**package** org.model.stock;

**public** **class** Userdetails {

**public** Userdetails() {

**super**();

}

**public** String username;

**public** String password;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** Userdetails(String username, String password) {

**super**();

**this**.username = username;

**this**.password = password;

}

}

**package** org.model.stock;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="Item\_table")

**public** **class** Itemdetails {

@Id

@Column(name="Item\_id")

@GeneratedValue

**int** itemId;

@Column(name="Item\_name")

String itemName;

@Column(name="Item\_price")

**int** itemPrice;

@Column(name="MFD")

String manufacturingDate;

@Column(name="QTY")

**int** itemQty;

@Column(name="Total\_price")

**double** totalprice;

**public** **double** getTotalprice() {

**return** totalprice;

}

**public** **void** setTotalprice(**double** totalprice) {

**this**.totalprice = totalprice;

}

**public** **int** getItemId() {

**return** itemId;

}

**public** **void** setItemId(**int** itemId) {

**this**.itemId = itemId;

}

**public** String getItemName() {

**return** itemName;

}

**public** **void** setItemName(String itemName) {

**this**.itemName = itemName;

}

**public** **int** getItemPrice() {

**return** itemPrice;

}

**public** **void** setItemPrice(**int** itemPrice) {

**this**.itemPrice = itemPrice;

}

**public** String getManufacturingDate() {

**return** manufacturingDate;

}

**public** **void** setManufacturingDate(String manufacturingDate) {

**this**.manufacturingDate = manufacturingDate;

}

**public** **int** getItemQty() {

**return** itemQty;

}

**public** **void** setItemQty(**int** itemQty) {

**this**.itemQty = itemQty;

}

**public** Itemdetails() {

**super**();

}

}

**package** org.model.stock;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.sql.SQLException;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**public** **class** Solution {

**private** **static** **final** **int** ***price*** = 0;

**public** **static** **void** main(String args[]) **throws** IOException{

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("1.Create new user\n2.Sign in");

**int** status=Integer.*parseInt*(br.readLine());

**if**(status==1){

Newuser newuser=**new** Newuser();

System.***out***.println("Enter New Username: ");

String Nname=br.readLine();

System.***out***.println("Enter the New User password: ");

String Npassword=br.readLine();

newuser.setNew\_username(Nname);

newuser.setNew\_password(Npassword);

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

session.save(newuser);

session.getTransaction().commit();

session.close();

*operation*();

}

**else** **if**(status==2){

System.***out***.println("Enter the username: ");

String username=br.readLine();

System.***out***.println("Enter the password: ");

String password=br.readLine();

Userdetails user=**new** Userdetails(username,password);

**try** {

**if**(Authentictaion.*check*(user)){

System.***out***.println("Login success");

}

}

**catch** (ClassNotFoundException | SQLException e) {

e.printStackTrace();

}

*operation*();

}

}

**private** **static** **void** operation() **throws** NumberFormatException, IOException {

System.***out***.println("1.Insert Items\n2.Update Item\n3.Delete Item\n4.View All");

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** check=Integer.*parseInt*(br.readLine());

**if**(check==1){

Itemdetails items=**new** Itemdetails();

System.***out***.println("Enter the Item Name: ");

String name=br.readLine();

items.setItemName(name);

System.***out***.println("Enter the Item price: ");

**int** price=Integer.*parseInt*(br.readLine());

items.setItemPrice(price);

System.***out***.println("Enter the Manufacturing date");

String mfd=br.readLine();

items.setManufacturingDate(mfd);

System.***out***.println("Enter the Item quantity: ");

**int** qty=Integer.*parseInt*(br.readLine());

items.setItemQty(qty);

**double** total=qty\*price;

items.setTotalprice(total);

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

session.save(items);

session.getTransaction().commit();

session.close();

System.***out***.println("Item Inserted");

}

**else** **if** (check==2){

Itemdetails items=**new** Itemdetails();

System.***out***.println("1.Update Item Name\n2.Update price\n3.Update MFD\n4.Update QTY");

**int** state=Integer.*parseInt*(br.readLine());

**if**(state==1){

System.***out***.println("Enter the item id to updated: ");

**int** upd=Integer.*parseInt*(br.readLine());

items.setItemId(upd);

System.***out***.println("Enter the item name: ");

String upname=br.readLine();

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

Itemdetails itemupdate=session.get(Itemdetails.**class**,upd);

itemupdate.setItemName(upname);

session.update(itemupdate);

session.getTransaction().commit();

session.close();

}

**else** **if**(state==2){

System.***out***.println("Enter the item id to updated: ");

**int** upd=Integer.*parseInt*(br.readLine());

items.setItemPrice(upd);

System.***out***.println("Enter the item name: ");

**int** upprice=Integer.*parseInt*(br.readLine());

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

Itemdetails itemupdate=session.get(Itemdetails.**class**,upd);

itemupdate.setItemPrice(upprice);

session.update(itemupdate);

session.getTransaction().commit();

session.close();

}

**else** **if**(state==3) {

System.***out***.println("Enter the item id to updated: ");

**int** upd=Integer.*parseInt*(br.readLine());

items.setItemPrice(upd);

System.***out***.println("Enter the item name: ");

String mfd=br.readLine();

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

Itemdetails itemupdate=session.get(Itemdetails.**class**,upd);

itemupdate.setItemName(mfd);

session.update(itemupdate);

session.getTransaction().commit();

session.close();

}

**else** **if**(state==4){

System.***out***.println("Enter the item id to updated: ");

**int** upd=Integer.*parseInt*(br.readLine());

items.setItemPrice(upd);

System.***out***.println("Enter the item name: ");

**int** qty=Integer.*parseInt*(br.readLine());

**int** total\_price=qty\****price***;

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

Itemdetails itemupdate=session.get(Itemdetails.**class**,upd);

itemupdate.setItemQty(qty);

session.update(itemupdate);

session.getTransaction().commit();

session.close();

}

}

**else** **if**(check==3) {

Itemdetails items=**new** Itemdetails();

System.***out***.println("Enter the Id to delete");

**int** del=Integer.*parseInt*(br.readLine());

SessionFactory sf = **new** Configuration().configure().buildSessionFactory();

Session session = sf.openSession();

session.beginTransaction();

items.setItemId(del);

session.delete(items);

session.save(items);

session.getTransaction().commit();

session.close();

}

**else** {

}

}

}

**package** org.model.stock;

**import** java.sql.SQLException;

**import** org.model.stock.DbFactory;

**import** com.mysql.jdbc.Connection;

**import** com.mysql.jdbc.ResultSet;

**import** com.mysql.jdbc.Statement;

**public** **class** Authentictaion {

**public** **static** **boolean** check(Userdetails user) **throws** ClassNotFoundException,SQLException{

**boolean** status = **false**;

Connection con = DbFactory.*getConn*();

Statement stmt=(Statement)con.createStatement();

String qury;

qury="";

ResultSet rs=(ResultSet) stmt.executeQuery("Select \* from Users\_table" + "where User\_name ='"+user.getUsername()+"' and User\_password ='"+user.getPassword()+"'");

**if**(rs.next()){

status = **true**;

}

stmt.close();

con.close();

**return** status;

}

}

**package** org.model.stock;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="Users\_table")

**public** **class** Newuser {

@Id

@Column(name="User\_name")

String new\_username;

@Column(name="User\_password")

String new\_password;

**public** String getNew\_username() {

**return** new\_username;

}

**public** **void** setNew\_username(String new\_username) {

**this**.new\_username = new\_username;

}

**public** String getNew\_password() {

**return** new\_password;

}

**public** **void** setNew\_password(String new\_password) {

**this**.new\_password = new\_password;

}

**public** Newuser() {

**super**();

}

}

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!-- ~ Hibernate, Relational Persistence for Idiomatic Java ~ ~ License: GNU Lesser General Public License (LGPL), version 2.1 or later. ~ See the lgpl.txt file in the root directory or <http://www.gnu.org/licenses/lgpl-2.1.html>. -->

<!DOCTYPE hibernate-configuration SYSTEM "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd" PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN">

-<hibernate-configuration>

-<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/test1</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*/>

<!-- JDBC connection pool (use the built-in) -->

<property name=*"connection.pool\_size"*>10</property>

<!-- SQL dialect -->

<property name=*"dialect"*>org.hibernate.dialect.MySQLInnoDBDialect</property>

<!-- Disable the second-level cache -->

<property name=*"cache.provider\_class"*>org.hibernate.cache.internal.NoCacheProvider</property>

<!-- Echo all executed SQL to stdout -->

<property name=*"show\_sql"*>true</property>

<!-- Drop and re-create the database schema on startup -->

<property name=*"hbm2ddl.auto"*>update</property>

<!-- Names the annotated entity class -->

<mapping class=*"org.main.Newuser"*/>

<mapping class=*"org.main.Items"*/>

</session-factory>

</hibernate-configuration>