

## Assignment 3

Title: Application development using JDBC & concurrency.

Problem Statement:

Enhance the system developed in previous assignments. by using JDBC, Multithreading, concurrency, synchronous & asynchronous callbacks, Thread Pools using Executor service.

Objective:

To understand database connectivity using java programs & to learn concurrency & it's application.

Outcome:

Students should be able to implement various types of JDBC drivers & concurrency in their application using classes & interfaces in java concurrent packages.

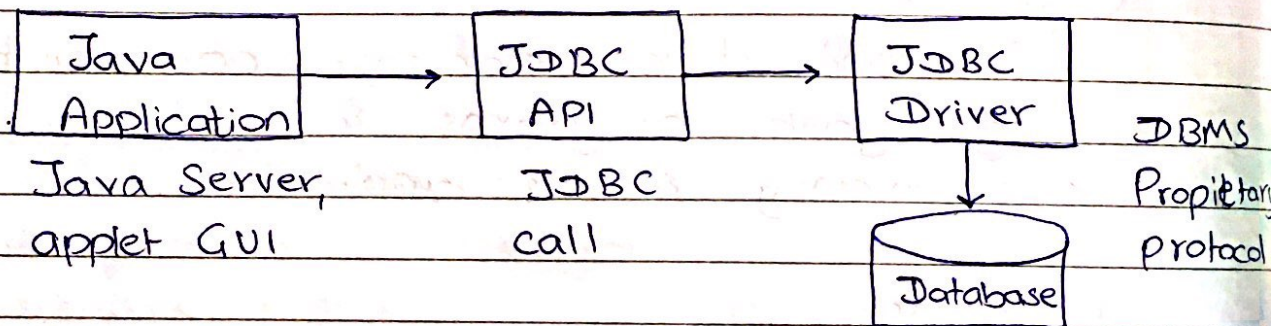
Software & Hardware Package:

- Java SE 11
- Java IDE
- 64 bit OS: ubuntu 20.04
- Processor: intel i5
- i/o devices



## Theory:

Java Database connectivity (JDBC) is an application programming Interface (API) for programming language JAVA which defines how a client may access database. It provides methods for query & update data in a database, & it's oriented towards relational database.



## JDBC Drivers:

JDBC drivers implement the defined interfaces in the JDBC API for interacting with database server.

JDBC driver implementation vary because of the wide variety of operating system & hardware platforms the implementation type can be divided into four categories.



## Type 1: JDBC - ODBC Bridge driver

This is the oldest JDBC driver mostly used to connect database like ms Access from windows. These days this is only used for experimental purposes.

## Type 2: JDBC - Native API

JDBC API calls are converted into native C/C++ API calls, which are unique to the database.

## Type 3: JDBC - Net Java

JDBC clients use standard network sockets to communicate with a middleware application system.

## Type 4: Pure Java

This type of driver is implemented in Java & directly connects to database using its native protocol.

## • Fundamental Steps in JDBC Application

- 1) import the packages
- 2) Load & Register the driver.
- 3) Establish a connection
- 4) Create statement object
- 5) Execute Query
- 6) Process resultSet
- 7) close connection



Concurrency:

It is the ability to run several programs on several parts of a program in parallel.

Thread: It is the facility to allow multiple activities within a single process, every thread has its own call stack but can access shared data of other threads in the same process.

Thread lifecycle

- 1) Newborn
- 2) Running - running on processor core
- 3) Runnable - waiting for access of processor core
- 4) Blocked - thread is suspended
- 5) Dead - Execution of thread is stopped.

Thread Pool-

All threads are in thread pool. Threads are assigned tasks & after completion are returned to the thread pool.

Thread Synchronizations:

- 1) Reentrant locks:

Implements lock interface & provides synchronization to methods while accessing shared resource.



2) lock():

call the lock() methods increments the hold count by 1 & gives the lock to thread if resource was initially free.

3) unlock();

decrements the hold count by 1 & frees the resource.

4) trylock();

if resource is not held by other thread then call to trylock() returns true & hold count is incremented by 1, if resource is not free then method returns false & thread is not blocked but it exists.

### • Testing:

testcases	expected output	result
1) - multiple clients Connecting multiple client programs	connects server to different threads	Success
2) - Database access Update/retrieve data operations.	Returns resultset or changes database accordingly	Success
3) - Synchronization login	accepts credentials & client can login	Success.



### Conclusion:

We learnt about concurrency & classes & interfaces. Database connectivity & types of ~~db~~ JDBC drivers. & thus added a MySQL database for the same, with multiple client support via Multithreading.

# Code

```
import java.sql.*;
import java.util.*;
```

```
public class account_handling {
    static market_database_handling mdh = new market_database_handling();
    Scanner in = new Scanner(System.in);

    boolean username_already_exists(String user_name){
        String query = "select user_name from users where user_name='"+user_name+ "'";

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            if (!rs.next()){
                con.close();
                return false;
            }
            else{
                System.out.println("** username exists **");
                con.close();
                return true;
            }
        }catch(Exception e){System.out.println(e);}

        return true;
    }
}
```

```
String new_user_login(String user_name, String password){
    if(username_already_exists(user_name)){
        return "err:usernameExists";
    }
}
```

```
try{
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con =
    DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
    Statement stmt = con.createStatement();
    String updt = "insert into users values('"+user_name+ "','"+ password + "')";
    stmt.executeUpdate(updt);
    con.close();
    System.out.println("*** account created ***");
} catch(Exception e){System.out.println(e);}
return "success";
}
```

```

String take_username(){
String user_name;
System.out.println("Enter Username : ");
user_name = in.nextLine();
return user_name;
}
String take_password(){
String password;
System.out.println("Enter Password : ");
password = in.nextLine();
return password;
}
String old_user_login(String user_name,String password){
String retrieved="";
if(!username_already_exists(user_name)){
return "err:username";
}
try{
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
Statement stmt = con.createStatement();
String query = "select password from users where user_name='"+user_name+"'";
ResultSet rs = stmt.executeQuery(query);
rs.next();
retrieved = rs.getString(1);
con.close();

}catch(Exception e){System.out.println(e);}

if( !password.equals(retrieved) ){
return "err:password";
}
System.out.println("** Successful login **");
if(user_name.equals("admin")){
return "admin";
}
return "customer";
}

}

```

```

import java.sql.*;
import java.util.*;
import org.apache.ibatis.jdbc.ScriptRunner;
import java.io.FileReader;
import java.io.Reader;
import java.io.BufferedReader;

```

```

public class market_database_handling {
Scanner in = new Scanner(System.in);
static account_handling acch = new account_handling();
static Client cli = new Client();

```



```

void get_market_status() {
    String query = "SELECT * FROM Market";

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("\\n\\n*** WELCOME TO BIG-BASKET ***");
        System.out.printf("\\n| %-15s| %10s |%10s | %10s |\\n","Item Name", "QTY", "Expires in", "Cost");
        System.out.println("_____");
        if(rs.next()) {
            do {
                System.out.printf("| %-15s| %10s |%10s days | %10s |\\n",
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4)
                );
            } while(rs.next());
        }
        else {
            System.out.println("Record Not Found...");
        }
        System.out.println("_____");
        con.close();
    } catch(Exception e) { System.out.println(e); }
}

```

```

void pass_day() {
    String updt_expiry = "update Market set expiry = IF(expiry<=1, 0, expiry-1);";
    String updt_quantity = "update Market set quantity = IF(expiry<1, 0, quantity)";
    String updt_cost = "update Market set cost = IF(expiry<1, 0, cost)";
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
        Statement stmt = con.createStatement();
        stmt.executeUpdate(updt_expiry);
        stmt.executeUpdate(updt_quantity);
        stmt.executeUpdate(updt_cost);
        System.out.println("A DAY HAS PASSED !!!");
        con.close();
    } catch(Exception e) { System.out.println(e); }
}

```

```

int continue_shopping() {
    System.out.println("login/leave shop? (1/0) : ");
    int repeat = Integer.parseInt(in.nextLine());
    return repeat;
}

```

```

int choose_login_options() {
    System.out.println("1. Login");
}

```



```

System.out.println("2. Signup");
System.out.print("\nChoice : ");
int new_acc = Integer.parseInt(in.nextLine());
return new_acc;
}

```

```

void display_respective_menu(String status){
if(status.equals("customer")){ customer_menu(); }
if(status.equals("admin")){ admin_menu(); }
}

```

```

void admin_menu(){
int choice;
System.out.println("*** Admin Mode ***");
System.out.println("Order default/ Order Specific ? (1/0)");
choice = Integer.parseInt(in.nextLine());
if(choice == 1){
run_default_market();
}
else{
System.out.println("Re-stock per category :");

```

```

int repeat = 1;
while(repeat != 0){

```

```

Integer q, exp, c; //quantity expiry cost
String item_name;

```

```

System.out.println("Item Name : ");
item_name = in.nextLine();
System.out.println("Restock Quantity : ");
q = Integer.parseInt(in.nextLine());
System.out.println("New Expiry : ");
exp = Integer.parseInt(in.nextLine());
System.out.println("New Cost : ");
c = Integer.parseInt(in.nextLine());
System.out.println("\nRestocking...");
String updt = "update Market set quantity = '"+q+"', expiry = '"+exp+"', cost = '"+c+" where name = '"+item_name+"';";
try{
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
Statement stmt = con.createStatement();
stmt.executeUpdate(updt);
System.out.println("Restocked !");
con.close();
}catch(Exception e){System.out.println(e);}

```

```

System.out.println("Restock another item ?(1/0) ");
repeat = Integer.parseInt(in.nextLine());
}
}
}

```

```

void customer_menu(){

```



```

Integer count;
System.out.print("\nNo. of items you want to BUY : ");
count = Integer.parseInt(in.nextLine());
int total_cost = 0;
String[] items = new String[20];
int[] quantities = new int[20];

for(int ct=0; ct<count; ct++){

String item_name;
Integer quan;
System.out.println("Item Name : ");
item_name = in.nextLine();
System.out.println("Quantity : ");
quan = Integer.parseInt(in.nextLine());

items[ct] = item_name;
quantities[ct]=quan;
total_cost = total_cost + buy_item(item_name, quan);
}

System.out.println("-----");
System.out.println("YOUR BASKET : \n");
for(int ct=0; ct<count; ct++){
System.out.println(items[ct] + " : " + quantities[ct] );
}
System.out.println("cost : " + total_cost);
System.out.println("\n\nThanks for supporting us!");
System.out.println("-----");
}

int buy_item(String name, Integer quan){
String updt_quantity = "update Market set quantity = IF(quantity-'"+quan+"'<1, 0, quantity-'"+quan+"'") where
name='"+name+"';";
String updt_cost = "update Market set cost = IF(quantity<1, 0, cost) where name='"+name+"';";
String updt_exp = "update Market set expiry = IF(quantity<1, 0, expiry) where name='"+name+"';";
String query_cost = "select cost from Market where name='"+name+"';";
int cost=0;
try{
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
Statement stmt = con.createStatement();

stmt.executeUpdate(updt_quantity);
stmt.executeUpdate(updt_cost);
stmt.executeUpdate(updt_exp);

ResultSet rs = stmt.executeQuery(query_cost);
rs.next();
cost = rs.getInt(1);

con.close();
}catch(Exception e){System.out.println(e);}
}

```



```

return (cost*quan);
}

void run_default_market(){
//Getting the connection
try{
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bb_accounts","root","Hello@123");
System.out.println("Connection established.....");
//Initialize the script runner
ScriptRunner sr = new ScriptRunner(con);
//Creating a reader object
Reader reader = new BufferedReader(new
FileReader("/home/pratt3000/Documents/College/PICT_TE-Labs/SDL/InventoryManagement/Database/
create_table.sql"));
//Running the script
sr.runScript(reader);
}catch(Exception e){System.out.println(e);}
}

String contact(){
// System.out.println("-----CONTACT DETAILS-----");
// System.out.println("\nHelp Desk : 9899998230");
// System.out.println("Complaints : 9899998231");
// System.out.println("Email : bgbskt@gmail.bgbskt.com");
// System.out.println("Donate : <DONATION LINK>");

return("-----CONTACT DETAILS-----\nHelp Desk : 9899998230\nComplaints : 9899998231\nComplaints :
9899998231\nEmail : bgbskt@gmail.bgbskt.com\nDonate : <DONATION LINK>");
}

```

```

String about(){
// System.out.println("----- ABOUT -----");
// System.out.println("\nBigbasket company was founded by V S Sudhakar, Hari Menon, Vipul Parekh, Abhinay
Choudhari and V S Ramesh in 2011. Initially, in 1999, they started India's first e-commerce site FabMart and then
went on to establish Fabmall-Trinethra chain of more than 200 grocery supermarket stores in southern India, the
business was later sold to Aditya Birla Group. It is popularly known as 'More' retail chain.[6] In 2011 they
launched bigbasket online grocery delivery service. On August 9, 2019, bigbasket partnered with a non-profit
organisation Goonj and through their Rahat flood programme provided relief materials to people who were
affected during the Kerala Flood");
return("----- ABOUT -----\nBigbasket company was founded by V S Sudhakar, Hari Menon, Vipul Parekh,
Abhinay Choudhari and V S Ramesh in 2011. Initially, in 1999, they started India's first e-commerce site FabMart
and then went on to establish Fabmall-Trinethra chain of more than 200 grocery supermarket stores in southern
India, the business was later sold to Aditya Birla Group. It is popularly known as 'More' retail chain.[6] In 2011
they launched bigbasket online grocery delivery service. On August 9, 2019, bigbasket partnered with a non-
profit organisation Goonj and through their Rahat flood programme provided relief materials to people who were
affected during the Kerala Flood");
}

```

```

String FAQ(){
// System.out.println("----- FAQs -----");
// System.out.println("1. when was bigbasket founded?");
// System.out.println("2007");
// System.out.println("1. who founded big basket?");
// System.out.println("someone did");
}

```

```

return ("----- FAQs -----\n1. when was bigbasket founded?\n2007\n2. who founded big basket?\nsomeone
did");
}

}

```

CLIENT :

```

import java.io.*;
import java.net.*;
import java.util.Scanner;
// Client class
public class Client
{
    static market_database_handling mdh = new market_database_handling();
    static account_handling acch = new account_handling();
    public static void main(String[] args) throws IOException
    {
        try
        {
            Scanner scn = new Scanner(System.in);
            // getting localhost ip
            InetAddress ip = InetAddress.getByName("localhost");
            // establish the connection with server port 5056
            Socket s = new Socket(ip, 5056);
            // obtaining input and out streams
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());

            int repeat = 1;
            while(repeat == 1){

                String user_name, password, password_chk;
                mdh.get_market_status();
                int new_acc = mdh.choose_login_options();
                String status = "";
                if(new_acc == 1){
                    while(status.equals("err:password") || status.equals("err:username") || status.equals("")){
                        System.out.println(status);
                        dos.writeUTF("old_user_login");
                        user_name = acch.take_username();
                        dos.writeUTF(user_name);
                        password = acch.take_password();
                        dos.writeUTF(password);
                        status = dis.readUTF();
                    }
                    mdh.display_respective_menu(status);
                }
                else if(new_acc == 2){
                    while(status.equals("err:usernameExists") || status.equals("")){

```



```

dos.writeUTF("new_user_login");
user_name = acch.take_username();
dos.writeUTF(user_name);
do{
password = acch.take_password();
System.out.print("REENTER ");
password_chk = acch.take_password();
}while(!password.equals(password_chk));
System.out.println("Successfully created account");
dos.writeUTF(password);
status = dis.readUTF();
}
mdh.display_respective_menu("customer");
}
repeat = mdh.continue_shopping();
}
dos.writeUTF("terminate");
// closing resources
scn.close();
dis.close();
dos.close();
s.close();
}catch(Exception e){
e.printStackTrace();
}
}
}
}

```

SERVER:

```

import java.io.*;
import java.text.*;
import java.net.*;
// Server class
public class Server
{
public static void main(String[] args) throws IOException
{
// server is listening on port 5056
ServerSocket ss = new ServerSocket(5056);
// running infinite loop for getting
// client request
while (true)
{
Socket s = null;
try
{
// socket object to receive incoming client requests
s = ss.accept();
System.out.println("A new client is connected : " + s);

```

```

// obtaining input and out streams
DataInputStream dis = new DataInputStream(s.getInputStream());
DataOutputStream dos = new DataOutputStream(s.getOutputStream());
System.out.println("Assigning new thread for this client");
// create a new thread object
Thread t = new ClientHandler(s, dis, dos);
// Invoking the start() method
t.start();
}
catch (Exception e){
ss.close();
e.printStackTrace();
}
}
}
}
}
// ClientHandler class
class ClientHandler extends Thread
{
DateFormat fordate = new SimpleDateFormat("yyyy/MM/dd");
DateFormat fortime = new SimpleDateFormat("hh:mm:ss");
final DataInputStream dis;
final DataOutputStream dos;
final Socket s;
market_database_handling mdh = new market_database_handling();
account_handling acch = new account_handling();
// Constructor
public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos)
{
this.s = s;
this.dis = dis;
this.dos = dos;
}
@Override
public void run()
{
String received;
while (true)
{
try {
String user_name, password, status;
received = dis.readUTF();

switch (received) {
case "old_user_login" :
user_name = dis.readUTF();
password = dis.readUTF();
status = acch.old_user_login(user_name, password);
dos.writeUTF(status);
break;
case "new_user_login" :
user_name = dis.readUTF();
password = dis.readUTF();
status = acch.new_user_login(user_name, password);
dos.writeUTF(status);
break;
default:

```



```
dos.writeUTF("Breaking");
break;
}
if(received.equals("terminate")){
break;
}
mdh.pass_day();
} catch (IOException e) {
e.printStackTrace();
}
}
try
{
// closing resources
this.dis.close();
this.dos.close();
}catch(IOException e){
e.printStackTrace();
}
}
}
```