Class FriendManagement Design Conception

Code is in Python3

Fields:

1. **friends\_dict**

this dictionary contains all the friend relations.

Each record is a user-friendList pair, “user”: [friend1, friend2, …], indicating that each user in the friend list is a friend of “user”

1. **subscribe**

this dictionary contains all the subscription relations

Each record is a requestor-targetList pair, “requestor”: [target1, target2, …], indicating that each target in targetlist is subscribed by this requestor

1. **block**

this dictionary contains all the block relations

Each record is a user-blockList pair, “user”: [blockUser1, blockUser2, …], indicating that each user in blocklist is blocked by this “user”

User Stories

1. As a user, I need an API to create a friend connection between two email addresses.

**Method name**: buildConenction

Condition 1: input is a dictionary, which contains key “friends” and corresponding value is a list with 2 users A and B

Condition 2: A and B didn't block each other

If both condition 1 and 2 satisfy, update ***friends\_dict*** (add B in the friend list of A, add A in the friend list of B)***,*** return true

Else return false

1. As a user, I need an API to retrieve the friends list for an email address.

**Method name**: retrieveFriends

Retrieve the friend list from ***friends\_dict*** based on the given email

If given email is not in ***friends\_dict*** or retrieved friend list is empty, return false.

1. As a user, I need an API to retrieve the common friends list between two email addresses.

**Method name**: retrieveCommonFriends

Refer to user story 2.

Return the intersection of friend lists of A and B

1. As a user, I need an API to subscribe to updates from an email address.

**Method name**: subscribeUpdate

Similar as user story 1, update ***subscribe*** for requestor

1. As a user, I need an API to block updates from an email address.

**Method name**: blockUpdates  
Firstly, update ***subscribe*** for requestor.

Secondly, remove target from friend list of requestor in ***friends\_dict***

Thirdly, remove remove target from friend list of requestor in ***subscribe***

1. As a user, I need an API to retrieve all email addresses that can receive updates from an email address.

**Method name**: retrieveRecipients

1. Sender’s friend

Check every sender’s friend, they are friends of each other, then this friend is a recipient

1. Those who subscribe sender

For every requestor in ***subscribe,*** if sender is one of the targets in the list, then this requestor is a recipient

For a) and b), don't have to retrieve ***block***relation, because story 5 will update the other two relation

1. Those who mentioned in text

For every mentioned user in text, if he/she doesn’t block sender, he/she should be added to recipient.

1. Elements in recipient list need to be unique