

➤ **As an unregistered user:**

- Register to the website using username, password, full address and name.

signature:

Name: signUp

Input: username varchar(30), user_password varchar(30), first_name varchar(30), last_name varchar(30), house_number int, postal_code int, street_name varchar(30), Output: errorBit bit

Output: errorBit bit

hint: success indicates that the username was not taken

➤ **As a registered user:**

- Login to the website using my username and password

signature:

Name: login

Input: username varchar(30), user_password varchar(30)

Output: errorBit bit

hint: success indicates that the user is found and the password matches with corresponding username.

- Update my name

signature:

Name: updateName

Input: member_id int, newFirstName varchar(30), newLastName varchar(30)

Output: nothing

- Update my password

signature:

Name: updatePassword

Input: member_id int, oldPassword varchar(30), newPassword varchar(30)

Output: error bit

hint: success indicates that the password matches with corresponding username.

- Viewing list of neighbours that are in the same house

signature:

Name: viewCloseNeighbours

Input: member_id int

Output: RECORDS

- Viewing list of neighbours that are in the same street

signature:

Name: viewStreetNeighbours

Input: member_id int

Output: RECORDS

- Send a friend request to a neighbour

signature:

Name: sendFriendRequest

Input: sender_id int, receiver_id int

Output: nothing

- Respond to a friend request received from a neighbour

signature:

Name: respondToFriendRequest

Input: sender_id int, receiver_id int, request_status varchar(30)

Output: nothing

- Review all friend request I sent to other neighbours

signature:

Name: viewRequestsSent

Input: member_id int

Output: RECORDS

- Review all friends requests received from other neighbours

signature:

Name: viewRequestsReceived

Input: member_id int

Output: RECORDS

- Preview the number of my friends

signature:

Name: numberOfFriends

Input: member_id int

Output: count int

- Create a new Event

signature:

Name: organizeEvent

Input: member_id int, eventName varchar(30), eventDescription text, is_public bool

Output: nothing

- View all private events I was invited to

signature:

Name: viewMyInvitedEvents

Input: member_id int

Output: RECORDS

- View all events I organized

signature:

Name: viewMyOrganizedEvents

Input: member_id int

Output: RECORDS

➤ **As a member invited to a private Event:**

- Respond to an event request sent to me

signature:

Name: resportToEventInvitation

Input: member_id int, event_id int, astatus varchar(30)

Output: nothing

➤ **As an event Creator:**

- Invite a member to an event I created

signature:

Name: inviteMember

Input: member_id INT, event_id INT, creator_id int, is_public bool

Output: errorBit bit

hint: success indicates that the event is private AND that the user that is trying to invite another member is the event creator AND the member that is to be invited is not already going to the event.

- Remove a member from an event I created or cancel an invitation request I sent to another member.

signature:

Name: removeMemberFromEvent

Input: member_id INT, event_id INT, creator_id int

Output: errorBit bit

hint: success indicates that the event is private AND that the user that is trying to invite another member is the event creator AND the member that is to be removed is already going to the event.

- Cancel an invitation I sent to another member

signature:

Name: cancellInvitation

Input: member_id INT, event_id INT, creator_id int

Output: errorBit bit

hint: success indicates that the event is private AND that the user that is trying to invite another member is the event creator AND the member that its invitation is to be cancelled is still pending.

- View all members I invited but they are still pending

signature:

Name: viewCreatorInvitedMembers

Input: organizer_id INT, event_id INT

Output: RECORDS

➤ **As a going member to an event:**

- View the going members to an event I am going

signature:

Name: viewGoingMembers

Input: member_id int, event_id int

Output: RECORDS

- Member can post news and concerns

signature:

Name: PostNews

Input: poster_id INT, content text, group_id INT

Output: errorBit bit

- Member can delete news and concerns of his own or in a group he is an admin in

signature:

Name: deletePostNews

Input: poster_id INT, post_id INT

Output: errorBit bit

- View neighbour news and concerns

signature:

Name: viewMemberNews

Input: subject_ID INT, object_ID INT

Output: RECORDS, errorBit bit

- View posts shared in a group

signature:

Name: viewGroupPosts

Input: viewer_ID INT, group_ID INT

Output: RECORDS, errorBit bit

- Comment on a post/concern

signature:

Name: commentOnNews

Input: member_ID INT, comment_content Text, post_ID INT

Output: errorBit bit

- like/unlike new/concerns

Signature: likeOnNews

Name: viewMyGoingEvents

Input: member_ID INT, post_ID INT

Output: errorBit bit

- Delete his/her own comment

signature:

Name: deleteCommentOnNews

Input: member_ID INT, _time DATETIME, post_ID INT, OUT

Output: errorBit bit

- Edit his/her own comment

signature:

Name: updateCommentOnNews

Input: member_ID INT, _time DATETIME, post_ID INT, content text

Output: errorBit bit

- Report another member

signature:

Name: reports

Input: subject_id INT, object_id INT, content TEXT

Output: errorBit bit

- block/unblock another member

signature:

Name: blockNeighbour

Input: subject_id INT, object_id INT

Output: errorBit bit

- Request an item

signature:

Name: requestItem

Input: member_id INT, supply_name VARCHAR(30), supply_description TEXT

Output: NOTHING

- Request a service

signature:

Name: requestService

Input: member_id INT, supply_name VARCHAR(30), supply_description TEXT, start_time DATETIME, end_time DATETIME

Output: NOTHING

- Make an offer on another member's request for a supply

signature:

Name: makeAnOffer

Input: supplier_id INT, requester_id INT, price INT, request_num INT

Output: NOTHING

- View supplies requested by a neighbour

signature:

Name: viewMemberRequestedSupplies

Input: subject_id INT, object_id INT

Output: RECORDS, error_bit BIT

- View their own requests for supplies

signature:

Name: viewMyRequests

Input: member_id INT

Output: RECORDS

- View Offers that a particular member has received

signature:

Name: viewOffersOnARequest

Input: subject_id INT, object_id INT, request_num INT

Output: RECORDS, error_bit BIT

- View All Offers they received on all supplies

signature:

Name: viewOffersOnMyRequests

Input: member_id INT

Output: RECORDS

- Search through requested supplies by name

signature:

Name: searchSuppliesByName

Input: member_id INT, search_term TEXT

Output: RECORDS

- Filter through requested services by time frame

signature:

Name: filterServiceByDuration

Input: member_id INT, search_term TEXT, start_date DATETIME, end_date DATETIME

Output: RECORDS

- Edit info for a posted request

signature:

Name: updateSupplyInfo

Input: new_name VARCHAR(30), new_description TEXT, request_num INT, editor_id INT

Output: RECORDS

- Delete their request for a particular supply

signature:

Name: deleteRequest

Input: request_num INT, editor_id INT

Output: error_bit BIT

- Accept or reject an offer on their request for a particular supply

signature:

Name: respondToOffer

Input: responder_id INT, requester_id INT, supplier_id INT, request_num INT, offer_response VARCHAR(30)

Output: error_bit BIT

- Rate another member
signature:
Name: rate
Input: subject_id INT, object_id INT, rate_value INT
Output: NOTHING
- Calculate rate of a particular member
signature:
Name: calculateRate
Input: member_id INT
Output: rating
- View their conversation with a particular member
signature:
Name: viewConversation
Input: subject_id INT, object_id INT
Output: RECORDS
- View list of chats
signature:
Name: viewChats
Input: member INT
Output: RECORDS
- Creating a group
signature:
Name: createGroup
Input: member_id INT, group_name VARCHAR(30)
Output: checkBIT BIT
- View members of the group
signature:
Name: viewGroupJoinedMembers
Input: member_id INT, group_id INT
Output: RECORDS, checkBIT BIT
- Leaving the group
signature:
Name: leaveGroup
Input: member_id INT, group_id INT
Output: checkBIT BIT
- View groups that the user is in
signature:
Name: viewMyGroups
Input: member_id INT
Output: RECORDS
- View mutual groups between 2 members

signature:

Name: viewMutualGroups

Input: member1_id INT, member2_id INT

Output: RECORDS, checkBIT BIT

- View invites to a group

signature:

Name: viewGroupInvitationRequests

Input: member_id INT

Output: RECORDS

- Responding to an invitation to a group

signature:

Name: respondToGroupInvitation

Input: member_id INT, group_id INT, response VARCHAR(30)

Output: checkBIT BIT

➤ **As a group admin:**

- Inviting someone to a group

signature:

Name: inviteToGroup

Input: admin_id INT, member_id INT, group_id INT

Output: RECORDS, checkBIT BIT

- Viewing invited members to a group

signature:

Name: viewInvited

Input: admin_id INT, group_id INT

Output: RECORDS, checkBIT BIT

- Removing a member from the group

signature:

Name: removeGroupMember

Input: admin_id INT, member_id INT, group_id INT

Output: checkBIT BIT

- Assigning a member as an admin

signature:

Name: assignGroupAdmin

Input: admin_id INT, member_id INT, group_id INT

Output: checkBIT BIT

- Delete a group

signature:

Name: deleteGroup

Input: member_id INT, group_id INT,

Output: checkBIT BIT

➤ **As a NSN administrator:**

- NSN administrator can ban a nsn member from joining the network

signature:

Name: AdministratorBan

Input: member_id INT

Output: Nothing

- NSN administrator can ban a nsn member from joining the network

signature:

Name: AdministratorUnBan

Input: member_id INT

Output: Nothing

- NSN administrator can view reports given for a specific member

signature:

Name: viewReport

Input: member_id INT

Output: records