

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



OOMD Mini Project Report

SOCIAL MEDIA NETWORK

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

KRISHN MALOO - 1BM21CS092

KSHITIJ S - 1BM21CS093

LIKHITH GS - 1BM21CS096

MANJIL RAJ PANTA - 1BM21CS103



Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2023-2024

B.M.S COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



DECLARATION

We, KRISHN MALOO (1BM21CS092) ,KSHITIJ S (1BM21CS093), LIKHITH GS (1BM21CS096), MANJIL RAJ PANTA (1BM21CS103) students of 6th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "SOCIAL MEDIA NETWORK" has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester March - June 2024. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

KRISHN MALOO - 1BM21CS092

KSHITIJ S - 1BM21CS093

LIKHITH G S - 1BM21CS096

MANJIL RAJ PANTA - 1BM21CS103

**BMS COLLEGE OF ENGINEERING DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING**



CERTIFICATE

This is to certify that the OOMD Mini Project titled “**SOCIAL MEDIA NETWORK**” has been carried out by KRISHN MALOO (1BM21CS092), KSHITIJ S (1BM21CS093), LIKHITH G S (1BM21CS096), MANJIL RAJ PANTA (1BM21CS103)

Signature of the Faculty in Charge

TABLE OF CONTENTS

Sl No	Title	Page No
1	Ch 1: Problem statement	1
2	Ch 2: Software Requirement Specification	2
3	Ch 3: Class Diagram	6
4	Ch 4: State Diagram	8
5	Ch 5: Interaction Diagram	10

Chapter 1: Problem Statement

In today's digital age, social media platforms are integral to connecting people, sharing information, and fostering communities. However, existing social media applications face significant challenges that hinder user experience and safety. Privacy and security concerns are paramount, as users increasingly worry about the protection of their personal information amidst frequent data breaches and unauthorized access incidents, leading to a decline in user trust. Additionally, content moderation remains a critical issue, with platforms struggling to prevent the spread of misinformation, hate speech, and inappropriate content, which can create a toxic environment detrimental to users' well-being.

Moreover, scalability is a persistent challenge; as user numbers grow, platforms must handle increased traffic and data loads without sacrificing performance. Ensuring scalability and reliability is crucial for maintaining a seamless user experience. Engaging and retaining users is another area where current platforms often fall short, as they must provide meaningful interactions and valuable content to keep users active and loyal. Furthermore, the diverse needs and preferences of users require platforms to offer customization options, as a one-size-fits-all approach is inadequate for catering to various user demographics and interests.

The platform will be designed to handle a large user base and high data traffic, maintaining optimal performance and uptime. To foster meaningful interactions and keep users engaged, the application will feature personalized content recommendations and customization options, allowing users to tailor their experiences according to their preferences. Additionally, the user interface will be user-friendly and accessible, offering a seamless experience across various devices. By addressing these critical issues, the new social media application aims to create a safe, engaging, and customizable platform that meets the evolving needs of users in the digital era.

Chapter 2: Software Requirement Specification

1. General description:

1.1 Purpose

The purpose of this document is to define the requirements for a social media application that allows users to create profiles, connect with friends, share content, and interact through comments and likes.

1.2 Scope

This social media application will provide a platform for users to:

- Create and manage profiles.
- Add and accept friend requests.
- Post text, images, and videos.
- Comment on and like posts.
- Send private messages.
- Receive notifications.

1.3 Definitions, Acronyms, and Abbreviations

- **User:** An individual who has created an account on the social media application.
- **Post:** Content shared by a user, including text, images, and videos.
- **Friend Request:** A request sent by one user to another to become friends on the platform.
- **Notification:** A message alerting the user to new activity such as likes, comments, or messages.

1.4 References

- Example Social Media Platform Documentation

1.5 Overview

The social media application aims to facilitate social interaction and content sharing among users. The following sections provide detailed functional, interface, performance, design constraints, non-functional attributes, and preliminary schedule and budget requirements.

2. Functional Requirements

2.1 User Registration and Authentication

- Users must be able to register with a unique username, email, and password.
- Users must be able to log in with their email and password.
- Password reset functionality must be available.

2.2 Profile Management

- Users must be able to create and update their profile information, including profile picture, bio, and personal details.

2.3 Friend Management

- Users must be able to send and accept friend requests.
- Users must be able to view a list of their friends.

2.4 Content Sharing

- Users must be able to create, edit, and delete posts containing text, images, and videos.
- Users must be able to view posts from their friends in a feed.

2.5 Interactions

- Users must be able to like and comment on posts.
- Users must be able to send private messages to their friends.

2.6 Notifications

- Users must receive notifications for friend requests, likes, comments, and messages.

3. Interface

3.1 User Interface

- The application must have a responsive web interface accessible via desktop and mobile browsers.
- The interface must be intuitive and easy to navigate.

3.2 Accessibility

- The application must adhere to accessibility standards to support users with disabilities.

4. Performance Requirements

4.1 Response Time

- The application must respond to user actions (e.g., posting, commenting, liking) within 2 seconds.
- Page loads must occur within 3 seconds for 95% of user requests.

4.

5. Design Constraints

5.1 Scalability

- The application must be designed to handle up to 10 million active users.
- The architecture must support horizontal scaling.

5.2 Reliability

- The application must have 99.9% uptime.
- Data integrity must be ensured through regular backups and robust error handling.

6. Non-Functional Attributes

6.1 Security

- The application must use HTTPS for all communications.
- User data must be encrypted at rest and in transit.
- Regular security audits must be conducted.

6.2 Usability

- The application must be user-friendly with a consistent design and layout.
- Users must be able to accomplish common tasks without extensive training.

6.3 Maintainability

- The codebase must follow standard coding practices and be well-documented.
- The application must be modular to facilitate updates and maintenance.

7. Preliminary Schedule and Budget

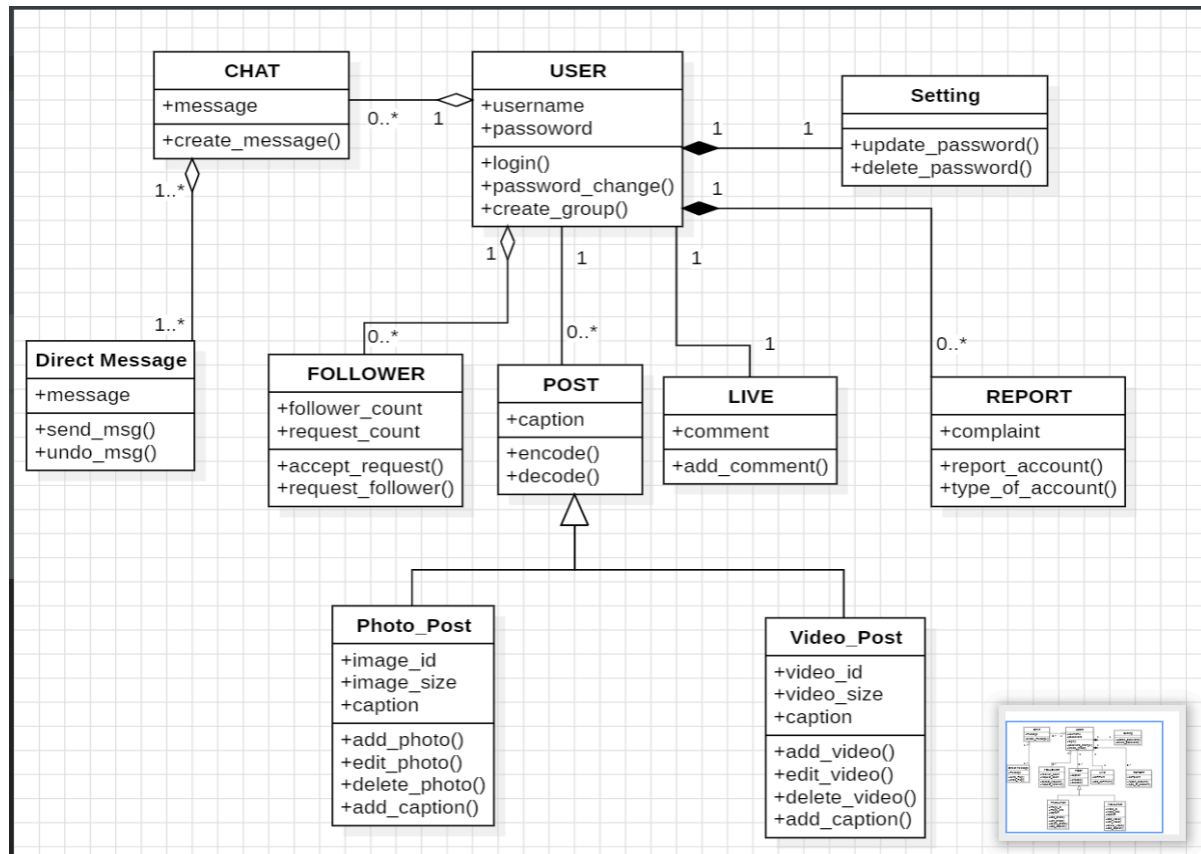
7.1 Schedule

- **Phase 1:** Requirements Gathering and Analysis (1 month)
- **Phase 2:** Design and Prototyping (2 months)
- **Phase 3:** Development (6 months)
- **Phase 4:** Testing (2 months)
- **Phase 5:** Deployment and Launch (1 month)
- **Total Duration:** 12 months

7.2 Budget

- **Personnel:** \$600,000
 - Project Manager: \$100,000
 - Developers: \$300,000
 - QA Testers: \$100,000
 - Designers: \$100,000
- **Infrastructure:** \$200,000
- **Software Licenses:** \$50,000
- **Miscellaneous:** \$50,000
- **Total Budget:** \$900,000

Chapter 3: Class Modeling



Class diagrams are a fundamental aspect of Unified Modeling Language (UML) used in software engineering to depict the static structure of a system. They provide a visual representation of classes, their attributes, methods, and the relationships between them. Here's a brief note on class diagrams: In the context of a social media platform, a class diagram can help illustrate the structural elements of the system. Here's a conceptual representation using a class diagram:

Class Diagram for Social Media Platform:

1. User Class:

- Attributes: UserID, Username, Email, Password, ProfilePicture
- Methods: Login(), Logout(), CreatePost(), Comment(), Like()

2. Post Class:

- Attributes: PostID, Content, Timestamp, LikesCount, CommentsCount
- Methods: AddLike(), AddComment(), Share()

3. Friendship Class:

- Attributes: FriendshipID, User1ID, User2ID, Status
- Methods: SendFriendRequest(), AcceptFriendRequest(), RemoveFriend()

4. Notification Class:

- Attributes: NotificationID, UserID, Content, Timestamp, IsRead
- Methods: MarkAsRead(), DeleteNotification()

5. Group Class:

- Attributes: GroupID, GroupName, Description, MembersCount
- Methods: AddMember(), RemoveMember(), PostInGroup()

6. SystemUser Class (for administrative purposes):**

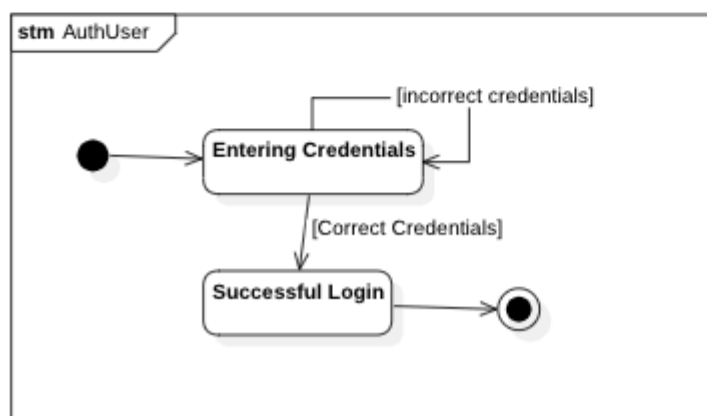
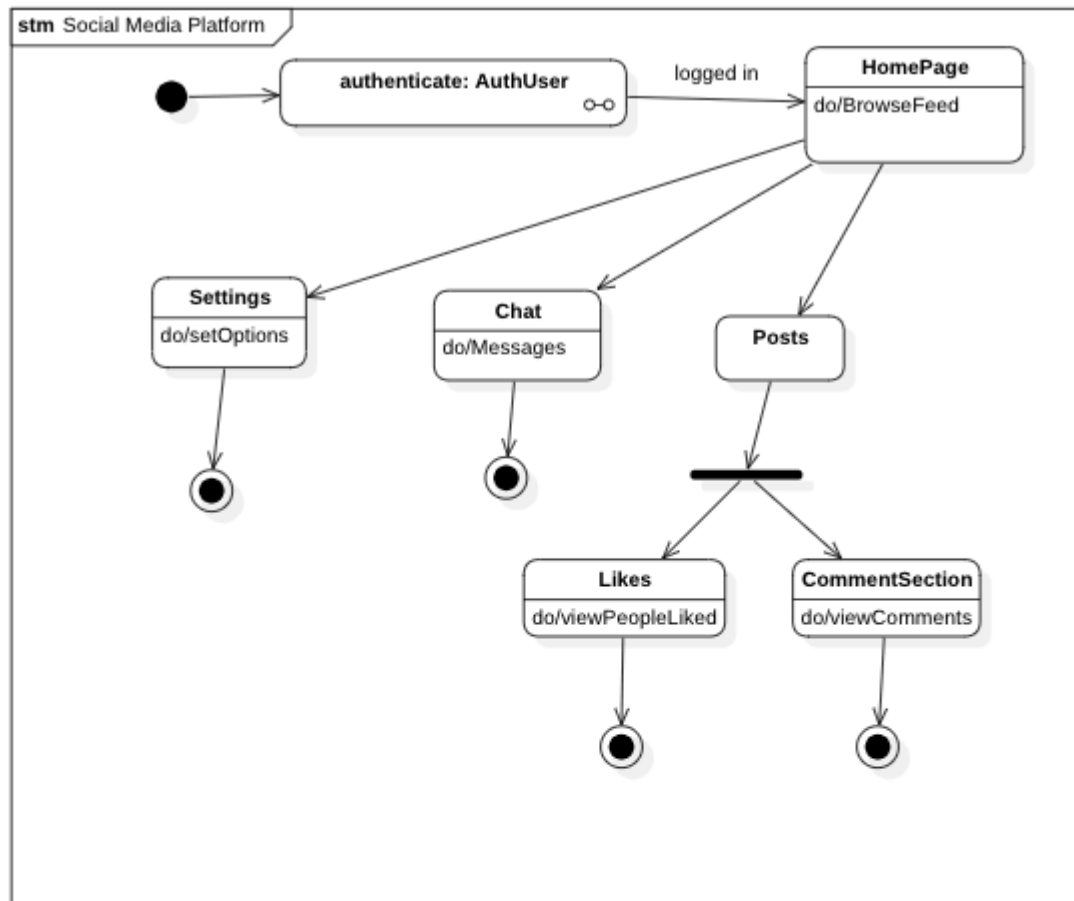
- Attributes: AdminID, Username, Email, Password
- Methods: ManageUsers(), ModerateContent(), ConfigureSystem()
- Relationships:
 - User-Post Association:
 - One User can have multiple Posts; One Post is associated with one User.
 - User-Friendship Association:
 - Users can have multiple Friendships; Friendship involves two Users.
 - User-Notification Association:
 - One User can receive multiple Notifications; One Notification is associated with one User.
 - User-Group Association:
 - Users can be part of multiple Groups; One Group has multiple Members.
 - Admin-User Association:
 - One Admin can manage multiple Users; One User is managed by one Admin. This class

diagram provides a foundational understanding of the relationships and responsibilities of key classes within a social media platform. It emphasizes the user-centric nature of such platforms, detailing the interactions between users, posts, friendships, notifications, and administrative functionalities. Keep in mind that the specifics of class attributes and methods can vary based on the design and requirements of the actual social media platform being developed.

Document: Represents a document. It has an ID, content, and can have multiple attachments.

Folder: Represents a folder of files. It has a name , ID, number of files and other attributes and operations within it.

Chapter 4: State diagrams



Screenshot

A state machine diagram, within the realm of a social media platform, serves as a valuable visual representation of the various states that entities within the system can undergo and the transitions between these states. In the context of social media, entities like users or posts can exhibit different states, and state machine diagrams illustrate how they transition from one state to another based on specific events or conditions.

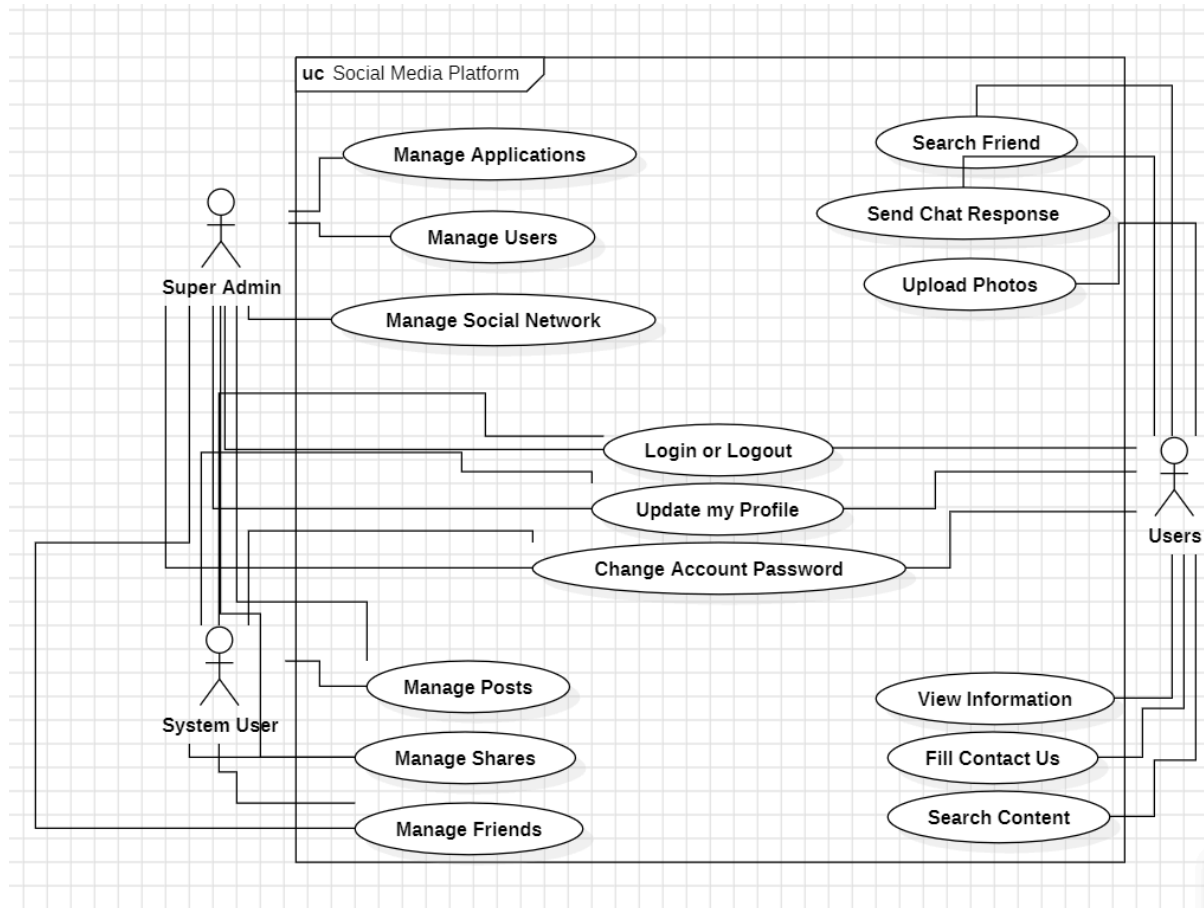
For example, consider a "User" entity on a social media platform:

1. User States:- States may include "Logged In," "Logged Out," "Inactive," or "Premium User."
2. State Transitions:- Events triggering state transitions can be depicted, such as "Login" leading to a transition from "Logged Out" to "Logged In."
3. Conditional Transitions:- Conditions for transitions, like "Upgrade to Premium," can be illustrated, showcasing how a user moves from a standard state to a premium state.
4. Post States:- A "Post" entity may have states like "Published," "Reported," or "Deleted."
5. Transitions Due to User Actions:- User actions, such as reporting a post, can be represented as events leading to state transitions.

State machine diagrams effectively capture the dynamic behavior of entities within the social media system, showing how they respond to user interactions and system events. These diagrams assist in identifying and modeling complex scenarios, ensuring a comprehensive understanding of the system's behavior over time. By visualizing state transitions, conditions, and events, stakeholders can gain insights into the lifecycle of various entities, aiding in system design, testing, and the overall improvement of user experiences on the social media platform.

Chapter 5: Interaction Modeling

5.1 USE CASE DIAGRAM



Use case diagrams are a type of Unified Modeling Language (UML) diagram used in software engineering to visually represent the interactions between various actors (users or external systems) and a system. These diagrams provide a high-level overview of the system's functionalities and the ways in which users or external entities interact with it.

Let's consider a simple scenario involving a system with three actors: User, System User, and Super Admin. Here are some use cases for each actor:

1. User:

- **Login:** The user should be able to log into the system using valid credentials.
- **View Profile:** Users can view and edit their profiles, updating personal information and preferences.
- **Post Content:** Users can create and publish text, images, or videos to share with others.
- **Interact with Content:** Users can like, comment on, or share posts created by themselves or others.

- Friend Connection: Users can send and accept friend requests to connect with others.

2. System User:

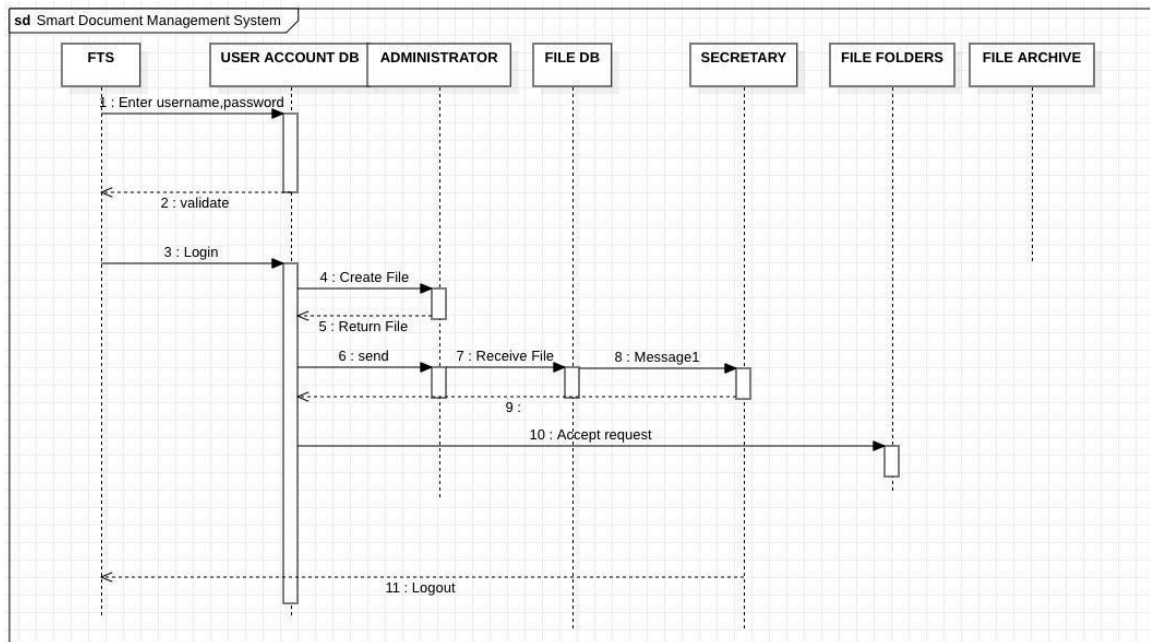
- Manage User Accounts: System users can create, modify, or deactivate user accounts.
- Content Moderation: System users can monitor and moderate user-generated content, ensuring compliance with community guidelines.
- Resolve User Issues: System users can address and resolve issues reported by regular users.
- System Configuration: System users can configure system settings, such as privacy controls and notification preferences.

3. Super Admin:

- User Management: Super Admins have the authority to manage all user accounts, including creating, modifying, and deactivating accounts.
- System Configuration: Super Admins can configure global system settings, affecting the entire platform.
- Access Logs and Analytics: Super Admins can access system logs and analytics to monitor user activities and system performance.
- Emergency Response: In case of critical issues, Super Admins can take emergency measures, such as system shutdown or applying security patches.

These use cases illustrate the various interactions and functionalities associated with each actor in the system, providing a basis for further detailed system modeling and development.

5.2 SEQUENCE DIAGRAM



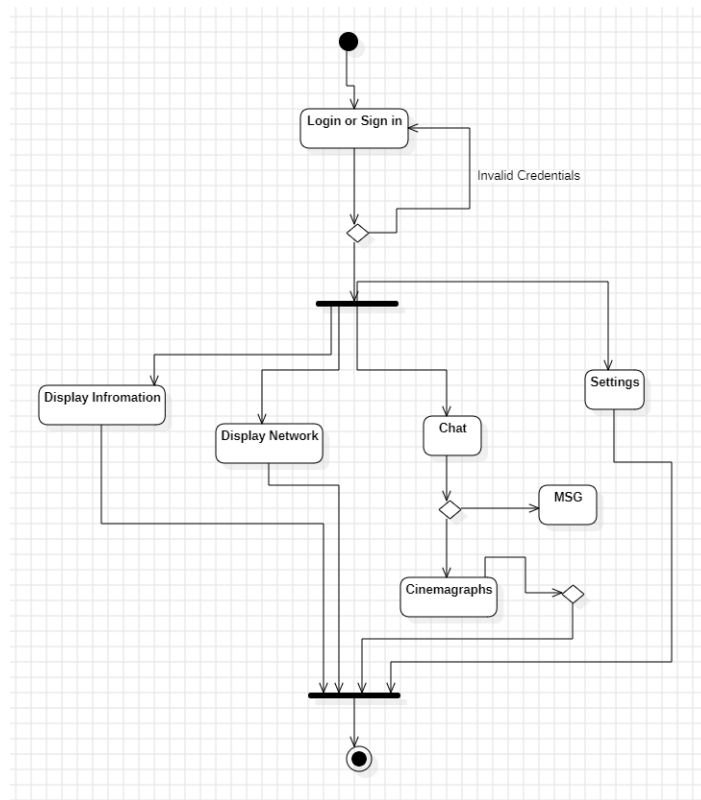
A sequence diagram in the context of a social media platform serves as a dynamic visualization tool that illustrates the interactions between various components, primarily focusing on the chronological order of messages and actions exchanged among users and the system. This diagram provides a step-by-step representation of how different entities collaborate in real-time. For instance, it can showcase the sequence of events when a user logs in, posts content, receives comments, and interacts with notifications.

In a social media sequence diagram:

1. **User Interaction Sequences:-** The diagram can detail how a user logs in, navigates the platform, and interacts with various features like posting content, liking, commenting, or sending messages.
2. **Message Exchange:-** It explicitly displays the messages exchanged between users and the system, highlighting the order of communication and the content of each message.
3. **Notification Flow:-** Users receiving notifications and responding to them can be visually represented, showing the flow of information and user engagement.
4. **Concurrent Actions:-** Concurrent activities, such as multiple users interacting simultaneously or reacting to the same content, can be illustrated to capture the real-time nature of social media interactions.
5. **System Responses:-** The diagram can include how the system responds to user actions, whether it involves updating the user interface, processing content, or triggering notifications.

By representing these interactions in a chronological manner, sequence diagrams offer a clear and intuitive understanding of the dynamic behavior of a social media platform. This helps developers and stakeholders identify potential bottlenecks, refine user flows, and ensure the seamless functioning of the system during real-world user interactions.

5.3 ACTIVITY DIAGRAM



An activity diagram is a UML diagram used to visually represent the flow of activities within a system, making it particularly useful in modeling the dynamic aspects of a social media platform. In the context of social media, an activity diagram can effectively illustrate the sequence of user interactions and system responses. For instance, the diagram may begin with a "User Login" activity, followed by branching pathways representing actions like "Create Post," "Interact with Content," or "Send Message." Decision nodes can depict conditional activities, such as checking user privacy settings before displaying certain content. It provides a clear visualization of the flow of activities and decisions, showcasing how users navigate through the platform. Additionally, activity diagrams can capture concurrent activities, like simultaneous posting and commenting, enhancing their utility in representing the real-time and interactive nature of social media interactions. Overall, activity diagrams offer a valuable tool for understanding and communicating the dynamic behavior of a social media system, aiding developers and stakeholders in refining the platform's functionality and user experience.

REFERENCES

- STARUML SOFTWARE
- Chat.openai.com
- <https://creatly.com>
- <https://www.coursera.org/learn/software-engineering-modeling-software-systems-using-uml>
- <https://ieeexplore.ieee.org/document/5260494>