Merchant.java

package org.jsp.merchantproductApp.dto;

import java.util.List;

import javax.persistence.\*;

@Entity

public class Merchant {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private long phone;

private String email;

private String password;

private String gst\_number;

@OneToMany(mappedBy = "merchant")

private List<Product> products;

public List<Product> getProducts() {

return products;

}

public void setProducts(List<Product> products) {

this.products = products;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public long getPhone() {

return phone;

}

public void setPhone(long phone) {

this.phone = phone;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getGst\_number() {

return gst\_number;

}

public void setGst\_number(String gst\_number) {

this.gst\_number = gst\_number;

}

@Override

public String toString() {

return "Merchant [id=" + id + ", name=" + name + ", phone=" + phone + ", email=" + email + ", password="

+ password + ", gst\_number=" + gst\_number + "]";

}

}

Product.java

**package** org.jsp.merchantproductApp.dto;

**import** javax.persistence.\*;

@Entity

**public** **class** Product

{

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**private** **int** id;

**private** String name;

**private** String brand;

**private** String category;

**private** **double** cost;

@ManyToOne

@JoinColumn

**private** Merchant merchant;

**public** Merchant getMerchant() {

**return** merchant;

}

**public** **void** setMerchant(Merchant merchant) {

**this**.merchant = merchant;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getBrand() {

**return** brand;

}

**public** **void** setBrand(String brand) {

**this**.brand = brand;

}

**public** String getCategory() {

**return** category;

}

**public** **void** setCategory(String category) {

**this**.category = category;

}

**public** **double** getCost() {

**return** cost;

}

**public** **void** setCost(**double** cost) {

**this**.cost = cost;

}

@Override

**public** String toString() {

**return** "Product [id=" + id + ", name=" + name + ", brand=" + brand + ", category=" + category + ", cost=" + cost

+ "]";

}

}

MerchantDao.java

**package** org.jsp.merchantproductApp.dao;

**import** javax.persistence.\*;

**import** org.jsp.merchantproductApp.dto.Merchant;

**public** **class** MerchantDao {

EntityManagerFactory fac=Persistence.*createEntityManagerFactory*("dev");

EntityManager man=fac.createEntityManager();

**public** Merchant saveMerchant(Merchant merchant)

{

EntityTransaction tran=man.getTransaction();

man.persist(merchant);

tran.begin();

tran.commit();

**return** merchant;

}

**public** Merchant updateMerchant(Merchant merchant)

{

EntityTransaction tran=man.getTransaction();

tran.begin();

//Find the database merchant based on id

Merchant dbMerchant=man.find(Merchant.**class**, merchant.getId());

**if**(dbMerchant!=**null**)

{

dbMerchant.setName(merchant.getName());

dbMerchant.setEmail(merchant.getEmail());

//dbMerchant.setPassword(merchant.getPassword());//Since you are passsing merchant

//object with null value for password ,same null will be updated if you use this line

tran.commit();

System.***out***.println("Merchant is updated");

//After Updating return updated merchant

**return** dbMerchant;

}

//if the id is not present return null;

**return** **null**;

}

**public** Merchant findMerchantById(**int** id)

{

**return** man.find(Merchant.**class**, id);

}

**public** Merchant verifyMerchantByEmailAndPassword(String email, String password)

{

Query q=man.createQuery("select m from Merchant m where m.email=?1 and m.password=?2");

q.setParameter(1, email);

q.setParameter(2, password);

**try** {

Merchant m=(Merchant) q.getSingleResult();//return (Merchant) q.getSingleResult();

**return** m;

} **catch** (NoResultException e)

{

**return** **null**;

}

}

**public** Merchant verifyMerchantByPhoneAndPassword(**long** phone, String password)

{

Query q=man.createQuery("select m from Merchant m where m.phone=?1 and m.password=?2");

q.setParameter(1,phone);

q.setParameter(2,password);

**try** {

Merchant m=(Merchant) q.getSingleResult();

**return** m;

} **catch** (NoResultException e)

{

**return** **null**;

}

}

}

ProductDao.java

**package** org.jsp.merchantproductApp.dao;

**import** java.util.Arrays;

**import** java.util.List;

**import** javax.persistence.\*;

**import** org.jsp.merchantproductApp.dto.Merchant;

**import** org.jsp.merchantproductApp.dto.Product;

**public** **class** ProductDao

{

**static** EntityManagerFactory *fac*=Persistence.*createEntityManagerFactory*("dev");

**static** EntityManager *man*=*fac*.createEntityManager();

**public** Product addProduct(Product p, **int** mid)

{

//Fetch the merchant based on the id ,if merchant is there add product else return null

Merchant m=*man*.find(Merchant.**class**, mid);

**if**(m!=**null**)

{

//If merchant is there then perform all the below things

//Get all list of Products

//List<Product> listproducts=m.getProducts();

//listproducts.add(p);

//m.setProducts(listproducts);

//add product to the list of products

m.getProducts().add(p); //Here I am adding the Product to the merchant

//set merchant to the product

p.setMerchant(m); //Here I am adding merchant to the product since

EntityTransaction tran=*man*.getTransaction(); //Bi-Directional Mapping

tran.begin();

*man*.persist(p);

tran.commit();

//Return the product

**return** p;

}

**return** **null**;

}

**public** **static** Product updateProduct(Product p)

{

//First find the Product info based on the Product id

//product id you get it from the Product object which is passed from Controller class

Product dbProduct=*man*.find(Product.**class**, p.getId());

**if**(dbProduct!=**null**)

{

EntityTransaction tran=*man*.getTransaction();

tran.begin();

//Fetch all the product info from passed Product object from Controller class then

//set it to dbProduct

dbProduct.setName(p.getName());

dbProduct.setBrand(p.getBrand());

dbProduct.setCategory(p.getCategory());

//I am not setting cost here

tran.commit();

**return** dbProduct;

}

**return** **null**;

}

**public** List<Product> findProductsByMerchantId(**int** mid) {

Query q=*man*.createQuery("select m.products from Merchant m where m.id=?1");

q.setParameter(1, mid);

List<Product> lps=q.getResultList();

**return** lps;

}

**public** List<Product> findProductsByBrand(String brand)

{

Query q=*man*.createQuery("select p from Product p where p.brand=?1");

//Pass the same user entered product brand from Controller class

q.setParameter(1,brand);

List<Product> lps=q.getResultList();

**return** lps;

}

**public** List<Product> findProductsByCategory(String category)

{

Query q=*man*.createQuery("select p from Product p where p.category=?1");

//Pass the same user entered product category from Controller class

q.setParameter(1, category);

List<Product> lps=q.getResultList();

**return** lps;

}

}

MerchantProductController.java

**package** org.jsp.merchantproductApp.controller;

**import** java.util.List;

**import** java.util.Scanner;

**import** javax.persistence.\*;

**import** org.jsp.merchantproductApp.dao.MerchantDao;

**import** org.jsp.merchantproductApp.dao.ProductDao;

**import** org.jsp.merchantproductApp.dto.Merchant;

**import** org.jsp.merchantproductApp.dto.Product;

**public** **class** MerchantProductController

{

**static** Scanner *sc*=**new** Scanner(System.***in***);

**static** MerchantDao *mdao*=**new** MerchantDao();

**static** ProductDao *pdao*=**new** ProductDao();

**public** **static** **void** main(String[] args)

{

System.***out***.println("1.Save Merchant");

System.***out***.println("2.Update Merchant");

System.***out***.println("3.Find Merchant by id");

System.***out***.println("4.Verify Merchant by email and password");

System.***out***.println("5.Verify Merchant by Phone and Password");

System.***out***.println("6.Add Product");

System.***out***.println("7.Update Product");

System.***out***.println("8.Find Products by Merchant id");

System.***out***.println("9.Find Products by Brand");

System.***out***.println("10.Find Products by category");

System.***out***.println("Enter your choice to perform some operation");

**int** choice=*sc*.nextInt();

**switch** (choice) {

**case** 1:

*saveMerchant*();

**break**;

**case** 2:

*updateMerchant*();

**break**;

**case** 3:

*findMerchantById*();

**break**;

**case** 4:

*verifyMerchantByEmailAndPassword*();

**break**;

**case** 5:

*verifyMerchantByPhoneAndPassword*();

**break**;

**case** 6:

*addProduct*();

**break**;

**case** 7:

*updateProduct*();

**break**;

**case** 8:

*findProductsByMerchantId*();

**break**;

**case** 9:

*findProductsByBrand*();

**break**;

**case** 10:

*findProductsByCategory*();

**break**;

**default**:

System.***err***.println("Invalid Choice");

}

}

//-------------------------------------------------------------------------------//

//main() method ends here

**public** **static** **void** saveMerchant()

{

System.***out***.println("Enter name,email,password,phone(enter without L),gst\_number");

String name=*sc*.next();

String email=*sc*.next();

String password=*sc*.next();

**long** phone=*sc*.nextLong();

String gst\_number=*sc*.next();

Merchant m=**new** Merchant();

m.setName(name);

m.setEmail(email);

m.setPassword(password);

m.setPhone(phone);

m.setGst\_number(gst\_number);

//saveMerchant() is Present in MerchantDAO class the return type is merchant

//Create a Merchant object and send it to saveMerchant() which is there in MercahntDao class

m=*mdao*.saveMerchant(m);//take the saved merchant back

System.***out***.println("Merchant is registered with id "+m.getId());

}

**static** **void** updateMerchant()

{

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

//

System.***out***.println("Enter the Merchant info to update here only merchant name and email ");

Merchant m=**new** Merchant();

//Don't forget to set the value for merchant id// it is mandatory since we are sending this merchant id to the MerchantDao class----as Merchant object so

m.setId(*sc*.nextInt());//To set id to the merchant object

m.setName(*sc*.next());

m.setEmail(*sc*.next());

//Pass the merchant object with different values to update to the updateMerchant() method which is there in MerchantDao

//Get back the saved merchant object from the updateMerchant() which is there in MerchantDao class

m=*mdao*.updateMerchant(m);

**if**(m!=**null**)

{

System.***out***.println("User is updated with merchant id "+m.getId());

}

**else**

{

//Based on the user entered merchant id ,it caries that id to the Merchant dao class using merchant object from here

System.***err***.println("Can not update since the merchant id is invalid");

}

}

**private** **static** **void** findMerchantById()

{

System.***out***.println("Enter the merchant id to verify");

**int** id=*sc*.nextInt();

Merchant m=*mdao*.findMerchantById(id);

**if**(m!=**null**)

{

System.***out***.println(m);

}

**else**

{

System.***err***.println("No Merchant found since id is invalid");

}

}

**private** **static** **void** verifyMerchantByEmailAndPassword()

{

System.***out***.println("Enter Merchant email and password for verification");

String email=*sc*.next();

String password=*sc*.next();

Merchant m=*mdao*.verifyMerchantByEmailAndPassword(email,password);

**if**(m!=**null**)

{

System.***out***.println("Merchant verification is Successful");

System.***out***.println(m);

}

**else**

{

System.***err***.println("Invalid emailId or password");

}

}

**private** **static** **void** verifyMerchantByPhoneAndPassword()

{

System.***out***.println("Enter Merchant Phone and Password ");

**long** phone=*sc*.nextLong();

String password=*sc*.next();

Merchant m=*mdao*.verifyMerchantByPhoneAndPassword(phone,password);

**if**(m!=**null**)

{

System.***out***.println("Merchant Verification is Successful");

System.***out***.println(m);

}

**else**

{

System.***err***.println("Invalid Phone and Password");

}

}

**private** **static** **void** addProduct()

{

System.***out***.println("Enter Merchant id to add product");

**int** mid=*sc*.nextInt();

System.***out***.println("Enter product name,brand,category,cost");

Product p=**new** Product();

p.setName(*sc*.next());

p.setBrand(*sc*.next());

p.setCategory(*sc*.next());

p.setCost(*sc*.nextInt());

//get that added product

p=*pdao*.addProduct(p,mid);

**if**(p!=**null**)

{

//Print the id od added product

System.***out***.println("Product added with id "+p.getId());

}

**else**

{

System.***err***.println("Can not add since the merchant id is invalid");

}

}

**static** **void** updateProduct()

{

System.***out***.println("Enter the product id to update ");

**int** pid=*sc*.nextInt();

System.***out***.println("Enter product name,brand,category,cost to update the product");

Product p=**new** Product();

p.setId(pid);

p.setName(*sc*.next());

p.setBrand(*sc*.next());

p.setCategory(*sc*.next());

//I dont want to set cost//If you dont set same value will be retained in the table for cost it will not be 0

//p.setCost(sc.nextDouble());

p=ProductDao.*updateProduct*(p);

//The above line returns the updated Product from DAO class after successful updation,else it returns null

**if**(p!=**null**)

{

System.***out***.println("Product is updated with the product id "+p.getId());

}

**else**

{

System.***out***.println("Can not update product since product id is invalid");

}

}

**private** **static** **void** findProductsByMerchantId()

{

System.***out***.println("Enter the merchant id to find Products");

**int** mid=*sc*.nextInt();

List<Product>lps=*pdao*.findProductsByMerchantId(mid);

**if**(lps.size()>0)

{

**for** (Product pro : lps) {

System.***out***.println(pro);

}

}

**else**

{

System.***err***.println("Products not found since merchant id is invalid");

}

}

**private** **static** **void** findProductsByBrand()

{

System.***out***.println("Enter the Brand name of the products to find Products");

//Sony and SONY both are same no problem

String brand=*sc*.next();

//Return the list of products from Dao class

List<Product>lps=*pdao*.findProductsByBrand(brand);

**if**(lps.size()>0)

{

**for** (Product pros : lps) {

System.***out***.println(pros);

}

}

**else**

{

System.***out***.println("No Products found since you have enetered invalid brand name");

}

}

**private** **static** **void** findProductsByCategory()

{

System.***out***.println("Enter the Product category to find Products");

String category=*sc*.next();

List<Product>lps=*pdao*.findProductsByCategory(category);

**if**(lps.size()>0)

{

**for** (Product pros : lps) {

System.***out***.println(pros);

}

}

**else**

{

System.***err***.println("No Products found since you have enetered invalid category name");

}

}

}