

Object Oriented Design

Last Revision Date: 11/3/11

1.0 Functions

2.0 Revision History

1.0 Functions

Module Name: **ConnActiv::login**

Module Type: **Method**

Return Type: **None**

Input arguments: **Email, Password**

Output arguments: **None**

Error messages: **If invalid email/password combination is provided**

Files accessed: **Users (Database)**

Files changed: **None**

Modules called: **None**

Narrative: **When the user enters their email address and password to login, it will be checked by accessing the users database to see if their email address exists and if the password is correct.**

Module Name: **ConnActiv::logout**

Module Type: **Method**

Return Type: **None**

Input arguments: **None**

Output arguments: **None**

Error messages: **None**

Files accessed: **None**

Files changed: **None**

Modules called: **None**

Narrative: **When the user decides to logout, he or she simple needs to click on the logout button and his or her session will end.**

Module Name: **ConnActiv::register_user**

Module Type: **Method**

Return Type: **None**

Input arguments: **Email, First/Last Name, Address, Password, Activities**

Output arguments: **None**

Error messages: **If invalid email or password is given**

Files accessed: **Activities (Database)**

Files changed: **Users (Database)**

Modules called: **None**

Narrative: **When the user registers on ConnActiv, they will be prompted to fill in as much information as they want to provide with a few mandatory fields such as email and password. The activities database will provide a list of already created activities to add to their profile and when the profile is complete, the information will be added to the database.**

Module Name: **Activity::Subscribe**
Module Type: **Method**
Return Type: **None**
Input arguments: **Activity Name, User ID**

Output arguments: **None**
Error messages: **If the user inputs an activity that isn't in the database**
Files accessed: **Activity (Database)**
Files changed: **User (Database)**
Modules called: **None**

Narrative: **A user wants to subscribe to an activity so they search for the activity and click to subscribe. This will add the activity ID to the user's information in the database.**

Module Name: **Activity::unsubscribe**
Module Type: **Method**
Return Type: **None**
Input arguments: **Activity ID, User ID**

Output arguments: **None**
Error messages: **None**
Files accessed: **None**
Files changed: **User (Database)**
Modules called: **None**

Narrative: **A user unsubscribes from an activity which will delete the activity ID from that user's information in the database.**

Module Name: **User::add_to_favorites**
Module Type: **Method**
Return Type: **None**
Input arguments: **Activity ID, User ID**

Output arguments: **None**
Error messages: **None**
Files accessed: **Activity (Database)**
Files changed: **User (Database)**
Modules called: **None**

Narrative: **A user wants to be able to quickly access a certain activity or other user via the Favorites section on his or her profile. On each page (activity, user, etc.), a user has the option to add that page to their favorites section.**

Module Name: **User::view_news_feed**
Module Type: **Method**
Return Type: **None**
Input arguments: **User ID**
Output arguments: **None**
Error messages: **None**
Files accessed: **User (Database), Activity (Database)**
Files changed: **None**

Modules called: **None**

Narrative: **The home page will access the user database to see what activities the user is subscribed to. Then the posts in the activity database will be parsed and displayed on the user's page.**

Module Name: **Review::approveUser**

Module Type: **Method**

Return Type: **None**

Input Arguments: **author, forUser, date, time, text, anonymous (Boolean)**

Output Arguments: **None**

Error Messages: **If the review addition was unsuccessful or invalid**

Files Accessed: **None**

Files Changed: **Reviews (Database)**

Modules Called: **None**

Narrative: **A user elects to submit a positive review (or approval) of another user or group with whom he/she has done an activity. The newly created review is added to the database, but it is kept anonymous if the submitting user wishes. When activated, this function formats the approval listing with necessary information and sends it to the database to save it.**

Module Name: **Review::disapprove**

Module Type: **Method**

Return Type: **None**

Input Arguments: **author, forUser, date, time, text, anonymous (Boolean)**

Output Arguments: **None**

Error Messages: **If the review addition was unsuccessful or invalid**

Files Accessed: **None**

Files Changed: **Reviews (Database)**

Modules Called: **None**

Narrative: **A user elects to submit a negative review (or disapproval) of another user or group with whom he/she has done an activity. The newly created review is added to the database, but it is kept anonymous if the submitting user wishes. When activated, this function formats the approval listing with necessary information and sends it to the database to save it.**

Module Name: **Activity::post_an_activity**

Module Type: **Method**

Return Type: **None**

Input Arguments: **author, public (Boolean), date, time, text, activityTags, recurring (Boolean)**

Output Arguments: **None**

Error Messages: **If the activity post was unsuccessful or invalid**

Files Accessed: **None**

Files Changed: **Activities (Database)**

Modules Called: **None**

Narrative: A user types a new post as a string and tags activities that it involves. He/she may mark it as public or private. The author, date, and time of the post are recorded as well. The user may mark the activity as recurring. The post is formatted properly and sent to the database to be saved and called upon later for display in news feeds.

Module Name: **Activity::ask_to_join**

Module Type: **Method**

Return Type: **None**

Input Arguments: **askingUser, author, date, time, text**

Output Arguments: **None**

Error Messages: **If the join request was unsuccessful or invalid**

Files Accessed: **Activities, User (Database)**

Files Changed: **Activities, User (Database)**

Modules Called: **None**

Narrative: A user sees an activity post in his/her news feed that is open to joiners and selects "join." The join request is sent to the database so that the authoring user receives it the next time he/she logs in.

2.0 Revision History

Person	Date Modified	Section	Description
Kim	11/1	Second half	First draft
Vince	11/1	First half	First draft