

# 鲜花订购系统-----蔡文柳负责部分：

---

## 一、订单表

---

前台：用户对订单表的查询（分全部及不同状态的订单）

### Orderform.java

```
package cn.edu.lingnan.pojo;

import java.util.Date;

public class Orderform {
    private Integer id;
    private String name;
    private String photo;
    private Integer num; //订单号码
    private Integer userId;
    private String conName;
    private String conSex;
    private String conPhone;
    private String address;
    private Date ordDate;
    private String status;
    private Double totalAmount;
    private String remark;

    public Orderform() {
    }

    public Orderform(Integer id, String name, String photo, Integer num, Integer
userId,
                    String conName, String conSex, String conPhone, String
address, Date ordDate, String status, Double totalAmount, String remark) {
        this.id = id;
        this.name = name;
        this.photo = photo;
        this.num = num;
        this.userId = userId;
        this.conName = conName;
        this.conSex = conSex;
        this.conPhone = conPhone;
        this.address = address;
        this.ordDate = ordDate;
        this.status = status;
        this.totalAmount = totalAmount;
        this.remark = remark;
    }

    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 String getPhoto() {
    return photo;
}

public void setPhoto(String photo) {
    this.photo = photo;
}

public Integer getNum() {
    return num;
}

public void setNum(Integer num) {
    this.num = num;
}

public Integer getUserId() {
    return userId;
}

public void setUserId(Integer userId) {
    this.userId = userId;
}

public String getConName() {
    return conName;
}

public void setConName(String conName) {
    this.conName = conName;
}

public String getConSex() {
    return conSex;
}

public void setConSex(String conSex) {
    this.conSex = conSex;
}

public String getConPhone() {
    return conPhone;
}

public void setConPhone(String conPhone) {
```

```

        this.conPhone = conPhone;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public Date getOrdDate() {
        return ordDate;
    }

    public void setOrdDate(Date ordDate) {
        this.ordDate = ordDate;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public Double getTotalAmount() {
        return totalAmount;
    }

    public void setTotalAmount(Double totalAmount) {
        this.totalAmount = totalAmount;
    }

    public String getRemark() {
        return remark;
    }

    public void setRemark(String remark) {
        this.remark = remark;
    }

    @Override
    public String toString() {
        return "Orderform{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", photo='" + photo + '\'' +
            ", num=" + num +
            ", userId=" + userId +
            ", conName='" + conName + '\'' +
            ", conSex='" + conSex + '\'' +
            ", conPhone='" + conPhone + '\'' +
            ", address='" + address + '\'' +
            ", ordDate=" + ordDate +
            ", status='" + status + '\'' +
            ", totalAmount=" + totalAmount +

```

```

        ", remark='" + remark + '\'' +
        '}'';
    }
}

```

## OrderformMapper.java

```

package cn.edu.lingnan.mapper;

import cn.edu.lingnan.pojo.Admin;
import cn.edu.lingnan.pojo.Orderform;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
import org.springframework.stereotype.Repository;

import java.util.List;

@Mapper
@Repository
public interface OrderformMapper {

    //前端：我的订单表
    List<Orderform> getByUserId(Integer userId);
    //前端：我的订单模糊查询
    List<Orderform> select(@Param("userId")Integer userId, @Param("param")String
param);
    //前端：按订单状态查询
    List<Orderform> getByStatus(@Param("userId")Integer userId,
@Param("status")String status);
    //订单查询，用于评价审核
    List<Orderform> getByBean(Orderform bean);

    //后台：模糊查询
    List<Orderform> backSelect(String param);
    //后台：删除订单
    boolean delete(Integer id);
    //后台：修改订单
    boolean update(Orderform orderform);
    //后台：修改订单状态
    boolean updateStatus(@Param("id")Integer id, @Param("status")String status);
    //后台：全部订单
    Orderform getById(Integer id);
    List<Orderform> getList();

    boolean insert(Orderform bean);
}

```

## OrderformMapper.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

```

```

<mapper namespace="cn.edu.lingnan.mapper.OrderformMapper">

    <select id="getById" resultType="Orderform">
        select * from orderform where id = #{id};
    </select>

    <!--前台：我的全部订单-->
    <select id="getByUserId" resultType="Orderform">
        select * from orderform where user_id = #{userId}
    </select>

    <!--前端模糊查询-->
    <!--订单编号，商品名-->
    <select id="select" resultType="Orderform">
        select * from orderform where user_id=#{userId} and (num LIKE
        CONCAT(CONCAT('%',#{num}),'%') OR name LIKE CONCAT(CONCAT('%',#{name}),'%'))
    </select>

    <!--前台：按状态查询我的订单-->
    <select id="getByStatus" resultType="Orderform">
        select * from orderform where status =#{status} and user_id=#{userId}
    </select>

    <select id="getByBean" resultType="Orderform">
        select * from orderform
        <where>
            <if test="id!=null">
                and id = #{id}
            </if>
            <if test="name!=null">
                and name like '%${name}%'
            </if>
            <if test="userId!=null">
                and user_id like '%${userId}%'
            </if>
        </where>
    </select>

    <!--后台全部订单-->
    <select id="getList" resultType="Orderform">
        select * from orderform;
    </select>

    <!--后台模糊查询-->
    <!--订单编号，商品名，用户编号，状态-->
    <select id="backSelect" resultType="Orderform">
        select * from orderform
        where num like '%${key}%'
        or user_id like '%${key}%'
        or status like '%${key}%'
        or name like '%${key}%'

    </select>

    <!--test="对应jsp中name字段!=null"
        数据库字段 = #{服务器字段}
    -->

```

```

<!--后台修改订单-->
<update id="update">
    update orderform
    <set>
        <trim suffixOverrides=",">
            <if test="conName != null ">
                con_name = #{conName},
            </if>
            <if test="conSex != null ">
                con_sex = #{conSex},
            </if>
            <if test="conPhone != null">
                con_phone = #{conPhone},
            </if>
            <if test="address != null">
                address = #{address},
            </if>
            <if test="status != null">
                status = #{status},
            </if>
            <if test="remark != null">
                remark = #{remark},
            </if>
        </trim>
    </set>
    where id = #{id}
</update>

<!--后台更新订单状态-->
<update id="updateStatus">
    update orderform set status =#{status} where id=#{id}
</update>

<!--后台删除订单-->
<delete id="delete">
    delete from orderform where id = #{id}
</delete>

<insert id="insert" >
    insert into orderform
    <trim prefix="(" suffix=")" suffixOverrides=",">
        <if test="id != null">
            id,
        </if>
        <if test="num != null">
            num,
        </if>
        <if test="user_id != null">
            user_id,
        </if>
        <if test="conName != null">
            con_name,
        </if>
        <if test="conSex != null">
            con_sex,
        </if>
        <if test="conPhone != null">
            con_phone,

```

```

        </if>
        <if test="address != null">
            address,
        </if>
        <if test="ordDate != null">
            ord_date,
        </if>
        <if test="status != null">
            status,
        </if>
        <if test="totalAmount != null">
            total_amount,
        </if>
    </trim>
    value
    <trim prefix="(" suffix=")" suffixOverrides=",">
        <if test="id != null">
            #{id},
        </if>
        <if test="num != null">
            #{num},
        </if>
        <if test="userId != null">
            #{user_id},
        </if>
        <if test="conName != null">
            #{con_name},
        </if>
        <if test="conSex != null">
            #{con_sex},
        </if>
        <if test="conPhone != null">
            #{con_phone},
        </if>
        <if test="address != null">
            #{address},
        </if>
        <if test="ordDate != null">
            #{ord_date},
        </if>
        <if test="status != null">
            #{status},
        </if>
        <if test="totalAmount != null">
            #{total_amount},
        </if>
    </trim>
</insert>

</mapper>

```

## OrderformService.java

```

package cn.edu.lingnan.service;

import cn.edu.lingnan.pojo.Orderform;

```

```

import java.util.List;

public interface OrderformService {
    //前端：我的订单表
    List<Orderform> getByUserId(Integer userId);
    //我的订单模糊查询
    List<Orderform> select(Integer userId,String param);
    //按订单状态查询
    List<Orderform> getByStatus(Integer userId,String status);

    List<Orderform> getByBean(Orderform bean);

    //后台：模糊查询
    List<Orderform> backSelect(String param);
    //删除订单
    boolean delete(Integer id);
    //修改订单
    boolean update(Orderform orderform);
    //修改订单状态
    boolean updateStatus(Integer id,String status);
    //全部订单
    Orderform getById(Integer id);
    List<Orderform> getList();

    boolean insert(Orderform bean);
}

```

## OrderformServiceImpl.java

```

package cn.edu.lingnan.service.impl;

import cn.edu.lingnan.mapper.OrderformMapper;
import cn.edu.lingnan.pojo.Orderform;
import cn.edu.lingnan.service.OrderformService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
@Service("orderformService")
public class OrderformServiceImpl implements OrderformService {

    @Autowired
    private OrderformMapper orderformMapper;

    @Override
    public Orderform getById(Integer id) {
        return orderformMapper.getById(id);
    }

    @Override
    public List<Orderform> getByUserId(Integer userId) {
        return orderformMapper.getByUserId(userId);
    }

    @Override
    public List<Orderform> getByBean(Orderform bean) {

```



```

        return orderformMapper.getByBean(bean);
    }

    @Override
    public List<Orderform> select(Integer userId, String param) {
        return orderformMapper.select(userId, param);
    }

    @Override
    public List<Orderform> backSelect(String param) {
        return orderformMapper.backSelect(param);
    }

    @Override
    public List<Orderform> getByStatus(Integer userId, String status) {
        return orderformMapper.getByStatus(userId, status);
    }

    @Override
    public boolean insert(Orderform bean) {
        return orderformMapper.insert(bean);
    }

    @Override
    public boolean delete(Integer id) {
        return orderformMapper.delete(id);
    }

    @Override
    public boolean update(Orderform orderform) {
        return orderformMapper.update(orderform);
    }

    @Override
    public boolean updateStatus(Integer id, String status) {
        return orderformMapper.updateStatus(id, status);
    }

    @Override
    public List<Orderform> getList() {
        return orderformMapper.getList();
    }
}

```

## OrderformController.java(前端)

```

package cn.edu.lingnan.controller.before;

import cn.edu.lingnan.pojo.Flower;
import cn.edu.lingnan.pojo.Orderform;
import cn.edu.lingnan.service.OrderformService;
import cn.edu.lingnan.util.MyUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;

```

```

import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.util.List;

@Controller
@RequestMapping("/orderform")
public class OrderformController {

    @Autowired
    private OrderformService orderformService;

    //我的订单表模糊查询
    @RequestMapping(value = "/select", method = RequestMethod.POST)
    public String select(String select, Model model, HttpSession session) {
        System.out.println(select);
        Integer userId = MyUtil.getUserId(session);
        List<Orderform> list = orderformService.select(userId, select);
        model.addAttribute("orderList", list);
        return "before/user-orderform";
    }

    //我的订单表
    @RequestMapping(value = "/userorder")
    public String userOrderform(Model model, HttpSession session) {
        List<Orderform> list1 =
orderformService.getByUserId(MyUtil.getUserId(session));
        model.addAttribute("orderList", list1);
        return "before/user-orderform";
    }

    @RequestMapping(value = "{status}/statusorder")
    public String status2( @PathVariable String status, Model model, HttpSession
session){
        System.out.println( status);
        //userId通过用户登录获取!!!
        Integer userId = MyUtil.getUserId(session);
        System.out.println("状态查找::"+userId);
        List<Orderform> list2 = orderformService.getByStatus(userId, status);
        model.addAttribute("orderList", list2);
        return "before/user-orderform";
    }

}

```

后台：管理员对订单的管理

## OrderformAdminController.java(后台)

```

package cn.edu.lingnan.controller.admin;

```

```

import cn.edu.lingnan.pojo.Orderform;
import cn.edu.lingnan.service.OrderformService;
import cn.edu.lingnan.util.MyUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;

import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.util.List;

@Controller
@RequestMapping("/admin/orderform")
public class OrderformAdminController {

    @Autowired
    private OrderformService orderformService;

    //后台订单表(全部)
    @RequestMapping("/order")
    public String queryOrder(Model model) {
        List<Orderform> list = orderformService.getList();
        model.addAttribute("orderList",list);
        return "admin/backstage-orfo";
    }

    //后台管理模糊查询
    @RequestMapping(value = "/backselect", method = RequestMethod.POST)
    public String select(String key, Model model) {
        System.out.println(key);
        List<Orderform> list = orderformService.backSelect(key);
        model.addAttribute("orderList",list);
        return "admin/backstage-orfo";
    }

    //后台删除某条订单记录
    @RequestMapping("/{id}/delete")
    public String deleteOrder(@PathVariable Integer id, Model model) {
        System.out.println(id);
        boolean ss = orderformService.delete(id);
        //        List<Orderform> list = orderformService.getList();
        //        model.addAttribute("orderList",list);
        //        return "admin/backstage-orfo";
        return queryOrder(model);
    }

    //后台修改某条订单记录
    @RequestMapping("/{id}/getId")
    public String updateIdOrder(@PathVariable Integer id, HttpSession session,
        HttpServletResponse response) {
        //        System.out.println(id);
        response.setContentType("UTF-8");
        Orderform orderform = orderformService.getById(id);
    }

```

```

        session.setAttribute("orderform",orderform);
        return "admin/orderform-update";
    }

    //后台修改某条订单记录
    @RequestMapping("/{id}/update")
    public String updateOrder(@PathVariable Integer id,Orderform orderform,
    Model model) {
        System.out.println(id);
        boolean ss = orderformService.update(orderform);
        if(ss){
            return queryOrder(model);
        }
        return "admin/orderform-update";
    }

    //后台修改某条订单状态
    @RequestMapping("/{id}/updateStatus")
    public String updateStatus(@PathVariable Integer id, @RequestParam("status")
    String status, Model model) {
        System.out.println("修改订单状态: : "+ id + status);
        boolean ss = orderformService.updateStatus(id, status);
        if(ss){
            return queryOrder(model);
        }
        return "orderform-update";
    }
}

```

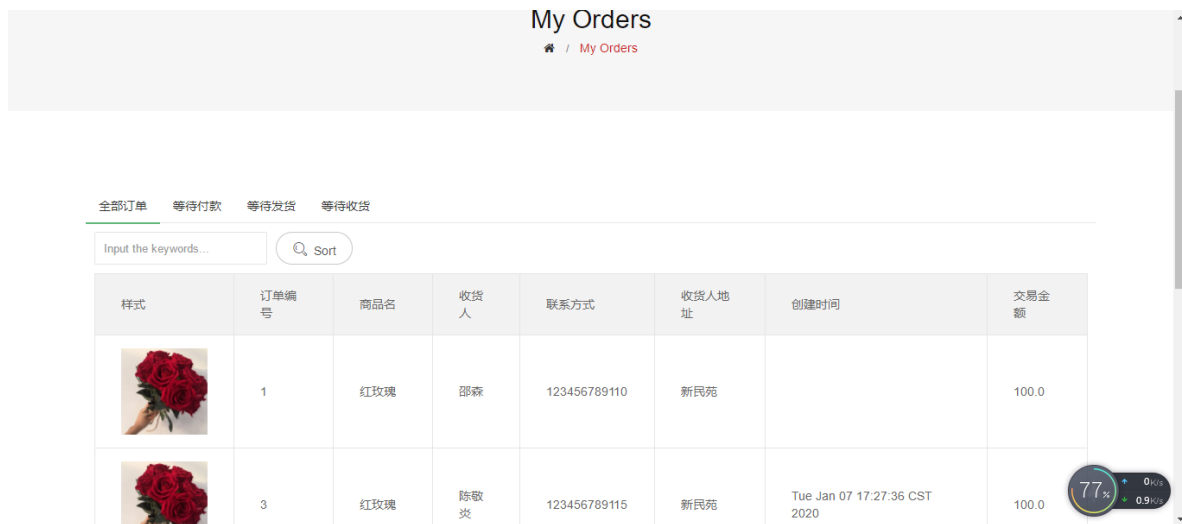
## user-orderform.jsp

### 我的订单

```

<c:forEach items="${requestScope.get('orderList')}" var="orderform">
    <tbody>
    <tr>
        <td></td>
        <td>${orderform.getId()}</td>
        <td>${orderform.getName()}</td>
        <td>${orderform.getConName()}</td>
        <td>${orderform.getConPhone()}</td>
        <td>${orderform.getAddress()}</td>
        <td>${orderform.getOrdDate()}</td>
        <td>${orderform.getTotalAmount()}</td>
    </tr>
    </tbody>
</c:forEach>

```



按状态查询我的订单

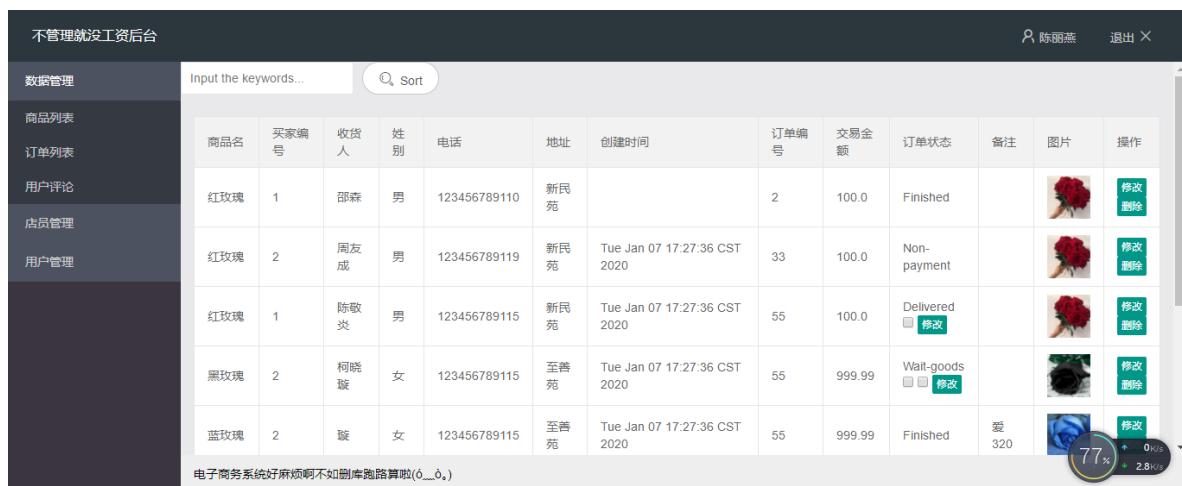
```
<div class="layui-tab layui-tab-brief lay-filter="docDemoTabBrief">
  <ul class="layui-tab-title" id="tabHeader">
    <li class="layui-this"><a href="{ctx}/orderform/userorder">全部订单</a></li>
    <li><a href="{ctx}/orderform/Non-payment/statusorder">等待付款</a></li>
    <li><a href="{ctx}/orderform/Not-delivery/statusorder">等待发货</a></li>
    <li><a href="{ctx}/orderform/Delivered/statusorder">等待收货</a></li>
  </ul>
```

####

## admin-orderform.jsp

```
<!--数据管理-->
<c:forEach items="{requestScope.get('orderList')}" var="orderform">
  <tr>
    <td>${orderform.getName()}</td>
    <td>${orderform.getUserId()}</td>
    <td>${orderform.getConName()}</td>
    <td>${orderform.getConSex()}</td>
    <td>${orderform.getConPhone()}</td>
    <td>${orderform.getAddress()}</td>
    <td>${orderform.getOrdDate()}</td>
    <td>${orderform.getNum()}</td>
    <td>${orderform.getTotalAmount()}</td>
  </tr>
```

### 页面展示



通过按钮修改订单状态（使用[c:when](#)与[c:if](#)结合）

```
<form action="${ctx}/admin/orderform/${orderform.getId()}/updateStatus" method="post">
  <c:choose>
    <c:when test="${orderform.getStatus()=='Finished'}">
      ${orderform.getStatus()}
    </c:when>
    <c:when test="${orderform.getStatus()=='Non-payment'}">
      ${orderform.getStatus()}
    </c:when>
    <c:when test="${orderform.getStatus() == 'Delivered'}">
      ${orderform.getStatus()}
      <br>
      <input type="checkbox" name="status" value="Finished" title="已完成">
      <input class="layui-btn layui-btn-xs" type="submit" value="修改">
    </c:when>
    <c:otherwise>
      ${orderform.getStatus()} <!--Not-delivery-->
      <br>
      <input type="checkbox" name="status" value="Delivered" title="已发货" ><!--name对应服务器pojo字段-->
      <input type="checkbox" name="status" value="Finished" title="已完成">
      <input class="layui-btn layui-btn-xs" type="submit" value="修改">
    </c:otherwise>
  </c:choose>
```

后台修改订单信息

```
<div class="layui-body" style="...">
  <!-- 内容主体区域 -->
  <div style="...">
    <h1 style="...">修改订单</h1>
    <div style="...">

      <form class="layui-form layui-form-pane" action="${ctx}/orderform/${sessionScope.orderform.id}/update">
        <div class="layui-form-item" style="...">
          <label class="layui-form-label">收件人姓名</label>
          <div class="layui-input-block">
            <input type="text" name="conName" value="${sessionScope.orderform.conName}" autocomplete="off" class="layui-input" style="width:100%;"/>
          </div>
        </div>
        <div class="layui-form-item" style="...">
          <label class="layui-form-label">收件人性别</label>
          <div class="layui-input-block">
            <input type="text" name="conSex" required lay-verify="required" value="${sessionScope.orderform.conSex}" autocomplete="off" class="layui-input" style="width:100%;"/>
          </div>
        </div>
        <div class="layui-form-item" style="...">
          <label class="layui-form-label">收件人电话</label>
          <div class="layui-input-block">
            <input type="text" name="conPhone" required lay-verify="required" value="${sessionScope.orderform.conPhone}" autocomplete="off" class="layui-input" style="width:100%;"/>
          </div>
        </div>
      </form>
    </div>
  </div>
```

修改订单

|       |              |
|-------|--------------|
| 收件人姓名 | 邵森           |
| 收件人性别 | 男            |
| 收件人电话 | 123456789110 |
| 地址    | 新民苑          |
| 备注    |              |

提交

二、用户（注册和完善/修改信息）

用户注册

## userMapper.java

```
@Mapper
@Repository
public interface UserMapper {

    public List<User> login(User user);

    //查询所有
    List<User> queryList();

    User getById(Integer id);

    List<User> getByKey(String key);

    List<User> getList();

    User deleteById(Integer id);

    User getByAccount(String account);

    boolean insert(User bean);

}
```

## userMapper.xml

```
<insert id="insert">
    insert into user
    <trim prefix="(" suffix=")" suffixOverrides=",">
        <if test="account != null">
            account,
        </if>
        <if test="password != null">
            password,
        </if>
        <if test="regDate == null">
            reg_date,
        </if>
    </trim>
    values
    <trim prefix="(" suffix=")" suffixOverrides=",">
        <if test="account!=null">
            #{account},
        </if>
        <if test="password!=null">
            #{password},
        </if>
        <if test="regDate==null">
            now(),
        </if>
    </trim>
</insert>
```

## userService.java

```
public interface UserService {

    public List<User> login(User user,String code);

    //查询所有
    List<User> queryList();

    User getByAccount(String account);
    boolean insert(User bean);

}
```

## userServiceImpl.java

```
@Override
    public boolean insert(User bean) {
        return userMapper.insert(bean);
    }
```

## userContorller.java

```
@PostMapping("/reg")
    public String reg(User bean, String cfmpassword, Model model) {
        System.out.println("reg::" + bean);
        System.out.println("reg::" + cfmpassword);
        //判断用户名重复
        if (userService.getByAccount(bean.getAccount()) != null) {
            model.addAttribute("msg", "该登录名已经被使用");
            return "forward:../reg.jsp";
        }
        //判断两次密码是否一致
        if(!bean.getPassword().equals(cfmpassword)) {
            model.addAttribute("msg", "两次输入密码不一致");
            return "forward:../reg.jsp";
        }

        boolean result = userService.insert(bean);
        System.out.println(result);

        if (result) {
            System.out.println("注册成功");
            return "forward:../login.jsp";
        }
        System.out.println("注册失败");
        model.addAttribute("msg", "注册失败");
        return "forward:../reg.jsp";
    }
```



## 用户修改信息

### userCenterMapper.java

```
@Mapper
@Repository
public interface UserCenterMapper {
    public List<Map<String, Object>> myOrder(Integer id); //我的订单
    public List<Map<String, Object>> myFocus(Integer id); //我的收藏
    boolean update(User bean);
    User userCenter(String account);
}
```

### userCenterMapper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="cn.edu.lingnan.mapper.UserCenterMapper">

    <select id="userCenter">
        select * from user where account = #{account}
    </select>

    <update id="update">
        update user
        <set>
            <if test="account != null and account != ''">
                account = #{account},
            </if>
            <if test="password != null and password != ''">
                password = #{password},
            </if>
            <if test="name != null and name != ''">
                name = #{name},
            </if>
            <if test="age != null and age != ''">
                age = #{age},
            </if>
            <if test="sex != null and sex != ''">
                sex = #{sex},
            </if>
            <if test="phone != null and phone != ''">
                phone = #{phone},
            </if>
            <if test="email != null and email != ''">
                email = #{email},
            </if>
            <if test="address != null and address != ''">
                address = #{address},
            </if>
        </set>
    </update>
</mapper>
```

```

        </if>
    </set>
    where id=#{id}
</update>

</mapper>

```

## userCenterService.java

```

public interface UserCenterService {

    public List<Map<String, Object>> myOrder(Integer id); //我的订单
    public List<Map<String, Object>> myFocus(Integer id); //我的收藏
    boolean update(User bean);
    User userCenter(String account);
}

```

## userCenterServiceImpl.java

```

@Override
public boolean update(User bean) {
    return userCenterMapper.update(bean);
}

```

## userCenterController.java

```

//我的信息
@RequestMapping("/userCenter")
public String userCenter(HttpSession session) {
    // session.getAttribute("loginUser");
    return "before/userCenter";
}

@RequestMapping(value = "/update")
public String update(User user, HttpSession session)
{
    Integer id = MyUtil.getUserId(session);
    user.setId(id);
    boolean ss = userCenterService.update(user);
    System.out.println(ss);
    //再重新注入一次
    session.setAttribute("loginUser", user);
    return userCenter(session);
}

```