Login System CRC Model

Anudev Gill, Rohit Shetty, Andrew Chun, Rana Al-Khulaidi

LoggedOutManager

uses InputHandler to get inputs from the terminal

uses UserInterface to print messages to the terminal

calls functions in UserManger to create new users

represents the home menu and the possible actions a user that is not logged can take

- menuSelector()

calls functions to represent the login process

- loginUser()

UserInterface

InputHandler

AuthenticateUser

UpdateUserHistory

UserManager

LoggedInManager

uses InputHandler to get inputs from the terminal

uses UserInterface to print messages to the terminal and a user’s login history

represent the menu for a user that is logged in and possible actions for them

- userScreen(), nonAdminScreen(), adminScreen()

function to represent the process of changing a user’s ban status

- updateUserBanStatus()

function to represent the process for deleting a user

- deleteUser()

calls function in UserManager to create a new Admin user

UserInterface

InputHandler

AuthenticateUser

RestrictUser

UpdateUserHistory

UserManager

UsernamePasswordFileManager

UserManager

uses InputHandler to get inputs from the terminal

uses UserInterface to print messages to the terminal

prompts a user for their username and password

- getUsernameAndPassword()

updates the users in memory from those saved in .csv files

- loadUsersFromCSV()

implements process for creating a new user

- createNewUser()

UserInterface

InputHandler

CreateUser

UsernamePasswordFileManager

UpdateUserHistory

InputHandler

Gets integer inputs with allowed values

- intInput()

Gets String inputs with or without allowed values

- strInput()

UserInterface

UserInterface

Prints welcome message

- printWelcomeMessage()

Prints exceptions

- HandleNumberFormatException(), printArbitraryException()

Prints login history

- printLoginHistory()

Prints menu options

- printNonAdminLogInMenu(), printAdminLogInMenu()

Prints username, password, and confirmation requests

- printUsernameInput(), printPasswordInput(), printUsernameInputForUserRestriction(), printRestrictUserConfirmation()

Prints success and failure messages

- printSignUpSuccess(), printLoginSuccess(), printSignUpFail(), printLoginFail()

CsvEditor

Create a .csv file

- createCSVfile()

Append to a .csv file

- addLine()

Read a .csv file

- readLines()

UsernamePasswordFileEditor

CsvEditor

UsernamePasswordFileEditor

Create a empty UsernamePassword.csv with three columns

- createUsernamePasswordFile()

Create a UsernamePassword.csv from a UserContainer class

- createUsernamePasswordFile()

Append Username,Password, and the admin status of a User to the UsernamePassword.csv file

- addUserInfo()

UsernamePassword.csv

User

UserContainer

AuthenticateUser

Logs a user in

- loginUser()

Logs a user out

- logoutUser()

Checks if a username exists

- checkUsernameExists()

Checks if a User is an admin

- checkUserAdmin()

User

UserContainer

RestrictUser

Bans user

- banNonAdminUser()

Unbans user

- unbanNonAdminUser

Checks if user is banned

- isUserBanned()

Deletes user

- deleteNonAdminUser()

User

NonAdminUser

UserContainer

CreateUser

Stores an instance of UserContainer

Creates a new Admin user with given username and password, and adds the user to HashMap.

-createAdminUser()

Creates a new NonAdmin user with given username and password, and adds the user to HashMap.

-createNonAdminUser()

Checks if given user already exists in HashMap, returns boolean.

-UsernameuniqueExists()

UserContainer

User

UpdateUserHistory

Stores an instance of UserContainer

Read a user’s history from UserHistory.csv and return an Arraylist of timestamps. Initialize the user’s loginHistory to this Arraylist

- readUserHistory()

Read each user’s history in an Arraylist of usernames and return the histories as a user, timestamp pair.

- readUserHistories()

Overwrite UserHistories.csv file with custom user histories

- overwriteUserHistories()

Write a new login for a given user to UserHistory.csv

- writeUserHistory()

Return a given user’s login history

- getLoginHistory()

User

UserContainer

HashMap

UserContainer

Stores a hashmap of usernames to User objects.

Getter for the user associated with a username

- get()

User

Abstract

User

Stores login times and dates

Stores usernames and passwords

Stores if it is admin

Stores if user is logged in

Getter for if a user is admin

- isAdmin()

Getter for username and password

- getUsername(), getPassword()

setter and getter for login times and dates

- setLoginHistory(), addToLoginHistory(), getLoginHistor()

Setter and getter for its logged in status

- setIsLoggedIn()

User

Admin

getter for admin status

- isAdmin()

+ All of User attributes and functions

Interface

Bannable

Stores if this user is banned

Be temporarily banned

- temporarilyBanUser()

Be unbanned

- unbanUser()

NonAdmin

User

NonAdmin

Bannable

getter for admin status

- isAdmin()

getter and setter for ban status

- checkUserBanStatus(), temporarilyBanUser(), unbanUser()

+ User methods