Jeremy Petty

IT 145

Final Project Draft

Option 1: Authentication System

**Problem Statement/Scenario:**

For this scenario, I have chosen “Option 1” which consists of developing a system to allow various zoo employees access to particular data within a database. This problem will require me to develop prompts to ask users for usernames and passwords. It will also require me establish parameters for utilizing credentials, limiting password attempts, assigning access roles, displaying content, and allowing users to log-out.

When the user is asked to enter their username and password credentials, which are checked against the credentials file before allowing access to the role account. Once the user accesses the role, they are presented with the information that identifies the users role. The user is then given the opportunity to either exit the program which is signified with a “y” or a “n”. This program is necessary to ensure the security of the zoo information system.

**Overall Process:**

For this project, I will start by creating a list of tasks that need to be accomplished by this program. Once I have created the list I will create a pseudocode that outlines the processes/steps needed within the code. From the pseudocode, I will begin building the actual code utilizing Java code and the NetBeans development software. After each stage I will utilize the debug functions of NetBeans to ensure functionality.

Creating this project has proven to be challenging and has resulted in some minor changes. When creating this project it started with creating an outline, .txt files, and reviewing various tutorials. This project involved implementing hashed passwords, classes, and file references.

I started the project by keeping it simple and creating a program that would simply output a user name and password request utilizing a scanner. I began adding to that little pieces at a time as we have discussed in our class. The next segment I began working on was referencing the .txt file for the credential. Once I added that segment I added in the variables from the MD5, password request, and attempt limiting. Originally, I added in the roles and descriptions directly into the Java code, but found that to be problematic with adding additional roles and users in the future. Once I finalized those processes I added in the exit process.

**Pseudocode (Authentication System):**

Start Program

//User starts the computer and enters the online portal

//PARAMETERS include establishing the credential file with roles and also creating rule for the number of attempts allowed

INITIATE username and password PARAMETERS (3 Attempts)

INITIALIZE prompt for Username

INITIALIZE prompt for Password

COLLECT username

IF username is entered proceed to password

ELSE, exit

COLLECT password

IF password is entered, proceed to verification (credential file)

ELSE, exit

IF password is attempted more than three (3),

Print (“User has been locked”)

OPEN credential file

//Credential file will be referenced in a .txt file

IF file matches, proceed to next step

ELSE, print error and stop program

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication does not match, THEN

PROMPT for username and password for a second (2) time

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication does not match, THEN

PROMPT for username and password for a third (3) time

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication does not match, THEN, PRINT “Not Successful” and exit program

ELSE,

PRINT assigned role information

PROVIDE OPTION for user to logout

**Error Documentation:**

This project was definitely more challenging than I would have thought. It looked like a straight forward program that could be easily laid out and implemented. What I found, however, was that it was complicated and required research on many levels. The first issue that ran into was referencing the credential.txt file. I was having trouble with placing the file in the correct file to be referenced. Once I was finally able to reference the file I had difficulty referencing the correct column utilizing delineators.

Implementing the MD5 was difficult and I don’t know that I correctly implemented it in the program. I attempted to add my name as a username and password. I found a MD5 string converter to get the correct hash code, but I was still unable to have it correctly work.

I had an issue with implementing the roles and user names. When I had laid out my original code my thought process was to create an IF statement that references the roles within the credential file and matches them with the roles in the IF loop. The issue that I had was the ability to add additional roles in the future. This lead me to create additional .txt files for each role to be referenced.

**Solution Documentation:**

The solution to the following problem:

1. Referencing the credentials file – For this issue it required me to change the (Scanner readCred=newScanner(newFile("/Users/jeremypetty/NetBeansProjects/Authentication/credentials.txt"));) file to correctly reference the right location. It also required me to move the additional role files into the same folder in order for it to correctly look for the roles based on the credential file role.
2. Referencing the correct roles within the creational file to locate the role file. This took a little bit of researching to find how to correctly reference the right column within the file and then based on the role find the correct role file.
3. Implementing the MD5 file was difficult because when referencing some of my other class mates pseudocode it appeared that it output the hash on the user screen. Although I did not get this to correctly work, I did get it to correctly reference the credentials files hash code.
4. As mentioned above I originally entered all the roles into the Java code and wanted to utilize an IF statement to loop and reference role defined within the Java code. When reading the directions for the project I realized that I could not make changes easily to roles.

**Changes to Pseudocode (Authentication System):**

Start Program

//User starts the computer and enters the online portal

//PARAMETERS include establishing the credential file with roles and also creating rule for the number of attempts allowed

INITIATE username and password PARAMETERS (3 Attempts)

INITIALIZE prompt for Username

INITIALIZE prompt for Password

COLLECT username

IF username is entered proceed to password

ELSE, exit

COLLECT password

IF password is entered, proceed to verification (credential file)

ELSE, exit

IF password is attempted more than three (3),

Print (“User has been locked”)

REFERECE MD5 code for HASH

OPEN credential file

//Credential file will be referenced in a .txt file

IF file matches, proceed to next step

ELSE, print error and stop program

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication is successful, THEN

OPEN .txt role file

OUTPUT Role file message

IF authentication does not match, THEN

PROMPT for username and password for a second (2) time

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication is successful, THEN

OPEN .txt role file

OUTPUT Role file message

IF authentication does not match, THEN

PROMPT for username and password for a third (3) time

CONFIRM user is authorized by comparing credential file to username and password entered.

IF authentication is successful, THEN

OPEN .txt role file

OUTPUT Role file message

IF authentication does not match, THEN, PRINT “Not Successful” and exit program

ELSE,

PRINT assigned role information

PROVIDE OPTION for user to logout