Object Oriented Programming

Term-End Project

Group members:
Sagar Bharti K004, Tanvi Dalvi K011, Dineshkumar Ganessan K016,
and Arushi Rai K047
B.Tech CSE (Cybersecurity)
Semester 3, Academic Year 2021-22

Contents

Objective

Creating a journaling app using JAVA and Java Database Connectivity JDBC

Language

Java (JDK 16.0.2)

Integrated Development Environment

Eclipse IDE for Java Developers (June 2021 version)

Other Software Used

MySQL Workbench
Notepad

Hardware Specifications

Windows 10 running on 8 GB RAM

Source Code

mainApp.java

```
package projectFinal;

//import java.io.*;

public class mainApp {


      public static void main (String args[]) {

            String username = " ", name = " ";

            loginAndSignup obj1 = new loginAndSignup();
            username = obj1.mainFunction();

            name = obj1.giveUserName(username);

            quotePrint q = new quotePrint();
            q.quotePrintFunction();


            vaultFilesSystem vf = new vaultFilesSystem(username, name);
            vf.mainFunction();
      }
}
```

loginAndSignup.java

```
package projectFinal;

import java.util.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.time.LocalDate;


public class loginAndSignup {
    @SuppressWarnings("finally")
    public int checkUsername (String s, int which) {
        if (which == 0) {
            //System.out.println("1done");
        int flag = 0;
        try {

            Connection connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root", 
"toor");
            Statement statement = connection.createStatement();
            ResultSet resultset = statement.executeQuery("select * from 
logincreds");

            while(resultset.next()) {
                if (s.equals(resultset.getString("username")) == true) {

                    flag = 1; break;
                }
            }

         }
         catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (flag == 0) return 0;
            else return 1;
        }

    }
        return 9;
}

    @SuppressWarnings("finally")
    public  int checkPassword (String p, String s) {
        int flag = 0;
        try {


            Connection connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root", 
"toor");
            Statement statement = connection.createStatement();
            ResultSet resultset = statement.executeQuery("select * from 
logincreds");

            while(resultset.next()) {
```

```
                      if (s.equals(resultset.getString("username")) == true &&
p.equals(resultset.getString("password")) == true) {

                            flag = 1; break;
                      }
                 }

           }
            catch (Exception e) {
                 e.printStackTrace();
           } finally {
                 if (flag == 0) return 0;
                 else return 1;
           }

      }

      public  String giveUserName (String s) {
           try {

                 Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root",
"toor");
                 Statement statement = connection.createStatement();
                 ResultSet resultset = (ResultSet) statement.executeQuery("select
logincreds.name, logincreds.username from logincreds where logincreds.username =
\"" + s + "\"");

                 while(resultset.next()) {

                            return (resultset.getString("name"));

                 }

           }
            catch (Exception e) {
                 e.printStackTrace();
           }

           return (" ");

      }

      public  char[] passwordGenerator (int length) {
                String caps = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
                String non_caps= "abcdefghijklmnopqrstuvwxyz";
                String chars = "!@#$^&*";
                String nos = "1234567890";
                String total = caps + non_caps + chars + nos;
                Random random = new Random();
                char[] password = new char[length];

                password[0] = non_caps.charAt(random.nextInt(non_caps.length()));
                password[1] = caps.charAt(random.nextInt(caps.length()));
                password[2] = chars.charAt(random.nextInt(chars.length()));
                password[3] = nos.charAt(random.nextInt(nos.length()));

                for(int i = 4; i< length ; i++) {
                   password[i] = total.charAt(random.nextInt(total.length()));
                }
                return password;
             }

     public  int createNewProfile(String name, String username, String password)
{
           try {
                Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root",
"toor");
                Statement statement = connection.createStatement();
                String addThese = "\"" + username + "\", \"" + password + "\",
\"" + name + "\"";
                statement.executeUpdate("INSERT INTO logincreds VALUES (" +
addThese +");");
                return 1;
           } catch (Exception e) {
                return 0;
           }
     }

     public String mainFunction () {
           String username = "";
           String newUsername = "";
           Scanner sc = new Scanner(System.in);


           System.out.print("Welcome to ProductiviTea!\t\t");
           LocalDate myObj = LocalDate.now();
           System.out.println(myObj);
           System.out.print("\t1. LOGIN (for existing users)\n\t2. SIGNUP (for
new users)\n\t > ");
           int ch = sc.nextInt();
           int passedOne = 1;
           while (passedOne == 1) {
           switch(ch) {
           case 1: {
            int cont = 1;
            while (cont == 1) {
            System.out.print("Enter your username > "); username = sc.next();
            //System.out.println(username);
            while (true) {
                if (checkUsername(username, 0) == 1) {
                    System.out.print("Enter your password > "); String 
password = sc.next();
                    if (checkPassword(password, username) == 1) {
                        String name = giveUserName(username);
                        System.out.println("Login successful! Welcome to 
your ProductiviTea account, " + name + "!" );
                        cont = 0;
                        passedOne = 0;
                        break;

                    } else {
                        System.out.println("Password incorrect for " + 
username);
                        System.out.print("Enter 1 to try again for " + 
username + ", 2 to try for another account, and 0 to exit > "); int ch2 = 
sc.nextInt();
                        if (ch2 == 1) continue;
                        else if (ch == 2) break;
                        else {cont = 0; break;}
                    }
                } else {
                    System.out.println("No account found with username - " 
+ username);
                    System.out.print("Enter 1 to try again, 0 to exit or 
\nNew user? Enter 2 to make a new account > "); int ch3 = sc.nextInt();
                    if (ch3 == 2 ) {
                        ch = ch3;
                        cont = 0;
                        break;
                    } else if (ch3 == 1) {
                        break;
                    } else {
                        cont = 0;
                        break;
                    }
                }
            }

            }
        break;
        } case 2: {
            System.out.println("Hey! Let's make you your own ProductiviTea 
account!");
            System.out.print("What's your name? > "); String newName = 
sc.next();
            System.out.print("Create a username for yourself > "); 
newUsername = sc.next();
            System.out.print("Now you have to make yourself a password, 
too.\nWant us to make one for you? Enter 1, else enter 0 > ");
            int ch4 = sc.nextInt();
            String newPassword;
            if (ch4 == 1) {
                newPassword = new String(passwordGenerator(6));
                System.out.println("Here's the password we made for you - " 
+ newPassword);
            } else {
                System.out.println("Enter your password > "); newPassword = 
sc.next();
            }

            while (true) {
                if (createNewProfile(newName, newUsername, newPassword) == 
1) {
                    System.out.println("User " + newName + " added - " + 
newUsername);
                    passedOne = 0;
                    break;

                } else {
                    System.out.print("Username " + newUsername + " is 
already in use\nPick another username, please > ");
                    newUsername = sc.next();
                    continue;
                }
            }


        break;
        }

    }

    }



        if (ch == 1) return (username);
        else return (newUsername);
```

```
     }

}
```

quotePrint.java

```
package projectFinal;

import java.util.Random;
import org.jsoup.*;
import org.jsoup.helper.*;
import org.jsoup.nodes.*;
import org.jsoup.select.*;

@SuppressWarnings("unused")
public class quotePrint {

     public String scraper() {

          Document doc = null;

          try {
               doc = Jsoup.connect("https://www.keepinspiring.me/positive-
quotes-and-sayings/").get();
               Elements quotes = doc.getElementsByTag("p");
               String quotesArr[] = new String[60];

               for (int i = 6, j = 0; i < 66; i++, j++)
                    quotesArr[j] = quotes.get(i).text();

               Random rand = new Random();
                return (quotesArr[rand.nextInt(60)]);
          } catch (Exception e) {
               System.out.println(" ");
          }

          return ("");

     }


     public void quotePrintFunction () {
          System.out.println("\n\n ~ " + scraper() + " ~ \n");
     }
}
```

vaultFilesSystem.java

```
package projectFinal;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.*;
import java.awt.Desktop;
import java.awt.desktop.*;


@SuppressWarnings("unused")
public class vaultFilesSystem {

      String username, name;

      vaultFilesSystem(String username, String name) {
            this.username = username;
            this.name = name;
      }

      public void makeNewEntry () {


            Scanner sc = new Scanner (System.in);

            System.out.print("Enter the name of the file you wish to add to your 
database > ");
            String fileName = sc.next();

            String ID = 
LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyyMMddHHmmss"));


            Connection connection;
            try {
```

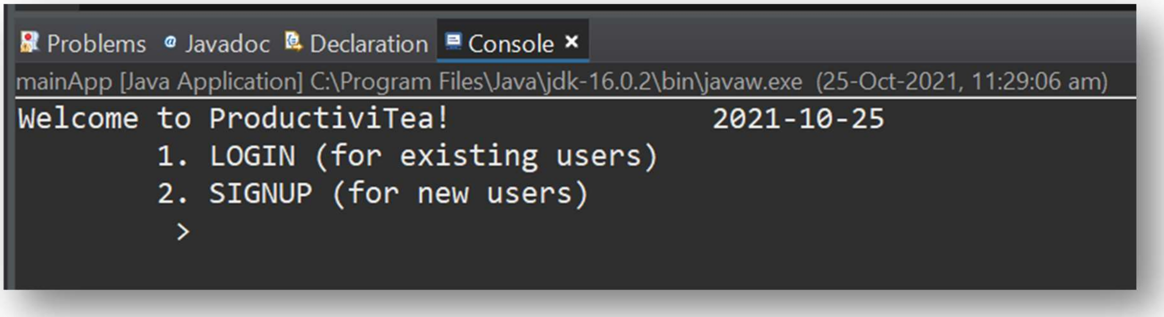
```
            connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root", 
"toor");
            Statement statement = connection.createStatement();
            String addThese = "\"" + ID + "\", \"" + fileName + "\", \"" + 
username + "\"";
            statement.executeUpdate("INSERT INTO files VALUES (" + addThese 
+");");

        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }


    }

    public void displayFileHeader(int n, String fileName) {
        String location = 
"C:\\Eclipse\\thirdTrial\\projectFinal\\src\\projectFinal\\Files\\";
        FileInputStream fin = null;

        System.out.print(n + " - ");

        try {
            try {
                fin = new FileInputStream(location + fileName);
            } catch (IOException e) {
                System.out.println(e);
            }

            int i = 0, stringPos = 0;

            do {
                    i = fin.read();
                    System.out.print((char)i);
                    stringPos++;

            } while (i!=-1 && stringPos<20);

            System.out.println("...");

        }   catch (IOException e8) {
            System.out.println("bleh");
        }   finally {
            try {
                if (fin!=null) fin.close();
            } catch (IOException e2) {
```

```
            System.out.println("Error closing the file");
            }
            }
      }

      public void openFiles (String ID) {
            Scanner sc = new Scanner(System.in);
            String location = 
"C:\\Eclipse\\thirdTrial\\projectFinal\\src\\projectFinal\\Files\\";
            Connection connection;

            String fileName = null;

            try {
                  connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root", 
"toor");
                  Statement statement = connection.createStatement();
                  ResultSet resultset = statement.executeQuery("select * from files 
where ID = \"" + ID + "\"");
                  int l = 0;
                  while (resultset.next()) {

                        fileName = resultset.getString("fileName");

                        char timestampArr[] = new char[20];
                        int j = 0, k = 0;

                        for (int i = 0; i < 19; i++) {
                              if (i == 4 || i == 7) timestampArr[j++] = '/';
                              else if (i == 10) timestampArr[j++] = ' ';
                              else if (i == 13 || i == 16) timestampArr[j++] = ':';
                              else {
                                    timestampArr[j++] = ID.charAt(k++);
                              }
                        }

                        String timestamp = new String(timestampArr);

                        FileInputStream fin = null;

                        if (l++ == 0) {

                              System.out.println("\n\nYour entry for " + timestamp + 
"... ");


                              try {
                                    try {
                                         fin = new FileInputStream(location +
fileName);
                                    } catch (IOException e) {
                                         System.out.println(e);
                                    }

                                    int i = 0;

                                    do {
                                               i = fin.read();
                                               System.out.print((char)i);

                                    } while (i!=-1);



                              }     catch (IOException e8) {
                                    System.out.println("bleh");
                              }     finally {
                                    try {
                                         if (fin!=null) fin.close();
                                    } catch (IOException e2) {
                                         System.out.println("Error closing the
file");
                                    }
                              }


                        }



                  }

                  System.out.print("\n\n\nWant to open this file for editing? Enter
1 to continue > ");
                  int ch = sc.nextInt();

                  if (ch == 1) {
                        File file = new File (location + fileName);
                        Desktop desktop = Desktop.getDesktop();
                        if (file.exists())
                              try {
                                    desktop.open(file);
                                    return;
                              } catch (IOException e) {
                                    // TODO Auto-generated catch block
                                    e.printStackTrace();
```

```
                    }


                }

            } catch (SQLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }


        }


        public void mainFunction () {

            Scanner sc = new Scanner(System.in);

            Connection connection;
            try {
                connection =
DriverManager.getConnection("jdbc:mysql://localhost:3307/first_jdbcthing","root",
"toor");
                Statement statement = connection.createStatement();
                ResultSet resultset = statement.executeQuery("select * from files
where username = \"" + username + "\"");
                int count = 1;
                while (resultset.next()) {
                    try {
                        displayFileHeader(count++,
resultset.getString("fileName"));
                    } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }

                resultset = statement.executeQuery("select * from files where
username = \"" + username + "\"");

                if (count>1) {
                    System.out.print("Wish to open these entries? Enter 1 > ");
                    int ch1 = sc.nextInt();
                    if (ch1 == 1) {
                        System.out.print("Enter the serial number of the entry
you wish to see > ");
                        int tempID = sc.nextInt();
                        int p = 0;
                        while (resultset.next()) {
                            try {
                                if (p == tempID-1) {
                                    openFiles(resultset.getString("ID"));
                                } else p++;
                            } catch (Exception e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                            }
                        }

                    }

                }

                if (count == 1) {
                    System.out.println("No files found! Let's make you a new 
one!");
                    makeNewEntry(); mainFunction();
                } else {
                    System.out.print("Let's make a new entry for today? Enter 1 
to continue > ");
                    int ch = sc.nextInt();
                    if (ch == 1) { makeNewEntry(); mainFunction();}

                }


            } catch (SQLException Illegaloperationonemptyresultset) {
                // TODO Auto-generated catch block
                makeNewEntry();
                mainFunction();
            }


        }

}
```

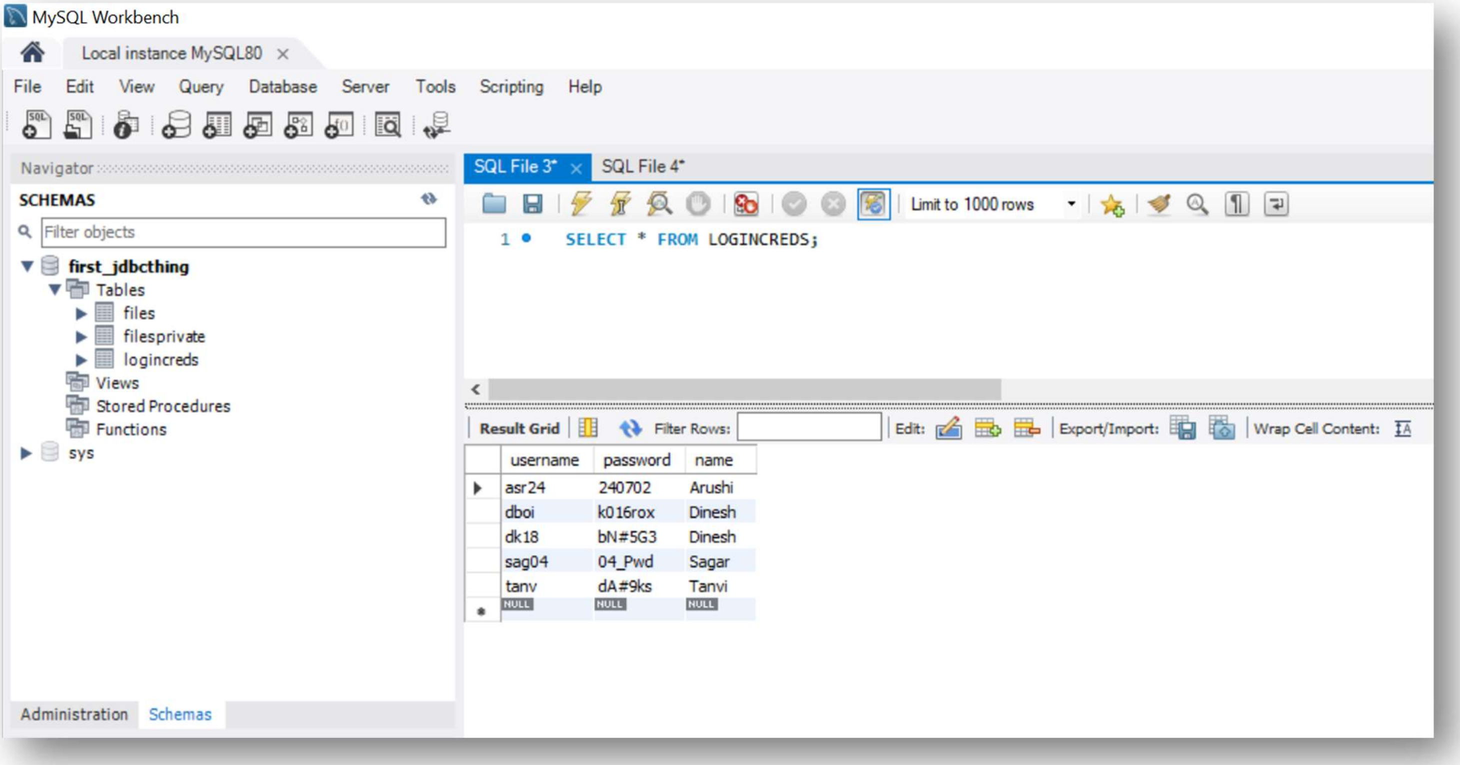
Flow of the Program

The program starts with a main menu, providing an option of login to an existing account or signing up by creating a new account, and proceeds with the account of the newly created account or the account the user logged in to. If the account is newly created, the login will not be asked to done separately.



```
mainApp [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (25-Oct-2021, 11:29:06 am)
Welcome to ProductiviTea!                2021-10-25
        1. LOGIN (for existing users)
        2. SIGNUP (for new users)
         >
```

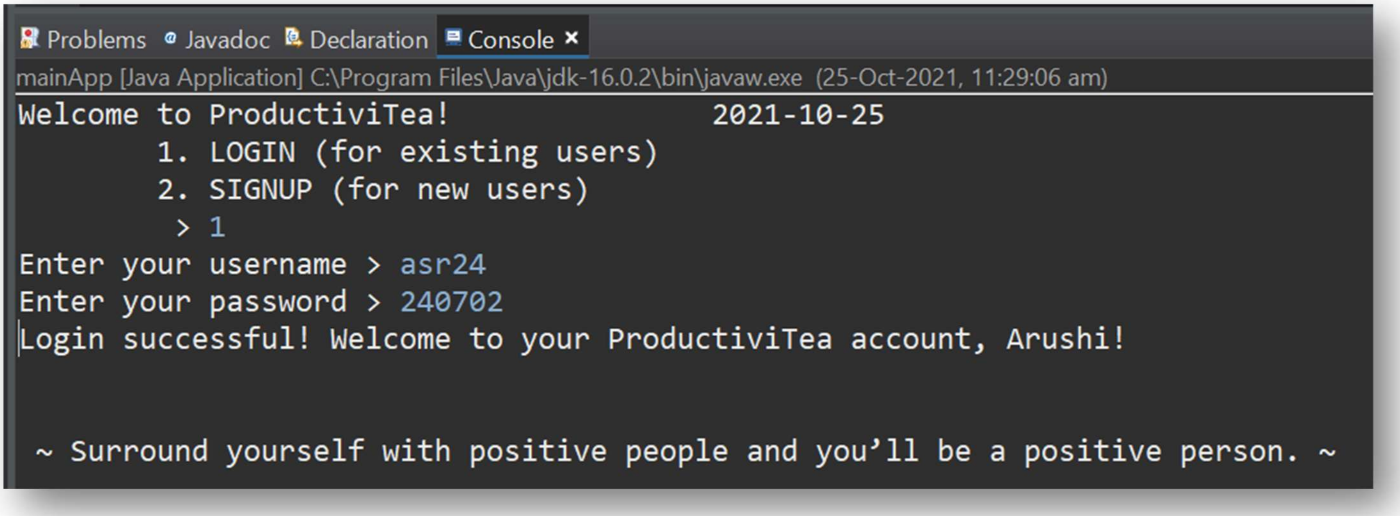
On the backend, the entered credentials are verified from the database.



```sql
SELECT * FROM LOGINCREDS;
```

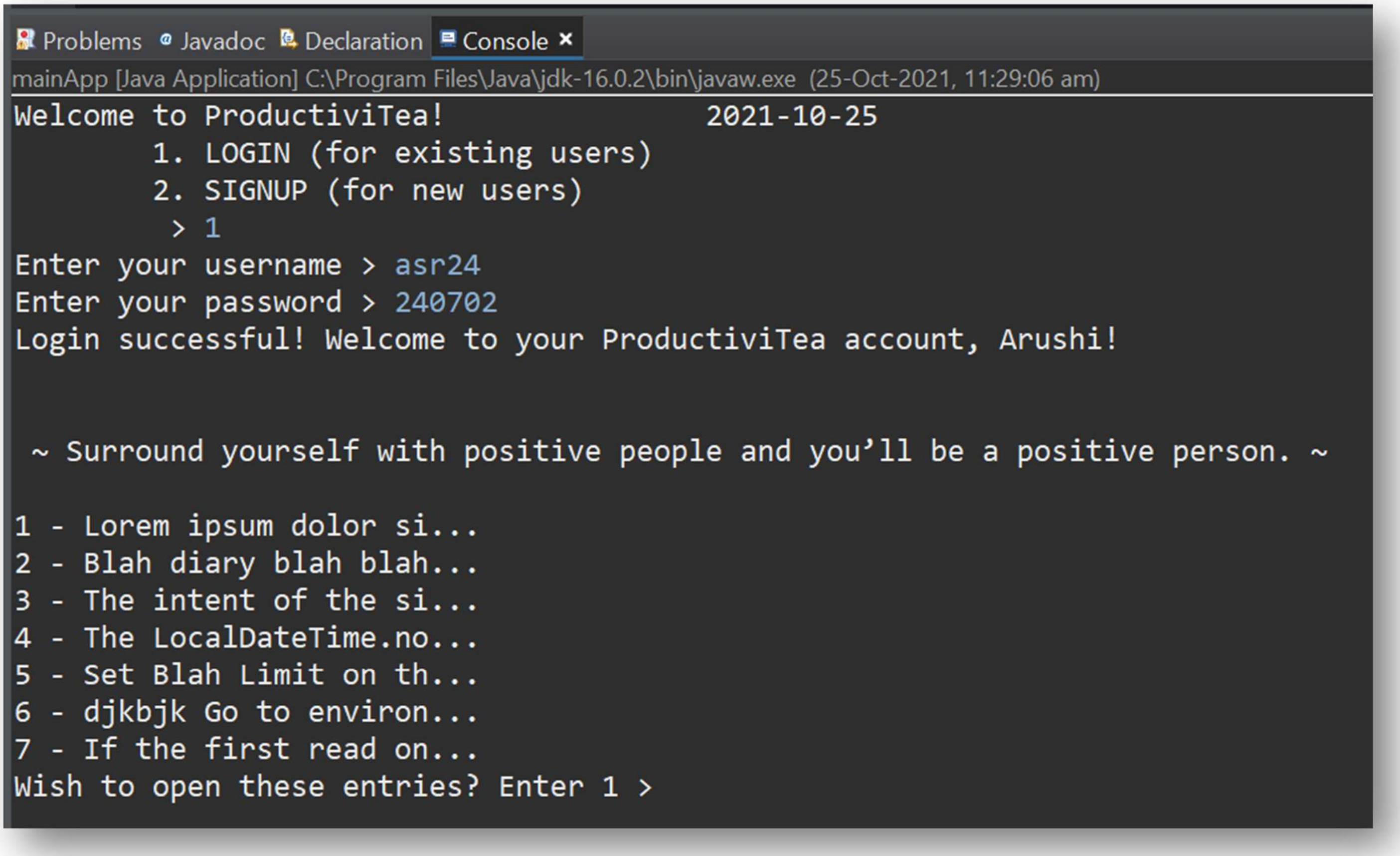
username	password	name
asr24	240702	Arushi
dboi	k016rox	Dinesh
dk18	bN#5G3	Dinesh
sag04	04_Pwd	Sagar
tanv	dA#9ks	Tanvi
NULL	NULL	NULL

Then, a random positive quote is displayed



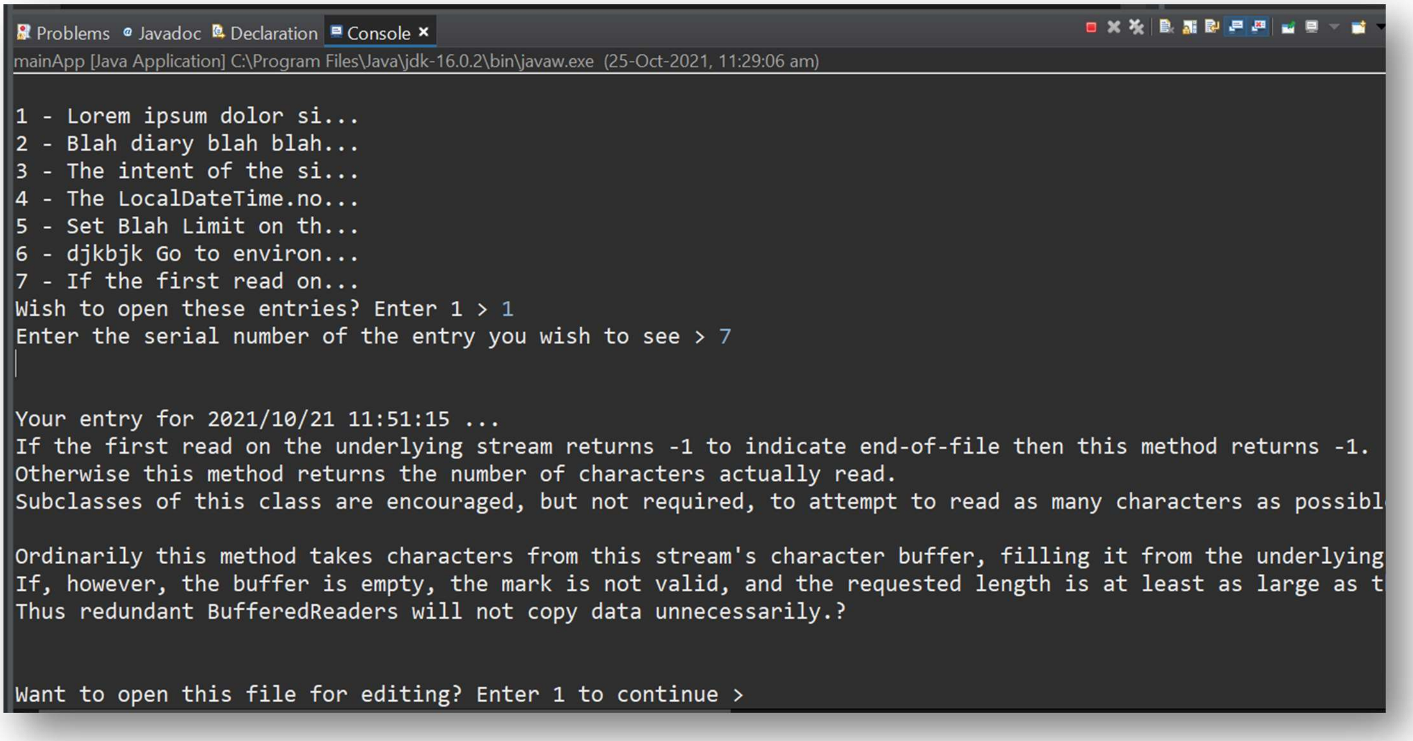
```
mainApp [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (25-Oct-2021, 11:29:06 am)
Welcome to ProductiviTea!                   2021-10-25
        1. LOGIN (for existing users)
        2. SIGNUP (for new users)
         > 1
Enter your username > asr24
Enter your password > 240702
Login successful! Welcome to your ProductiviTea account, Arushi!


 ~ Surround yourself with positive people and you’ll be a positive person. ~
```

After that, we see a list of all the journal entries that the particular user has made.



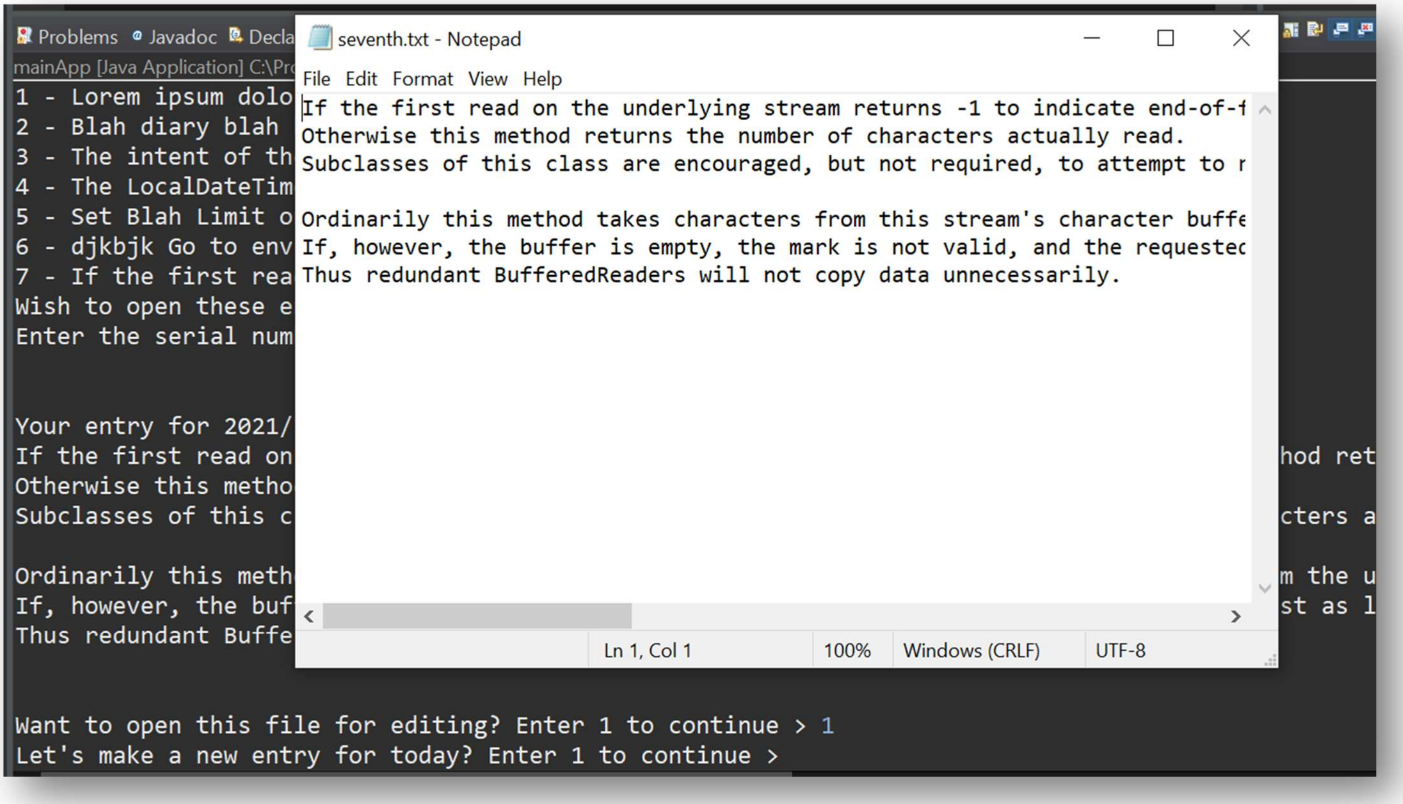
```
mainApp [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (25-Oct-2021, 11:29:06 am)
Welcome to ProductiviTea!                   2021-10-25
        1. LOGIN (for existing users)
        2. SIGNUP (for new users)
         > 1
Enter your username > asr24
Enter your password > 240702
Login successful! Welcome to your ProductiviTea account, Arushi!


 ~ Surround yourself with positive people and you’ll be a positive person. ~

1 - Lorem ipsum dolor si...
2 - Blah diary blah blah...
3 - The intent of the si...
4 - The LocalDateTime.no...
5 - Set Blah Limit on th...
6 - djkbjk Go to environ...
7 - If the first read on...
Wish to open these entries? Enter 1 >
```

We have the choice to open up any one of these entries, too.



```
mainApp [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (25-Oct-2021, 11:29:06 am)

1 - Lorem ipsum dolor si...
2 - Blah diary blah blah...
3 - The intent of the si...
4 - The LocalDateTime.no...
5 - Set Blah Limit on th...
6 - djkbjk Go to environ...
7 - If the first read on...
Wish to open these entries? Enter 1 > 1
Enter the serial number of the entry you wish to see > 7


Your entry for 2021/10/21 11:51:15 ...
If the first read on the underlying stream returns -1 to indicate end-of-file then this method returns -1.
Otherwise this method returns the number of characters actually read.
Subclasses of this class are encouraged, but not required, to attempt to read as many characters as possibl

Ordinarily this method takes characters from this stream's character buffer, filling it from the underlying
If, however, the buffer is empty, the mark is not valid, and the requested length is at least as large as t
Thus redundant BufferedReaders will not copy data unnecessarily.?


Want to open this file for editing? Enter 1 to continue >
```

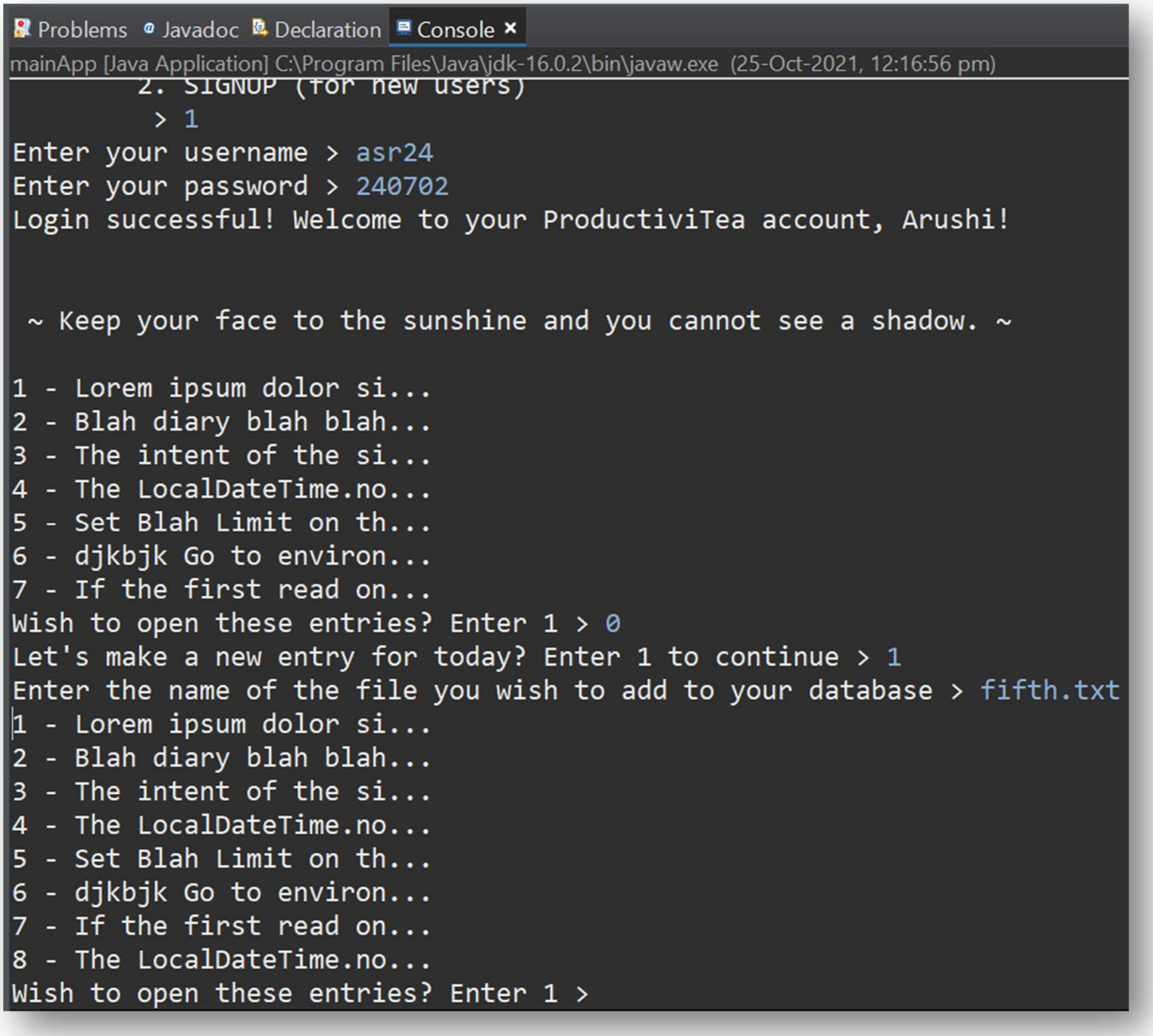
If we wish to edit this file, we can press 1 here, which will automatically open up the text file on Notepad for us.



```
mainApp [Java Application] C:\Pro
1 - Lorem ipsum dolo
2 - Blah diary blah
3 - The intent of th
4 - The LocalDateTim
5 - Set Blah Limit o
6 - djkbjk Go to env
7 - If the first rea
Wish to open these e
Enter the serial num


Your entry for 2021/
If the first read on                                                  hod ret
Otherwise this metho
Subclasses of this c                                                  cters a

Ordinarily this meth                                                  m the u
If, however, the buf                                                  st as l
Thus redundant Buffe


Want to open this file for editing? Enter 1 to continue > 1
Let's make a new entry for today? Enter 1 to continue >
```

```
If the first read on the underlying stream returns -1 to indicate end-of-f
Otherwise this method returns the number of characters actually read.
Subclasses of this class are encouraged, but not required, to attempt to r

Ordinarily this method takes characters from this stream's character buffe
If, however, the buffer is empty, the mark is not valid, and the requestec
Thus redundant BufferedReaders will not copy data unnecessarily.
```

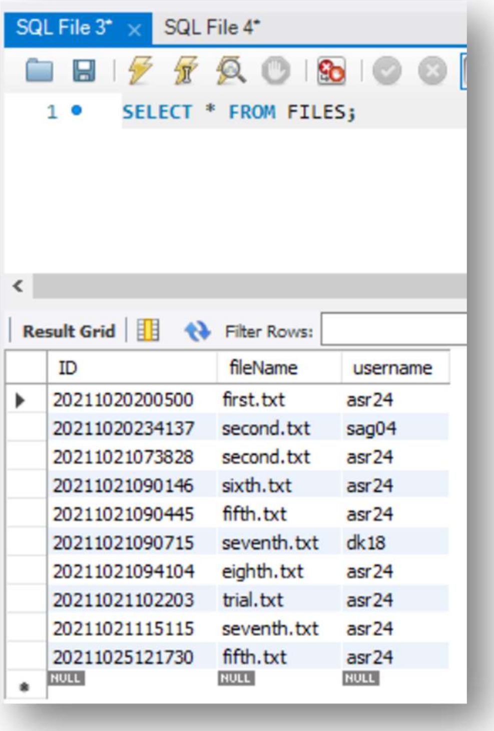
We can then also add a new entry by entering the name of the file it is saved on.



mainApp [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (25-Oct-2021, 12:16:56 pm)

```
        2. SIGNUP (for new users)
         > 1
Enter your username > asr24
Enter your password > 240702
Login successful! Welcome to your ProductiviTea account, Arushi!


 ~ Keep your face to the sunshine and you cannot see a shadow. ~

1 - Lorem ipsum dolor si...
2 - Blah diary blah blah...
3 - The intent of the si...
4 - The LocalDateTime.no...
5 - Set Blah Limit on th...
6 - djkbjk Go to environ...
7 - If the first read on...
Wish to open these entries? Enter 1 > 0
Let's make a new entry for today? Enter 1 to continue > 1
Enter the name of the file you wish to add to your database > fifth.txt
1 - Lorem ipsum dolor si...
2 - Blah diary blah blah...
3 - The intent of the si...
4 - The LocalDateTime.no...
5 - Set Blah Limit on th...
6 - djkbjk Go to environ...
7 - If the first read on...
8 - The LocalDateTime.no...
Wish to open these entries? Enter 1 >
```

The new entries are added to the database like so:



```
SELECT * FROM FILES;
```

ID	fileName	username
20211020200500	first.txt	asr24
20211020234137	second.txt	sag04
20211021073828	second.txt	asr24
20211021090146	sixth.txt	asr24
20211021090445	fifth.txt	asr24
20211021090715	seventh.txt	dk18
20211021094104	eighth.txt	asr24
20211021102203	trial.txt	asr24
20211021115115	seventh.txt	asr24
20211025121730	fifth.txt	asr24
NULL	NULL	NULL