Project Document
Title: Social Networking System
1. Introduction
The Social Networking System is a C++ application that simulates basic social networking functionalities. It allows users to manage their profiles, establish friendships, send messages, and interact in groups. The system includes features for user authentication, message management, and group interactions.

2. Project Overview
The project comprises several classes and functions designed to handle user profiles, messages, and social groups. The main components are:

a)User Class: Manages user information and authentication.

b)Message Class: Handles the creation and display of messages.

c)SocialGroup Class: Manages group membership and message distribution within the group.

d)Global Functions: Provide functionality for adding users, managing friendships, and displaying information.

3. Class Definitions

3.1 User Class

Attributes:

~ firstName: User’s first name.
~lastName: User’s last name.
~age: User’s age.
~gender: User’s gender.
~contactNumber: User’s contact number.
~address: User’s address.
~email: User’s email.
~password: User’s password.
~hobbies: List of user’s hobbies.

Methods:

~Constructor: Initializes a user with given attributes.
~authenticate: Validates user credentials.

3.2 Message Class

Attributes:

~sender: Username of the message sender.
~receiver: Username of the message receiver.
~content: Content of the message.
~timestamp: Time when the message was sent.

Methods:

~Constructor: Initializes a message with sender, receiver, and content.
~displayMessage: Displays message details with timestamp.

3.3 SocialGroup Class

Attributes:

~groupName: Name of the social group.
~messages: List of messages in the group.
~members: Set of members in the group.
~Methods:

Constructor: Initializes a group with a name.

~addMember: Adds a user to the group.
~sendMessage: Sends a message to all members except the sender.
~addMessage: Adds a general message to the group.
~displayMessages: Displays all messages in the group.
~displayMembers: Displays all group members.

4. Global Variables and Functions

Global Variables:

~mapUserName: Map of usernames to User objects.
~friends: Map of usernames to their list of friends.

Functions:

~addUser: Adds a new user to the system.
~makeThemFriends: Establishes a friendship between two users.
~displayAllUsers: Displays all users in the system.
~displayAllFriendships: Displays all friendships.
~displayMutualFriends: Displays mutual friends between two users.
~authenticateUser: Authenticates a user based on username and password.

5. Main Function

The main function provides a menu-driven interface allowing users to:

~Add new users.
~Make friends.
~Display all users and friendships.
~Show mutual friends.
~Send messages.
~Add members to a group.
~Display group messages.
~Authenticate users.
~Exit the program.

6. Sample Output

Below is a sample output demonstrating the functionalities of the system:

mathematica
Copy code

```
1. Add User
2. Make Friends
3. Display All Users
4. Display All Friendships
```

```
5. Display Mutual Friends
6. Send Message
7. Add Group Member
8. Display Group Messages
9. Authenticate User
10. Exit

Enter your choice: 1
Enter username: alice
Enter first name: Alice
Enter last name: Smith
Enter age: 25
Enter gender: Female
Enter contact number: 1234567890
Enter address: 123 Elm Street
Enter email: alice@example.com
Enter password: password123
Enter hobbies (type ’end’ to stop):
Reading
Traveling
end
Username: alice, Name: Alice Smith, Age: 25, Gender: Female, Contact: 1234567890, Address: 123 Elm Street, Email: alice@example.com, Hobbies: Reading Traveling
```

7. Conclusion

This Social Networking System is a basic implementation of social networking features, providing functionalities for user management, messaging, and group interactions. It demonstrates fundamental principles of object-oriented programming in C++ and serves as a foundation for more advanced social network applications.

8. Future Enhancements

Enhanced Security: Implement password hashing and more secure authentication methods.
Database Integration: Store user data and messages in a database for persistence.
GUI Interface: Develop a graphical user interface for a more user-friendly experience.
Extended Features: Add functionalities like user profiles, photo sharing, and notifications.