Ricky Tucker JR

Final Project

Found in App Development 19EW5

Professor Lovejoy

Contents

[Problem Statement/Scenario: 3](#_Toc12149063)

[Pseudocode 3](#_Toc12149064)

[Methods and Classes: 5](#_Toc12149065)

[Revised pseudocode 5](#_Toc12149066)

[Error Documentation: 9](#_Toc12149067)

[Solution Documentation: 9](#_Toc12149068)

# Problem Statement/Scenario:

The selected Scenario is A ZOO Authentication System. This Authentication System will handle a user entered username and password(Hashed with MD5 ash) which will be cross-referenced with the authentication document. The user will have three attempts to provide valid credentials. Once the credentials have been verified, the appropriate role file will be displayed to the user.

Pseudocode:

Load credentials file into credentialsIN

INITIALIZE a list of credentials as credentialsList

WHILE line of credentials left in credentiasIN

READ username from line

READ hashed password from line

READ userRole from line

CREATE a new entry into credentialsList with username , hashed password, userRole

ENDWHILE

INITIALIZE loginAttempts to 0

FOR loginAttempts is less than 3

PROMPT user for username or ‘q’ for quit

READ username to username

IF username equals ‘q’ THEN

DISPLAY “Goodbye”

EXIT program

END IF

PROMPT user for password

READ password and hash(MD5) to userPassword

IF credentialsList has credentials with matching username and userPassword

DISPLAY userRole file from credentialsList

DISPLAY “Press any key to exit/logout”

WAIT for user to press key

EXIT program

ELSE If loginAttempts is 2

Display “Max failed login attempts, good bye”

EXIT program

ELSE

DISPLAY “Failed login”

ENDIF

# Methods and Classes:

Class Authentication – Handles authentication for zoo system

Main – provides Primary interface

LoadUserRoleFile – loads the User Role from the database

Class Credentials – Handles loading, processing, verification of credentials.

LoadCredentials – function handlng loading credentials

HashMD5 – helper function to perform password hashing  
 MatchCredentials – function handling matching credentials from list

GetUserRole – function handling acquiring role information from list

# Revised pseudocode

Authentication.Main()

INITIALIZE loginAttempts to 0

CALL Credentials.LoadCredentials()

FOR loginAttempts is less than 3

PROMPT user for username or ‘q’ for quit

READ username to username

IF username equals ‘q’ THEN

DISPLAY “Goodbye”

EXIT program

ENDIF

PROMPT user for password

READ password to userPassword

CALL Credentials.HashMD5( userPassword)

SET userPassword to result of HashMD5 funtion

IF Credentials.MatchCredentials(username, userPassword) equals “Match” THEN

SET userRoleName TO result of Credentials.GetUserRole(username,userPassword)

SET userRole TO results of Authentication.LoadUserRoleFile(userRoleName)

DISPLAY userRole

DISPLAY “Press any key to exit/logout”

WAIT for user to press a key

EXIT program

ELSE THEN

DISPLAY “Invalid user Credentials”

END ELSE

IF loginAttempts equals 2

Display “Max failed login attempts, good bye”

EXIT program

ENDIF

Authentication.LoadUserRoleFile(userRole)

OPEN userRoleFile with name userRole

READ userRole into userRoleOut

RETURN userRoleOut

END DEF

Credentials .LoadCredentials

Load credentials file into credentialsIN

INITIALIZE a list of credentials as credentialsList

WHILE line of credentials left in credentiasIN

READ username from line

READ hashed password from line

READ userRole from line

CREATE a new entry into credentialsList with username , hashed password, userRole

ENDWHILE

END DEF

Credentials .HashMD5(password)

Return hashMD5 password

END DEF

Note on implementation, used the collections interface Stream() and anyMatch to find if there is a match.

Credentials .MatchCredentials(username,userPassword)

FOR each credentialPair in CredentialList

IF credentialPair matches username and userPassword THEN

RETURN “Match”

ENDIF

END FOREACH

RETURN “NO Match”

END DEF

Credentials .GetUserRole (username , userPassword)

FOR each credentialPair in CredentialList

IF credentialPair matches username and userPassword THEN

RETURN userRoleName

ENDIF

END FOREACH

Return “No userRole”

END DEF

# Error Documentation:

1. An error on file input because didn’t include throws IOException

2. patterning matching for using split argument on the string, was using “ “ not “\t”.

# Solution Documentation:

1.Implemented throw IOException in function declaration

2.Changed matching string from “ “ to “\t” to get correct splits.