Course Project – Final Submission

Jeramy Dichiera

Rasmussen College

Author Note

This research is being submitted on May 7th, 2017, for Anastasia Rashtchian’s COP2268C Java Programming course.

This program is a console interface that allows users to connect to and manage a user stock profile.

Abstract User Class File

public abstract class AbstractUser {

String userID;

String firstName;

String lastName;

String password;

int admin;

public AbstractUser(String newUserId, String newFirstName, String newLastName,

String newPassword){

userID = newUserId;

firstName = newFirstName;

lastName = newLastName;

password = newPassword;

}

public String getUserID() {

return userID;

}

public void setUserID(String userID) {

this.userID = userID;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

abstract void printUserType();

}

Admin User Class File

*/\*\**

*\* Created by udichje on 5/7/17.*

*\*/*

public class AdminUser extends AbstractUser{

int admin;

public AdminUser(String newUserId, String newFirstName, String newLastName,

String newPassword, int admin){

super(newUserId, newFirstName, newLastName, newPassword);

admin = admin;

}

public int getIsAdmin() {

return admin;

}

public void setAdmin(int admin) {

this.admin = admin;

}

public void printUserType(){

System.*out*.println("Admin user");

}

}

User Class File

public class User extends AbstractUser{

int admin;

public User(String newUserId, String newFirstName, String newLastName,

String newPassword, int admin){

super(newUserId, newFirstName, newLastName, newPassword);

admin = admin;

}

public int getIsAdmin() {

return admin;

}

public void setAdmin(int admin) {

this.admin = admin;

}

public void printUserType(){

System.*out*.println("Normal user");

}

}

Index HTML File For Servlet

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Stock Application Login</title>

</head>

<body>

<form name="logonScreen" method="POST" action="../servlet/Servlet">

<table width="328" height="206">

<tr>

<td width="126">User ID

<input type="text" name="userID">

</td>

<td width="126">Password

<input type="password" name="password">

</td>

<td>

<input type="button" value="Log On"

onClick="validate(logonScreen)">

</td>

</tr>

</table>

</form>

</body>

</html>

Servlet file

*/\*\**

*\* Created by udichje on 5/7/17.*

*\*/*

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

import java.sql.\*;

import java.util.\*;

import java.net.\*;

public class Servlet extends HttpServlet{

StockDatabase database;

public void init() throws ServletException{

try {

database = new StockDatabase();

} catch (SQLException e) {

System.*out*.println("Servlet : SQL Exception creating new DB " +

"object.");

e.printStackTrace();

System.*exit*(1);

} catch (ClassNotFoundException e){

System.*out*.println("Servlet : Class NotFoundException creating " +

"new DB object");

e.printStackTrace();

System.*exit*(1);

}

}

public void destroy(){

try

{

database.close();

}

catch (IOException e) {

System.*out*.println("IOException closing database:");

e.printStackTrace();

System.*exit*(1);

}

catch (SQLException e) {

System.*out*.println("SQLException closing database:");

e.printStackTrace();

System.*exit*(1);

}

catch (ClassNotFoundException e) {

System.*out*.println("ClassNotFoundException closing database:");

e.printStackTrace();

System.*exit*(1);

}

}

public void doGet(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException{

doPost(request, response);

}

public void doPost(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException{

User user = null;

String userID;

String password;

boolean validAction = true;

response.setContentType("text/html");

PrintWriter out = response.getWriter();

response.setHeader("Expires", "Tues, 01 Jan 1980 00:00:00 GMT");

HttpSession session = request.getSession(true);

}

}

StockDatabase Class File

*/\*\**

*\* Created by udichje on 5/7/17.*

*\*/*

import java.io.\*;

import java.sql.\*;

import java.util.\*;

public class StockDatabase

{

private Connection connection = null;

public StockDatabase() throws ClassNotFoundException, SQLException{

if(connection == null){

String driverPath = "org.sqlite.JDBC";

String databasePath = "jdbc:sqlite:StockDatabase.db";

try

{

Class.*forName*(driverPath);

}

catch (ClassNotFoundException e)

{

System.*out*.println("Connection error ... :");

throw new ClassNotFoundException(e.getMessage());

}

try

{

connection = DriverManager.*getConnection*(databasePath);

}

catch (SQLException e)

{

throw new SQLException(e.getMessage());

}

}

}

public void close() throws SQLException,IOException,ClassNotFoundException{

connection.close();

connection = null;

}

private byte[] serializeObject(Object obj) throws IOException{

ByteArrayOutputStream baOStream = new ByteArrayOutputStream();

ObjectOutputStream objOStream = new ObjectOutputStream(baOStream);

objOStream.writeObject(obj);

objOStream.flush();

objOStream.close();

return baOStream.toByteArray();

}

private Object deserializeObj(byte[] buf) throws IOException,

ClassNotFoundException{

Object obj = null;

if (buf != null){

ObjectInputStream objIStream = new ObjectInputStream((new

ByteArrayInputStream(buf)));

obj = objIStream.readObject();

}

return obj;

}

public void addStock(String stockSymbol, String stockDescription) throws

SQLException, IOException, ClassNotFoundException{

Statement statement = connection.createStatement();

statement.executeUpdate("INSERT INTO Stocks VALUES ('" + stockSymbol

+ "'" + ",'" + stockDescription + "')");

statement.close();

}

public boolean addUser(User user) throws SQLException, IOException,

ClassNotFoundException{

boolean result = false;

String dbUserID;

String dbFirstName;

String dbLastName;

String dbPassword;

int isAdmin;

dbUserID = user.getUserID();

if (getUser(dbUserID) == null){

dbFirstName = user.getFirstName();

dbLastName = user.getLastName();

dbPassword = user.getPassword();

isAdmin = user.getIsAdmin();

PreparedStatement pStatement = connection.prepareStatement

("INSERT INTO Users VALUES (?,?,?,?,?)");

pStatement.setString(1, dbUserID);

pStatement.setString(2, dbFirstName);

pStatement.setString(3, dbLastName);

pStatement.setString(4, dbPassword);

pStatement.setInt(5, isAdmin);

pStatement.executeUpdate();

pStatement.close();

result = true;

}

else {

throw new IOException("User exists - cannot add.");

}

return result;

}

public boolean updateUser(User user) throws SQLException, IOException,

ClassNotFoundException{

boolean result = false;

String dbUserID;

String dbFirstName;

String dbLastName;

String dbPassword;

int isAdmin;

dbUserID = user.getUserID();

if (getUser(dbUserID) == null){

dbFirstName = user.getFirstName();

dbLastName = user.getLastName();

dbPassword = user.getPassword();

isAdmin = user.getIsAdmin();

PreparedStatement pStatement = connection.prepareStatement

("UPDATE Users SET firstName = ?, lastName = ?, password " +

"= ?, admin = ? WHERE userID = ?");

pStatement.setString(1, dbUserID);

pStatement.setString(2, dbFirstName);

pStatement.setString(3, dbLastName);

pStatement.setString(4, dbPassword);

pStatement.setInt(5, isAdmin);

pStatement.executeUpdate();

pStatement.close();

result = true;

}

else {

throw new IOException("User cannot be updated.");

}

return result;

}

private void delStock(String stockSymbol) throws SQLException,

IOException, ClassNotFoundException{

Statement statement = connection.createStatement();

statement.executeUpdate("DELETE FROM Stocks WHERE symbol = '" +

stockSymbol + "'");

statement.close();

}

public void delUser(User user) throws SQLException, IOException,

ClassNotFoundException{

String dbUserID;

String stockSymbol;

Statement statement = connection.createStatement();

try

{

connection.setAutoCommit(false);

dbUserID = user.getUserID();

if(getUser(dbUserID) != null){

ResultSet result1 = statement.executeQuery("SELECT userID, " +

"symbol FROM UserStocks WHERE userID = '" + dbUserID

+ "'");

while(result1.next()){

try

{

stockSymbol = result1.getString("symbol");

delUserStocks(dbUserID, stockSymbol);

}

catch (SQLException e)

{

throw new SQLException("Delection of user stock " +

"holding failed: " + e.getMessage());

}

}

try

{

statement.executeUpdate("DELETE FROM Users WHERE " +

"userID = '" + dbUserID + "'");

}

catch (SQLException e)

{

throw new SQLException("User deletion failed: "

+ e.getMessage());

}

}

else{

throw new IOException("User not found in database - cannot " +

"delete.");

}

try

{

connection.commit();

}

catch (SQLException e)

{

throw new SQLException("Transaction commit failed : "

+ e.getMessage());

}

}

catch (SQLException e)

{

try

{

connection.rollback();

}

catch (SQLException se)

{

throw new SQLException(("Transaction failed then rollback " +

"failed: " + se.getMessage()));

}

throw new SQLException("Transaction failed; was rolled back: " +

e.getMessage());

}

statement.close();

}

public void delUserStocks(String userID, String stockSymbol) throws

SQLException, IOException, ClassNotFoundException{

Statement statement = connection.createStatement();

ResultSet results;

statement.executeUpdate("DELETE FROM UserStocks WHERE " +

"userID = '" +

userID + "'" + "AND symbol = '" + stockSymbol + "'");

results =statement.executeQuery("SELECT symbol FROM UserStocks WHERE " +

"symbol = " + stockSymbol + "'");

if(!results.next()){

delStock(stockSymbol);

}

statement.close();

}

public void addUserStock(String userID, String stockSymbol) throws

SQLException, IOException, ClassNotFoundException{

Statement statement = connection.createStatement();

System.*out*.println("INSERT INTO UserStocks VALUES ("+

userID + ",'" + stockSymbol + "'");

statement.executeUpdate("INSERT INTO UserStocks VALUES ("+

userID + ",'" + stockSymbol + "')");

statement.close();

}

public String getStockDescription(String stockSymbol) throws

SQLException, IOException, ClassNotFoundException{

Statement statement = connection.createStatement();

String stockDescription = null;

ResultSet results = statement.executeQuery("SELECT symbol, name FROM " +

"Stocks WHERE symbol = '" + stockSymbol + "'");

if(results.next()){

stockDescription = results.getString("name");

}

results.close();

statement.close();

return stockDescription;

}

public User getUser(String userID) throws SQLException, IOException,

ClassNotFoundException{

Statement statement = connection.createStatement();

String dbUserID;

String dbFirstName;

String dbLastName;

String dbPassword;

int isAdmin;

byte[] buf = null;

User user = null;

ResultSet results = statement.executeQuery("SELECT \* FROM Users WHERE" +

" userID = '" + userID + "'");

if(results.next()){

dbUserID = results.getString("userID");

dbFirstName = results.getString("firstName");

dbLastName = results.getString("lastName");

dbPassword = results.getString("password");

isAdmin = results.getInt("admin");

user = new User(dbUserID, dbFirstName, dbLastName, dbPassword,

isAdmin);

}

results.close();

statement.close();

return user;

}

public ArrayList listUsers() throws SQLException, IOException,

ClassNotFoundException{

ArrayList list = new ArrayList();

Statement statement = connection.createStatement();

ResultSet results = statement.executeQuery("SELECT userID, firstName," +

" lastName, admin FROM Users ORDER BY userID");

while(results.next()){

list.add(results.getString("userID"));

list.add(results.getString("firstName"));

list.add(results.getString("lastName"));

list.add(results.getInt("admin"));

}

results.close();

statement.close();

return list;

}

public ArrayList listUserStocks(String userID) throws SQLException,

IOException, ClassNotFoundException{

ArrayList list = new ArrayList();

Statement statement = connection.createStatement();

ResultSet results = statement.executeQuery("SELECT \* FROM UserStocks " +

"WHERE userID = " + userID + " ORDER BY symbol");

System.*out*.println("++ User Stock List ++");

while(results.next()){

list.add(results.getString("symbol"));

}

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

results.close();

statement.close();

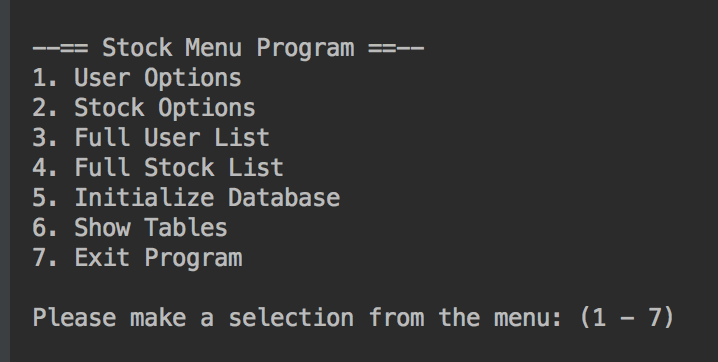
return list;

}

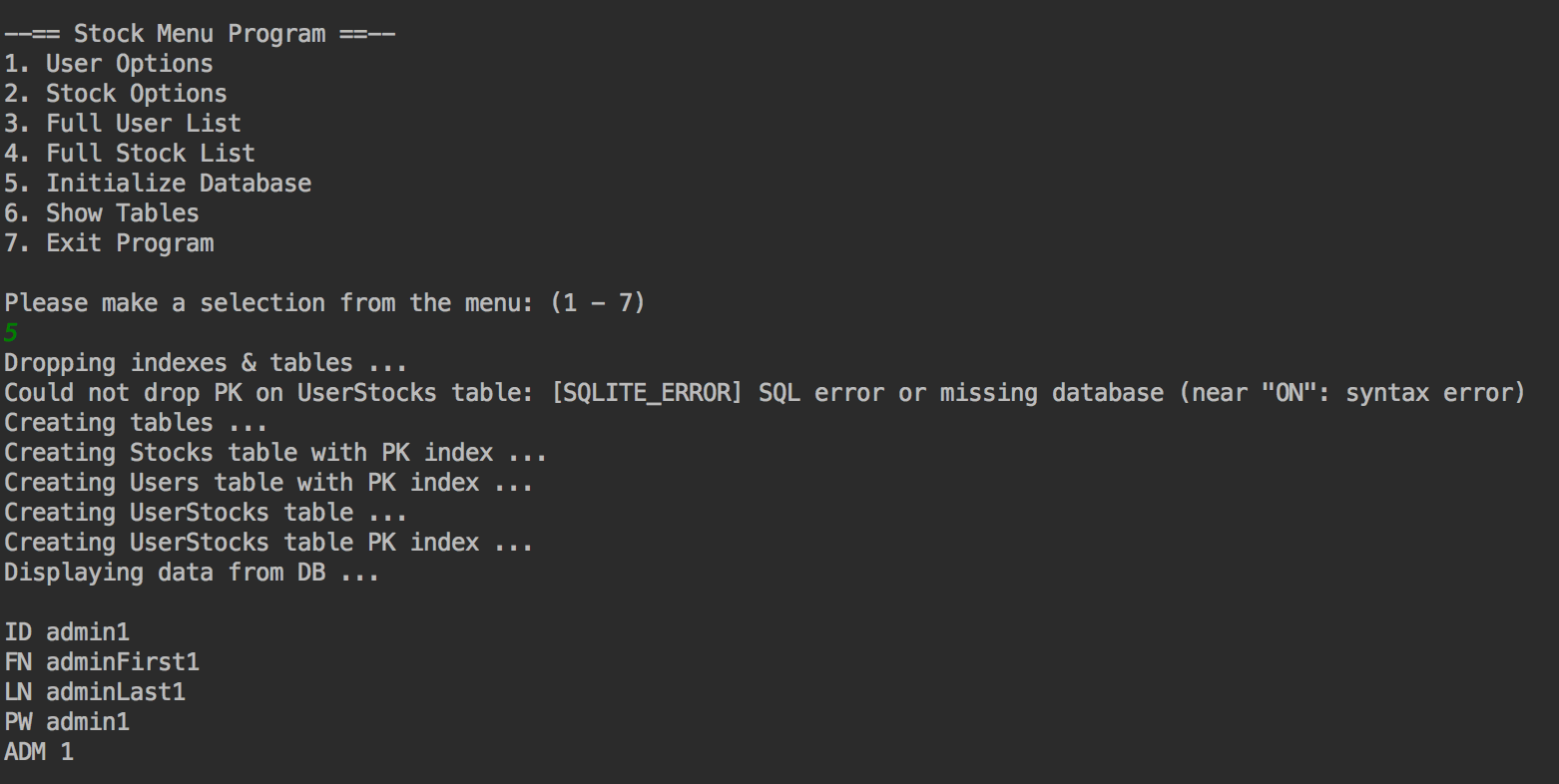
}

Program in Action

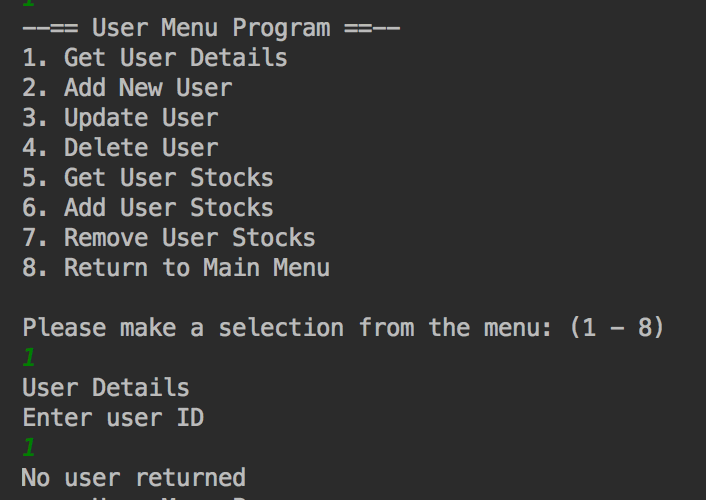
Main Menu:



Initialize Database:



Retrieve User From DB



Add user to DB

