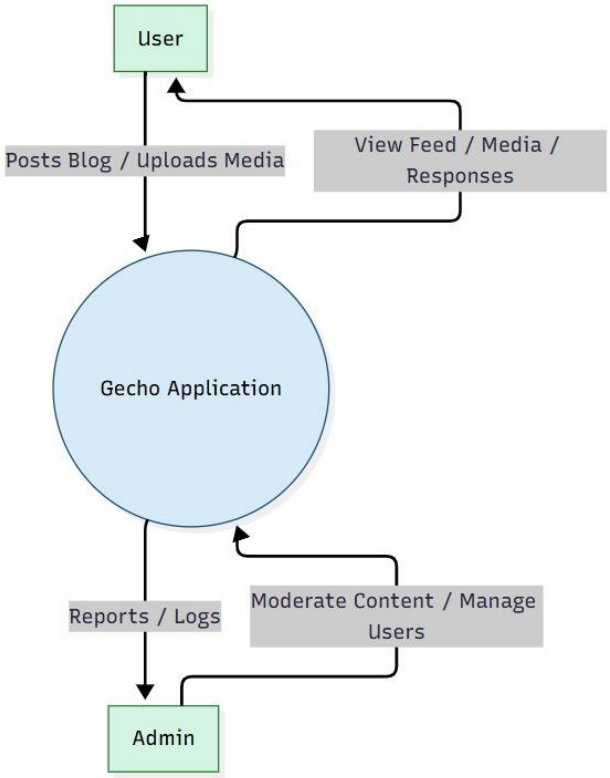

Figure 3.1.1 Level 0 Data Flow Diagram

3.1.2 Level 1 DFD:

The Level 1 Data Flow Diagram of G'echo breaks the main system process into multiple detailed sub-processes such as User Authentication, Post Management, Interaction Handling, Profile Management, and Notification Service. When a Student, Faculty member, Club, or Administrator logs in, the Authentication process verifies their credentials using Firebase Auth and assigns the correct role. The Post Management process handles creating, viewing, deleting, and updating posts, while Interaction Handling manages likes and comments exchanged between users. Profile Management allows users to update their personal details, and the Notification Service retrieves and pushes real-time alerts from Firestore. These processes work together by exchanging data with Firebase databases and returning processed information back to the users, ensuring smooth and structured communication within the system.

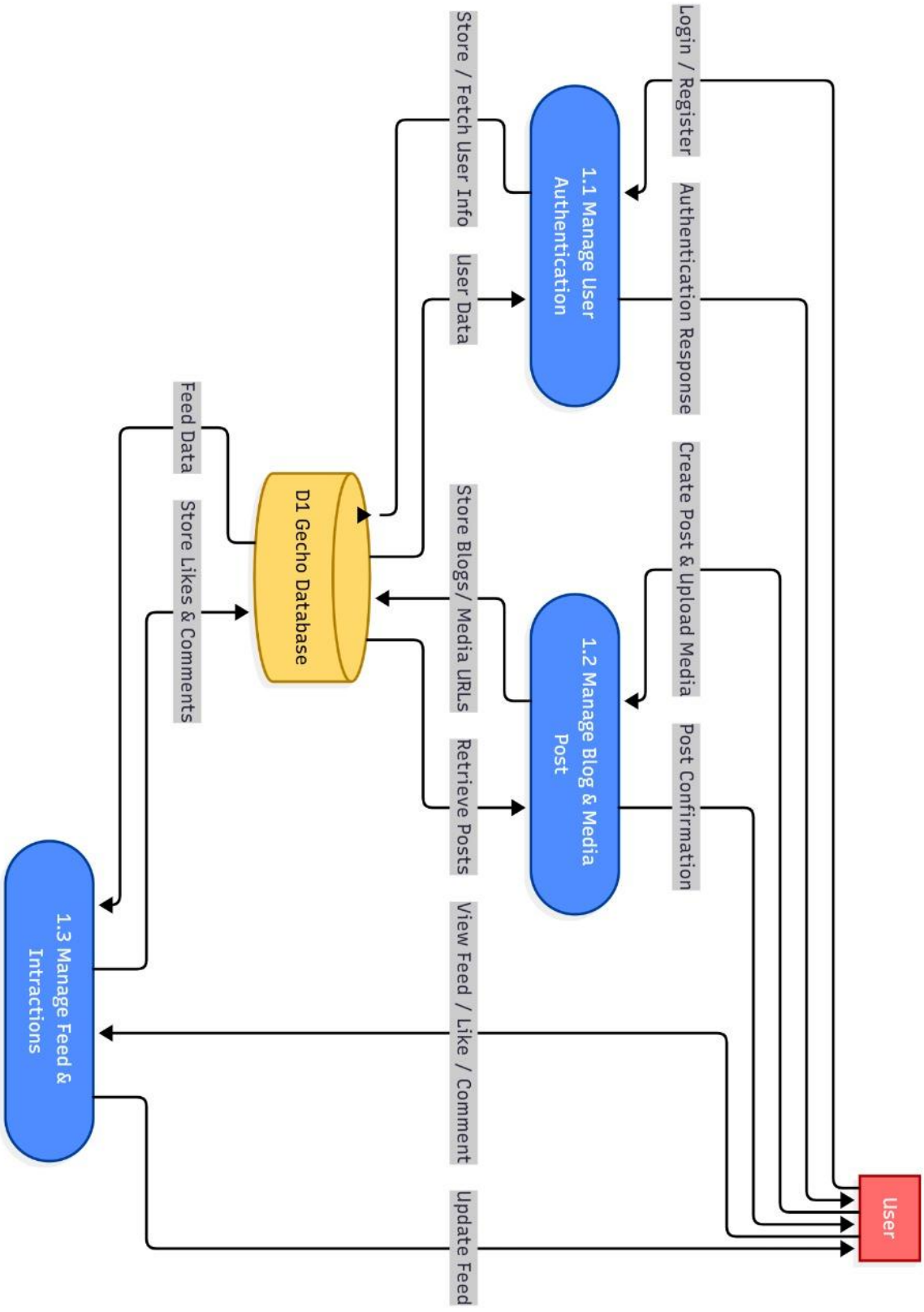

Figure 3.1.2 Level 1 Data Flow Diagram

3.1.3 Level 2 DFD:

The Level 2 Data Flow Diagram of G'echo provides a deeper breakdown of each major sub-process into more detailed operations. In the Authentication process, user credentials are validated, roles are assigned, and secure session tokens are generated. The Post Management process is further divided into creating posts, uploading media to Firebase Storage, retrieving event posts from Firestore, and deleting or updating existing content. Interaction Handling is split into recording likes, saving comments, and fetching engagement data in real time. The Profile Management process covers updating profile photos, editing display names, and retrieving user identity details from the database. Finally, the Notification Service includes generating alerts, storing notification logs, and delivering push notifications instantly to users. These detailed processes collectively ensure seamless data flow, real-time updates, and secure communication throughout the G'echo platform.

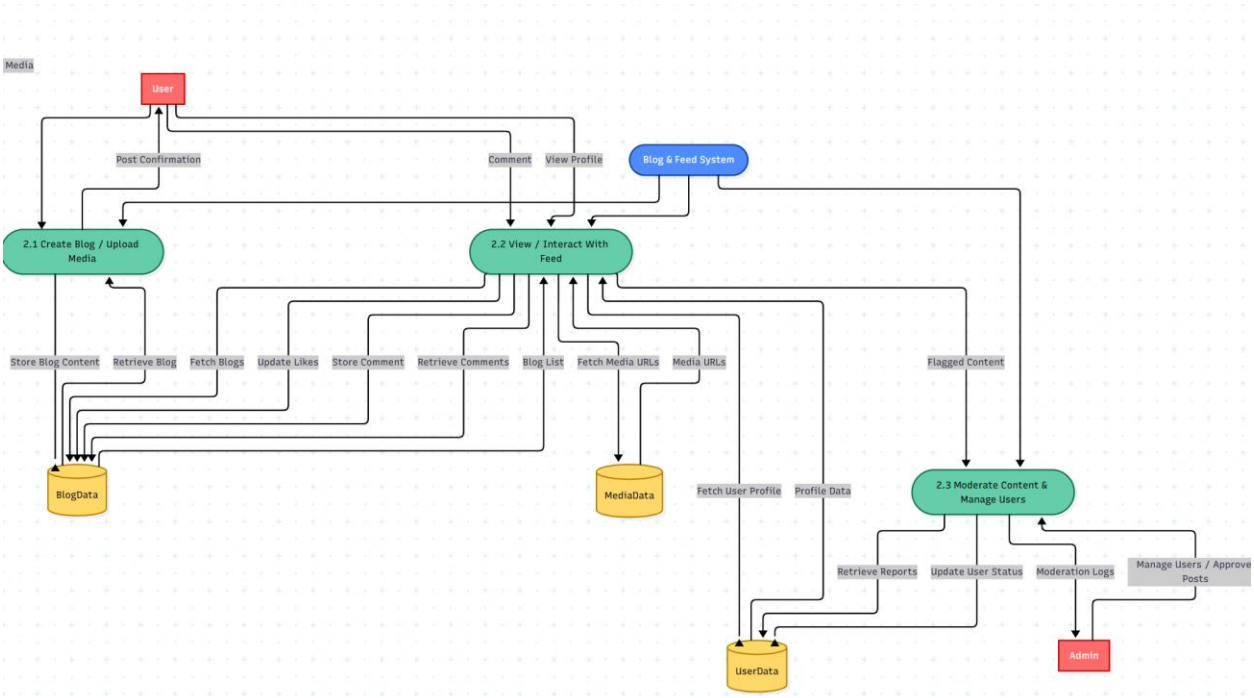

Figure 3.1.3 Level 2 Data Flow Diagram

3.2 UML Diagrams:

We use UML (Unified Modelling Language) diagrams to model the system's structure and behaviour:

3.2.1 Class Diagram

The class diagram of G'echo represents the core structure of the system by defining the main classes, their attributes, and the relationships between them. The key classes include **User**, **Post**, **Comment**, **Notification**, and **Media**, each representing essential components of the platform. The *User* class is further specialized into subclasses like *Student*, *Faculty*, *Club*, and *Administrator*, each inheriting common properties such as userID, name, email, and role, while containing unique behaviors based on their permissions. The *Post* class contains attributes like postID, content, timestamp, and media references, and is linked to users through associations. Comments and likes are managed through the *Comment* and *Interaction* classes, which connect users to posts. The class diagram helps visualize the system's object-oriented structure, showing how different components

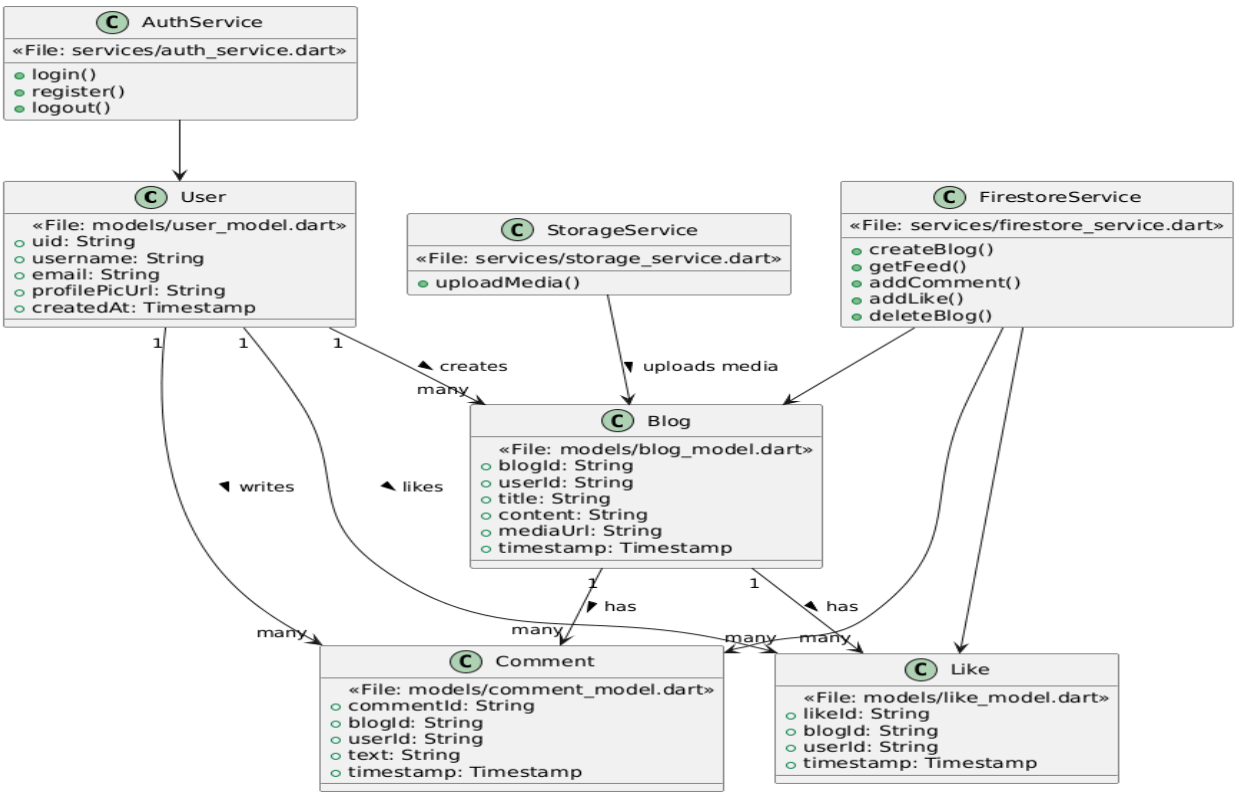

Figure 3.2.1 Class Diagram

3.2.2 Sequence Diagram

The sequence diagram illustrates the complete workflow of how a user interacts with the system while creating and viewing a post on the G'echo application. It shows the communication between four main components: User, Frontend Application, Firebase Backend, and Firestore Database. The flow is as follows:

1. **User Opens the Home Page:** The interaction begins when the *User* (Student/Faculty/Club/Admin) launches the application and navigates to the home feed. The Frontend (Mobile App) receives this request.
2. **Frontend Requests Post Data:** The Frontend sends a request to the Firebase Backend to fetch all recent posts, images, and event details stored in the Firestore Database.
3. **Backend Fetches Posts from Database:** Firebase Backend queries the **Firestore Database** to retrieve the list of posts, along with author details, timestamps, likes, and comments. The database returns the complete dataset to Firebase.
4. **Backend Sends Post Data to Frontend:** The Firebase Backend forwards the post data to the Frontend, which then renders the posts, media, and interactions for the user to view.
5. **User Creates a New Post:** If the user is a Club or Administrator, they can create a new post. The Frontend sends this data (text + image/video) to Firebase using a POST-like request.
6. **Backend Stores Post and Media:** Firebase stores the media file in **Firebase Storage**, saves the post details in **Firestore**, and generates real-time updates for all connected users.
7. **Database Sends Real-Time Update to Frontend:** Firestore pushes the updated post list back to the Frontend. The new post appears instantly on every user's home feed through Firebase's real-time listener.

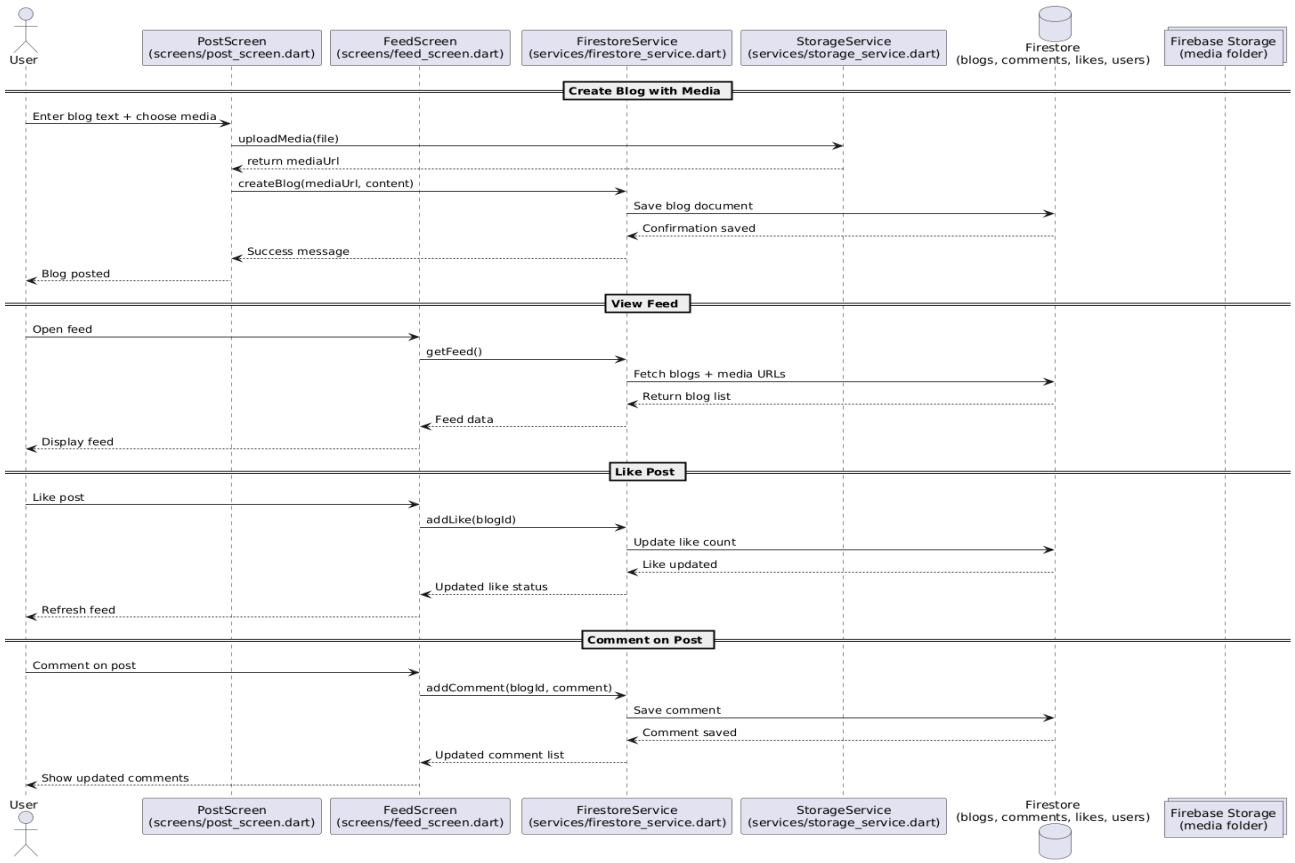

Figure 3.2.2 Sequence Diagram

3.2.3 Object Diagram

An Object Diagram is a snapshot of system instances at a particular point in time. For instance, an object diagram may show:

- The object diagram shows a **UserObj instance** with attributes such as *uid = "user123"*, *username = "Tech Club"*, and *email = "techclub@gla.ac.in"*.
- The **UserObj creates a BlogObj**, representing a specific post with values like *blogId = "post001"*, *title*, *content*, and *mediaUrl*.
- The **BlogObj is associated with a Comment1 object**, which represents an actual comment on the post with fields like *commentId = "c001"* and *text = "Great post!"*.
- The **BlogObj is also linked to a Like1 object**, showing a like interaction with attributes such as *likeId = "l111"* and *userId = "user789"*.

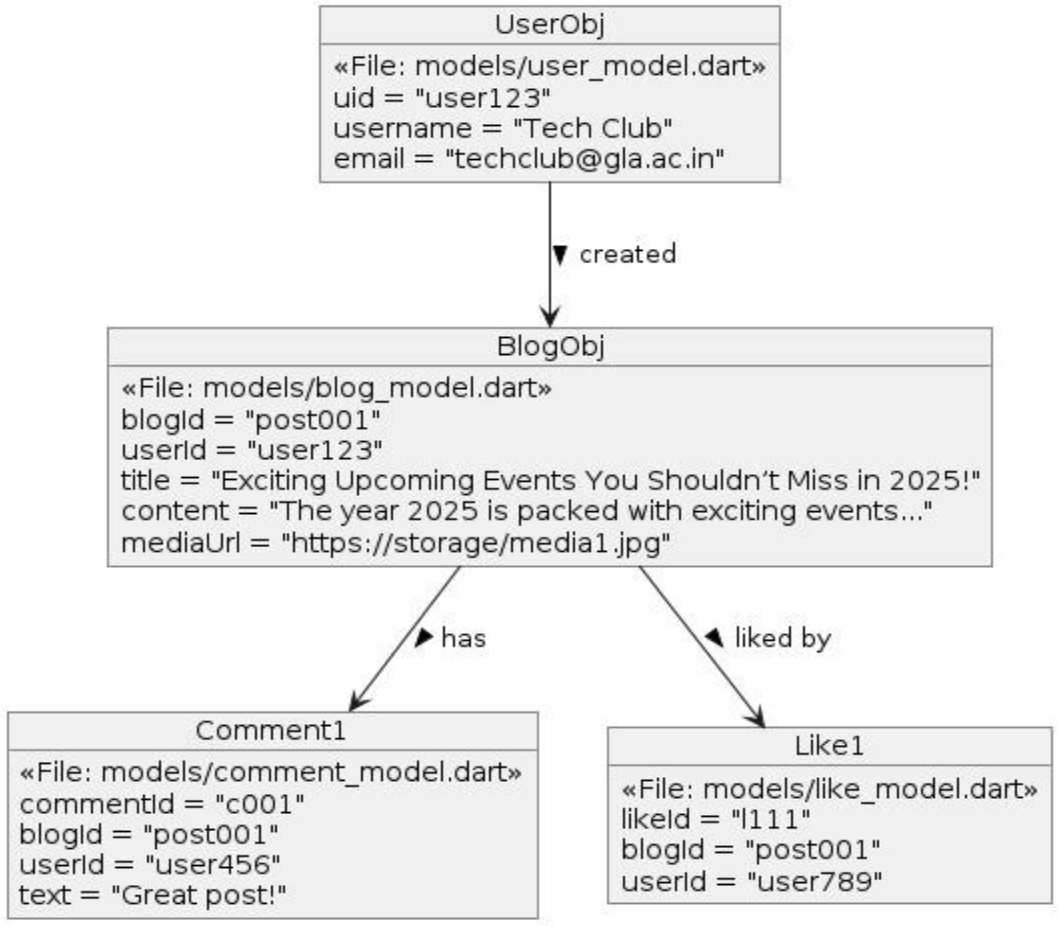

Figure 3.2.3 Object Diagram

3.2.4 Collaboration Diagram

A Collaboration Diagram (also called a Communication Diagram) emphasizes object relationships and the order of interactions.

- The collaboration diagram shows how the **UI**, **User**, **FirestoreService**, **StorageService**, and **FirestoreDB** interact when a user creates a blog post.
- After login, the **User calls createBlog()**, and FirestoreService coordinates the workflow by uploading media to StorageService and saving the file in FirestoreDB.
- Once the media and blog data are successfully stored, the services return success responses, and the **UI displays a notifySuccess() message**, completing the blog creation process.

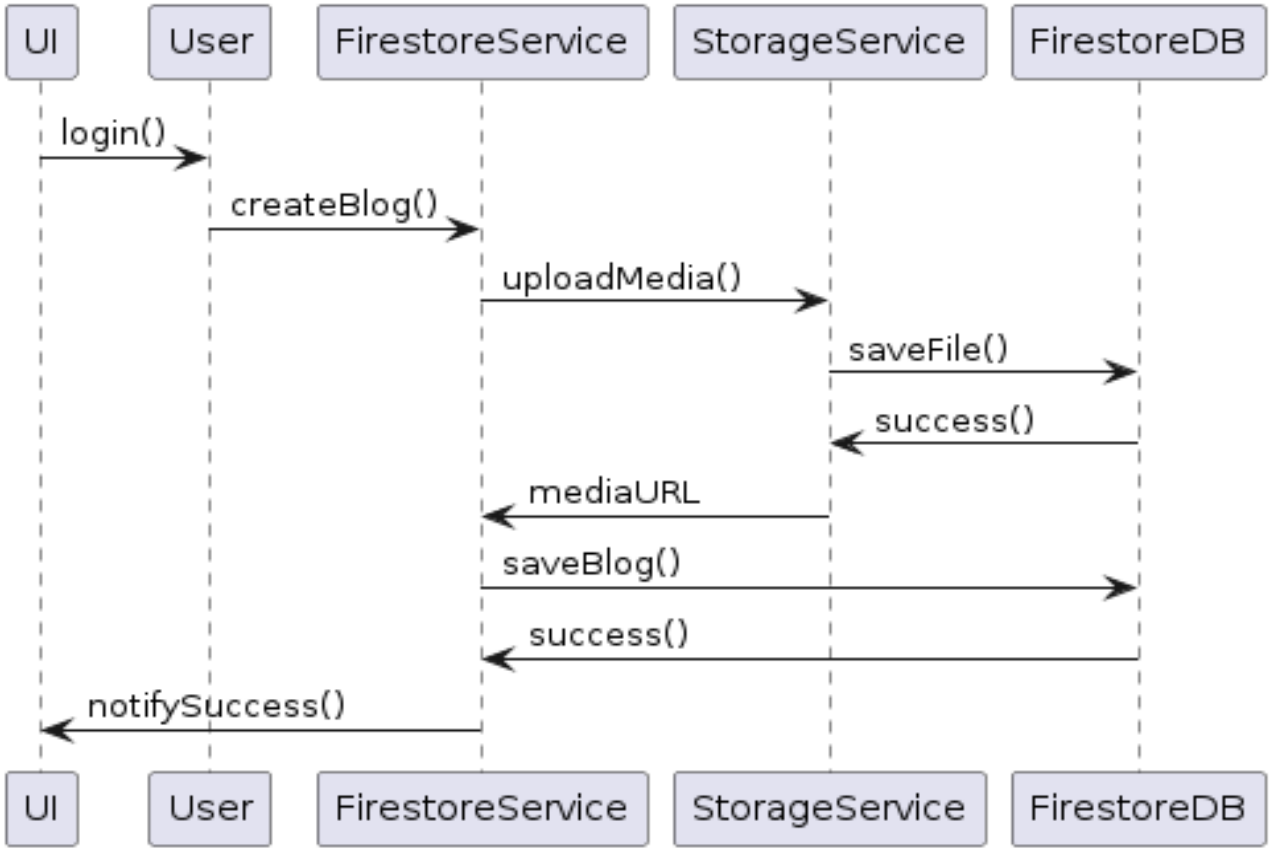

Figure 3.2.4 Collaboration Diagram

3.3 ER Diagram

The ER diagram of G'echo represents the core entities of the system and the relationships between them. The main entities include **User**, **Post**, **Comment**, **Like**, and **Notification**, each containing essential attributes such as userID, postID, commentID, and timestamps. The *User* entity maintains a one-to-many relationship with *Post*, *Comment*, and *Like*, indicating that each user can create multiple posts and interact with various posts through comments and likes. The *Post* entity is linked to media elements and has relationships with both *Comment* and *Like* entities, representing user engagement on each post. Notifications are associated with users and posts to deliver real-time updates. Overall, the ER diagram provides a high-level view of how data is structured and interconnected within the G'echo platform, ensuring efficient data organization and retrieval.

Major Entities

1. User

The *User* entity represents every registered member of the G'echo system, including students, faculty, clubs, and administrators. Each user can create multiple posts and interact with other posts through likes and comments.
Attributes: uid (PK), username, email, profilePicUrl, createdAt

2. Blog

The *Blog* entity stores all posts created by users. Each blog is linked to a user and contains the title, content, media file, and timestamp. A single blog can have many comments and many likes.
Attributes: blogId (PK), userId (FK → users.uid), title, content, mediaUrl, timestamp

3. Comment

The *Comment* entity represents all comments made by users on blog posts. Each comment is associated with both a specific blog and the user who wrote it.
Attributes: commentId (PK), blogId (FK → blogs.blogId), userId (FK → users.uid), text, timestamp

4. Like

The *Like* entity records every like given by users on blog posts. Each like is linked to both the user who liked the post and the blog that received the like.
Attributes: likeId (PK), blogId (FK → blogs.blogId), userId (FK → users.uid), timestamp

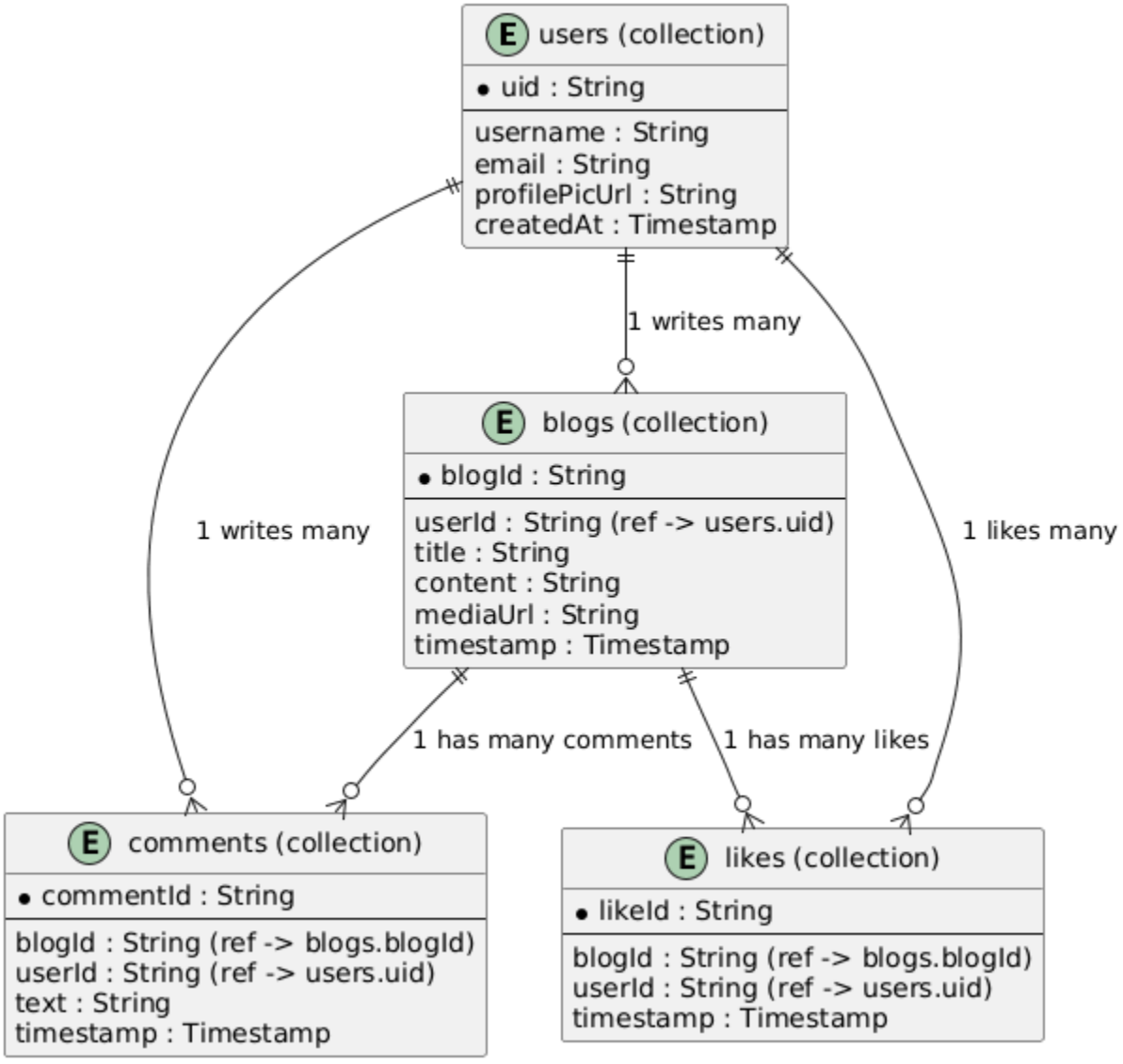

Figure 3.3 ER Diagram

Chapter 4

Technical Implementation

Implementation is the phase where the system design is transformed into a working application. This chapter explains the technologies used, system modules, backend and frontend development processes, and how all components are integrated to form the final interact to clubs.

4.1 Technologies Used

The **G'echo** is developed using a combination of modern frontend and backend technologies. These tools ensure high performance, scalability, and a smooth user experience. The technologies are organized below into two main categories: Frontend Technologies and Backend Technologies.

4.1.1 Frontend Technologies1

1. Flutter 3.8.1+

- Cross-platform mobile app development framework using Dart language
- Provides native performance on both Android and iOS platforms
- Material Design 3 components for consistent UI/UX

2. Dart Programming Language

- Object-oriented language optimized for UI development
- Strong typing system with null safety features
- Asynchronous programming support with Future and Stream APIs

3. Material Design 3

- Google's design system for modern, accessible interfaces
- Adaptive theming and responsive layouts
- Built-in animations and transitions

4. Image Picker & Cached Network Image

- Native device camera and gallery integration
- Efficient image caching and loading mechanisms

5. Flutter Dotenv

- Environment variable management for secure configuration
- Separation of development and production settings
- API key and credential protection

4.1.2 Backend Technologies

1. Firebase Authentication
    - Secure user authentication and session management
    - Multi-platform authentication support
    - Token-based authentication with automatic refresh
2. Cloud Firestore (NoSQL Database)
    - Real-time document database with offline support
    - Scalable, serverless database architecture
    - Real-time synchronization across all connected devices
3. Cloudinary (image CDN)
    - Cloud-based image and video management service
    - Automatic image optimization and transformation
    - Global CDN for fast image delivery
4. Firebase security Rules
    - Server-side security enforcement
    - Role-based access control(RBAC)
    - Real-time validation of database operation
5. Firebase Hosting
    - Static web hosting for configuration files
    - SSL certificate management
    - Global CDN distribution

4.2 System Modules

The University Portal System is divided into four major functional modules, each designed to handle specific roles and responsibilities within the university ecosystem.

4.2.1 Student Module

The Student Module provides limited access focused on content consumption and interaction within the university portal.

Key Features:

- Students can securely log in using their official GLA University email.
- They can view all event posts, announcements, photos, and videos in the main feed.
- Students can interact with posts by liking and commenting.
- They receive real-time notifications for new posts and important updates.
- Students can view their profile details fetched directly from the university database.
- The module helps students stay updated with all academic and extracurricular activities.

4.2.2 Club Module

The Club Module empowers student organizations to create and manage content for the university community.

Key Features:

1. Secure Authentication: Club representative login with verified credentials
2. Content Creation: Create both photo posts and blog articles
3. Media Management: Upload and manage images through Cloudinary integration
4. Post Scheduling: Plan and schedule content publication
5. Engagement Analytics: View likes, comments, and reach metrics
6. Tag Management: Categorize content with relevant tags and topics
7. Profile Customization: Manage club information, description, and branding
8. Content Moderation: Edit and delete own posts with version control

4.2.3 Admin module

The Admin Module provides comprehensive control over the platform with enhanced privileges for university administration

Key features:

1. User Account Management: Approve, activate, or deactivate user registrations
2. Content Moderation: Review, edit, or remove any post across the platform
3. Priority Posting: Create priority posts that appear at the top of feeds for 2 days
4. Registration Approval System: Review and approve pending user registration requests
5. System Analytics: Access comprehensive usage statistics and engagement metrics
6. Bulk Operations: Perform mass actions on users, posts, and content
7. Security Monitoring: Track user activities and system access logs
8. Notification Management: Send system-wide announcements and alerts

4.2.4 Registration Request Module

A specialized module handling the user onboarding process with approval workflows.

Key features:

1. Self-Registration: Allow users to submit registration requests
2. Document Verification: Upload and verify identity documents
3. Approval Workflow: Three-day expiry system for pending requests
4. Automated Notifications: Email alerts for registration status updates
5. Bulk Approval: Admin interface for processing multiple requests
6. Request Tracking: Real-time status updates for applicants
7. Data Validation: Automatic verification of university email domains

4.3 Backend Implementation

The backend architecture leverages Firebase services and Cloudinary for a serverless, scalable solution.

4.3.1 Database Architecture

Firestore Collections Structure:

- admin_users
- student_users
- club_users
- posts
- registration_requests
- users

Document Schema Design:

1. User Collections: Role-based separation with specific fields per user type
2. Posts Collection: Unified structure supporting both photo and blog content
3. Registration Requests: Temporary storage with automatic expiry
4. Indexing Strategy: Optimized queries for real-time performance

4.3.2 Authentication System

Hybrid Authentication Approach:

1. Custom Credential Validation: Check against Firestore user collections
2. Firebase Auth Integration: Create session tokens for app navigation
3. Role-Based Access Control: Enforce permissions at database level
4. Token Management: Automatic refresh and secure storage

Security Implementation:

1. Password validation against stored credentials
2. Temporary Firebase Auth tokens for session management
3. Environment variable protection for sensitive configurations
4. Firestore security rules for data access control

4.3.3 Image Management System

Cloudinary Integration:

1. Upload Pipeline: Direct client-side uploads with signed URLs
2. Automatic Optimization: Format conversion and quality adjustment
3. Folder Organization: Structured storage with user and post identification
4. CDN Delivery: Global content delivery for fast image loading
5. Transformation API: Dynamic image resizing and cropping

4.3.4 Real-time Data Synchronization

Firestore Streams:

1. Live Feed Updates: Automatic UI refresh when new posts are created
2. User Status Changes: Real-time approval status notifications
3. Comment Threading: Instant comment updates across all users
4. Offline Support: Local caching with automatic sync when online

4.4 Frontend Implementation

The Flutter frontend provides a native mobile experience with platform-specific optimizations

4.4.1 User Interface Development

Screen Architecture:

- Authentication Screens: Login and registration with form validation
- Home Screen: Bottom navigation container with role-based tabs
- Feed Screen: Infinite scroll with real-time post updates
- Create Post Screen: Multi-type content creation (photo/blog)
- Profile Screen: User information and settings management
- Admin Screens: Approval interface and system management tools

Responsive Design:

- Adaptive layouts for different screen sizes
- Platform-specific UI elements (Material Design)
- Accessibility compliance with screen readers
- Dark/light theme support

4.4.2 State Management Architecture

1. State Management Layers:
2. Widget State: Local component state with setState()
3. Service Layer: Business logic and API communication
4. Stream State: Real-time data from Firestore
5. Persistent State: User preferences and cached data

4.4.3 Service Layer Architecture

1. AuthService: User authentication and session management
2. PostService: CRUD operations for posts and content
3. ImageService: Cloudinary integration and image processing
4. RegistrationService: User registration and approval workflows

4.4.4 Navigation and Routing

1. Dynamic bottom navigation based on user type
2. Students: Feed, Explore, Profile (3 tabs)
3. Clubs/Admins: Feed, Explore, Create, Profile (4 tabs)
4. Conditional UI elements based on permissions.

Chapter 5

Testing & Results

Testing is a crucial phase of software development used to evaluate the functionality, performance, reliability, and security of the system. This chapter explains the testing methods used, the detailed test cases, and the final results with screenshots.

5.1 Testing Methods

To ensure that the G'echo works as intended, multiple testing techniques were applied:

5.1.1 Unit Testing

Individual functions, API endpoints, and components were tested separately. Examples:

- Student authentication API
- Post creation and validation logic
- Image upload to Cloudinary functions
- Admin approval workflow functions
- Registration request processing

This ensures that each function behaves correctly in isolation.

5.1.2 Integration Testing

This testing verified that combined modules work together smoothly. Examples:

- Student login → feed viewing → post interaction
- Club creates post → image upload → post appears in feed
- Admin approves registration → user account creation → login access
- Registration request → admin notification → approval workflow

It ensures proper interaction between modules and databases.

5.1.3 Functional Testing

Performed to check whether the system meets the functional requirements. Major features tested:

- Role-based authentication (Student, Club, Admin)
- Post creation flow (Photo and Blog posts)
- Image upload and optimization
- Real-time feed updates
- Registration approval system
- Priority post functionality

Each feature was tested end-to-end.

5.1.4 User Interface (UI) Testing

Ensured that:

- Screens load correctly on different devices
- Bottom navigation functions properly based on user roles
- Forms, buttons, and input fields work as expected
- Layout is responsive on various screen sizes

5.1.5 Performance Testing

Checked how the system performs under load:

- Multiple users creating posts simultaneously
- Large image upload and processing time
- Real-time feed updates with multiple concurrent users
- Database query performance with large datasets

5.1.6 Security Testing

Tested for:

- Password validation and secure storage
- Unauthorized access attempts to restricted features
- Role-based access control (Student/Club/Admin)
- API key protection through environment variables

5.2 Test Cases

Below are sample test cases used during testing.

Table 5.2.1 Test Case 1 – Login Test Case

Test Case ID	TC-01
Feature	User Login
Description	Verify Login with Valid Credentials
Input	Email & Password
Expected Output	User should be redirect to dashboard
Actual Output	As Expected,
Status	Passed

Tabel 5.2.2 Test Case 2 – Invalid Login Test Case

Test Case ID	TC-02
Module	Authentication
Test Scenario	Login with Incorrect Password
Precondition	User exists
Input	Wrong Password
Expected Output	Error Message “Invalid Credentials”
Status	Passed

Table 5.2.3 Test Case 3 – Post Creation Test Case

Test Case ID	TC-03
Module	Post Management
Test Scenario	Create photo post and submit
Precondition	Club user authenticated
Input	Image file, caption, tags
Expected Output	Post created and appears in feed
Actual Output	Post successfully created
Status	Passed

Table 5.2.4: Test Case 4 - Image Upload Test Case

Test Case ID	TC-04
Module	Image Management
Test Scenario	Upload image to Cloudinary
Precondition	Valid image file selected
Input	Image file from gallery
Expected Output	Image uploaded and URL returned
Actual Output	Working fine
Status	Passed

Table 5.2.5 Test Case 5 - Admin Approval Test Case

Test Case ID	TC-05
Module	Registration Management
Test Scenario	Approve pending registration request
Precondition	Admin authenticated
Input	Registration request selection
Expected Output	User account created successfully
Actual Output	Account created and user can login
Status	Passed

Table 5.2.6 Test Case 6 - Priority Post Test Case

Test Case ID	TC-06
Module	Admin Module
Test Scenario	Create priority post
Precondition	Admin login
Input	Post content with priority flag
Expected Output	Post appears at top of feed for 2 days
Actual Output	Working fine
Status	Passed

Tabel 5.2.7 Test Case 7 - Unauthorized Access Restriction

Test Case ID	TC-07
Module	Security
Test Scenario	Student trying to access admin features
Precondition	Student logged in
Input	Attempt to access admin approval screen
Expected Output	"Unauthorized access" or feature hidden
Actual Output	Access correctly blocked
Status	Passed

Table 5.2.8 Test Case 8 - Real-time Feed Update

Test Case ID	TC-08
Module	Feed Management
Test Scenario	New post appears in real-time
Precondition	Multiple users viewing feed
Input	Club creates new post
Expected Output	Post appears instantly in all user feeds
Actual Output	Real-time update working correctly
Status	Passed

Table 5.2.9 Test Case 9 - Role-based Navigation

Test Case ID	TC-09
Module	Navigation
Test Scenario	Different navigation for different user types
Precondition	Users of different types logged in
Input	Login as Student, Club, Admin
Expected Output	Post Students see 3 tabs, Clubs/Admins see 4 tabs
Actual Output	Navigation correctly adapted
Status	Passed

Table 5.2.10 Test Case 10 - Registration Request Expiry

Test Case ID	TC-10
Module	Registration Management
Test Scenario	Registration request expires after 3 days
Precondition	Registration request submitted
Input	Wait for 3 days
Expected Output	Request automatically expires
Actual Output	Expiry mechanism working correctly
Status	Passed

Chapter 6

Conclusion

The G'echo platform successfully transforms the traditional communication system of GLA University into a modern, unified, and interactive digital environment. By bringing students, faculty, clubs, and administrators onto a single platform, G'echo ensures that information flows smoothly and reaches every member of the campus community in real time. Students can now stay connected with various clubs, explore upcoming events, interact with posts, and remain updated about all academic and extracurricular activities without depending on physical notices or scattered communication channels. This enhanced visibility helps students discover new opportunities, participate actively in events, and engage more closely with campus life.

For clubs and administrators, G'echo provides an efficient and centralized way to share announcements, showcase events, and connect with students. Through features like creating posts, adding media, commenting, and sending notifications, the platform enables them to communicate instantly and effectively. Faculty members also remain informed and engaged, contributing to an environment where information is transparent and accessible to all.

Overall, the project achieves its objective of improving connectivity, promoting student involvement, and strengthening the bridge between students and campus organizations. G'echo not only enhances communication but also builds a sense of community by ensuring that every event, activity, and update is visible and easily accessible. It represents a significant step toward digital transformation within the university, making campus interactions more organized, interactive, and student-centric.

Chapter 7

Summary

The G'echo platform is designed to create a seamless communication channel within GLA University by connecting students, faculty, clubs, and administrators through a single, centralized application. In many universities, students often miss important events, updates, and club activities because information is scattered or communicated through multiple platforms. G'echo solves this problem by bringing all campus announcements and interactions together, allowing students to stay informed about everything happening around them. From cultural events and academic seminars to club recruitment and sports activities, students can quickly access details, view posts, and stay engaged with university life.

Clubs benefit greatly from this platform as well. Instead of relying on manual promotion or offline posters, they can create posts, upload images and videos, and share important updates instantly with all students. This improves visibility for clubs and helps them connect with interested students more effectively. Faculty and administrators also remain connected, as they can review announcements, comment on posts, and monitor campus activities in real time. The platform supports easy interaction through likes, comments, and notifications, creating a dynamic and interactive campus environment.

The summary of the system reflects its core purpose—strengthening communication within the university and ensuring that students never miss an opportunity. Through real-time updates powered by Firebase and a user-friendly interface built with Flutter, G'echo delivers fast, secure, and reliable information to every user. Students can now explore upcoming events, track club activities, and stay connected with different groups, enhancing participation and campus engagement. Ultimately, G'echo helps create a more active, informed, and connected university community, where every student has access to the information they need to grow academically and socially.

Chapter 8

Screen Shots

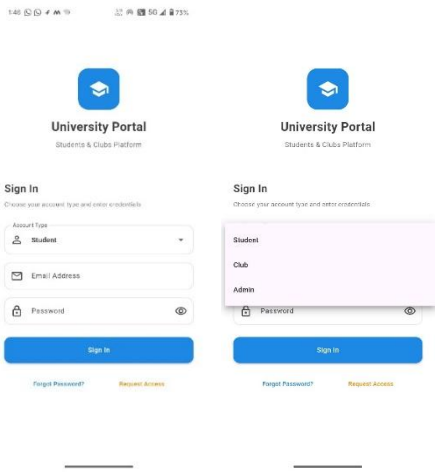

Fig 8.1 Login Page

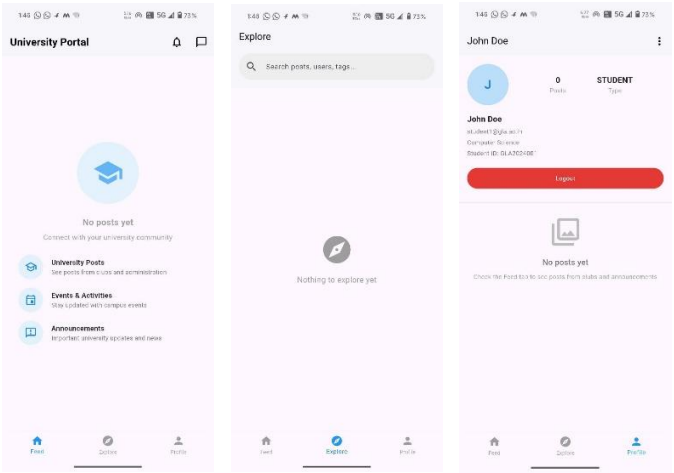

Fig 8.2 Student Accessed Application Page (Feed , Explore , Profile)

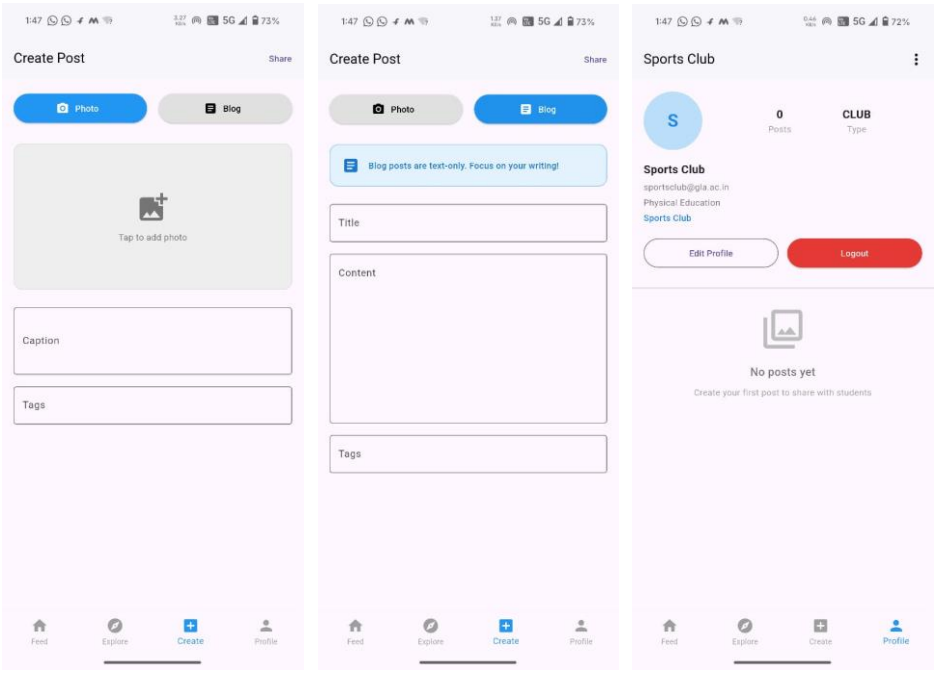

Fig 8.3 Club Accessed Application Page (Create Post , Profile)

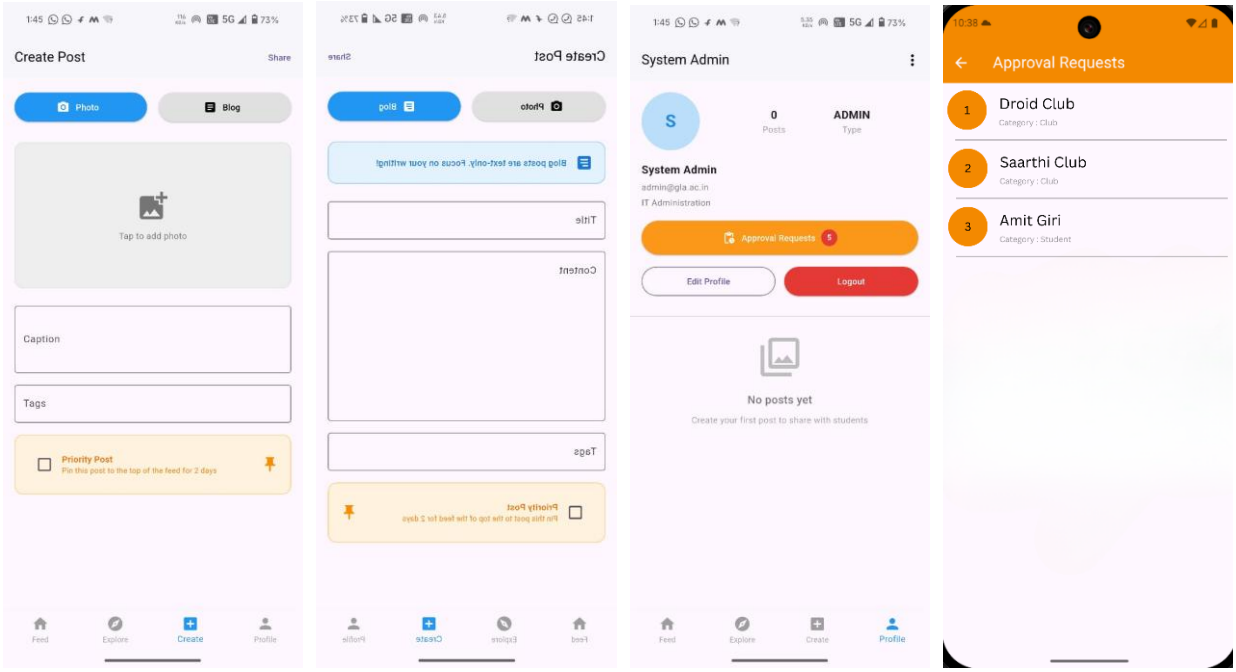

Fig 8.4 Admin Accessed Application Page

(Create Post, Profile , Approved Request)

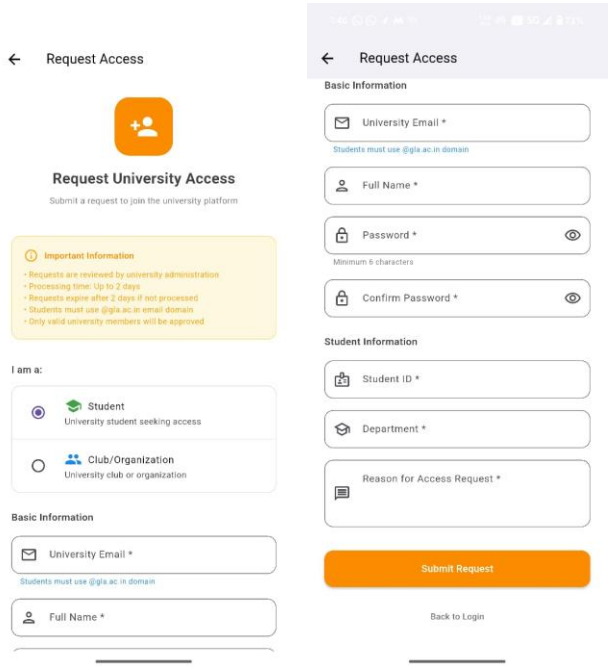

Fig 8.5 Request Access Page

Chapter 9

Meeting with Mentor