Assignment 2

Designing the Follows, Bookmarks, and Messages Restful Web Service APIs

Follows

Analysis: A new following is created for the current user. Follows here is a verb. Followers is also a list of users itself. This is a many to many relation that is each user can have many followers and each follower can be followed by many users. The crud operation here is to follow someone and not retrieve new information so it is a Create operation.
If the users wish to see everyone that are following them they can do so by retrieving a list of all followers. These followers are also just other users. This is a retrieve operation.
If a user decides to unfollow someone they are already following, then we need to decrease the following count by 1 and remove that user from the list.

Use case	Http method	Path	Request	Response
User ***follows*** another user	POST	/users/:uid1/follows/:uid2	Users primary key	New follow JSON
User ***retrieves*** users that are following them	GET	/users/:uid1/followers/	Users primary key	All followers in JSON array
User ***unfollows*** follower	DELETE	/users/:uid1/follows/:uid2	Users primary key	Delete status
User ***retrieves*** users they are following	GET	/users/:uid1/following	Users primary key	All following in JSON array
User ***unfollows*** all users in their list	DELETE	/users/:uid1/follows/	Users primary key	Delete all followers
User ***counts*** all follower in the list	GET	/users/:uid1/followers/:uid2	Users primary key	JSON of count of followers

Bookmarks

Analysis: A bookmark tuit cannot exist if the tuit itself does not exist. The bookmarks should be not visible to other users and is only associated with the current user's profile. If the tuit is deleted, then the bookmark also has to be deleted which means the bookmark depends on the

tuits. Since this is not a creation task, as the tuit already exists this is a Retrieval operation. This is again a many to many operations. One user can bookmark many tuits and each tuit can be bookmarked by many users.
A user can decide to remove a tuit from bookmarks. So they will not be able to see this tuit on their bookmarks page anymore so this is a delete operation. Tid: 633de4ab2ae68b2e1e0430e4
Uid : 633dde9d0b81d782233d080d
uid2:633ddee20b81d782233d0812

Use case	Http method	Path	Request	Response
User ***bookmarks*** a tuit	GET	/users/:uid/bookmark/:tid	Users primary key Tuit primary key	Bookmark JSON
User ***removes bookmark*** from list	DELETE	/users/:uid/bookmark/:tid	Users primary key Tuit primary key	Delete status
User ***views*** bookmarked tutis	GET	/users/:uid/bookmark	Users primary key	All bookmarks in JSON array
Retreive ***users*** that bookmarked a tuit	GET	/tuits/:tid/bookmark	Tuit primary key	Tuit json array
Get ***count of*** all bookmarks	GET	/tuits/:tid/bookmark	Tuit primary key	Tuit json array with count

Messages

Analysis: A user can send a message to another user provided that user exists. This will be a create operation since a user is creating a new message to another user. This is a many to many relation since one user can send messages to many users and a user can receive messages from many users.
A user can also view all the messages that has been sent to them from different users.
A user can delete the conversation or message that they have received as well.

Use case	Http method	Path	Request	Response
User ***sends a message*** *to* another user	POST	/users/:uid1/message/:uid2	Users primary key	Message Json
User ***retrieves*** all messages	GET	/users/:uid1/messages/	Users primary key	All messages in JSON array
User ***deletes*** message	DELETE	/users/:uid1/message/:uid2	Users primary key	Delete status

User ***retreivies*** all sent messages	GET	/users/:uid1/messages/	Users primary key	User sent messages JSON
User ***deletes*** all messages	DELETE	/users/:uid1/messages	Users primary key	Delete status