CSC207 Phase 0

Task 1: CRC Model

Abstract: no

Superclass: NA

User: Entity

Subclasses: NA

- username
  - getters/setters
- password
  - getters/setters
- loginHistory: list of dates they logged in on
  - Default empty list
  - getters/setters
- banned
  - Default false
  - getters/setters
- roles: a Set of strings to represent roles such as Regular, Admin
  - Default [“Regular”]
  - getters/setters
- addLogin(date/time) that appends a date to loginHistory
- addRole(String role): appends the role to the roles set if not already in the array

- UserAccess

Abstract: no

Superclass: NA

UserAccess: Use Case

Subclasses: NA

- List containing User objects
- checkLogin method that takes a username and password to see if a username and password pair exist.
    - If an account exists, call user.addLogin(date/time)
    - Returns a HashMap<String, boolean> of the form [“Exists”: true/false, “IsBanned”: true/false, “IsAdmin”: true/false]
- exists(username): return true if this user exists, false otherwise
- findUser(username): finds and returns user with username
- addUser(username, password): creates a new user with this info and appends to the list of Users returns true if successfully created
- delete(username): find user with username in the array and removes the user from array. Returns true if successfully removed, returns false if user does not exist
- ban(username): find user with username and then call user.setBanned(true) returns true if successful, false otherwise
- unban(username): find user with username and then call user.setBanned(false) returns true if successful, false otherwise
- giveRole(username, String role): calls user.addRole(role) return false if user doesn’t exist true otherwise
- outputUserInfo(): output all existing user info in a HashMap with the format {“userName”: {“Password”: String, “IsBanned”: String, “IsAdmin”: String}}
- outputUserLogInHistory(): output all existing user log in history in a HashMap with format {“userName”: [Timestamp1, Timestamp2,...]}
- getPrevioustLogIn(username): return the previous login times and dates
- isAdmin(username): check whether a user with username is admin

- User
- LoginPage

Abstract: no

Superclass: NA

MainMenu: Controller

Subclasses: NA

- A method that starts the program and gives people the option to log in or create account “to create an account enter 1 to log in enter 2”
  - If log in, call LoginPage.startLogin()
  - If create account, call CreateAccountPage.createAccount()

- LoginPage
- CreateAccountPage

Abstract: no

Superclass: NA

CreateAccountPage: Controller

Subclasses: NA

- createAccount(): “enter the username” store username, calls UserAccess.exists(username)
  - if a user exists, “username is already taken. Enter 1 to try again or enter 2 to log in instead”
    - if 1, calls CreateAccountPage.createAccount()
    - if 2, calls LoginPage.startLogin()
  - Otherwise “enter the password” store password and call UserAccess.addUser(username, password) returns true if successfully created

- MainMenu
- UserAccess
- LoginPage

Abstract: no

Superclass: NA

LoginPage: Controller

Subclasses: NA

- startLogin method that launches the login system.
  1. Will return a string like “enter username”, “enter password” (prompts person to type)
  2. Calls UserAccess.checkLogin(username, password)
  3. returns “Incorrect username or password, press 1 to create an account or press 2 to try login again” or “You are now logged in as <username>. Welcome.” or “This account is banned”
     a. If Admin, call AdminPage.selectAccountOption()
     b. If create account, call CreateAccountPage.createAccount()

- MainMenu
- UserAccess
- AdminPage
- CreateAccountPage

Abstract: no

Superclass: NA

AdminPage: Controller

Subclasses: NA

- selectAccountOption() “press 1 to delete a user, press 2 to ban a user, press 3 to unban a user, press 4 to make an existing user an admin”
- Four methods delete(), ban(), unban(), makeAdmin()
- delete(): “enter the username of the user to delete” calls UserAccess.delete(username) returns “successfully deleted” or “user does not exist”
- ban() “enter the username of the user to ban” calls UserAccess.ban(username) returns “successfully banned” or “user does not exist”
- unban() “enter the username of the user to unban” calls UserAccess.unban(username) returns “successfully unbanned” or “user does not exist”
- makeAdmin() “enter the username of the user to grant admin rights” calls UserAccess.giveRole(username, “Admin”) returns “admin rights granted” or “user does not exist”

- LoginPage
- UserAccess