Zachary Strickler

Game Wizard

Pseudocode

**FORUMS**

**createForum(forumName, forumID)**

set the name of the forum = forumName

set the formID

**removeForum(forumID)**

if permitted -

delete the forum equal to the forum id

else-

void

**editForum(forumID)**

if permitted-

allow user to change forum content and title

else-

void

**readForum(forumID)**

display the contents and title of the forum

**saveForum(forumID)**

if save forum selected-

save the forum to the users saved forums

**reportForum(forumID, userID)**

if user selects report forum –

report the forumid and userid who created it

else -

void

**USERS**

**createUser(userID)**

creates a user, asks for string for username, and gives a user id

**editUser(userID)**

edit a userID

**editUserName(userID, userName)**

if username does not exist -

edit the string of the users username

verify that the username is appropriate

save

else-

void

**changeEmail(userID, userEmail)**

if email is not linked to another account –

edit the string of the users email

verify that it is a real email

save

else –

void

**changePicture(userID, userProfilePic)**

if new picture meets guidelines –

change userprofile pic

save

else-

void

**changePassword(userID, userPassword)**

if new user password meets password criteria –

change user password

save

else-

void

**deleteUser(userID)**

shows a prompt “do you wish to delete your account?”

if user selects “Yes”

delete account

else

void

**viewUser(userID, username, userDescription)**

shows the users profile and description

**privateUser(userID, userPrivacy)**

changes the users privacy setting

if userPrivacy is true

ask user if they want to set it to false

save

elif userPrivacy is false

ask user if they want to set it to true

save

**changeUserSettings(userID)**

opens menu to allow user to change appearance settings

change text size

change appearance mode

**reportUser(userID)**

reports user associated with a userid

**POSTS**

**createPost(postID, postTitle, postTime, postForum, postContent)**

if postContent and postTitle is appropriate

create postID

else

void

**editPost(postID, postContent)**

if permitted

edits a postID

else

void

**deletePost(postID)**

if permitted

delete a post associated with a post id

else

void

**showPost(postID)**

user clicks on post

shows a post associated with a post id to the user

**reportPost(postID)**

report = true

reports a post associated with a post id

**hidePost(postID)**

user clicks on hide post button

hides post associated with a postID

**upvotePost(postID, postUpvotes)**

user clicks upvote = true

adds +1 to the upvote int

**downvotePost(postID, postDownvotes)**

user clicks downvote = true

adds -1 to the upvote int

**commentOnPost(postID)**

if permitted

user comments on post associated with a post id

comment is created with a comment id

comment is linked to user

else

void

**COMMENTS**

**createComment(commentID, commentContent, commentUser)**

reads keyboard input of comment

if comment is appropriate

post comment

else

void

**editComment(commentID, commentContent)**

edit comment

If permitted

Read keyboard input of comment

If comment is appropriate

Edit comment

Else

Void

Else

void

**deleteComment(commentID)**

if permitted

print “Do you want to delete comment”

if true

delete the comment

else

void

else

void

**showComment(commentID)**

if on post and user clicks comment

show comment content

else

void

**reportComment(commentID)**

report commentID

sends a report to administrator

**hideComment(commentID)**

hide commentID

stops displaying comment to user

**DIRECT MESSAGES**

**typeMessages(messageID, messageContent)**

read input from user

if message input is appropriate

type the message

else

void

**editMessage(messageID, messageContent)**

read input from user

if edited message is appropriate

edit message

else

void

**deleteMessage(messageID)**

if permitted

print “Do you want to delete message”

if true

delete the message

else

void

else

void

**viewMessage (messageID)**

shows message content to user

**sendMessage(messageID, messageUser)**

send typed message input

**INTEGRATION**

**integrateAccount(userID)**

prompts user to link account

if discord and true

open discord link website

if twitch and true

open twitch link website

else

void

**integrateType(application)**

display users integration

**removeIntegration(application)**

prompts user to remove integration

if true and user has twitch and discord integrated

prompt user if they want to remove twitch, discord, or both

if both

remove both integrations

if discord or twitch

remove application integration