

Method Descriptions

Parent Class Methods:

PasswordUsernameChange(username, pass):

Precondition: username exists and password is correct. As well as the user has logged into their account

Postcondition: username and or password has been changed

Side-effects: database has recorded changes

user: valid string variabel that may be initiated to change or stay the same

pass: string variable that may be passed in and be changed or stay the say

PersonallInfo(nameID, role):

Precondition: name is a valid name, and the role being assigned to it is valid

Postcondition: name will be created for the account with the specifying role

Side-effects: name and their role will be added to database

Name: string variable that is passed in to create an object of an account that will be created with this name

Role: string variabel passe din associated with whether this person is a TA, instructor, or admin

ContactInfo(address, email, number):

Precondition: address, email, and phone numbers are valid

Postcondition: address, email, number are updated or inserted

Side-effects: contact info is now in database

Address: address will be added to that account

Email : email variable will be added to the account

number: phone number will be added to the account

Profiledesign(image, description, status, account name):

Precondition: an account exists

Postcondition: account has been updated with a description of that account, profile picture, and online status

side-effects: database has recorded image, description, status

image: image for a profile picture being passed into method

description: string variable to pass in as a description

status: will shows the status of the account

PrintContactInfo(accountid, role):

Precondition: an account exists for contact info

Postcondition: contact info of the account is readable depending on the role

side-effects: none

account id: variable of person who has the contact info

role: role of person trying to obtain the contact info, different for TA and instructor and admin

SendNotifications(senderemail, recieveemails, message):

Precondition: emails exist of the people being sent the message

Postcondition: email is sent from the instructor

Side-effects: none

senderemail: email passed in for the sender

recieveemails: email of people or persons receiving the email passed in

Message: message of what is being sent to recievers.

Login/Logout Class Methods:

SetupPasswordsUsernames(changepass,changeuser,forpass,foruser):

Precondition: account exists

Postcondition: password or username is updated to new version

side-effects: updated passord or username in database

changepass: variable to pass back as true if password needs to be changed

changuser: variable to pass back as true if username was changed

forpass: vairable to pass back as true if password was forgotten

foruser: variable to pass back as true if username was forgotten

SetUpLogOut(logout):

Precondition: account has logged in to an account

Postcondition: back to login screen afer logout

side-effects: none

logout: variable tha returns true if logout button was initiated.

Create Account Class:

AccountSetUp(name, accountid, role, changename):

Precondition: accountid is valid and not used,

Postcondition: name of account is created, with an acountid, and their role

side-effects: new account added to database

name: string variable of the account name

accountid: int variable of an accountid number

role: role of the account TA, instructor, or admin

changename: variable to return true if the name has changed for the account

DeleteAccount(accountid):

precondition: accountid is valid

postcondition: accountid no longer exists in databse, accountid is now valid to be used

side-effect: account has been deleted from database

accountid: deletes the accountid that was passed in

AssignAccount(accountid, courseid,labid)

precondition: account ID, courseID, and or labID, is valid

postcondition: account is now assigned to a course or lab depending on role

side-effects: saves account with the course or lab id

accountid: accountid passed in to get the role of the person to assign to course or lab

courseid: variable of the course ID to assign the account ID to

labid: variable of the Lab ID to assign lab to the account if the role of the account is a TA

Create Assignments Class Methods:

CreateName(assignmentname):

Precondition: instructor has logged in and is creating an assignment

Postcondition: assignment has been created and given a name

Side-effects: creates an assignment with a name

Assignmentname: variable passed in of assignment name.

SetDueDate(assignmentname, date):

Precondition: assignment has been created

Postcondition: assignment now paired with a due date

Side-effects: saves due date to database

Assignmentname: variable passed in of assignment name

Date: due date passed in to be paried with assignmentname

SetDescription(assignmentname, description):

Precondition: assignment has been created

Postcondition: description now paired with the assignment

Side-effects: saves description to database

Assignmentname: variable passed in of assignment name.

Description: variable passed in to give assignment a description

SetPoints (assignmentname, points):

Precondition: assignment has been created

Postcondition: assignment given points

Side-effects: saves points to database

Assignmentname: variable passed in of assignment name.

Points: variable passed in to give assignment points it is worth

SetType (assignmentname, type):

Precondition: assignment has been created

Postcondition: assignment given a type

Side-effects: saves type to database

Assignmentname: variable passed in of assignment name.

Type: variable passed in to give the assignment a type (quiz,homework, etc)

FileUp (assignmentname, file):

Precondition: assignment name has been created

Postcondition: assignment file has been updated

Side-effects: saves file to database

Assignmentname: variable passed in of assignment name.

File: variable passed in to upload the file of the assignment name

TA Class Methods:

ViewAssignments(assignment):

Precondition: assignment is existing

Postcondition: TA can now see the assignments of theirs and other TA's

Side-effects: none

Assignment: variable passed in to tell the software to print the assignments to screen

ObtainingContactInfo(info, userID):

Precondition: information is available to read

Postcondition: software is displaying the information to screen

Side-effects: none

Info: variable passed in to tell the software to print contact info

UserID: variable passed in of user id that is being used to get the info

Instructor Class Methods:

AssigningTA(labid, userID):

Precondition: lab exists, and TA ID exists

Postcondition: TA now assigned to a lab

Side-effects: TA now saved to database

LabID: Lab ID passed in to pair with the TA

UserID: TA ID passed in to pair with the lab

ViewAssignments(assignment):

Precondition: assignment is existing

Postcondition: Instructor can now see the assignments of their TA's and course assignments

Side-effects: none

Assignment: variable passed in to tell the software to print the assignments to screen

Admin Class Methods:

DataView(view)

Precondition: there is data to see

Postcondition: data is now printed to a screen

Side-effects: none

View : variable passed in to tell the software to print data to screen

CreateClass(name, classID), type):

Precondition: admin has logged in to their account and has selected to create a class

Postcondition: class is now created with its name, ID, and type

Side-effects: Class is now saved to database with its ID and type

Name: name of class passed in, this can be changed

ClassID: course ID passed in for the class, this can be changed

Type: type of course passed in to give it a type, this can be changed