

## SOLUTION CLASS:

```
package org.main;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;

import javax.persistence.Query;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.model.Item;
import org.model.User;

public class Solution {

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

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

        User user = new User();

        user.setUsername("admin");

        user.setPassword("admin@123");

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

        Session session = sf.openSession();

        session.beginTransaction();

        //session.save(user);

        session.getTransaction().commit();

        Query query=session.createQuery("from User");
```

```
User user1=(User) query.getSingleResult();
```

```
System.out.print("Username :");
```

```
String username=cons.readLine();
```

```
System.out.print("Password :");
```

```
String password=cons.readLine();
```

```
if (user1.getUsername().equals(username) &&  
user1.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(cons.readLine())) {
```

```
case 1:
```

```
Item item = new Item();
```

```
System.out.print("Enter the item id:");
```

```
item.setld(Integer.parseInt(cons.readLine()));
```

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

```
item.setName(cons.readLine());
```

```
System.out.println("Enter the item prize:");
```

```
item.setPrice(Double.parseDouble(cons.readLine()));
```

```
session.beginTransaction();
```

```
session.save(item);
```

```
session.getTransaction().commit();
```

```
break;
```

```
case 2:
```

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

```
String id=cons.readLine();
```

```
query=session.createQuery("from Item where id=:id");
```

```

        query.setParameter("id",Integer.parseInt(id));
        Item item_ret=(Item)query.getSingleResult();
        if(item_ret.getId()==Integer.parseInt(id))
        {
            System.out.println("item id is: "+item_ret.getId());
            System.out.println("item name is: "+item_ret.getName());
            System.out.println("item prize is: "+item_ret.getPrice());
        }
        else
            System.out.println("Item Not found with this id #" +
id);

        break;
case 3:
    System.out.println("Enter the itemid to update");
    String id1=cons.readLine();
    query=session.createQuery("from Item where id=:id");
    query.setParameter("id",Integer.parseInt(id1));
    Item item1=(Item)query.getSingleResult();

    if(item1.getId()==Integer.parseInt(id1))
    {
        System.out.println("Enter the item name:");
        Item item2 = new Item();
        item1.setName(cons.readLine());
        System.out.println("Enter the item prize:");
        item1.setPrice(Double.parseDouble(cons.readLine()));

        query=session.createQuery("UPDATE Item item SET
item.name=:name_up,item.prize=:prize_up WHERE item.id=:id");

        query.setParameter("id",Integer.parseInt(id1));
        query.setParameter("name_up", item2.getName());
        query.setParameter("prize_up",item2.getPrice());

```

```

        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=cons.readLine();

    session.beginTransaction();

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

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

    query.executeUpdate();

    //Item item_del=(Item)query.getSingleResult();

    //session.save(item_del);

    session.getTransaction().commit();

    break;

case 5:

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

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

    for(Item each:item_all)
    {

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

```

```

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

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

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

```

---

## ITEM CLASS

```

package org.model;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQuery;

@Entity

```

```
@NamedQuery(name = "User.getByuserid", query = "select username,password from User where id=:id")
```

```
public class Item {
```

```
    @Id
```

```
    private Integer id;
```

```
    private String name;
```

```
    private double price;
```

```
    public Integer getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(Integer id) {
```

```
        this.id = id;
```

```
    }
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

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

```
        this.name = name;
```

```
    }
```

```
    public double getPrice() {
```

```
        return price;
```

```
    }
```

```
    public void setPrice(double price) {
```

```
        this.price = price;
```

```
    }
```

```
public Item(Integer id, String name, double price) {  
    super();  
    this.id = id;  
    this.name = name;  
    this.price = price;  
}
```

```
public Item() {  
    super();  
}
```

```
}
```

---

## USER CLASS

```
package org.model;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
@Entity
```

```
public class User {
```

```
    @Id
```

```
    private String username;
```

```
    private 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 User(String username, String password) {
        super();
        this.username = username;
        this.password = password;
    }

    public User() {
        super();
    }
}
```