Mini social network system in c++

```
#include <iostream>
#include <vector>
#include <string>
#include <map>

using namespace std;

class Post {
public:
    string content;
    string author;

    Post(string content, string author) : content(content), author(author) {}

    void display() const {
        cout << author << ": " << content << endl;
    }
};

class User {
public:
    string name;
    vector<User*> friends;  // Storing pointers to User objects
    vector<Post> posts;

    User(string name) : name(name) {}

    void addFriend(User* newFriend) {
        // Only add if the user is not already in the friends list
        for (size_t i = 0; i < friends.size(); i++) {
            if (friends[i] == newFriend) {
                return;  // Already friends, do nothing
            }
        }
        friends.push_back(newFriend);
    }

    void createPost(const string& content) {
        Post newPost(content, name);
        posts.push_back(newPost);
    }
```

```
    void showPosts() const {
        cout << name << "'s Posts:" << endl;
        for (size_t i = 0; i < posts.size(); i++) {
            posts[i].display();
        }
    }

    void showFriends() const {
        cout << name << "'s Friends:" << endl;
        if (friends.empty()) {
            cout << name << " has no friends yet." << endl;
        } else {
            for (size_t i = 0; i < friends.size(); i++) {
                cout << "- " << friends[i]->name << endl;
            }
        }
    }
};

class SocialNetwork {
public:
    map<string, User*> users;

    void addUser(const string& userName) {
        if (users.find(userName) == users.end()) {
            users[userName] = new User(userName);
            cout << userName << " has joined the network." << endl;
        } else {
            cout << userName << " already exists in the network." << endl;
        }
    }

    void addFriendship(const string& userName1, const string& userName2) {
        if (users.find(userName1) != users.end() && users.find(userName2) != users.end()) {
            User* user1 = users[userName1];
            User* user2 = users[userName2];

            // Add each user to the other's friend list
            user1->addFriend(user2);
            user2->addFriend(user1);

            cout << userName1 << " and " << userName2 << " are now friends." << endl;
        } else {
```

```
            cout << "One or both users do not exist." << endl;
        }
    }

    void createPost(const string& userName, const string& content) {
        if (users.find(userName) != users.end()) {
            users[userName]->createPost(content);
            cout << userName << " posted: " << content << endl;
        } else {
            cout << userName << " does not exist in the network." << endl;
        }
    }

    void showUserPosts(const string& userName) {
        if (users.find(userName) != users.end()) {
            users[userName]->showPosts();
        } else {
            cout << userName << " does not exist." << endl;
        }
    }

    void showUserFriends(const string& userName) {
        if (users.find(userName) != users.end()) {
            users[userName]->showFriends();
        } else {
            cout << userName << " does not exist." << endl;
        }
    }

    ~SocialNetwork() {
        for (map<string, User*>::iterator it = users.begin(); it != users.end(); ++it) {
            delete it->second;
        }
    }
};

// Function to display the main menu
void showMenu() {
    cout << "\n--- Social Network Menu ---\n";
    cout << "1. Add User\n";
    cout << "2. Add Friend\n";
    cout << "3. Create Post\n";
    cout << "4. Show User's Posts\n";
    cout << "5. Show User's Friends\n";
```

```
    cout << "6. Exit\n";
    cout << "Enter your choice: ";
}

int main() {
    SocialNetwork network;
    int choice;
    string userName, userName2, content;

    do {
        showMenu();
        cin >> choice;

        switch (choice) {
            case 1:
                // Add user
                cout << "Enter the name of the new user: ";
                cin >> userName;
                network.addUser(userName);
                break;

            case 2:
                // Add friendship
                cout << "Enter the name of the first user: ";
                cin >> userName;
                cout << "Enter the name of the second user: ";
                cin >> userName2;
                network.addFriendship(userName, userName2);
                break;

            case 3:
                // Create a post
                cout << "Enter the name of the user posting: ";
                cin >> userName;
                cout << "Enter the post content: ";
                cin.ignore();  // To clear the input buffer
                getline(cin, content);  // To get the entire line including spaces
                network.createPost(userName, content);
                break;

            case 4:
                // Show user's posts
                cout << "Enter the name of the user: ";
                cin >> userName;
```

```
            network.showUserPosts(userName);
            break;

        case 5:
            // Show user's friends
            cout << "Enter the name of the user: ";
            cin >> userName;
            network.showUserFriends(userName);
            break;

        case 6:
            // Exit
            cout << "Exiting the social network...\n";
            break;

        default:
            cout << "Invalid choice! Please choose a valid option.\n";
            break;
    }

  } while (choice != 6);

  return 0;
}
```