VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

NGUYEN HOANG PHUC– 521H0509

KIEU THANH PHAT – 521H0125

DEVELOPING A SOCIAL NETWORK APP USING ASP.NET CORE AND REACT JS

INFORMATION TECHNOLOGY PROJECT

COMPUTER SCIENCE – SOFTWARE ENGINEERING

Advised by

MSc. Duong Huu Phuoc

HO CHI MINH CITY, 2024

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

NGUYEN HOANG PHUC– 521H0509

KIEU THANH PHAT – 521H0125

DEVELOPING A SOCIAL NETWORK APP USING ASP.NET CORE AND REACT JS

INFORMATION TECHNOLOGY PROJECT

COMPUTER SCIENCE – SOFTWARE ENGINEERING

Advised by

MSc. Duong Huu Phuoc

HO CHI MINH CITY, 2024

ACKNOWLEDGEMENT

We would like to express my sincere thanks to MSc. Duong Huu Phuoc of the subject "Information Technology Project" and the Faculty of Information Technology for making conditions for me to do this report so that I can add the final score column in my process, as well as helping and guiding us throughout the process of studying and making reports so that I and my friends won't be late the process.

Writing the report has helped me practicing more presentation skills, as well as many other skills. Due to lack of experience in report writing as well as limited knowledge and reasoning ability, in this report there will certainly be errors, looking forward to receiving comments and suggestions from my grateful teacher to help me improve myself.

Ho Chi Minh city, August 1st 2024

Authors

(Sign and write full name)

Nguyen Hoang Phuc, Kieu Thanh Phat

DECLARATION OF AUTHORSHIP

We hereby declare that this is my own research project and is under the scientific guidance of MSc. Duong Huu Phuoc. The research content and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation were collected by the author from different sources and clearly stated in the reference section.

In addition, the Project also uses a number of comments, assessments as well as data from other authors and other organizations, all with citations and annotations of the original source.

If any fraud is detected, I will take full responsibility for the content of my Project. Ton Duc Thang University is not involved in copyright violations caused by me during the implementation process (if any).

Ho Chi Minh City, August 1st 2024

Authors

(Sign and write full name)

Nguyen Hoang Phuc, Kieu Thanh Phat

SOCIAL NETWORK APP FOR SHARING POSTS THAT FOCUSES ON CONNECTING WITH FRIENDS.

The social media network similar to it predecessor Facebook and Instagram , is built on the basis of enhancing the connectivity of friends with an endless reach around the world. User can posts feed and post describing their experiences while maintaining a length full and informal conversation through their chat group

As a result of building the application with Asp.Net Core and React JS, the application have fulfill all criterial of a high quality application. using React JS interactive libraries with additional help of Prime React component libraries for the seamless UI/UX design. Additionally, users' identity and information security are maintained with Asp.Net Core Identity and Entity Framework

Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CRUD	Create, Read, Update, Delete
ASP.NET	Active Server Pages .NET
JWT	JSON Web Token
SQL	Structured Query Language
API	Application Programming Interface
IDE	Integrated Development Environment
UI/UX	User Interface / User Experience
SSMS	SQL Server Management Studio
MVC	Model-View-Controller
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
REST	Representational State Transfer
DB	Database
VS Code	Visual Studio Code

Chapter 1 INTRODUCTION

1.1 Justifying Topic Choice

In today's interconnected world there is a great need for digital communication, communication that will enable friends and family to connect with each other through a wide range of distances. Moreover, with information being so redundant people are gaining much more information and knowledge through a wide range of mediums than they used to before, but that doesn't mean that the information being received is informative. An example of this is disinformation which can potentially cause harm depending on how the individual perceives it.

The social media network of today is creating solutions for harnessing our world interconnected infrastructure. These social networks are creating a vast database of users and users' content with both flexibility and growth in mind, in addition to preventing bad actors from creating a hostile environment for other users. The forefront of These social media network is Facebook and Instagram, these two websites have created an environment that encourages users to stay engaged with the latest trend and connect with their friend personal experiences, with added benefit of discovering their own hobby and communities. Even with the vast function of these social media networks, they still lacking the localization and feature tailor to the preferences and cultural aspect of a specific country like Vietnam.

Our social network application is set on fixing these issues, by customizing the UI/UX design to meet the preferences of Vietnamese users and incorporating features that are tailors toward their culture. Resulting in a more friendly application for Vietnamese users enhancing their social tie, sense of nationality and comfortability when using our social media networks.

1.2 Topic Implementation Goals

The main goal for this application is to provide a social network platform that is tailor for the specific need of Vietnamese users, to post and share their experiences and maintain connection through features like chat messaging. The system is further enhanced by the customization feature. For example, users can choose who they want to share post with, what group of friends would the users allow to see the posts, and which post will be saved for later viewing. The application is also committed with enhancing data security and integrity. This is implemented by incorporating ASP.NET identity frameworks for the authorization and authentication method, which decide what users can and cannot do. These protected barrels help keep user accounts protected from bad actors. In addition to these main features, there will also be additional advanced communication features like creating and managing group chat, and assigning role and user privilege in each group chat. Creating a moderation system that allows moderators to monitor user activity and detect bad actors in the system.

1.3 Subject And Scope of Research.

The main users of the application are people ranging from the age of 18-65 who would like to post their experiences, admins who are assigned with managing groups, and companies who want to expand their band to a wider audience. The software that the team will use for collaboration are GitHub, GitLab and Git kraken. Each member used a wide range of IDE including but not limited to Microsoft VS, VS Code. The Frontend is programmed using JS with the main library being React JS, using primary Prime React for the UI. On the other hand, the back of the application will use C# using the ASP.NET CORE framework incorporating Identity and managing the database server hosted on a local SQL database using SQL database manager. The application is tested with 10-20 users who have used social media platforms before for better feedback.

1.4 Research Method

System implementation: the system will use an authentication method using username and password, users can register for their new account or login with the right password and credentials. In the situation that the user forgets their password the user can use the recover password function that will send an OTP request, prompting the user to check their email in order to follow the necessary step required to create a new password for their account. When Login, the user can perform common functionality seen in other social media websites like CRUD operation on their posts and additional interaction operation to another user. All users and their users' posts can be reported on the basis of preventing bad actors and maintaining a welcoming community for the application. On the admins side, admin can monitor user activity and posts to filter for bad actors in the system, have a detailed description of user post activity and decide which post to ban or delete if it violates community guidelines.

1.5 Practical Significance of The Topic.

The social media networks are tailor for cultural preferences of the Vietnamese people through UI/UX and additional feature, targeting audience between the age of 18-65. The user of the social network can post about their journey and experience places that they have visited through medium like picture and text. With the added benefit of messaging and maintaining friendships with the network chatting feature. Admins can moderate and check for disturbances with the site, seeing if there are any bad actors in the system.

Chapter 2 LITERATURE REVIEW

This chapter is used to explain, understand and contextualize the technology used in application development. This explanation will include details such as why the technology was used, what it is used for and how we implement it in the application operation.

2.1 Vite

Is a complied build tools that provide faster development experience for developer . Vite used native ES modules and modern browsers APIs to quickly comply your code hence the name Vice which in French mean “quick”, with no need to incorporate bundle. The building development server in Vite is equipped with an optimized system for faster reloading and hot module replacement, allowing for changes that developer made to happen in real time. In our project we used Vice to optimize our development cycle, allowing us to make rapid changes in the front end without having to worry about dependency between models or waste time rebuilding the application by using hot reload.

Figure 1 Vite Logo

2.2 Axios

Axios is a promise-based HTTPS client that is used for node bases applications and applications that run on the browser. Some key feature of Axios is it isomorphic nature this mean that it have the capability to run on both the server side and client side of an application. The main difference between running on the server side is that it uses native node.js http module as opposed to XMLHttpRequests when running on the client side. There are many features in the Axios library, but our application used Axios for its ability to handle https requests, interpreted and handle these requests when sending them from the client to the server. As a result, making it easier for our application to handle the request and how to response to them.

Figure 2 Axios Logo

2.3 React JS

React is a JavaScript library created by Facebook to simplify the process of creating and managing the frontend interface. React optimize your rendering and frontend application by using the virtual Dom when an element render instead of changing the entered DOM, react will create a virtual DOM to apply the initial change then using a Diff algorithm apply the needed change on the real DOM, this process used to optimize rendering speed of your applications. React is also built on the principle of teaching one write anywhere with it agile development and compatibility with other frameworks and library it can be added into your application anytime anywhere. We used React to simplify or frontend development thought the use of many features like components, JSX, state and props making our UI adaptable to the changing UX of the application.

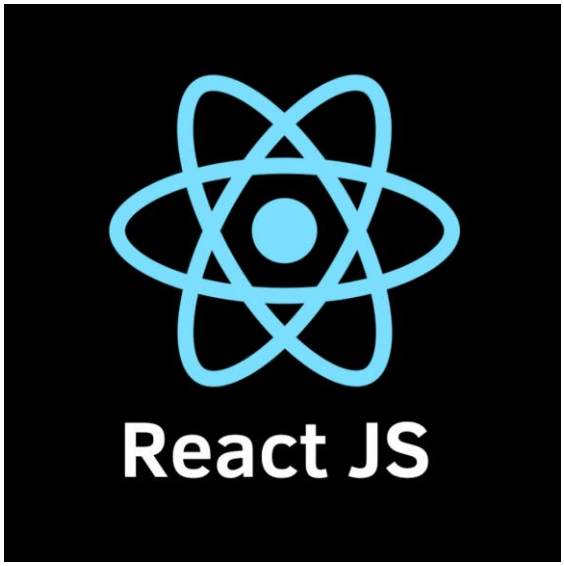

Figure 3 React Logo

2.3.1 **TSX**

The language that we used in your application is TSX, TSX is similar to JSX in react with the main difference being that it is written in typescript instead of JavaScript. In the same way as JSX, TSX is an extension of JavaScript that also comes with the full power of ES6, this power translates to its main feature which is the using a {} expression to combine JavaScript dropdown tag in a single file for separation of concern. This separation of concern has created a unit of call component.

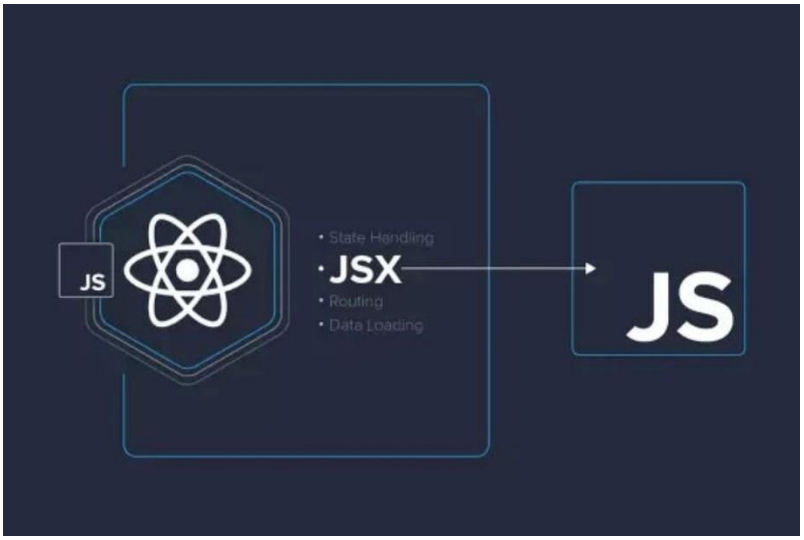

Figure 4 JSX flowchart

2.3.2 **Components**

Components are encapsulated pieces of react code that represent a UI element by managing the output render in the DOM. Each component can be reused multiple time in multiple different file bases on the developer need. In addition to being reusable, components can pass

information to each other to meet the adaptation of the UI. This information are represented in two main ways state and props.

2.3.3 **Props**

Props are object that are passed thought by its parent components specify using it child components argument, every parent component can pass state and information to its child component using props.

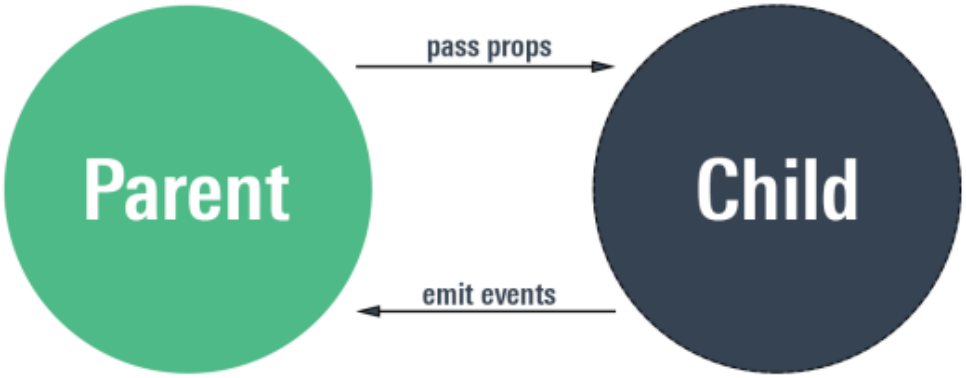

Figure 5 Props Flowchart

2.3.4 **State**

React have a build in state object assign to each component storing the data an information of a component, when the state changes the component will be rerender to adjust with the new UI changes. Moreover, state can also help pass information between layer in the react tree hierarchy, this mean that it is able to pass down the information from a component to it children using props, or lift the state up to it ancestor so that it dependencies can affect it in the desirable way.

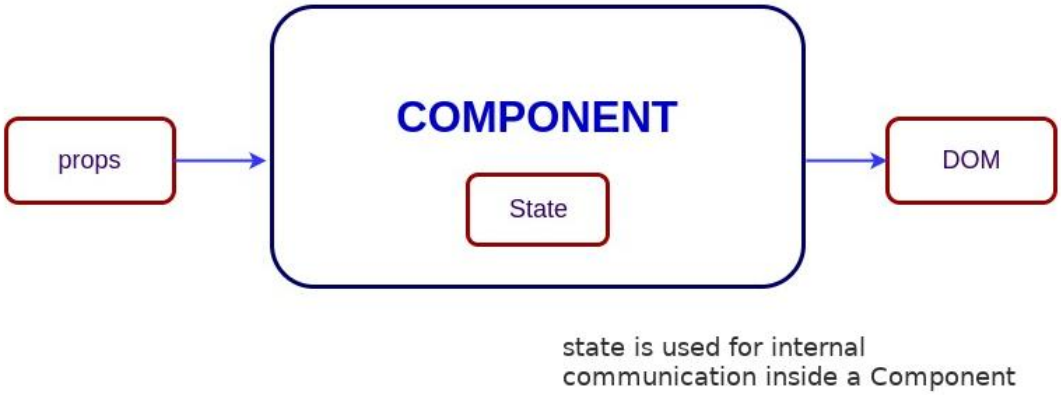

Figure 6 State Flowchart

2.3.5 **Prime React**

Prime React is an open-source UI component for React containing over 80 components. Developers using prime React can choose a wide range of prebuild components and style to use in their project optimizing the time necessary to build a full-scale UI/UX interface. Moreover, Prime React is also customizable, you can incorporate a CSS library of your choice to enhance or change the theme of a Prime component. With these features in mind, our team has used Prime for over 50% of our components, this includes thinking like button, sidebar, avatars making our UI/UX development quicker and easier.

Figure 7 Prime React Logo

2.4 Asp.Net Core

Asp.Net Core is a cross-platform open-sources and top performance development framework for the .Net ecosystem. Enabling developers to build applications from IOT to websites. Asp.Net Core was first released in 2016 as the child of closed sources development Asp.Net. is built to run on many popular OS like window and Linux, with the added benefit of having the highest performance speed when compared to its rival Node.js and Java.

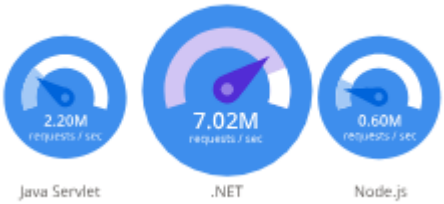

Figure 8 Performance Indicator

ASP.NET Core offers a wide range of features to streamline development. For our project, we utilized some of these features to enhance functionality and efficiency:

- Dependency injection
- Model-View-Controller
- Entity Framework
- Identity

2.4.1 Dependency injection

Dependency injection is a design pattern where a dependency does not need to create a dependent object to use it functionality instead it will call the object when needed. This design pattern is used to achieve Inversion of control between the classes and their dependencies making it satisfy the loosely couple principle when designing application.

2.4.2 Direct Dependency

Dependency inversion is a technique to combat direct dependency, take for example class A, B and C in the picture each class call the other class to use the objects' function or variable in the dependent class. A is dependent on B so it has to create a new Object B for it to use B function, B is dependent on C, so it has to create a new Object C to used it function and so on and so on. This is reflected in the run time chart where each class will be created linearly when call.

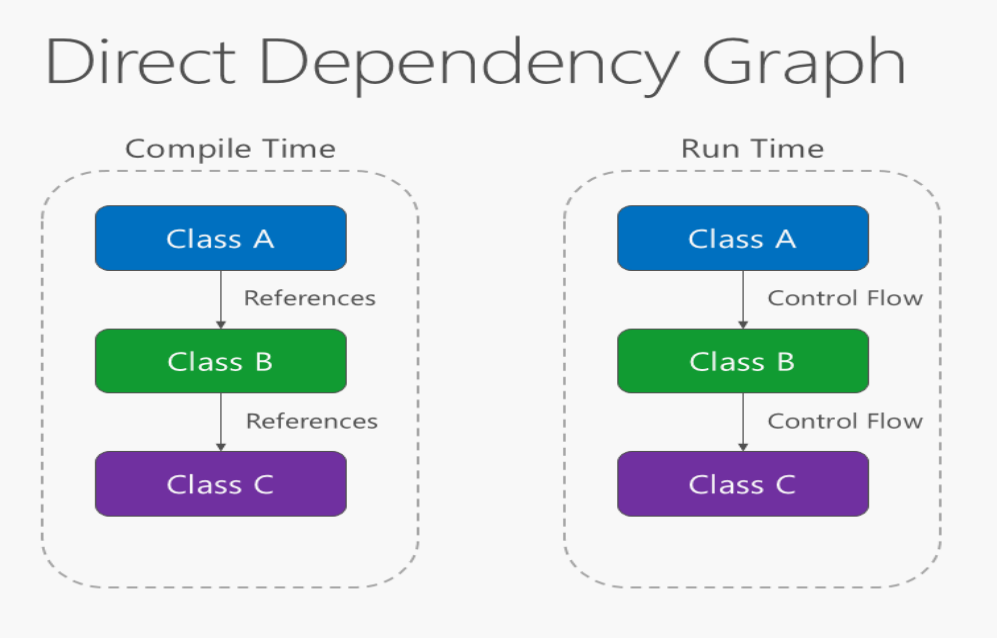

Figure 9 Dependency Graph

2.4.3 **Inverted Dependency**

Inverted dependency fixes this by instead of calling the class directly call the interface that implement this class instead, class A call the interface of class B, class B call the interface of class C and so on. Making the process of changing class more adaptable and easily customized than if we were to call the class directly.

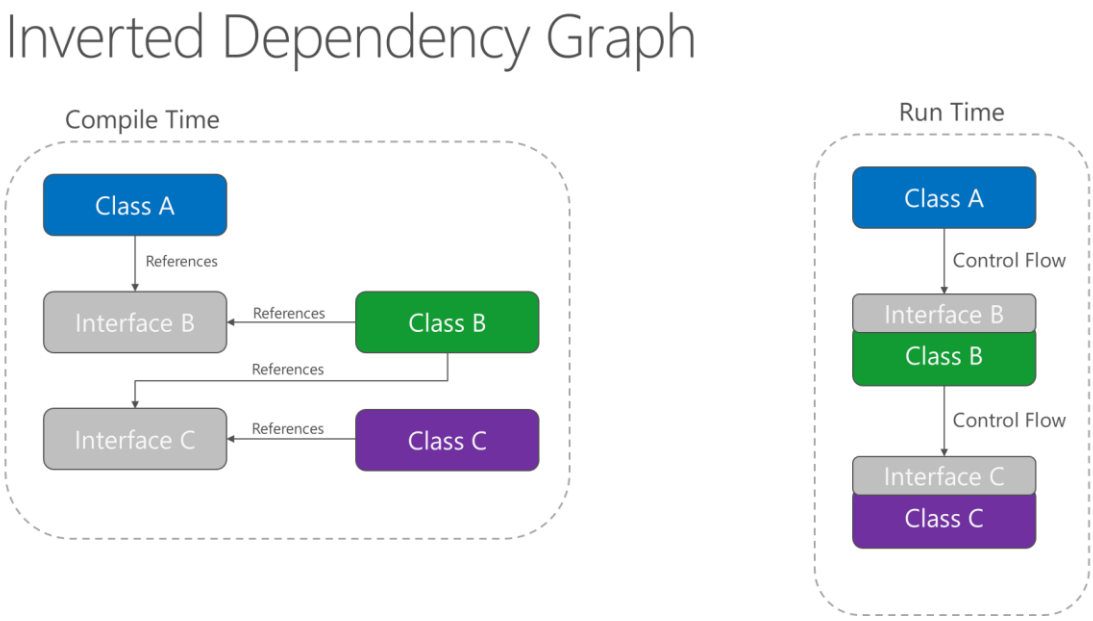

Figure 10 Inverted Dependency Graph

2.4.4 Model-View-Controller

Model-view-Controller is a software design patter to decouple user-interface into three main parts:

Model: for handling data and business logic

View: defines the UI or how the data should be present

Controller: used to control and deal with user command like forwarding command to the model and view handler

The pattern used “separation of concern” to provides for a better division of work and create clean code for maintenance. The flow of work goes like this the requests are routed to the Controller which will then work with the Model to perform an action or to retrieve data to be display to the View. In our application, we used the Model and Controller to define objects, mapped them to the database using Entity Framework for automatic creation, and managed them through routes and controllers.

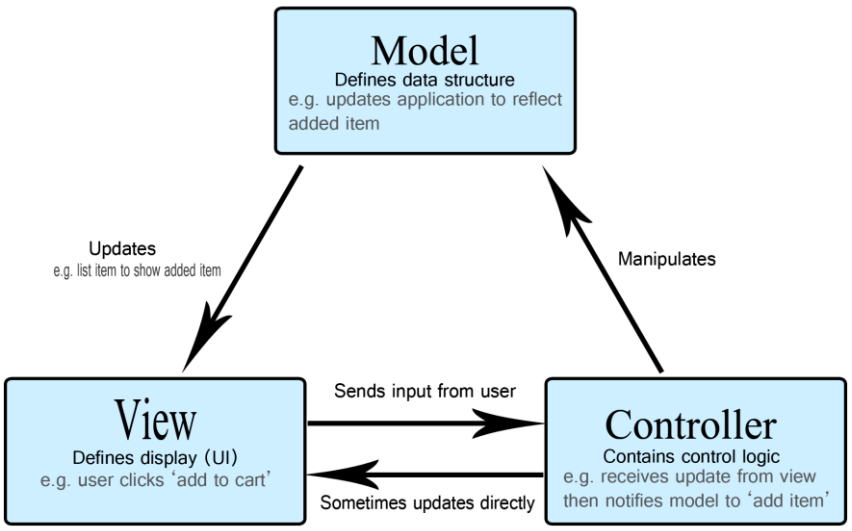

Figure 11 MVC Graph

2.4.5 Asp.Net Core Identity

Asp.Net Core Identity is a tag along membership system within Asp.Net Core that adds login features to your application. It manages services like user sign up, login, password renewal, Roles and claim-based authorization. The Roles and claim-based authorization is one of the main functionalities of the system dealing with authentication (checking the user’s identity) and authorization (controlling the user’s privilege). Identity configuration is usually with a

SQL database Server to store user info like name password and personal data. Moreover, identity is not contrast to SQL server database but can also be move to another persistent storage like Azure Table Storage

2.4.6 Asp.Net Core Jwt

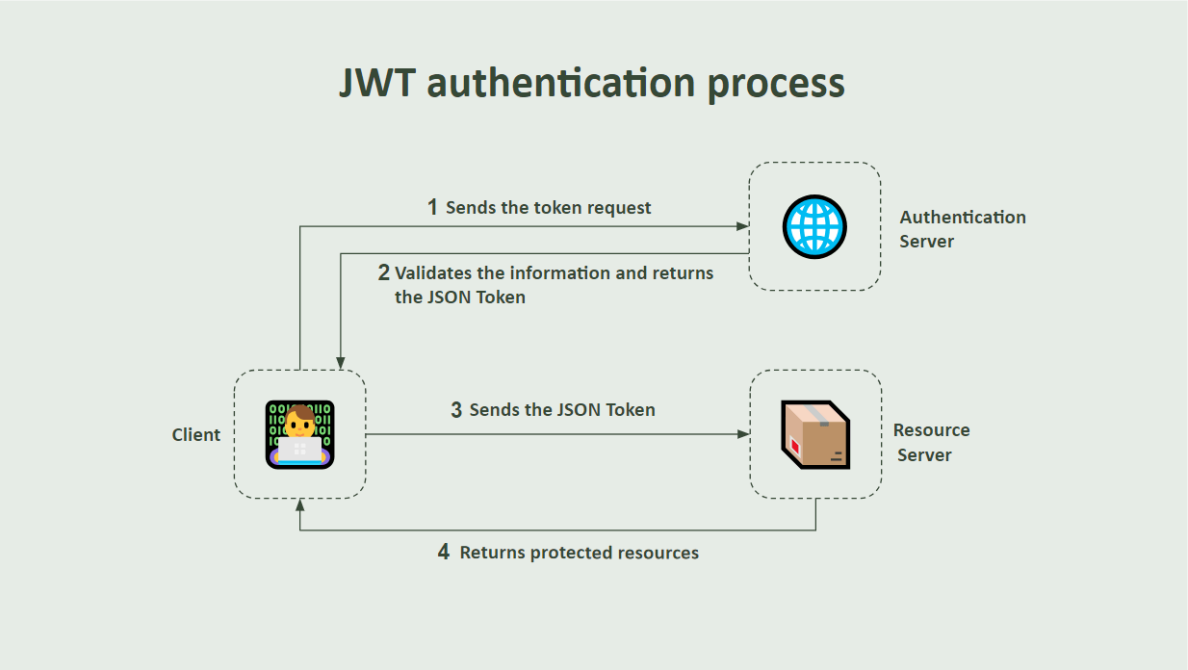

Figure 12 JWT Flowchart

Asp.Net Core provides a way to create a JWT token, a JWT token is a simple and secure practice for JSON object to be transmitted between different parties. Think of a token as a access key for the JSON object, the key is used first when encrypting the token, second when decrypting the token on the backend server. The JWT token is encrypted into 3 different sections split by a dot(.)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Figure 13 Token Image

Each part is encrypted, starting with the first red segment, which serves as the header. This section includes details such as the signature or algorithm used for the encryption process. The second part contains the payload, which holds the data within the JSON object. Finally, the third part is the secret key, used for both encryption and decryption.

When a user sends their JWT token containing their credentials to the backend server it will then be decrypted by the server secret revealing the payload content:

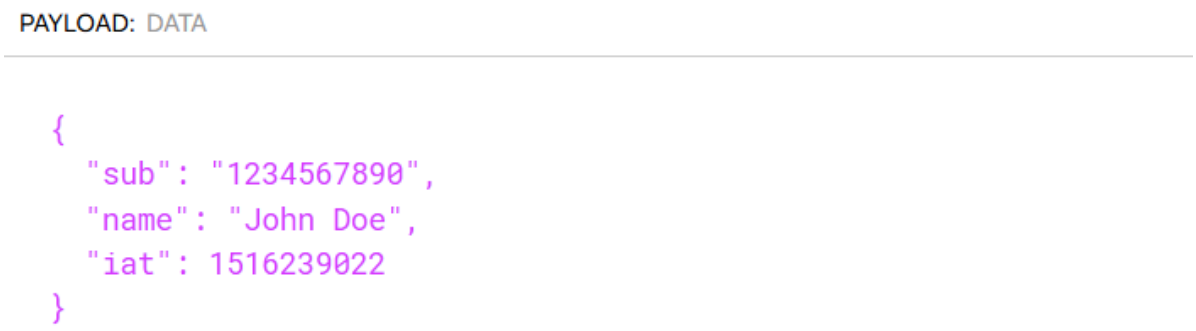

Figure 14 Body Image

Enabling the server to Authorize and Authenticate the user of the application using Asp.Net Identity Framework

2.5 Rest Api

Rest Api is the principle of web application development, following the principles REST architectural approach. Created by Roy Fielding in the year 2000, Rest Api quickly became popular because of its main benefit being its simplicity, stateless, scalability and lightweight. Client can interact with the backend sever thought http request to get return data in convenient format like JSON or XML, these methods include:
GET: get the requested data.

POST: create a new data object

PUT: update an existing data object

DELETE: Remove a data object

Figure 15 Rest Flowchart

2.6 SQL server management studio

Figure 16 SQL Server Management Logo

SQL server management studio was created by Microsoft in the year 2005 with the purpose of creating an integrated environment for handling any SQL infrastructure from SQL Server to cloud and local Azure SQL database. It allows developers to configure and manage the database engine, deploy, oversee, update components data-tier used by the application. SQL Server can also be used to build queries and scripts for designing and managing your data warehouse both local and on the cloud

2.7 WebSocket

WebSocket is a bidirectional, full duplex protocol. Bidirectional mean to allow communication from Two network devices, full duplex protocols mean that each devices will have 2 communication channels dedicated to receiving and sending information. Unlike it counterpart Http, WebSocket is a stateful protocol, which means that the connection will be maintain until terminated by either party and all session information will be store in the connection. Based on all these attributes, we can conclude that WebSocket is best used in real time communication like call or chat group, which requires a constant stream of data to be receive by both the client and server .As a result, we used WebSocket for the creation of our chatroom use cases ,which will allow the user to interact, chat and message with friend and love ones

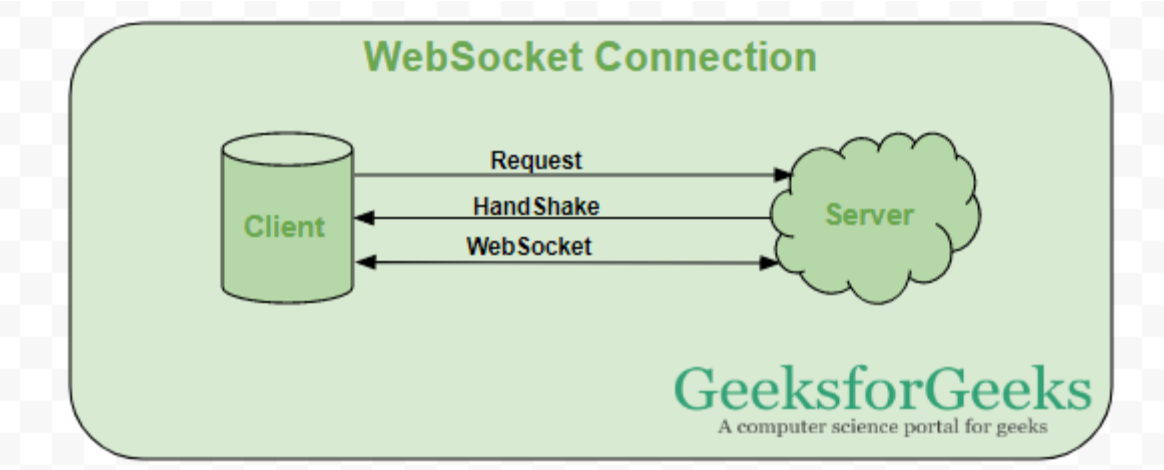

Figure 17 WebSocket Flowchart

WebSocket flow:

1. The client will initiate a request to the server.
2. The server will create a HandShake confirming and verifying the creation of a connection.
3. This connection will then be known as a WebSocket, which allow bidirectional and full duplex protocol
4. The WebSocket remains as a dedicated communication channel for this session, handling message request until either the server or client explicitly closes it or the connection from one of them end.
5. Status code 101 is used to notify a protocol switch to WebSocket.

2.8 SignalR

SignalR is an ASP.NET library that provides real-time web functionality for applications. This functionality allows the server to push contents to the client directly instead of waiting

for a response. As a result, this functionality helps with applications that require a constant stream of data like when user messages each other, refresh page or if a page implements long polling. This approach will decrease the amount of network traffic the server receives, decrease pooling time for receiving data and maintain a dedicated pipeline for request.

SignalR work by being an abstraction over some of the transport that is used to do real time communication between server and client. Real time communication is possible using a RCP API to invoke function in the client from the server. RCP support "server push" functionality, which means that changes that happened in the database to be push back to the client. Moreover ,SignalR supports the following transport technique for handling real time communication :

- WebSocket
- Server Sent Event
- Long Polling

But prefer long WebSocket due to it optimal transport due to it efficient used for server memory , lowest response rate and containing the most underlying features. Additionally, SignalR abstraction helps with maintaining version control and shields developers from compatibility issues with WebSocket, you do not need to worry about whether WebSocket update or the application using a different order version.

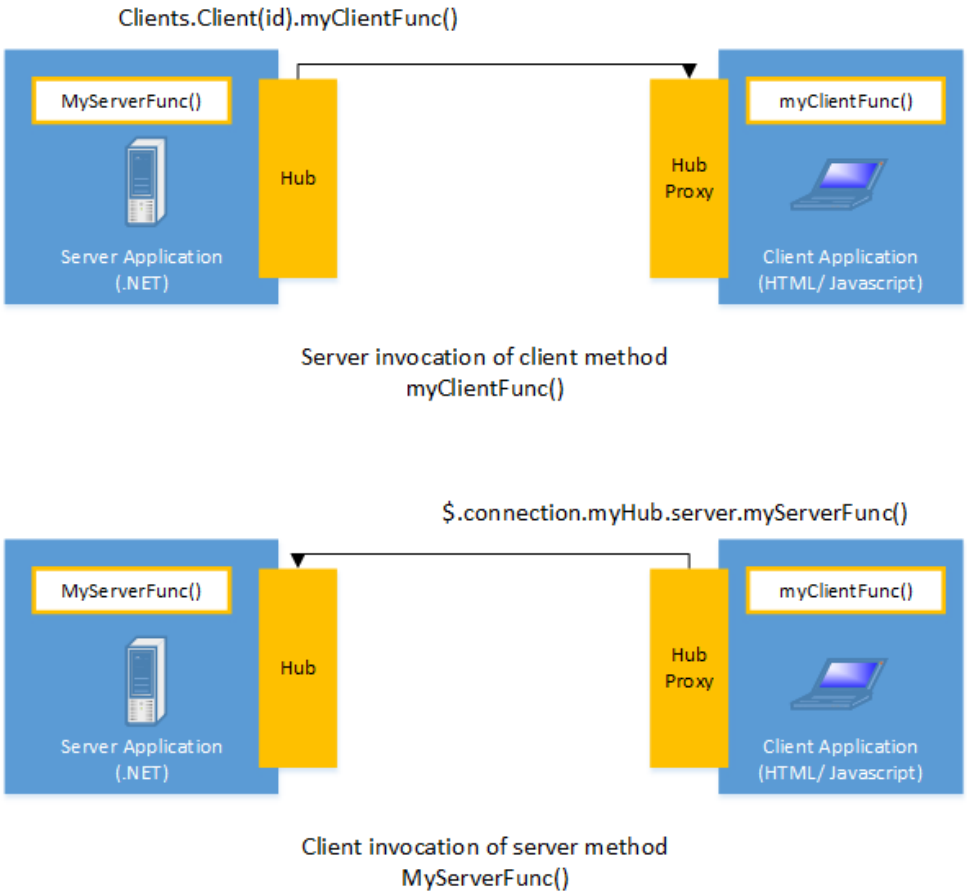

Figure 18 SignalR Flowchart

Signals used Hub to communicate between client and server. Hub is a high level pipeline that is established for client and server communication, Hub contain 2 build in protocol text bases JSON for messages and binary for binary file data. Each time a request is send signal will dispatch the request from hub between client and server. In our application, we use **Signal** to manage backend responses. Each time a chat group message is sent, a WebSocket is established for that specific chat group, enabling real-time communication.

Chapter 3 REQUIREMENT ANALYSIS

3.1 Specification requirement

The social media platform is customized to tailor to the custom and culture experience of Vietnamese user. The functionality of the application includes interconnected story telling through posts and smooth communication via chat messages in addition to a moderation system that detects and manages bad actors, ensuring a safely moderated system.

For user, the application is design to simplify sharing their experience thought content like post. On the other hand, for admins the application has given them powerful tools to efficiently moderately large volume of data, ensuring platform security and integrity.

The application is primarily image based, fostering visually engaging social media experiences inspired by its predecessor Instagram.

User Features

Account Management:

Personal profile:

- User information (name, bio, profile picture, etc.).
- Friends list.
- Follow/Unfollow system.
- Suggested friends.
- Account settings:
- Privacy settings (Public/Private profile).
- Active status visibility.
- Blocked users list.

Activity history:

- Saved posts.
- Comment and like history.
- Blocked accounts.
- Account recovery:
- Password reset via email.

Posts & Interactions:

Post creation:

- Upload photos/videos.

- Add captions.
- Tag friends.
- Post engagement:
- Like, comment, and share posts.
- View engagement details (likes, comments, shares, timestamps).

Post management:

- Edit captions, tags, and media.
- Delete or archive posts.
- Disable comments.
- Set privacy settings for posts.
- Report inappropriate content.

Search Functionality:

Search for:

- Users by name or profile.
- Posts by hashtags.

Friend & Follow System:

- Friend requests.
- Follow/unfollow users.
- Block/unblock users.

Messaging & Chat System:

- Private messaging.
- Group chat creation.
- Voice & video calls.
- Send media and files.

Chat settings:

- Notifications.
- Manage sent files, media, and posts.
- Block users in chat.
- Change chat themes

Admin & Moderator Feature

Admin System:

- Full control over the platform.

Manage:

- User accounts (edit, ban, delete).
- Posts (hide, delete).
- Group chats.
- Sub-admin roles.
- Assign and manage moderators.

Moderator System

- Limited administrative privileges.

Manage:

- User accounts.
- Posts.
- Group chats.
- Receive and review user reports.
- Keep a log of moderation actions.
- View statistics and analytics.

Reports & Moderation

- Receive user reports.
- Review and resolve complaints.
- Take actions (e.g., warn, ban, restrict users).
- Notify users of report outcomes.

Analytics & Insights

- Track and analyze:
- User growth and account creation.
- Post engagement and trends.
- Filter statistics by time or hashtag.

3.2 Database

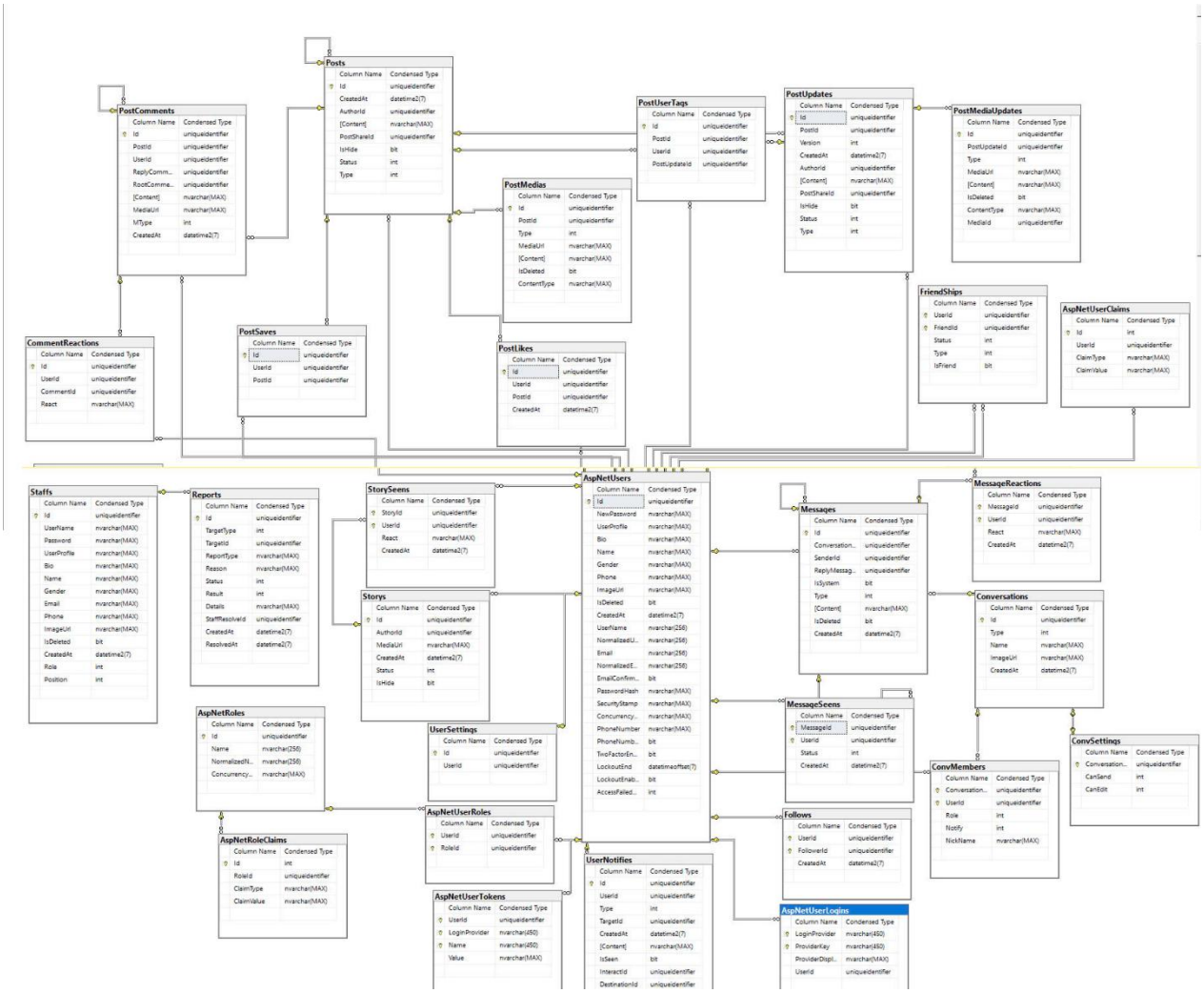

Figure 19 ERD Diagram

3.2.1 Post Relationship

PostComments

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** Each post can have multiple comments, and each comment belongs to a single post.

PostSaves

- **Relationship:** One-to-Many
- **Foreign Key:** PostId

- **Description:** A user can save multiple posts, and each saved post belongs to a single post.

PostTags

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** Each post can have multiple tags associated with it.

PostLikes

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** Each post can be liked by multiple users.

PostMedia

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** Each post can have multiple media files (images, videos, etc.).

PostUpdates

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** A post can have multiple updates tracking changes.

PostMediaUpdates

- **Relationship:** One-to-Many
- **Foreign Key:** PostId
- **Description:** Tracks updates made to media associated with posts.

AspNetUsers (Users Table)

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** Each post belongs to a single user, but a user can create multiple posts.

3.2.2 User Relationship

Posts

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can create multiple posts, but each post belongs to a single user.

PostComments

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can comment on multiple posts, but each comment is made by a single user.

PostLikes

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can like multiple posts, but each like is made by a single user.

PostSaves

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can save multiple posts, but each saved post is linked to a single user.

Friendships

- **Relationship:** One-to-Many (Self-Referencing)
- **Foreign Keys:** UserId, FriendId
- **Description:** Tracks friendships between users. Each user can have multiple friends.

Messages

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can send multiple messages.

Conversations

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can be part of multiple conversations.

MessageReactions

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user can react to multiple messages.

Follows

- **Relationship:** One-to-Many (Self-Referencing)
- **Foreign Keys:** UserId, FollowingId
- **Description:** A user can follow multiple users, and multiple users can follow a single user.

UserSettings

- **Relationship:** One-to-One
- **Foreign Key:** UserId
- **Description:** Each user has specific settings for their account.

UserNotifies

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** A user receives multiple notifications.

AspNetUserTokens

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** Tracks authentication tokens for user logins.

AspNetUserRoles

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** Tracks user roles and permissions.

AspNetUserClaims

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** Tracks claims assigned to users.

AspNetUserLogins

- **Relationship:** One-to-Many
- **Foreign Key:** UserId
- **Description:** Stores login credentials from different authentication providers.

3.3 Use case diagram.

3.3.1 Use case for Users.

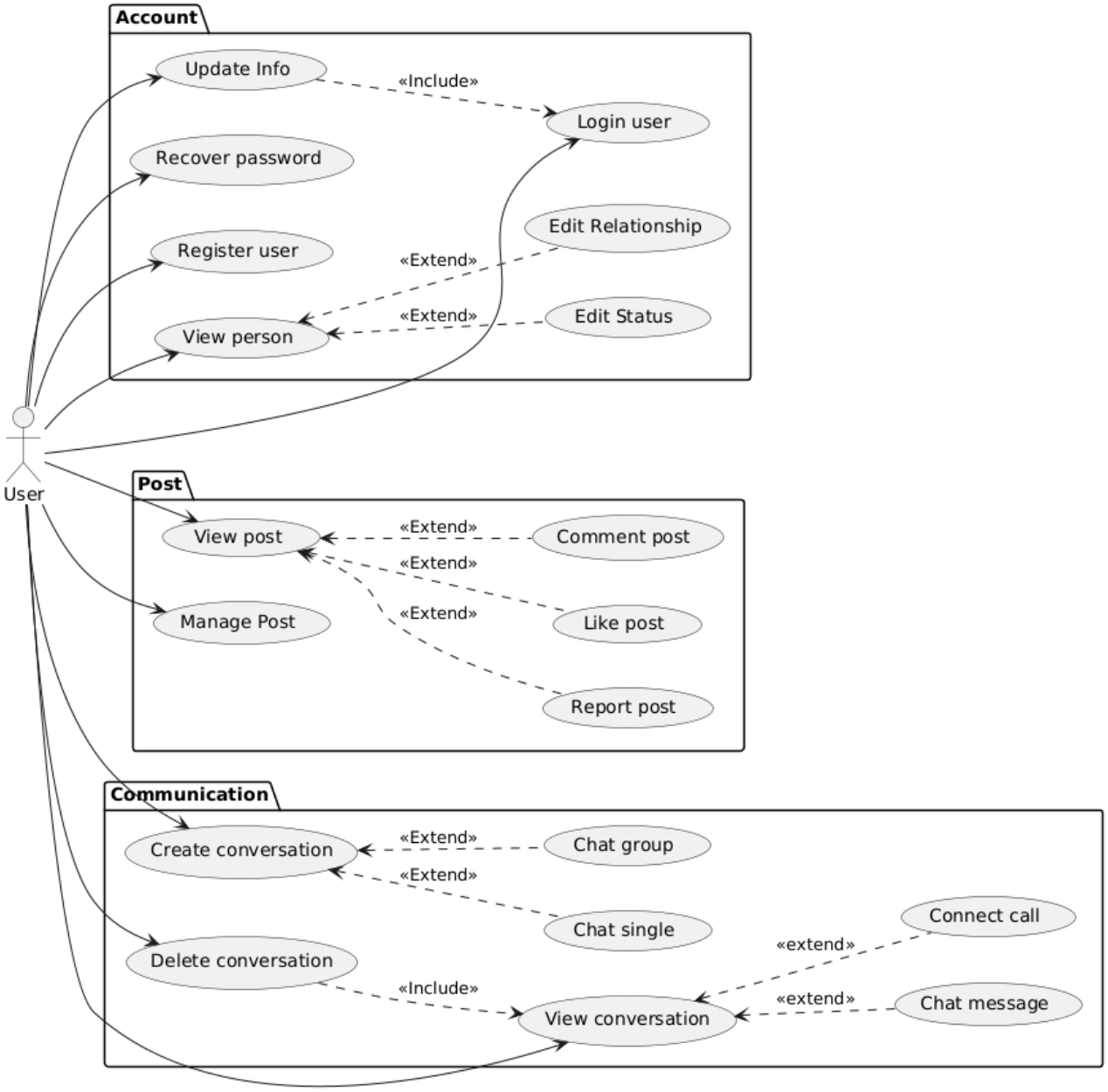

Figure 20 User Use Case Diagram

In the **Account Subsystem**, users can manage their own personal information through the functionality like " Register User" to create a new account , "Login User" to access their account functionality, "Recover Password" to restore their password and with it their account access, and "Update Infor" function which include a "Login User" call making sure the user authenticates before any changes can be made. User can also use the "View Person" to check another user's

profile, which extent to “Edit Relationship” to modify their relationship status and “Edit Status” to update their personal connection status.

In the **Post Subsystem**, User can interact with content by using the “View Post” function enabling them to browse shared stories, which extent into additional feature like “Comment Post to leave feedback and review, “Like Post” to express their interest in the content, and “Report Post” to notify the moderator of community guild line violation. The manage post functionality gives the user full control with their post like CRUD operation and so on

In the **Communication subsystem**, users can “Create Conversation” to initiate chats, which extends to “Chat Group” for group discussion and “Chat Single” for private one on one messaging. The “Delete Conversation” function allow user to remove chat along with all chat information like chat history, message, file and etc. “Chat Message” allow user to send and receive message vie text or if user want to have an advancement of this method, they can use the extended “Connect Call” instead.

3.3.2 Use case for admins.

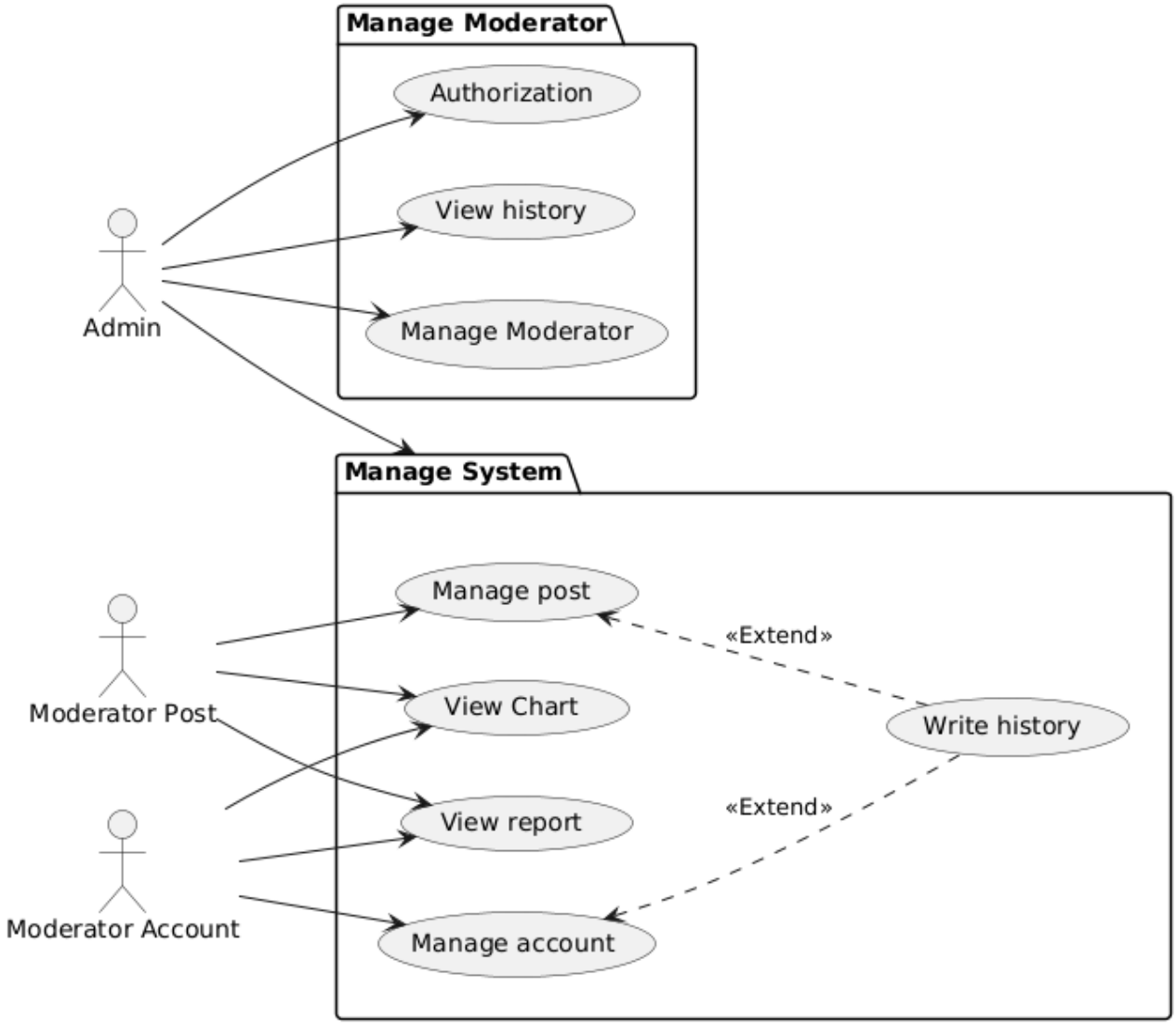

Figure 21 Admin and Moderator Use Case Diagram

The Manage Moderate subsystem. Allow admins the capability to “Authorization” granting permissions to different group moderator. “View History” Allows admins to review pass actions like actions of the user from their post, reported history or actions of the admins like their permissions changes and so on. “Manage Moderator” admins can oversee all moderators that is in the system enabling them to revoke permissions or change the permissions status of a moderator.

Manage System Subsystem. This subsystem is dedicated mostly to Moderator who can “Manage Post” which will extent “Write History” to write or log action taken on posts. View chart illustrated a visual diagram of activity like post interaction, number of like , number of

comment , the percentage of report and etc. “View Report” allows moderator to access reports extend “Write History” to see Logs report related interactions.

3.3.3 List Usecases In The System

ID	Name usecase	Description	Actor
User			
UC-1	Login	Login allow login with role as user or admin system.	User Admin
UC-2	Register user	User can register account to access application with basic info	User
UC-3	Recover password	User can get new password via email was register account	User
UC-4	Edit account	Edit info account and setting notification	User Admin
UC-5	Get notify	User can get notify about post, reply comment, result of report, from system	User
UC-6	Create post	User can create post with at least 1 media	
UC-7	Edit post	Can edit caption, user tag and media of user post	User
UC-8	Interact post	Person can like, comment, share, hide when view post	User
UC-9	Search post	User can search post by content	User

UC-10	Search account	Search account by user_profile or name	User
UC-11	Interact person	Can send make friend or unfriend, block/unblock with person	User
UC-12	Search conversation	User can search conversation by name group or through out find friend	User
UC-13	Create conversation	User can create group to chat or start chat with dicrect person	User
UC-14	Edit conversation	User can edit name, image of group with access from leader Leader can edit and setting permission	User Admin
UC-15	Send chat	User can send chat text/file or start call with dicrect chat or group chat	User
Admin			
UC-16	Manage Moderator	Admin can manage mod Authorization role for mod	Admin
UC-17	Get report	Admin and mod can receive correct report with their role from User	Admin Mod User
UC-18	Handle report	Admin and mod can resolve report and notice to User	Admin Mod User

UC-19	Manage account	Admin and Mod-Account can manage account as band, delete, limit comment of this account	Admin Mod-Account User
UC-20	Manage post	Admin and Mode-Post can manage post as hide, delete this post	Admin Mod-Post Post
UC-21	View chart	Admin and Mod can view chart about analysis account, post in duration	Admin Mod

3.3.4 Usecase Specification

Use Case: Login	ID: UC-1	Priority: High
Actor	User	
Description	Users can log in with their credentials to access the system.	
Pre-Condition	The user must have a valid account.	
Post-Condition	User is redirected to the home page.	
Basic Path	1. User enters username and password. 2. System validates credentials. 3. On success, the user is redirected to the home page.	
Alternative Paths	Invalid credentials: System shows an error message.	

Use Case: Register User	ID: UC-2	Priority: High
Actor	User	
Description	Users can create an account with basic information.	
Pre-Condition	None.	
Post-Condition	Account is created and confirmation email is sent.	
Basic Path	1. User fills in the registration form. 2. System validates the input. 3. On success, the system creates the account and sends a confirmation email.	
Alternative Paths	Invalid input: System highlights errors in the form	

Use Case: Recover Password	ID: UC-3	Priority: Medium
Actor	User	
Description	Users can reset their password via a registered email.	
Pre-Condition	The user must have a registered email.	
Post-Condition	Password is successfully reset.	
Basic Path	1. User requests a password reset. 2. System sends a reset link to the user's email. 3. User resets the password using the link.	
Alternative Paths	Invalid email: System shows an error message	

Use Case: Edit Account	ID: UC-4	Priority: Medium
Actor	User	
Description	Users can update personal information and settings.	
Pre-Condition	The user must be logged in.	
Post-Condition	Account information is updated.	
Basic Path	1. User accesses account settings. 2. User updates the information. 3. System saves changes.	
Alternative Paths	Invalid input: System shows validation errors.	

Use Case Get Notifications	ID: UC-5	Priority: Medium
Actor	User	
Description	Users can receive notifications about posts, comments, and reports.	
Pre-Condition	Notifications must exist for the user.	
Post-Condition	Notifications are displayed to the user.	
Basic Path	1. System send notify 2. User opens the notification panel. 3. System displays a list of notifications.	

Use Case: Create Post	IOD: UC-6	Priority: High
Actor	User	
Description	Users can create a new post with media and captions.	
Pre-Condition	User must be logged in.	
Post-Condition	Post is created and visible on the user’s profile.	
Basic Path	User uploads media and adds captions. User tags friends (optional). System saves the post.	
Alternative Paths	Invalid media format: System shows an error message.	

Use Case: Edit	Post ID: UC-7	Priority: Medium
Actor	User	
Description	Users can modify their posts.	
Pre-Condition	The post must exist, and the user must own it.	
Post-Condition	Post is updated with new content.	
Basic Path	1. User selects a post to edit. 2. User modifies captions, tags, or media. 3. System updates the post.	

Use Case: Interact with Post	ID: UC-8	Priority: Medium
Actor	User	
Description	Users can like, comment, save, or share posts.	
Pre-Condition	The post must exist.	
Post-Condition	Interaction is recorded and reflected on the post.	
Basic Path	1. User performs an action (like, comment, save, or share). 2. System updates the interaction.	

Use Case: Search Post	ID: UC-9	Priority: Medium
Actor	User	
Description	Users can search posts by content.	
Pre-Condition	Posts must exist.	
Post-Condition	Matching posts are displayed.	
Basic Path	1. User enters keywords. 2. System retrieves matching posts.	

Use Case: Search Account	ID: UC-10	Priority: Medium
Actor	User	
Description	Users can search for accounts by name or profile.	
Pre-Condition	Accounts must exist.	
Post-Condition	Matching accounts are displayed.	
Basic Path	1. User enters search criteria. 2. System retrieves matching accounts.	

Use Case: Interact with Person	IOD: UC-11	Priority: Medium
Actor	User	
Description	Users can send friend requests, unfriend, block, or unblock other users.	
Pre-Condition	Target account must exist.	
Post-Condition	Relationship status is updated.	
Basic Path	1. User navigates to the profile of another person. 2. User selects an interaction option (e.g., send friend request, block). 3. System updates the relationship status.	
Alternative Paths	Action fails due to system error: Display an error message.	

Use Case: Search Conversation	IOD: UC-12	Priority: Medium
Actor	User	
Description	Users can search for conversations by group name or by finding friends.	
Pre-Condition	User must logged and conversations must exist.	
Post-Condition	Matching conversations are displayed.	
Basic Path	1. User enters search criteria (e.g., group name or friend’s name). 2. System retrieves matching conversations.	
Alternative Paths	No matches found: Display a 'no results' message.	

Use Case: Create Conversation	IOD: UC-13	Priority: Medium
Actor	User	
Description	Users can create group chats or initiate direct messages.	
Pre-Condition	User must logged and make friend with person and person must exists	
Post-Condition	A new conversation is created.	
Basic Path	1. User selects participants from a list. 2. User optionally adds a group name and image or chat directly with person. 3. System creates the conversation.	
Alternative Paths	Invalid participants: Display an error message.	

Use Case: Edit Conversation	IOD: UC-14	Priority: Medium
Actor	User (Leader)	
Description	Group leaders can edit group names, images, and settings.	
Pre-Condition	User must be the group leader.	
Post-Condition	Group settings are updated.	
Basic Path	1. Leader navigates to the group settings. 2. Leader modifies settings (e.g., group name, permissions). 3. System saves changes.	
Alternative Paths	Unauthorized access: Display a permission error message.	

Use Case: Send Chat	IOD: UC-15	Priority: High
Actor	User	
Description	Users can send messages, files, and media in direct or group chats.	
Pre-Condition	The conversation must exist.	
Post-Condition	Message is delivered to the recipient(s).	
Basic Path	1. User composes a message or selects a file/media to send. 2. System delivers the message to the recipient(s).	
Alternative Paths	Message delivery fails: System displays an error message.	

Use Case: Manage Moderators	IOD: UC-16	Priority: High
Actor	Admin	
Description	Admins can manage moderator accounts and their permissions.	
Pre-Condition	Admin role is required.	
Post-Condition	Moderator roles and permissions are updated.	
Basic Path	1. Admin navigates to the moderator management page. 2. Admin assigns, edits, or revokes moderator roles. 3. System updates the changes.	
Alternative Paths	Invalid role assignment: Display an error message.	

Use Case: Get Reports	IOD: UC-17	Priority: Medium
Actor	Admin, Moderator	
Description	Admins and moderators receive reports from users based on their roles.	
Pre-Condition	Reports must exist in the system.	
Post-Condition	Reports are available for review.	
Basic Path	1. System sends relevant reports to the admin/moderator dashboard. 2. Admin/Moderator views the reports.	
Alternative Paths	No reports available: Display a 'no reports' message.	

Use Case: Handle Reports	IOD: UC-18	Priority: Medium
Actor	Admin, Moderator	
Description	Admins and moderators resolve user reports and notify users of the outcome.	
Pre-Condition	Reports must be assigned to the admin/moderator.	
Post-Condition	Report is resolved, and relevant users are notified.	
Basic Path	1. Admin/Moderator reviews the report. 2. Admin/Moderator takes appropriate action (e.g., warning, banning account). 3. System updates the report status and notifies the relevant users.	
Alternative Paths	Action fails: System displays an error message.	

Use Case: Manage Account	IOD: UC-19	Priority: High
Actor	Admin, Moderator (Account Role)	
Description	Admins and moderators manage user accounts by banning, deleting, or limiting comment permissions.	
Pre-Condition	Must have role admin or mod-account anduser account must exist.	
Post-Condition	Account status is updated.	
Basic Path	1. Admin/Moderator selects an account to manage. 2. Admin/Moderator performs an action (e.g., ban, delete, limit). 3. System updates the account status.	
Alternative Paths	Unauthorized access: Display a permission error message.	

Use Case: Manage Post	IOD: UC-20	Priority: High
Actor	Admin, Moderator (Post Role)	
Description	Admins and moderators manage posts by hiding or deleting them.	
Pre-Condition	Must have role Admin or mod-post and the post must exist.	
Post-Condition	Post status is updated.	
Basic Path	1. Admin/Moderator selects a post to manage. 2. Admin/Moderator performs an action (e.g., hide, delete). 3. System updates the post status.	
Alternative Paths	Unauthorized access: Display a permission error message.	

Use Case: View Chart	IOD: UC-21	Priority: Medium
Actor	Admin, Moderator	
Description	Admins and moderators can view analytical charts about accounts and posts over a selected duration.	
Pre-Condition	Must have role admin and mod. Analytical data must exist.	
Post-Condition	Charts are displayed.	
Basic Path	1. Admin/Moderator navigates to the analytics page. 2. Admin/Moderator selects the desired duration and filters (e.g., hashtags). 3. System generates and displays the charts.	
Alternative Paths	No data available: Display a 'no data' message.	

3.4 Activity Diagram

3.4.1 Post diagram

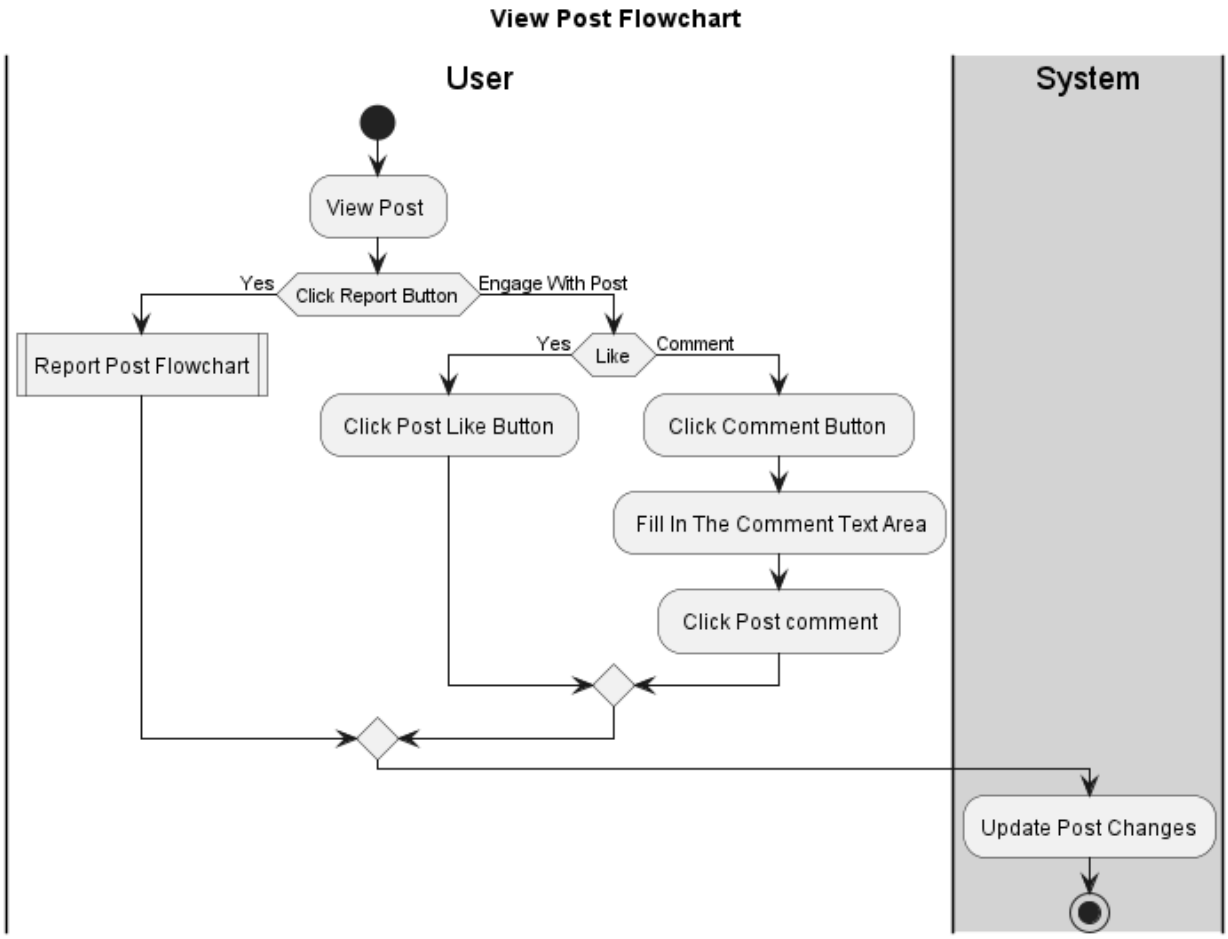

Figure 22 View Post Activity Diagram

When the users “View Post” they can choose to report the post by “Click Report Button” leading to the report post options. On the other hand, if the users chooses to not click the button, they will then be engaging with the post content, this can be a simple operation “Like” or a more complicated operation like “Comment, which will then promt the users to “Click Comment Button” then “Fill in The Comment Button” and finally “Click Post Comment”. In the final operation the System will then change the database bases on users posts interaction

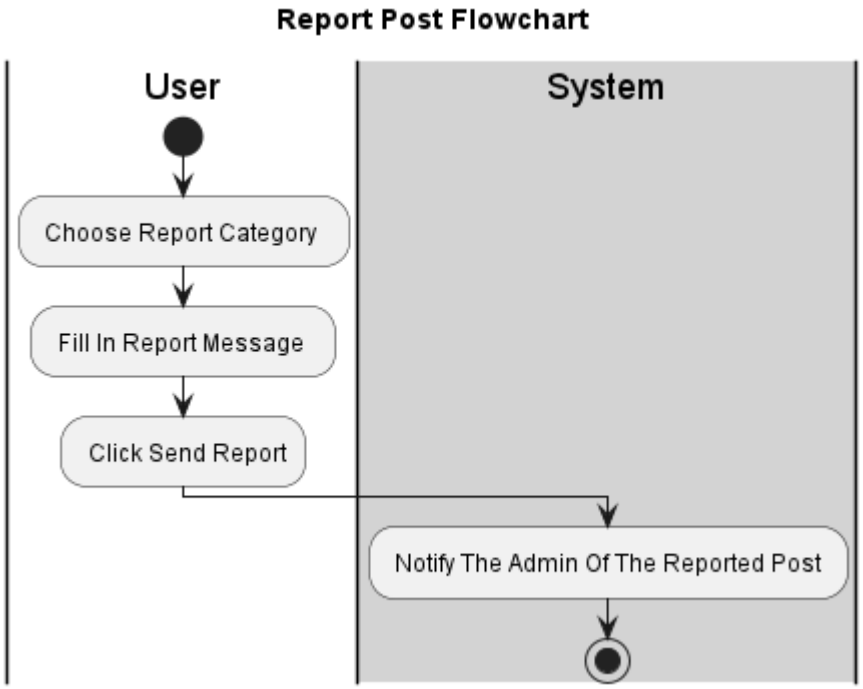

Figure 23 Report Post Activity Diagram

When users “Report Post” users need to fill out additional criterial like the category of the reported post, and the report message so that the Moderator will know the reason why the post was reported, finally “Click Send Report” to send the reported to Moderators. The system will notify the reported post the admin or moderator.

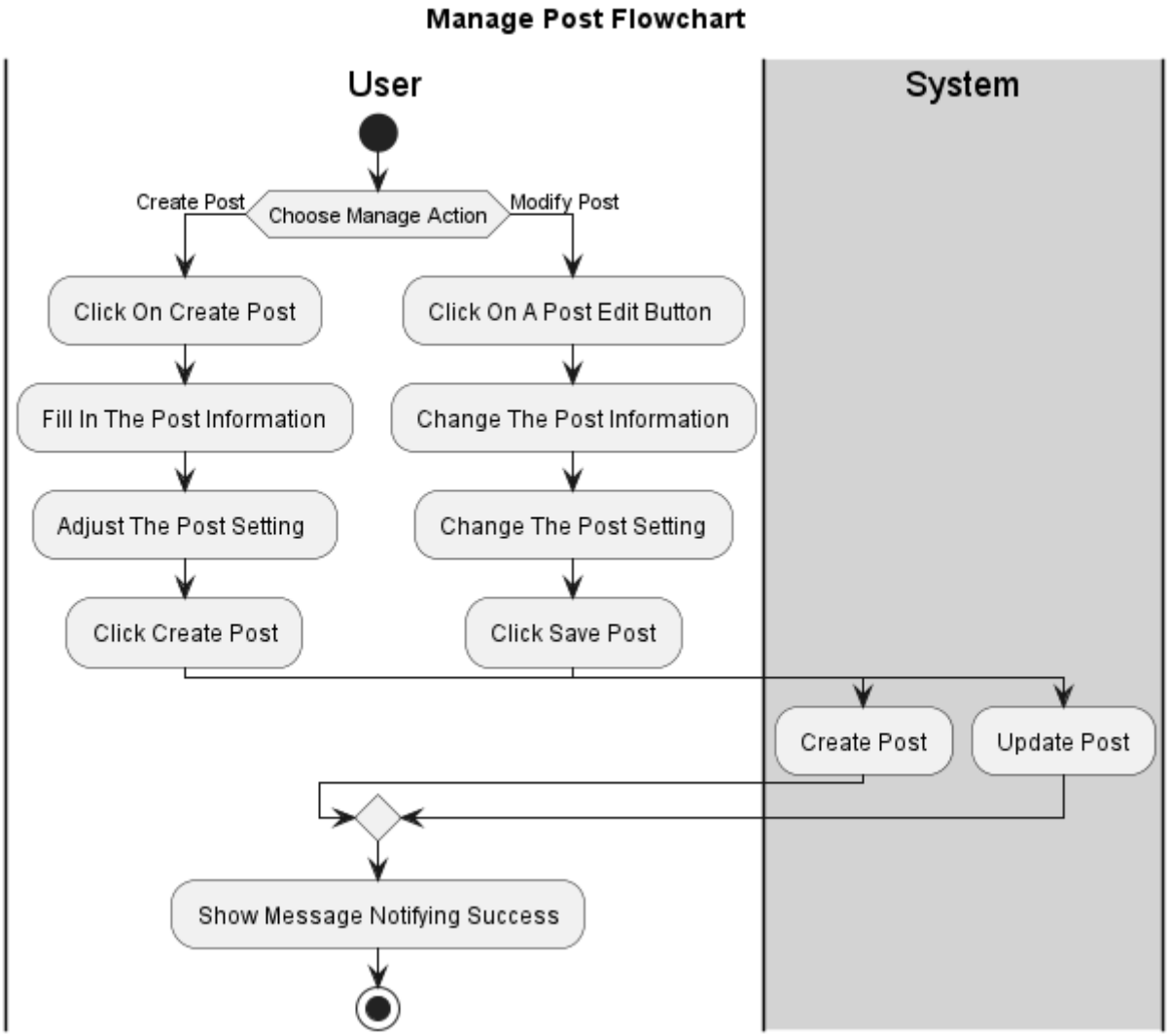

Figure 24 Manage Post Activity Diagram

When users “Manage Post” the user will have two main action path bases on the “Choose Manage Action” the “Create Post” action path and the “Modify Post” action path

In the “Create Post” action path the user will first “Click On Create Post” button then they will be prompted to “Fill Post Information” this can be anything from the description of the post to the location or friend that the user would like to add to the posts. After which, the user can “Adjust the Post Setting” this can decide who can see or interact with the post. Finally, the user can then “Click Create Post” to upload the post to their account handle by the system “Create Post” action.

In the “Modify Post” action path the users will “Click On A Post Edit Button” to edit their post by “Change The Post Information” ,“Change the Post Setting” and finally “Click save Post” to update to post handle by the system “Update Post” action.

Finally, the two paths will merge into a common step where the system processes the request and displays a success message notifying the user that their post has been created or updated successfully.E

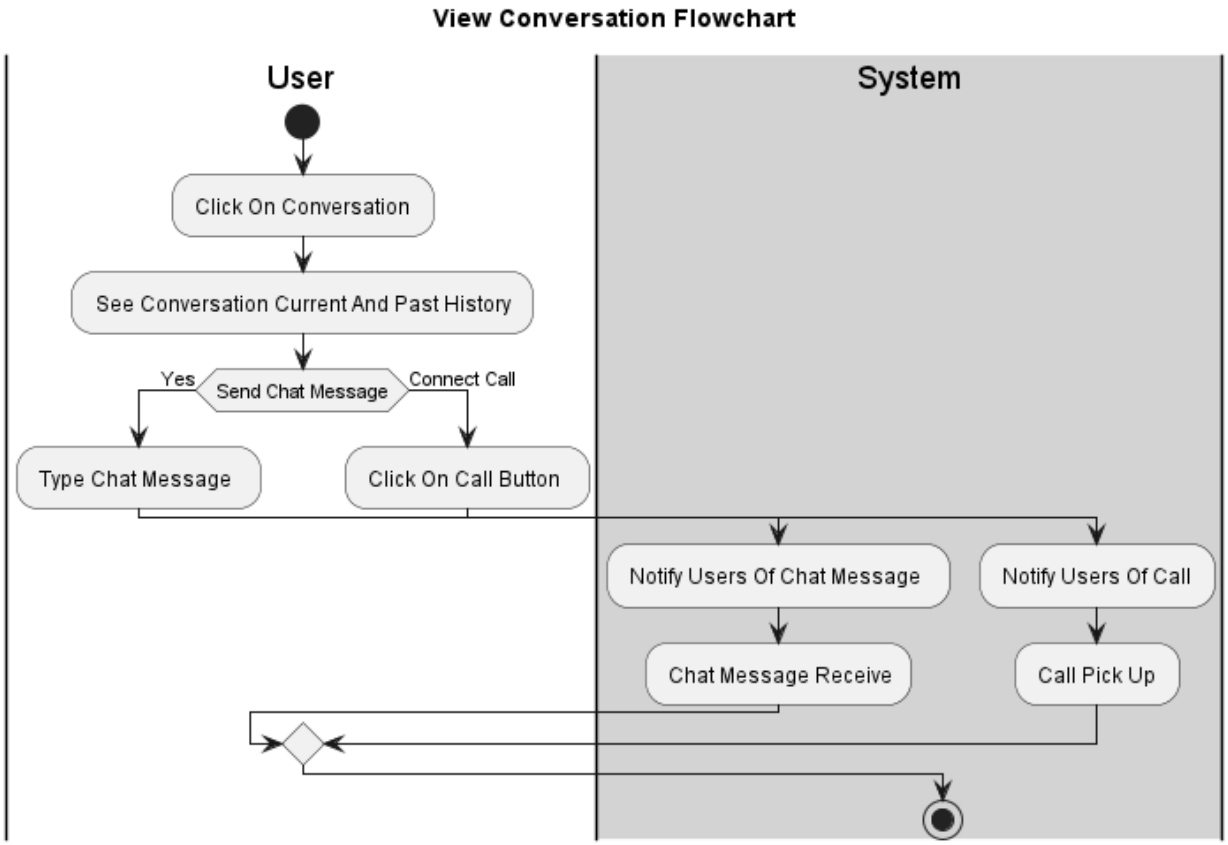

Figure 25 View Conversation Activity Diagram

When users “View Conversation” they see the conversation current and past history, from then on the user can used two action to communicate with other users in the communication channel. One is by “Connect Call” the other is by “Send Chat Message” both of these actions will notify other user vie the system and end when the “Chat Message Receive” or “Call Pick Up”

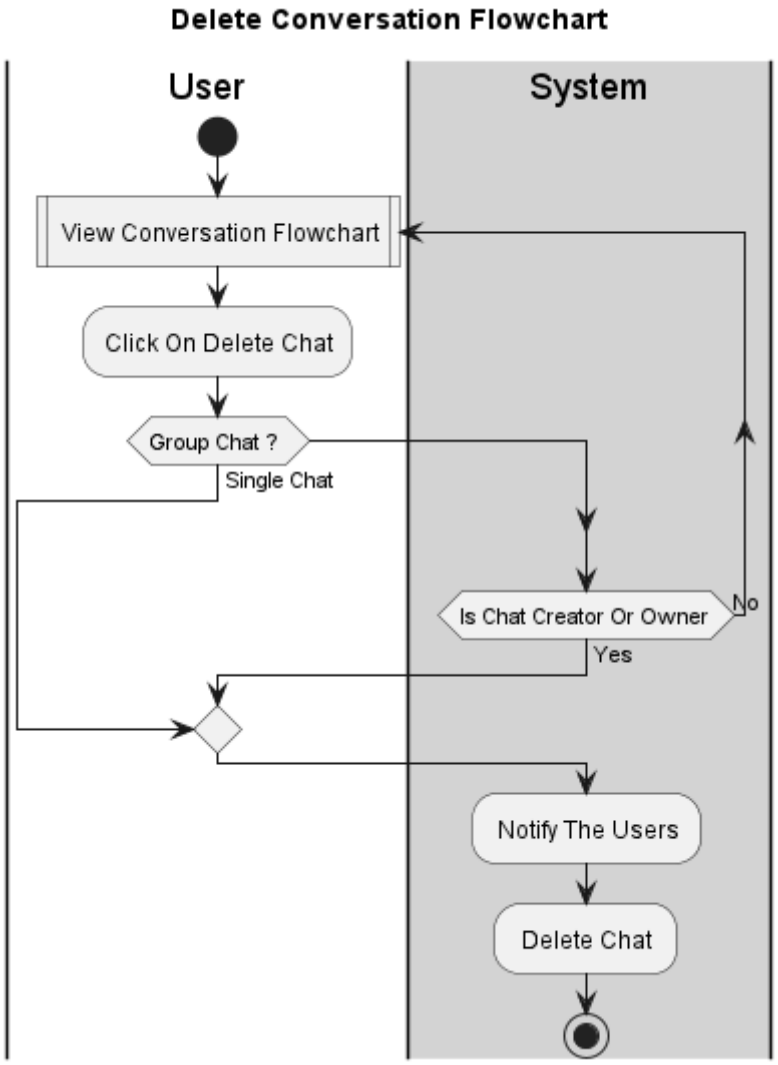

Figure 26 Delete Conversation Activity Diagram

When users “Delete Conversation” the users will need to “View Conversation Flowchart” first then “Click On Delete Chat” button if the chat is a group chat then the system need to check if the Users is the chat create or Owner , if the credentials for group chat is met or the chat is a “Single Chat” then all Users involves in the chat will be notify of the deletion

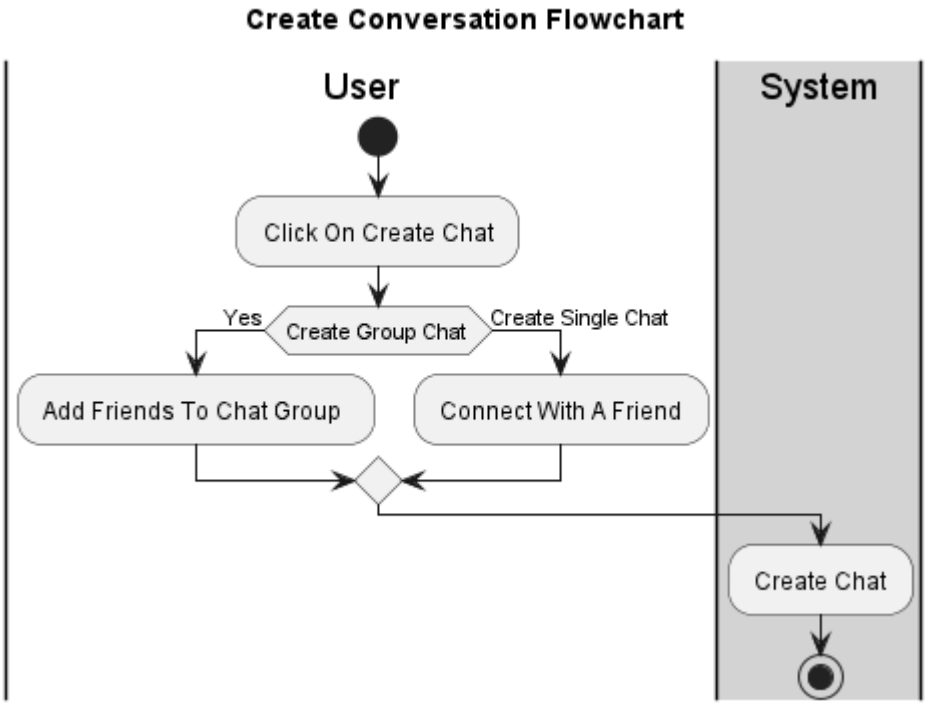

Figure 27Create Conversation Activity Diagram

When users “Create Conversation” users “Click On Create Chat” which prompt the user for two options, either “Create Group Chat “or “Create Single Chat”. For group chat users can “Add Friends to Chat Group” and for single chat users can “Connect With A Friend” both actions will merge into a single “Create Chat”

3.4.2 Account

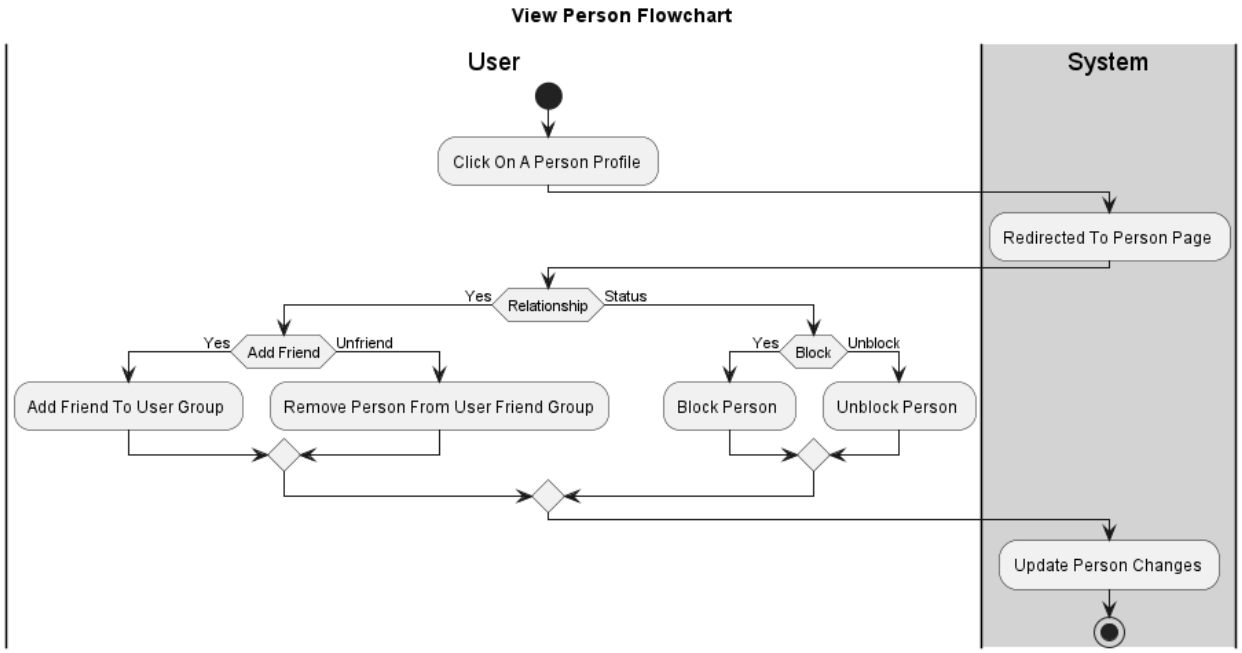

Figure 28 View Person Activity Diagram

When users “View Person” users “Click on A Person Profile” prompting the system to “Redirected to Person Page”. After being redirected to the person’s page, the user can choose between managing their relationship status or blocking/unblocking the person’s. When the users “Add Friend” or “Unfriend” will result in a change in the user’s friend group. In both cases, the changes will be reflected by the system vie the “Update Person Changes” action.

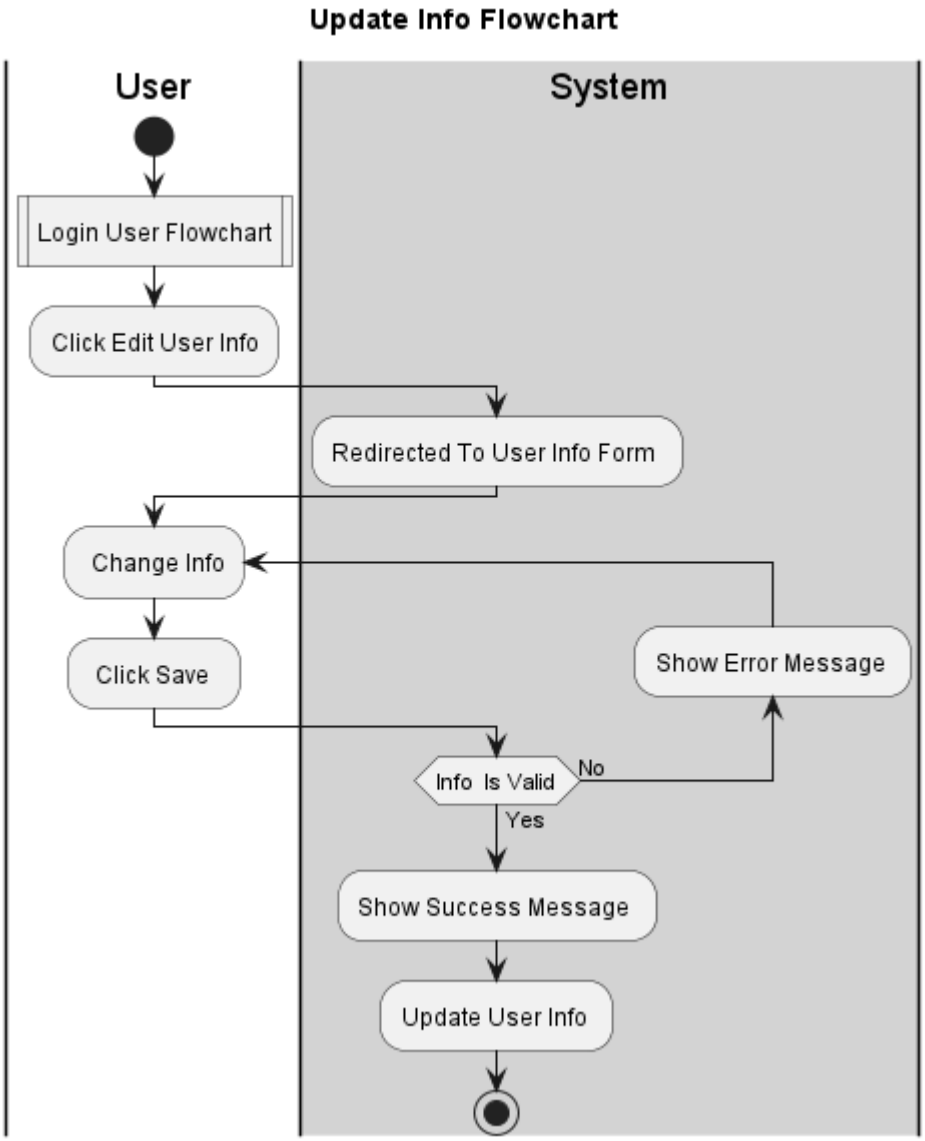

Figure 29 Update InfoActivity Diagram

When users “Update Info” users much first follow the “Login Users Flowchart” then “Click Edit User Infor” for the system to “Redirected To User Info Form. Here in the info form users can “Chane Info” like phone number name etc.” Click Save” , the systems will check if “Info is Valid” the system “Show Success Message “ and “Update User Info” else the system will notify by “Show Error Message”.

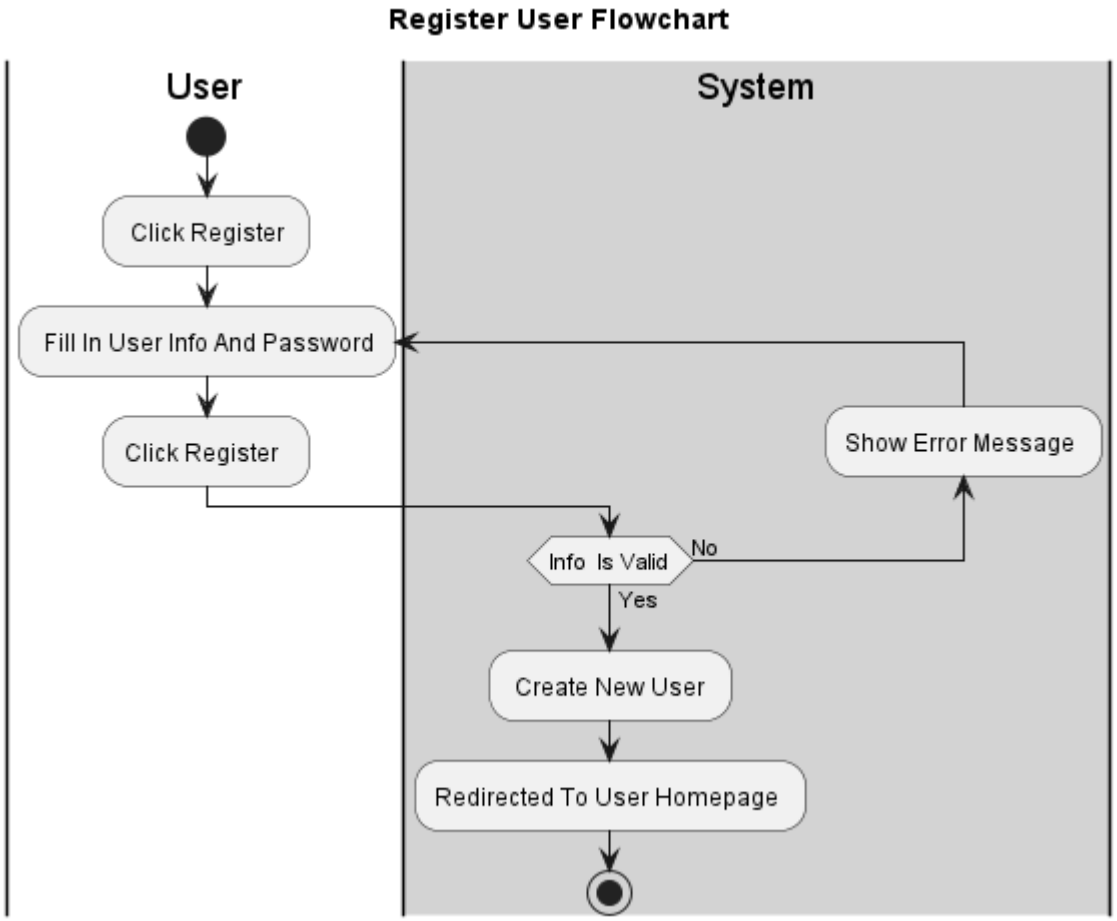

Figure 30Register User Activity Diagram

When users “Register”, they much “Click Register” to create a new account first then “Fill in User Info And Password”. After entering the necessary credentials, they will be prompted to “Click Register”. For the process to be successful the system check if the information provided is valid. If the “Info Is Valid” system will “Create New User” and finally “Redirected to User Homepage”. However, if the validation fail then the system will “Show Error Message”

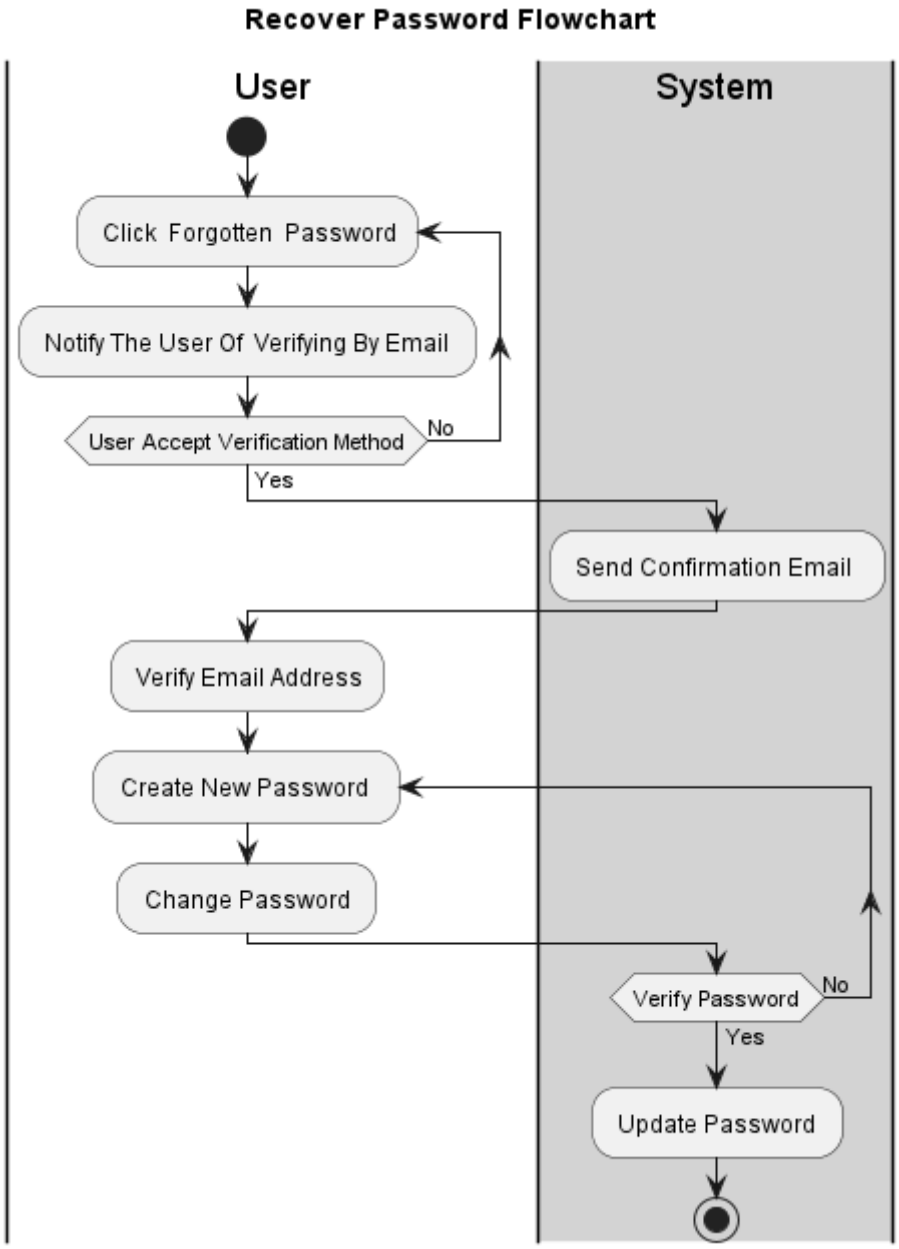

Figure 31 Recover Password Activity Diagram

When user “Click Forgotten Password” a prompt will be shown to the users “Notify the User of Verifying by Email” if “User Accept Verification Method” then a “Send Confirmation Email” action will be executed. The users will then go to their email inbox to “Verify Email Address” then redirected password creation site where they will “Create New Password” and finally “Change Password. The system will “Verify Password” then “Update Password” if successful, if the validation fails then the user much enter another password

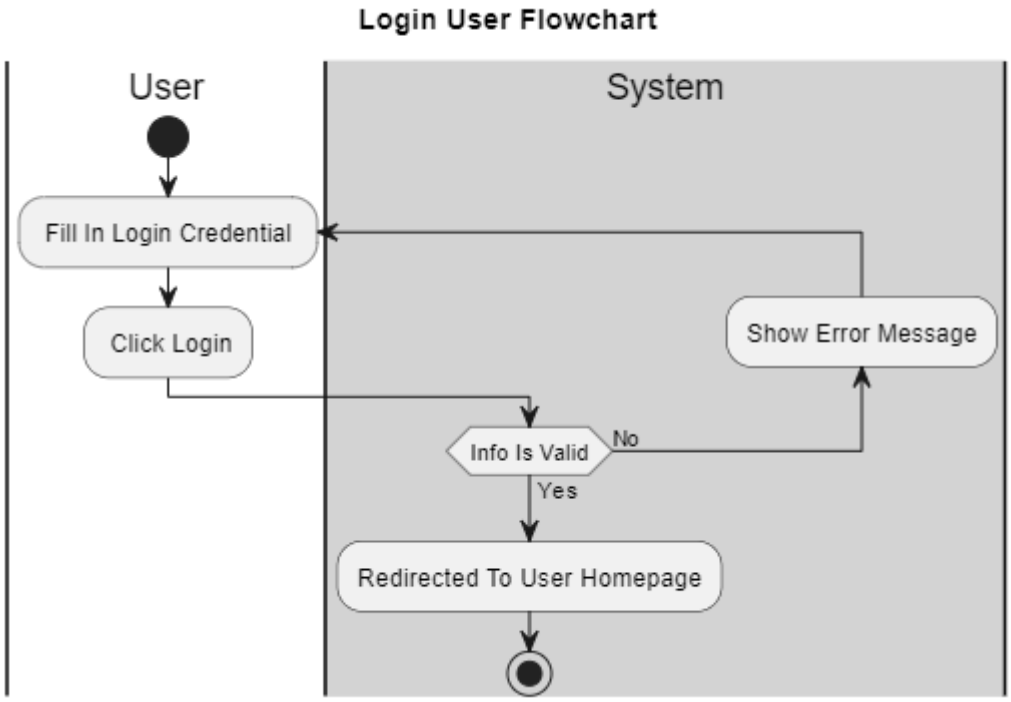

Figure 32 Login User Activity Diagram

When users “Login User” the user will need to “Fill In Login Credential” then “Click Login” the System will check if “Info is Valid” Then “Redirected To User Homepage” Prompt if the info is not the system will “Show Error Message” to prompt the user to re enter their credential

Chapter 4 Conclusion

4.1 Results archived.

Successfully developed the social media website: The website is integrated with complete CRUD functionality for post and user interaction. Users can create, read, update post while interacting with other users' post thoughts like and comment. In addition, the platform includes messaging features, allowing users to connect which each other in order to connect with loved one.

Successfully developed messaging functionality: user can text loved one with zero-time delay, create and manage their own chat group or message another user directly. In addition, users have diverse options in communication like messaging and voices call.

Successfully developed a moderation function: enabling admin and moderator to review reported posts and remove content that violates the community guidelines. User can report inappropriate posts, safeguarding social media environments, while moderators are equipped with the appropriate tools to deal with bad actors

4.2 Advantages and disadvantages

Advantages:

Easy To Use User Experience: the application have an easy to user interface, making it easy to navigate. Users can create post to share their story, send messages to friend and love one without the need of a guild or tutorial

Customizable post and interface: The application offers prominent level of customization, allowing users to personalize their post by text or pictures to express their feelings and interests. In addition, users can also choose to customize their own setting and information.

Diverse Messaging Capabilities: The application supports both texting and calling, enabling the user to message their friends and loved ones. Users have control over who they can chat with and can manage privacy settings, ensuring encrypted and secure conversations.

Disadvantages:

Limited Database Capacity: current the system database run on a local SQL server managed by SQL Server Management studio , so it cannot handle a large number of users

Security Concern: the application currently only used basic authentication method provided by ASP.NET , so it is in danger of high level attack that is not protected by the implemented frame work

Limited Functionality: the application currently only have basic social media network implementation like CRUD post and so on , it does not have more advance feature than the current social media app in the market

4.3 Difficult during work

Difficulty in managing the amount of work being distributed among teammate due to the vast function an overlap between function

Difficulty in managing timeline between teammates some functionalities arrive pass the decided time while other arrive too late to meet deadline.

Difficulty in managing the codebases and code quality , some code block need to be overwritten while other need to be completely dismantle in order to build the application

4.4 Future development

Enhance Database capacity: now the application could only handle at maximum 10 to 20 users. With added database capacity like AWS dynamo DB and Azura database we can scale our users bases more effectively

Enhance Security detection: security for the database and application will be enhanced to it fullest potential incorporating different frameworks and security prevention to protect the application

Optimized Performance and Load Handling: the application will be optimized to handle high traffic and concurrent users efficiently by implementing caching mechanisms, load balancing, and performance tuning, ensuring a smooth user experience even during peak usage times.

4.5 REFERENCES

- **ASP.NET Authentication**: O. (n.d.). Overview of ASP.NET Core Authentication. Microsoft Learn. Retrieved February 8, 2025, from https://learn.microsoft.com/en-us/aspnet/core/security/authentication/?view=aspnetcore-9.0

- **ASP.NET Entity Framework**: O. (n.d.). Entity Framework documentation hub. Microsoft Learn. Retrieved February 8, 2025, from https://learn.microsoft.com/en-us/ef/

- **SQL Server Management Studio**: O. (n.d.). Download SQL Server Management Studio (SSMS). Microsoft Learn. Retrieved February 8, 2025, from https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16