

## UE21CS351A: Database Management System

# MINI PROJECT USER REQUIREMENT SPECIFICATION

## Social Networking System

### Table of Contents

#### 1.Introduction

- The primary purpose of the project is to build a platform that helps people to connect. Users can find friends, chat with them and build communities based on common interests or causes. It facilitates user engagement, information sharing and relationship building.
- The project's scope encompasses the development of a social networking platform with core functionalities including user registration, profile management, friend connections, post creation, commenting, story sharing, and private/group messaging. The project excludes advanced features such as video calls and e-commerce functionality.

#### 2.Project Description

- Project overview - This project involves the development of a social networking platform whose primary objective is to create an environment for people to connect, communicate and engage with others.
- Major project functionalities - Major project functionalities include user registration and authentication, user profile configurations, connecting with friends, creating posts and commenting on them, creating stories, chatting privately or in a group and support for exporting chats.

#### 3.System Features and Function Requirements

- **User Registration:**  
Description: Provides user registration functionality.  
Functional Requirements: Allows users to sign up with a unique username, email, and secure password. Validates and stores user information securely, checking for duplicate usernames and emails.
- **User Login:**

Description: Authenticates users to access the platform.

Functional Requirements: Validates user credentials (email and password) against registered users to ensure secure login.

- **User Profile Management:**

Description: Allows users to create and customize their profiles.

Functional Requirements: Enables users to fill out profile information, including name, bio, picture, gender, and mobile number. Provides an option to set profiles as public or private. Tracks the number of connections (friends) a user has.

- **Friendship Management:**

Description: Enables users to connect with others.

Functional Requirements: Allows users to send and receive friend requests. The recipient can accept or decline friend requests. Users can chat with the recipient only if the friend request is accepted.

- **Post Creation and Management:**

Description: Allows users to create and manage posts.

Functional Requirements: Enables users to create posts with captions, pictures/videos, and optional music. Supports likes, comments, and editing of posts. Posts are visible only to a user's friends.

- **Commenting on Posts:**

Description: Enables users to comment on posts.

Functional Requirements: Allows users to add comments to posts, including text, likes, and timestamps.

- **Story Sharing:**

Description: Allows users to share temporary stories.

Functional Requirements: Enables users to post stories with pictures/videos, optional music, likes, and timestamps. Stories automatically expire after 24 hours.

- **Chatting:**

Description: Facilitates messaging between users and other users/groups.

Functional Requirements: Allows users to send messages to friends or groups, with content, timestamps, and group IDs where applicable. Supports both individual and group messaging.

- **Group Interaction:**

Description: Enables users to join and participate in groups.

Functional Requirements: Allows users to join groups through invitations or accepted requests. Manages group memberships, requests, and invitations.

- **Export chats**

Description: The Chat Export feature allows users to export their chat conversations for data backup, reference, or personal archiving.

Functional Requirements: Users shall have the ability to export individual chat conversations or group chats, selecting specific chats for export.

### **Functional Requirement:**

#### 1) **SignUp** -

**Inputs** - username, email and password

- A new user will have to SignUp to the platform to be able to use it.
- If the entered username and email already exists in the database records, the user will be asked to enter a different username and email.
- The password field will be encrypted before storing it onto the database.

#### 2) **Login** -

**Inputs** - email and password

- The entered credentials will be validated against the list of registered users.

#### 3) **User Profile**

**Inputs** – User\_ID, Username, Email, Password(encrypted), Name, Bio, Profile\_Pic, Gender, private\_or\_public.

- The users can set their profile in which they are expected to fill up some information about themselves.
- Users have a choice to set their account as public or private. If it is private then the user's friends are the only ones allowed to look at their posts.
- The Picture and Gender fields are optional.
- The username, email and password fields will be taken from the table holding user registration information.

#### 4) **Friend** -

**Inputs** - User1\_ID, User2\_ID, Time, Status

- Users are allowed to connect with others.
- The first user sends a friend request to the second user.

- The second user gets notified about the friend request from user 1.
- The second user is allowed to accept or decline the friend request. The status field tracks whether the request has been accepted or declined.
- The two users can chat only if they have accepted each other's friend requests.

#### 5) **Post** -

**Inputs** - User\_ID, Post\_ID, Caption, Picture/Video(source), music, Likes, Time.

- The users can create posts. The User\_ID foreign key indicates which user has made the post.
- Each post has a uniquely generated post\_ID. It must also have a caption, a picture or a video. If the post has a picture the user is allowed to add music.
- Each post also has comments and likes under it.
- The users can change the contents of their posts .
- The posts are only visible to the user's friends.

#### 6) **Comments** -

**Inputs** - User\_ID, Post\_ID, Text, Likes, Time.

- Users can comment under any post.
- Each comment has a POST\_ID as a foreign key to indicate the post under which the comment has been made and also a User\_ID field to know the user who has made that comment.
- Comments must mandatorily have a message text. Users can also like the comments under a post.

#### 7) **Stories** -

**Inputs** - User\_ID, Picture/Video(Source), Music, Likes, Time.

- Users can post pictures or videos called stories that automatically disappear after 24 hours.
- Similar to posts, stories have a user\_ID as a foreign key .
- They can have a video or a picture. Users can add music to the story if it is a picture.
- Time field specifies the time at which the story is created. The story will automatically be deleted after 24 hours.

#### 8) **Chats** -

**Inputs** - User\_ID, Receiver\_ID, Message Content/Text, Time

- Users are allowed to send messages to their friends
- Each message has some content associated with it.
- Time at which the message is sent is also tracked.

9) **Group\_Chats**

**Inputs** – User\_ID, Group\_ID, Message Content/Text, Time

- Users can also send messages in the group chats.
- Users\_id, Group\_ID and time act as a composite primary key.
- Hence users can post more than a single message in the same group.

10) **Groups** - Users can be a part of any number of groups.

**Inputs** - User\_ID, Group\_ID, Request/Invitation to join the group.

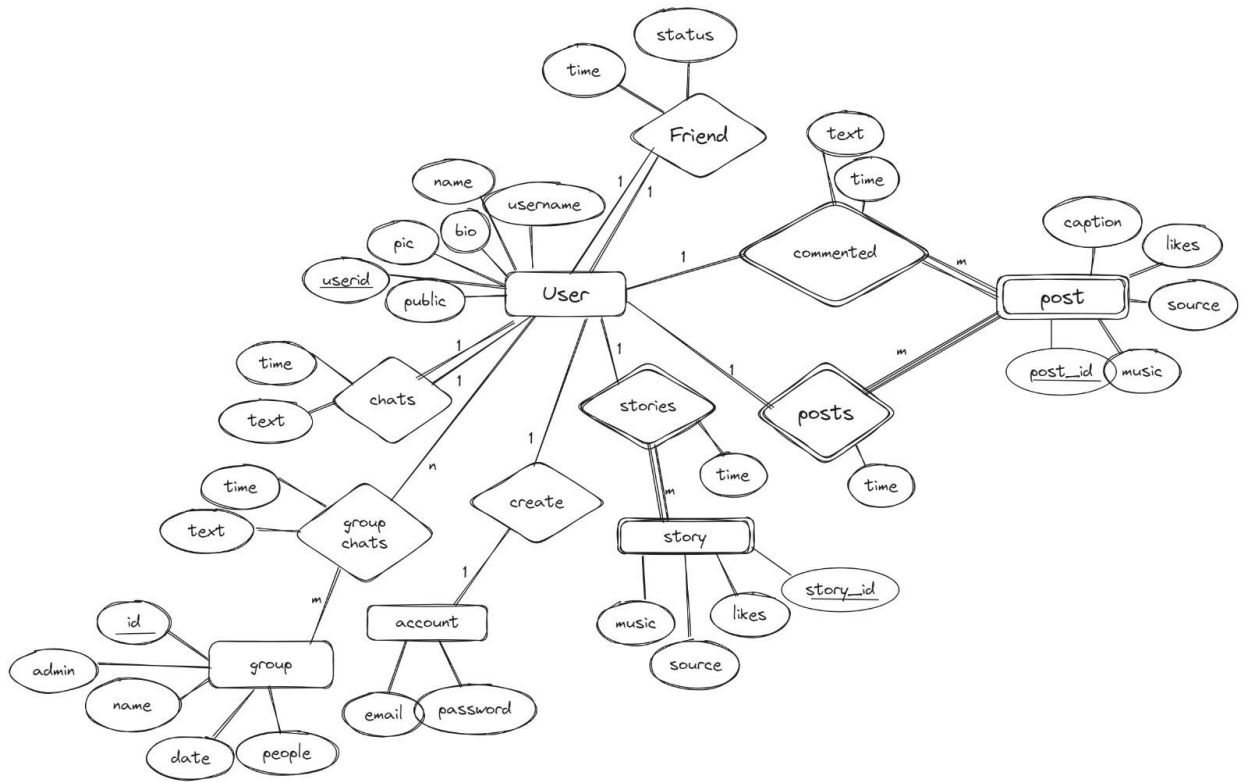
- Users can join a group if they have been sent an invite or if their request to join the group has been accepted.
- User\_ID, Group\_ID act as a composite foreign key here.
- Users can hence be a part of multiple groups.

11) **Export** - Users are allowed to export chats.

**Inputs** - User\_ID, Receiver\_ID, Message Content/Text.

- Users can export their messages by selecting them and clicking on the export option.
- Upon export the users will be notified about the exported chat file.

## ER Diagram



## Relational schema

