武汉纺织大学

Web应用开发课程设计

**个人信息维护**

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 徐雨薇**

**学 号： 1804280437**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月17日**

目 录

[1 需求分析 1](#_Toc8979)

[1.1登入程序 1](#_Toc31608)

[1.2主界面 1](#_Toc27307)

[1.3各项功能 1](#_Toc28060)

[1.3.1 修改手机号码 1](#_Toc1568)

[1.3.2 修改密保邮箱 2](#_Toc24509)

[1.3.3 修改头像 2](#_Toc5086)

[2 系统设计 2](#_Toc30018)

[2.1用例图 2](#_Toc21938)

[2.2 ER图 3](#_Toc27634)

[2.3 UML类图（Class Diagram） 3](#_Toc27371)

[2.4 UML时序图（Sequence Diagram） 4](#_Toc7974)

[2.4.1 系统登录 4](#_Toc9407)

[2.5 UML活动图（Activity Diagram） 4](#_Toc3019)

[2.5.1 登陆 4](#_Toc4877)

[2.5.2 修改邮箱 4](#_Toc7234)

[2.5.3 修改手机号 5](#_Toc26406)

[2.5.4 修改头像 5](#_Toc18596)

[3 系统实现 5](#_Toc15512)

[3.1 项目结构 5](#_Toc15007)

[3.2 配置文件 6](#_Toc2386)

[3.2.1 db.properties文件 6](#_Toc32659)

[3.2.2 spring-application.xml文件 6](#_Toc32691)

[3.2.3 spring-mvc.xml文件 7](#_Toc30938)

[3.2.4 mybatis.xml文件 7](#_Toc31294)

[3.3 VO类 8](#_Toc25399)

[3.3.1 VO类User.java 8](#_Toc1485)

[3.3.2 VO类UserInfo.java 8](#_Toc22196)

[3.4 DAO接口类 9](#_Toc1655)

[3.4.1 DAO接口类UserDAO.java 9](#_Toc27256)

[3.4.2 DAO接口类UserDAO.java 9](#_Toc5252)

[3.5Controller控制类 9](#_Toc17509)

[3.5.1 Controller控制类UserController.java 9](#_Toc23076)

[3.5.2 Controller控制类UserInfoControler.java 10](#_Toc23157)

[3.6 Service服务层UserService.java 10](#_Toc16326)

[3.6.1 Service服务层UserService.java 10](#_Toc7219)

[3.6.2 Service服务层UserInfoService.java 11](#_Toc25667)

[3.7 前端页面代码 11](#_Toc11454)

[4 系统总结 16](#_Toc1846)

# 1 需求分析

设计一个人信息维护，实现个人信息的显示与相关信息的修改功能（修改头像，手机，密保邮箱），登录历史信息的表格显示。登录历史信息与个人信息来自于数据库。具体要求如下：

## 1.1登入程序

当程序运行时，主显示个人信息维护界面，右边显示登入历史信息；左边有3个是可以修改的按键、修改头像图片和手机号码、密保邮箱。

## 1.2主界面

当index.html里面设置了直接登入，所以程序里面直接进入主界面，如下图1.1：



图1.1主界面个人信息维护界面

## 1.3各项功能

### 1.3.1 修改手机号码

点击界面上手机后面的修改，出现提示框，修改手机号之后会出现修改成功的提示框，同时界面上的手机号也更换了。如图1.2所示：

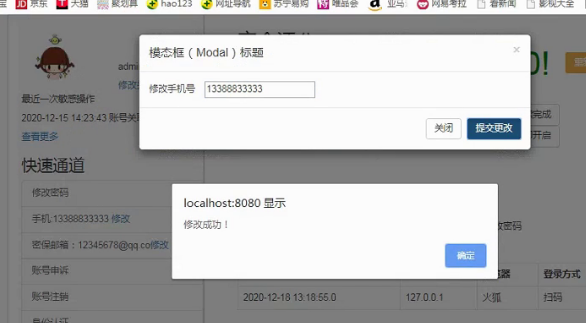


图1.2修改手机号码成功

### 1.3.2 修改密保邮箱

点击界面上密保邮箱的修改，出现提示框，修改密保邮箱之后会出现修改成功的提示框，同时界面上的密保邮箱也更换了。如图1.3所示：



图1.3修改密保邮箱成功

### 1.3.3 修改头像

点击界面上图像旁边的修改，出现提示框，更改图片之后会出现修改成功的提示框，同时界面上的头像图片也更换了，但是由于技术不达标这项功能没有完成。

**1.3.4 登录历史信息**

登录历史信息，采用了mysql。其表单userinfo.sql 如图1.4所示：

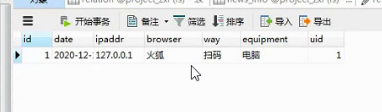


图1.4 userinfo.sql表单

# **2 系统设计**

## 2.1用例图

## 其个人信息维护用例如下图2.1：

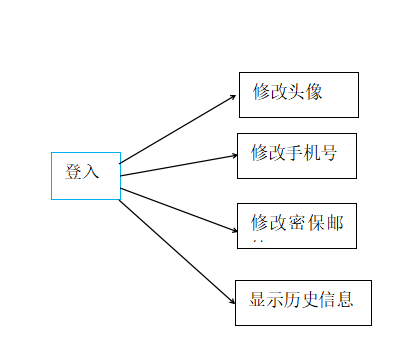


图2.1个人信息维护功能

## 2.2 ER图

## MySQL数据库，有2个表单。user.sql个人信息修改表单用于保存用户、密码、手机邮箱、与图片，userinfo.sql登录历史信息表单如下图2.1所示：

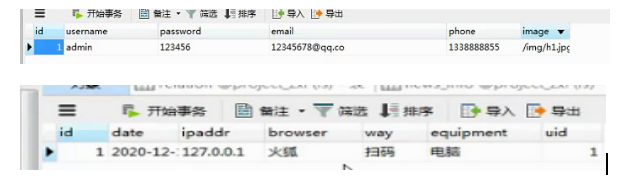


图2.1数据库表单

## 2.3 UML类图（Class Diagram）

对于用户功能模块，共设计如下4个类。

* 数据库连接类db.properties：连接数据库并获取连接对象.
* VO实体类User：与数据库结构进行映射的类。主要由属性，Integer,String方法组成，VO类中的属性与表中的字段相对应，每一个VO类的对象都表示表中的每一条记录。
* DAO接口UserDAO：主要定义操作的接口，定义一系列数据库的原子性操作，例如增删改等。
* DAO实现类UserInfoDaol：DAO接口的真实实现类，主要完成具体数据库操作，但不负责数据库的打开和关闭。
* 业务逻辑实现类UserService：对于数据层的原子操作进行整合。还要负责数据库的打开与关闭（不管是否出异常，数据库都要关闭）
* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*：

各类的结构及类之间的关系如图2-2所示：

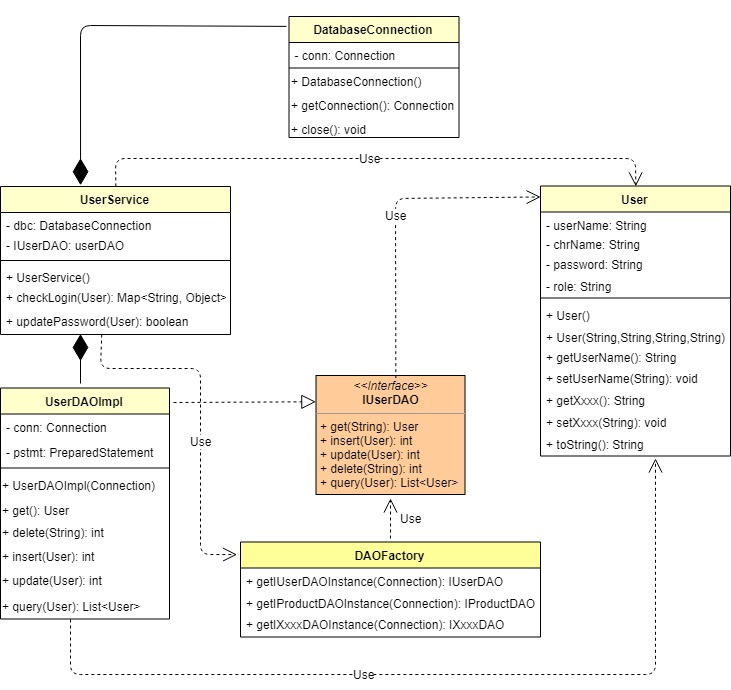


图2-2用户功能类图

## 2.4 UML时序图（Sequence Diagram）

### 2.4.1 系统登录

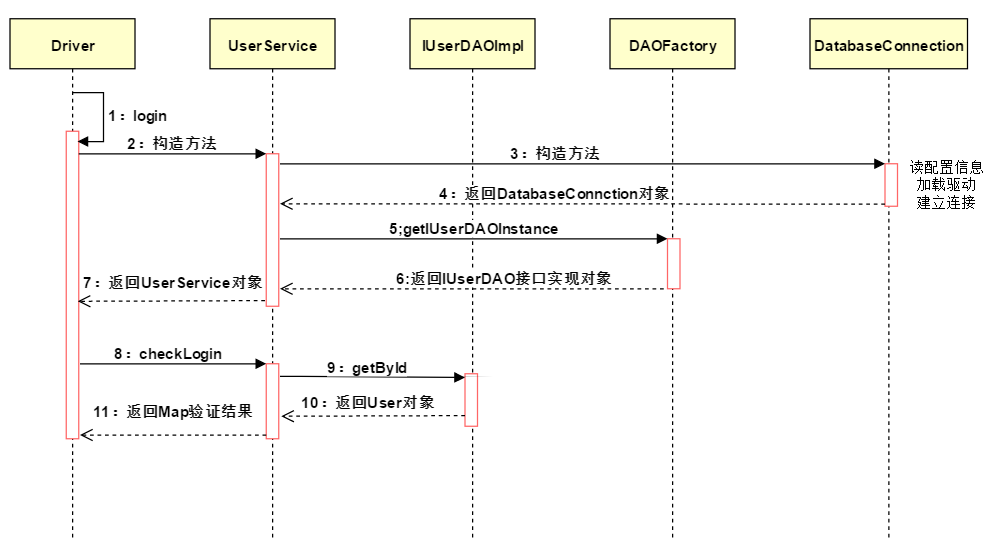


图2.3时序图

## 2.5 UML活动图（Activity Diagram）

### 2.5.1 登陆

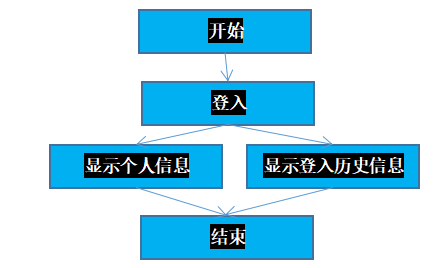


图2-4 登入界面活动图

### 2.5.2 修改邮箱

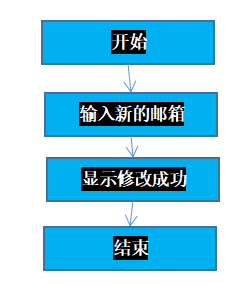


图2-5 修改邮箱界面活动图

### 2.5.3 修改手机号

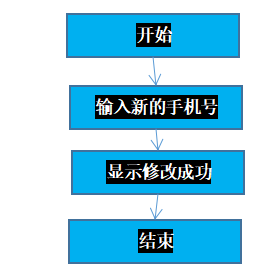


图2-6 修改手机号界面活动图

### 2.5.4 修改头像

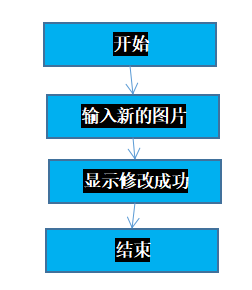


图2-7 修改头像界面活动图

# **3 系统实现**

## 3.1 项目结构

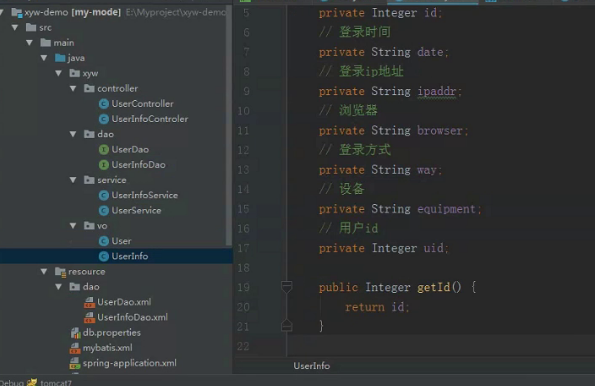


图3.1项目结构图

## 3.2 配置文件

在项目下新建Source Folder，名为resources

### 3.2.1 db.properties文件

该配置文件主要是以文件形式保存数据库的驱动类名称，连接数据库的URL地址，访问数据库的用户名及对应的密码，程序运行时会读取该文件相关信息，避免硬编码，当相关信息发生变化时，只需修改配置文件而不用修改源代码，增加程序的可扩展型。

jdbc.driver=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://127.0.0.1/xyw?characterEncoding=utf-8

jdbc.username=root

jdbc.password=123456

### 3.2.2 spring-application.xml文件

开启注解扫描

<?xml version="1.0" encoding="UTF-8" ?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<!--注解扫描-->

<context:component-scan base-package="xyw">

<context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

<context:property-placeholder location="classpath:db.properties"/>

<!--连接池-->

<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource">

<property name="driverClassName" value="${jdbc.driver}"/>

<property name="url" value="${jdbc.url}"/>

<property name="username" value="${jdbc.username}"/>

<property name="password" value="${jdbc.password}"/>

</bean>

<!--Mybatis工厂-->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource"/>

<property name="configLocation" value="classpath:mybatis.xml"/>

<property name="mapperLocations" value="classpath:dao/\*.xml"/>

</bean>

<!--配置Mapper扫描-->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="xyw.dao"/>

</bean>

</beans>

### 3.2.3 spring-mvc.xml文件

开启注解扫描

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd

http://www.springframework.org/schema/tx

http://www.springframework.org/schema/tx/spring-tx.xsd">

<!--开启注解扫描-->

<context:component-scan base-package="xyw">

<!--只扫描控制器-->

<context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

</context:component-scan>

<mvc:default-servlet-handler/>

<mvc:annotation-driven/>

</beans>

### 3.2.4 mybatis.xml文件

打印sql

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE configuration

PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<!--打印sql-->

<settings>

<setting name="logImpl" value="STDOUT\_LOGGING"/>

</settings>

<typeAliases>

<!--包扫描-->

<package name="xyw.vo"/>

</typeAliases>

</configuration>

## 3.3 VO类

### 3.3.1 VO类User.java

显示表单登入用户信息

public class User {

private Integer id;

private String username;// 用户名

private String password;// 密码

private String email;// 邮箱

private String phone;// 手机号

private String image;// 头像地址

}

### 3.3.2 VO类UserInfo.java

显示登入历史信息的表单

public class UserInfo {

private Integer id;

// 登录时间

private String date;

// 登录ip地址

private String ipaddr;

// 浏览器

private String browser;

// 登录方式

private String way;

// 设备

private String equipment;

// 用户id

private Integer uid;

}

## 3.4 DAO接口类

## **3.4.1 DAO接口类UserDAO.java**

登入后，对手机号码与邮箱进行修改

public interface UserDao {

// 登录

User loginUser(User user);

// 修改用户手机

int updatePhone(User user);

// 修改邮箱

int updateEmail(User user);

}

## **3.4.2 DAO接口类UserDAO.java**

登入后，显示登入历史信息。

public interface UserInfoDao {

List<UserInfo> queryAll();

UserInfo queryByUid(Integer uid);

}

## 3.5Controller控制类

## **3.5.1 Controller控制类UserController.java**

控制器实现登入，修改电话，修改邮箱功能

public class UserController {

@Autowired

private UserService userService;

@RequestMapping("/loginUser")// 登录功能

@ResponseBody

public User loginUser(User user){

User newUser = userService.loginUser(user);

return newUser;

} // 修改电话

@RequestMapping("updatePhone")

@ResponseBody

public int updatePhone(User user){

return userService.updatePhone(user);

} // 修改邮箱

@RequestMapping("updateEmail")

@ResponseBody

public int updateEmail(User user){

return userService.updateEmail(user);

}

}

## **3.5.2 Controller控制类UserInfoControler.java**

控制器查询使用信息与显示

public class UserInfoControler {

@Autowired

private UserInfoService userInfoService;

// 查询所有

@RequestMapping("/queryAll")

@ResponseBody

public List<UserInfo> queryAll(){

List<UserInfo> userInfos = userInfoService.queryAll();

return userInfos;

}

// 根据userId查询

@RequestMapping("/queryByUid")

@ResponseBody

public UserInfo queryByUid(Integer uid){

UserInfo userInfo = userInfoService.queryByUid(uid);

return userInfo;

}

}

## 3.6 Service服务层UserService.java

## **3.6.1 Service服务层UserService.java**

Service查询所有登录，手机，邮箱信息

**ppublic class UserService {**

**@Autowired**

**private UserDao userDao;**

**public User loginUser(User user) {**

**User newUser = userDao.loginUser(user);**

**return newUser;**

**}**

**public int updatePhone(User user) {**

**return userDao.updatePhone(user);**

**}**

**public int updateEmail(User user) {**

**return userDao.updateEmail(user);**

**}**

**}**

## **3.6.2 Service服务层UserInfoService.java**

Service查询所有历史登录记录

**public class UserInfoService {**

**@Autowired**

**private UserInfoDao userInfoDao;**

**// 查询所有登录记录**

**public List<UserInfo> queryAll() {**

**List<UserInfo> userInfos = userInfoDao.queryAll();**

**return userInfos;**

**}**

**// 根据uid查询**

**public UserInfo queryByUid(Integer uid) {**

**UserInfo userInfo = userInfoDao.queryByUid(uid);**

**return userInfo;**

**}**

**}**

## 3.7 前端页面代码

**<!DOCTYPE html>**

**<html lang="en" xmlns:v-on="http://www.w3.org/1999/xhtml">**

**<head>**

**<meta charset="UTF-8">**

**<title>Title</title>**

**<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">**

**<link rel="stylesheet" href="/css/index.css">**

**<script type="text/javascript" src="./js/jquery-3.4.1.min.js"></script>**

**<script type="text/javascript" src="./js/vue.js"></script>**

**<script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>**

**</head>**

**<body>**

**<div id="app">**

**<!--左边的Div-->**

**<div id="leftDiv">**

**<span><img :src="user.image" style="width: 100px; height: 100px"></span>**

**<div id="headerDiv">**

**<p>{{user.username}}</p>**

**<p><a href="#">修改头像</a> </p>**

**</div>**

**<p>最近一次敏感操作</p>**

**<p>2020-12-15 14:23:43 账号关联微信账号</p>**

**<a href="#">查看更多</a>**

**<h3>快速通道</h3>**

**<div class="list-group">**

**<button type="button" class="list-group-item">修改密码</button>**

**<button type="button" class="list-group-item">手机:{{user.phone}} <a href="#" v-on:click="update(0)">修改</a></button>**

**<button type="button" class="list-group-item">密保邮箱：{{user.email}}<a href="#" v-on:click="update(1)">修改</a></button>**

**<button type="button" class="list-group-item">账号申诉</button>**

**<button type="button" class="list-group-item">账号注销</button>**

**<button type="button" class="list-group-item">身份认证</button>**

**</div>**

**</div>**

**<!--右边的Div-->**

**<div id="rightDiv">**

**<h2>安全评分</h2>**

**<div id="rightTop">**

**<div class="progress jiduDiv">**

**<div class="progress-bar progress-bar-success" role="progressbar" aria-valuenow="90" aria-valuemin="0" aria-valuemax="100" style="width: 90%;">**

**</div>**

**</div>**

**<div id="jinduRight">**

**<span style="color: green; font-size: 50px">90!</span>**

**<button type="button" class="btn btn-warning dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="margin-top: -30px; margin-left: 20px">重新检测 </button>**

**</div>**

**</div>**

**<h4>账号存在隐患，立即修复</h4>**

**<span>建议完成人脸识别，提供安全保障</span><span><button type="button" class="btn btn-default" style="margin-left: 200px">继续完成</button></span><br>**

**<span>建议开启登录保护，防止盗号风险</span><span><button type="button" class="btn btn-default" style="margin-left: 200px">立即开启</button></span><br>**

**<hr style="margin-top: 50px; color: grey">**

**<h2>**

**登录历史**

**</h2>**

**<p>以下为您最近的登录记录，若存在异常情况，请在核实后尽快修改密码</p>**

**<table class="table table-bordered">**

**<caption>基本的表格布局</caption>**

**<thead>**

**<tr>**

**<th>时间</th>**

**<th>IP</th>**

**<th>浏览器</th>**

**<th>登录方式</th>**

**<th>设备</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>{{userInfo.date}}</td>**

**<td>{{userInfo.ipaddr}}</td>**

**<td>{{userInfo.browser}}</td>**

**<td>{{userInfo.way}}</td>**

**<td>{{userInfo.equipment}}</td>**

**</tr>**

**</tbody>**

**</table>**

**</div>**

**<!-- 模态框（Modal） -->**

**<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">**

**<div class="modal-dialog">**

**<div class="modal-content">**

**<div class="modal-header">**

**<button type="button" class="close" data-dismiss="modal" aria-hidden="true">**

**&times;**

**</button>**

**<h4 class="modal-title" id="myModalLabel">**

**模态框（Modal）标题**

**</h4>**

**</div>**

**<div class="modal-body" v-if="flag == 0">**

**修改手机号&nbsp;&nbsp;&nbsp;&nbsp;<input v-model="user.phone">**

**</div>**

**<div v-else>**

**修改邮箱&nbsp;&nbsp;&nbsp;&nbsp;<input v-model="user.email">**

**</div>**

**<div class="modal-footer">**

**<button type="button" class="btn btn-default" data-dismiss="modal">关闭**

**</button>**

**<button type="button" class="btn btn-primary" v-on:click="mySubmit()">**

**提交更改**

**</button>**

**</div>**

**</div><!-- /.modal-content -->**

**</div><!-- /.modal -->**

**</div>**

**</div>**

**</body>**

**<script type="text/javascript">**

**var vue = new Vue({**

**el: '#app',**

**data: {**

**user:{},**

**userInfo:{},**

**flag:''**

**},**

**methods:{**

**queryUser:function () {**

**$.ajax({**

**url:"/loginUser",**

**type:"POST",**

**data:{username : "admin", password: '123456'},**

**success:function(result){**

**vue.user = result;**

**console.log(result);**

**}**

**});**

**},**

**queryUserInfo:function () {**

**$.ajax({**

**url:"queryByUid",**

**type:"GET",**

**data:{uid : '1'},**

**success:function(result){**

**console.log(result);**

**vue.userInfo = result**

**}**

**});**

**},**

**update:function (val) {**

**// 0是手机号 1是邮箱**

**this.flag = val**

**$('#myModal').modal('show')**

**},**

**mySubmit:function () {**

**var \_this = this;**

**var flag = this.flag**

**if(flag == 0){**

**// 修改手机**

**$.ajax({**

**url:"updatePhone",**

**type:"GET",**

**data:\_this.user,**

**success:function(result){**

**console.log(result);**

**vue.userInfo = result**

**}**

**});**

**}else{**

**// 修改邮箱**

**$.ajax({**

**url:"updateEmail",**

**type:"GET",**

**data:\_this.user,**

**success:function(result){**

**console.log(result);**

**vue.userInfo = result**

**}**

**});**

**}**

**$('#myModal').modal('hide')**

**alert("修改成功！")**

**}**

**},**

**create:function () {**

**},**

**mounted:function () {**

**this.queryUser();**

**this.queryUserInfo()**

**}**

**})**

**</script>**

**</html>**

# **4 系统总结**

本次实验我后端用了：SpringMVC，Spring，Mybatis。Tomcat作为服务器，maven项目管理工具，数据库是mysql，然后项目实现了修改手机号，修改密保邮箱与登入后历史信息显示，但是没有完成修改图片点操作。我所用点js代码都是在网上百度搜的。这学期web点课程，因为没有基础很多都更不上，这次实验所写代码基本上是一边看书，在网上搜，最后还有问题时是问web学习比较好点同学，我还有很多不足，即使写出了代码还有个别个地方有瑕疵。但是我会继续对web这门功课进行深造，希望能把之前老师所讲知识都能弄会。