getUser

|  |  |
| --- | --- |
| Pre-Condition: | $user[] |
|  |
| Post-Condition: | $user [”<attribute>” = “<value>”] |
|  |
| Result: | Assigns User attributes to User |
|  |

registerNormal

|  |  |
| --- | --- |
| Pre-Condition: | $email = <value>, $username = <value>, $password = <value> |
|  |
| Post-Condition: | $validationKey = “<value>” |
|  |
| Result: | $email containing user validation key is sent to $user upon successful validation.  Valid $email, $username and $password stored in database. |
|  |

registerSocial

|  |  |
| --- | --- |
| Pre-Condition: | $username, $email, $authToken |
|  |
| Post-Condition: | $\_Session = [userName = “<value>”, id = “<authToken>”]  Valid Username, Email |
|  |
| Result: | Social API provides authentication.  User provides credentials.  Validated credentials are stored into database. |
|  |

loginNormal

|  |  |
| --- | --- |
| Pre-Condition: | $username = “<value>”, $password = “<value>” |
|  |
| Post-Condition: | $\_Session[username = “<value>”, $id = “<authToken>”] |
|  |
| Result: | $username and $password is verified with the database; User session is created or user is prompted to correct errors. |
|  |

authCheck

|  |  |
| --- | --- |
| Pre-Condition: | $authToken = <$\_Session[$id]>, $authToken = <other>, $\_Session[id = ”<authToken>”],  $Session[id = “<other>”] |
|  |
| Post-Condition: |  |
|  |
| Result: | Comparison between $authToken and $\_Session[$id], returning true only if they both contain matching values. |
|  |

logout

|  |  |
| --- | --- |
| Pre-Condition: | $\_Session = [$username = “<value>”, $id = “<value>”] |
|  |
| Post-Condition: | $\_Session = [id = <null>] |
|  |
| Result: | Breaks user’s session |
|  |

updateCert

|  |  |
| --- | --- |
| Pre-Condition: | $schoolEmail = “<null>” |
|  |
| Post-Condition: | $schoolEmail = “<value>” |
|  |
| Result: | -User has a validated institutional email associated with themselves. |
|  |

updateAccount

|  |  |
| --- | --- |
| Pre-Condition: | user = true, $\_Session[$id = “<authToken>”] |
|  |
| Post-Condition: | user[attr] = “<value>” |
|  |
| Result: | -Validated user attributes are updated. |
|  |

requestPassReset

|  |  |
| --- | --- |
| Pre-Condition: | $userName = “<value>”, $password = “<value>”, $email = “<value>” |
|  |
| Post-Condition: | $password = <new value> |
|  |
| Result: | -User is sent an email containing the new value for $password |
|  |

getFavorites

|  |  |
| --- | --- |
| Pre-Condition: | $userId = “<value>”, $entryId = “<value>” |
|  |
| Post-Condition: | $favorite[$userId = “<value>”, $entryId = “<value>”] |
|  |
| Result: | User favorites a $post; the association is stored into Favorites  Called into $favorite[] |
|  |

getLikes

|  |  |
| --- | --- |
| Pre-Condition: | $userId = <value>, $entryId = <value> |
|  |
| Post-Condition: | $liked[$userId = “<value>”, $entryId = “<value>”] |
|  |
| Result: | -User rates a $post; the association is stored into Rating.  -Called into $liked[] |
|  |

disable

|  |  |
| --- | --- |
| Pre-Condition: | userId = <value>, userName = <value> |
|  |
| Post-Condition: | userName = <null> |
|  |
| Result: | -userName of User is set to <null>, disables account |
|  |

getCollegeList

|  |  |
| --- | --- |
| Pre-Condition: | $institutions[1][“id” = “0129382”] |
|  |
| Post-Condition: | $institution[][“id” = “<Institution name>”, “name” = “<Institution name>”, “description” = “<Institution Description>”] |
|  |
| Result: | -Returns an assoc. array of institutions |
|  |

getProgramList

|  |  |
| --- | --- |
| Pre-Condition: | $program[#][] |
|  |
| Post-Condition: | $program[#][“id” = “<Program Id>” , “name” = “<Program Name>”], |
|  |
| Result: | -Returns an assoc. array of college programs |
|  |

getClassified

|  |  |
| --- | --- |
| Pre-Condition: | $ads[] |
|  |
| Post-Condition: | $ads[“id” = “<ad token>” |
|  |
| Result: | -Returns array of ad keys |
|  |

getEntries

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][] |
|  |
| Post-Condition: | $entries[#][“id” = “<Entry Id>” , “parent” = “<Entry Parent Id>” , “name” = “<Entry Name>” , “type” = “<Entry type>” , “owner” = “<Entry Owner>” , “description” = “<Entry Description>” , “rating” = “<Entry Rating>” , “datePost” = “<Entry DatePosted>” , “dateLastEdit” = “<Entry DateLastEdited>” , “crn” = “<Entry CRN>”] = “<Entry ID>” |
|  |
| Result: | -Populates $entries[] with Entry data |
|  |

getEntry

|  |  |
| --- | --- |
| Pre-Condition: | $entries[][] |
|  |
| Post-Condition: | $entry = “<Entry ID>” |
|  |
| Result: | Assigns an Entry ID to $entry |
|  |

getComments

|  |  |
| --- | --- |
| Pre-Condition: | $entries[][] |
|  |
| Post-Condition: | $entryType = comment  $entries[#][“<attribute>” = “<value>”] |
|  |
| Result: | -Assigns entry id where Entry is Type “comment” |
|  |

postEntry

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][“<attribute>” = “<value>”] |
|  |
| Post-Condition: | $query = “insert values $entries[#][...] into entry” |
|  |
| Result: | -$query is used to add a new Entry |
|  |

addComment

|  |  |
| --- | --- |
| Pre-Condition: | $entryId = “<value>”, $parentId = “<Id of comment parent>”, $type = “comment” |
|  |
| Post-Condition: | $query = “insert into Entry values $entry[#][“id” = <Entry Id>, “parentId = “<Id of comment’s parent>” … “type” = “comment””] |
|  |
| Result: | -Adds an Entry of type comment to the database with the id of the parent Entry as parentId |
|  |

addRating

|  |  |
| --- | --- |
| Pre-Condition: | $commended && $userId = false |
|  |
| Post-Condition: | $query = “insert $rating[#][] into entry where $rating[#][“commended” = true]” |
|  |
| Result: | -Updates the rating table where Commended is true under UserId |
|  |

addFile

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][“Id” = “<value>”], $files[#][“<attributes>” = “<value>”] |
|  |
| Post-Condition: | $fileEntry[#][entryId = “$entries[#][“Id” = “<value>”], fileId = $files[#][“Id” = “<value>”]”] |
|  |
| Result: | -Finds the id of the entry and files  -Creates a record on the File table  -Record is added to File\_Entry Table |
|  |

flag

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][..., “type” != “flagged” ,...] |
|  |
| Post-Condition: | $entries[#][..., “type” = “flagged” ,...] |
|  |
| Result: | -Sets entry type as flagged |
|  |

favorite

|  |  |
| --- | --- |
| Pre-Condition: | $favorites[#][“UserId” = “<value || null>”, “EntryId” = “<null>”] |
|  |
| Post-Condition: | $query = “insert $favorites[#][] into Favorites where $favorites[#][“UserId” = “<value>”, “EntryId” = “<value>”]” |
|  |
| Result: | -UserId and EntryId are associated through the favorites table |
|  |

edit

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][..., “<Entry UserId>” = “<User UserId>” ,<description> = <value>,...], |
|  |
| Post-Condition: | $entries[#][...,<description> = <new value>,...] |
|  |
| Result: | -replaces the description attribute with a new/edited value if the Entry UserId matches the User UserId |
|  |

searchEntry

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][“<attribute>” = “<value>”], $value != “<entryId>” || “<valueId>” |
|  |
| Post-Condition: |  |
|  |
| Result: | Attempts to find relevant Entry records based on <attribute> $value pairs |
|  |

sortEntry

|  |  |
| --- | --- |
| Pre-Condition: | $entries[#][“<attribute>” != “<null>”], $attribute != “<entryId>” || “<valueId>” |
|  |
| Post-Condition: |  |
|  |
| Result: | Attempts to sort by relevant non null $attribute values |
|  |