CONSOLE PROJECT-15E103

USER CLASS:

```
package org.model;


import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;


@Entity

public class User {

@Id

@GeneratedValue

@Column(name="User_ID")

private int userId;

private String userName;

private String password;


public int getUserId() {

		return userId;

}

public void setUserId(int userId) {

		this.userId = userId;

}

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 User() {
        super();
}


}
```

ITEM CLASS:

```
package org.model;


import java.sql.Date;


import javax.persistence.Column;


public class Item {
        @Column(name="Item_Id")
        private int itemId;
        @Column(name="Item_Name")
        private String itemName;
        @Column(name="Item_Price")
        private double itemPrice;
        @Column(name="Item_Quantity")
        private int itemQuantity;
```

```
@Column(name="M_Date")
private Date manufacturingDate;


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 double getItemPrice() {
		return itemPrice;
}
public void setItemPrice(double itemPrice) {
		this.itemPrice = itemPrice;
}
public int getItenQuantity() {
		return itemQuantity;
}
public void setItenQuantity(int itemQuantity) {
		this.itemQuantity = itemQuantity;
}
public Date getManufacturingDate() {
		return manufacturingDate;
```

```
        }

        public void setManufacturingDate(Date manufacturingDate) {

                this.manufacturingDate = manufacturingDate;

        }

        public Item() {

                super();

        }



}
```

SOLUTION CLASS:

```
package org.model;


import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;


import javax.persistence.Query;


import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.mapping.List;


public class Solution {

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

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

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

                Session session = sf.openSession();
        session.beginTransaction();
        Query query=session.createQuery("from User");
        User user=(User) query.getResultList();
        System.out.println("Enter Username : ");
        String username=br.readLine();
        System.out.println("Enter Password : ");
        String password=br.readLine();
if (user.getUserName().equals(username) && user.getPassword().equals(password)){
System.out.println("Login Success");
boolean flag = false;
while (true) {
System.out.print("1.Create 2.Retrieve 3.Update 4.Delete 5.View All\n Choice:");
switch (Integer.parseInt(br.readLine())) {
case 1:
        Item item = new Item();
        System.out.print("Enter the item id:");
        item.setItemId(Integer.parseInt(br.readLine()));
        System.out.println("Enter the item name:");
        item.setItemName(br.readLine());
        System.out.println("Enter the item prize:");
        item.setItemPrice(Double.parseDouble(br.readLine()));
        session.beginTransaction();
        session.save(item);
        session.getTransaction().commit();
        break;
case 2:
        System.out.println("Enter the itemid to Retreview");
        String id=br.readLine();
        query=session.createQuery("from Item where id=:id");
```

```
                query.setParameter("id",Integer.parseInt(id));
                Item item_ret=(Item)query.getSingleResult();
                if(item_ret.getItemId()==Integer.parseInt(id))
                {
                System.out.println("item id is: "+item_ret.getItemId());
                System.out.println("item name is: "+item_ret.getItemName());
                System.out.println("item prize is: "+item_ret.getItemPrice());
                }
                else
                System.out.println("Item Not found with this id #" + id);
                break;
        case 3:
                System.out.println("Enter the itemid to update");
                String id1=br.readLine();
                query=session.createQuery("from Item where id=:id");
                query.setParameter("id",Integer.parseInt(id1));
                Item item1=(Item)query.getSingleResult();
                if(item1.getItemId()==Integer.parseInt(id1))
                {
                System.out.println("Enter the item name:");
                Item item2 = new Item();
                item1.setItemName(br.readLine());
                System.out.println("Enter the item prize:");
                item1.setItemPrice(Double.parseDouble(br.readLine()));
                query=session.createQuery("UPDATE Item item SET item.name=:name_up,prize=:prize_up
WHERE id=:id");
                query.setParameter("id",Integer.parseInt(id1));
                query.setParameter("name_up", item2.getItemName());
                query.setParameter("prize_up",item2.getItemPrice());
```

```
            session.beginTransaction();

            session.save(item1);

            session.getTransaction().commit();

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

            }

            else

            {

            System.out.println("Item Not found with this id #" + id1);

            }

            break;

case 4:

            System.out.println("Enter the itemid to Delete");

            String id4=br.readLine();

            session.beginTransaction();

            query=session.createQuery("DELETE FROM Item item WHERE item.id=:id");

            query.setParameter("id",Integer.parseInt(id4));

            query.executeUpdate();

            session.getTransaction().commit();

            break;

case 5:

            query=session.createQuery("from Item");

            /*List<Item> item_all=query.getResultList();

            for(Item each:item_all)

            {

            System.out.println("item id is: "+each.getItemId());

            System.out.println("item name is: "+each.getItemName());

            System.out.println("item prize is: "+each.getItemPrice());

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

            }
```

```
                break;*/
        default:
        flag = true;
        break;
        }
        if(flag)
        break;
        }
        } else {
        System.out.println("Invalid Login Details");
        }
        session.close();
    }
    }
```