

```

Stock.class
package HibernateTask;

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 Stock {
    @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 Stock(Integer id, String name, double price) {
        super();
        this.id = id;
        this.name = name;
        this.price = price;
    }

    public Stock() {
        super();
    }
}

```

User.Class

package HibernateTask;

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();

 }

}

Solution.Class

package HibernateTask;

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 HibernateTask.Stock;

import HibernateTask.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("akash98");
        user.setPassword("123456789");
        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\n
Choice:");
        switch (Integer.parseInt(cons.readLine())) {
            case 1:
                Stock Stock = new Stock();
                System.out.print("Enter the Stock id:");
                Stock.setId(Integer.parseInt(cons.readLine()));
                System.out.println("Stock name:");
                Stock.setName(cons.readLine());
                System.out.println("Stock prize:");
                Stock.setPrice(Double.parseDouble(cons.readLine()));
                session.beginTransaction();
                session.save(Stock);
                session.getTransaction().commit();
                break;
            case 2:
                System.out.println("Stock id to Retreview");
                String id = cons.readLine();
                query = session.createQuery("from Stock where
id=:id");
                query.setParameter("id", Integer.parseInt(id));
                Stock Stock_ret = (Stock) query.getSingleResult();
                if (Stock_ret.getId() == Integer.parseInt(id)) {
                    System.out.println("Stock id is: " +
Stock_ret.getId());
                    System.out.println("Stock name = " +
Stock_ret.getName());

```

```

        System.out.println("Stock prize = " +
Stock_ret.getPrice());
    }
    else
        System.out.println("Not found id #"+id);
    break;
case 3:
    System.out.println("Stock id to update");
    String id1 = cons.readLine();
    query=session.createQuery("from Stock where id=:id");
    query.setParameter("id",Integer.parseInt(id1));
    Stock Stock1 = (Stock) query.getSingleResult();
    if(Stock1.getId()==Integer.parseInt(id1))
    {
        System.out.println("Stock name:");
        Stock Stock2 = new Stock();
        Stock1.setName(cons.readLine());
        System.out.println("Stock prize:");

        Stock1.setPrice(Double.parseDouble(cons.readLine()));
        query = session.createQuery("UPDATE Stock Stock
SET Stock.name=:name_up,prize=:prize_up WHERE id=:id");
        query.setParameter("id",
Integer.parseInt(id1));
        query.setParameter("name_up",
Stock2.getName());
        query.setParameter("prize_up",
Stock2.getPrice());

        session.beginTransaction();
        session.save(Stock1);
        session.getTransaction().commit();
        System.out.println("Stock updated");
    }else
    {
        System.out.println("Not found id #" + id1);
    }
    break;
case 4:
    System.out.println("Stock id to Delete");
    String id4 =
cons.readLine();session.beginTransaction();
    query=session.createQuery("DELETE FROM Stock Stock
WHERE Stock.id=:id");
    query.setParameter("id",Integer.parseInt(id4));
    query.executeUpdate();
    break;

    }
    session.close();
}
}

```