武汉纺织大学

Web应用开发课程设计

**网页的账号注册操作**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 刘广川**

**学 号： 1814260214**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月20日**

目 录

[1 需求分析 1](#_Toc17888)

[2 系统设计 1](#_Toc13861)

[2.1 UML类图（Class Diagram） 1](#_Toc29933)

[2.2 UML活动图（Activity Diagram） 2](#_Toc13693)

[2.2.1 登陆 2](#_Toc14846)

[3 系统实现 3](#_Toc25370)

[3.1 项目结构 3](#_Toc26716)

[3.2 VO类User.java 3](#_Toc10630)

[private String passWord; 3](#_Toc15142)

[3.3 VO类PhoneCode.java 3](#_Toc28447)

[3.4 RegisterController.java 4](#_Toc8212)

[3.5 SendCheckCodeController.java 5](#_Toc20552)

[3.6 CheckSumBuilder.java 6](#_Toc14688)

[3.7 SendCheckCode.java 7](#_Toc2091)

[3.8 SaveToSql.java 9](#_Toc22527)

[4 系统测试 10](#_Toc25750)

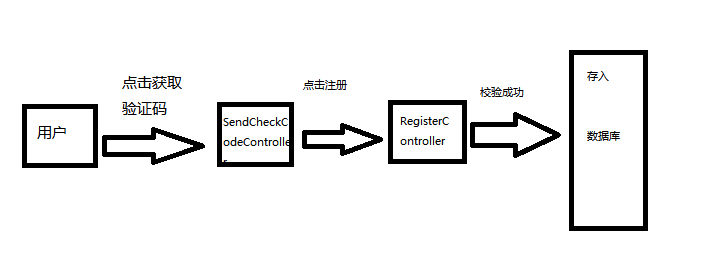
[5 系统总结 13](#_Toc30858)

# 1 需求分析

·

按上图完成用户注册，存入数据库中。

# **2 系统设计**



## 2.1 UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

* VO实体类User：与数据库结构进行映射的类。主要由属性，setter, getter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
* VO实体类PhoneCode：在session中携带信息的类。主要由属性，setter, getter方法组成。
* Controller类RegisterController：进行验证码校验并将用户信息存储进数据库。
* Controller类SendCheckCodeController：进行调用验证码发送类进行向用户键入手机号发送验证码。
* Dao类CheckSumBuilder:发送验证码接口所依赖的验证码计算方法。
* Dao类SendCheckCode:调用网易云信接口进行手机验证码发送。
* Dao类SaveToSql:用于连接数据库并将注册的用户信息存入数据库中。

各类的结构及类之间的关系如图2-5所示：

图2-5 用户功能类图

## 2.2 UML活动图（Activity Diagram）

### 2.2.1 登陆

登录操作需要验证用户名，密码，只有当用户名和密码同时存在即用户存在的情况下才能成功登录系统，否则会提示错误信息，具体流程如图2-6所示。

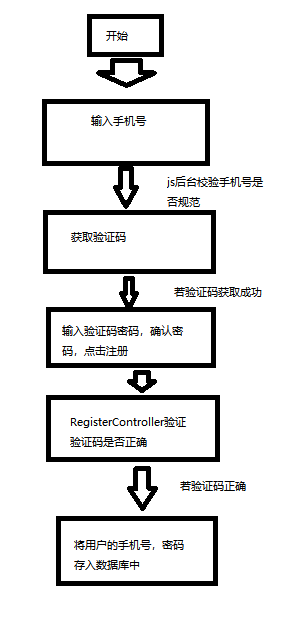
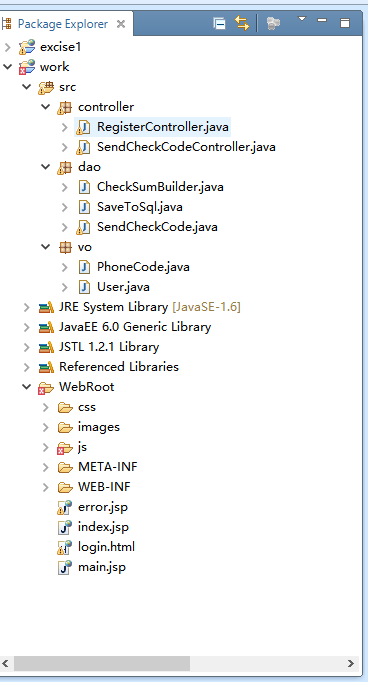


图2-2 登录流程图

# **3 系统实现**

## 3.1 项目结构



## 3.2 VO类User.java

描述该文件作用，并对里面的关键代码进行解释

**public** **class** User {

**private** String phone;

## private String passWord;

//构造方法，getter,setter,toString方法，自行补充

}

## 3.3 VO类PhoneCode.java

**public** **class** PhoneCode {

**private** Integer code;

**private** String msg;

**private** String obj;

//构造方法，getter,setter,toString方法，自行补充

}

## 3.4 RegisterController.java

package controller;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import dao.SaveToSql;

import vo.User;

@WebServlet(urlPatterns = "/register.do")

public class RegisterController extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

String phone = request.getParameter("phone");

String password = request.getParameter("password1");

String checkCode = request.getParameter("checkCode");

String forwardPath = "";

HttpSession session = request.getSession();

String saveCheckCode = (String)session.getAttribute("checkCode");

if (!checkCode.equals(saveCheckCode)) {

forwardPath = "/error.jsp";

request.setAttribute("info", "验证码不正确！");

} else {

User user = new User(phone, password);

SaveToSql saveToSql = new SaveToSql();

if (saveToSql.save(user)) {

forwardPath = "/main.jsp";

request.setAttribute("info", "该手机号已注册！");

}else {

forwardPath = "/error.jsp";

}

}

RequestDispatcher rd = request.getRequestDispatcher(forwardPath);

rd.forward(request, response);

}

}

用于获取用户所提交的注册表单，提取session中已发送的验证码进行校验用户所输入的验证码正确性，正确后开始存入数据库。

## 3.5 SendCheckCodeController.java

package controller;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;

import com.fasterxml.jackson.databind.ObjectMapper;

import dao.SendCheckCode;

import vo.PhoneCode;

import vo.User;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

public class SendCheckCodeController extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse response) {

String phone = null;

String result= null;

PhoneCode phoneCode = null;

phone = request.getParameter("phone");

try {

result = SendCheckCode.sendCode(phone, "19469751");

ObjectMapper om = new ObjectMapper();

phoneCode = om.readValue(result, PhoneCode.class);

response.setContentType("text/plain");

if(phoneCode.getCode()==200){

HttpSession session = request.getSession();

session.setAttribute("checkCode",phoneCode.getObj());

session.setMaxInactiveInterval(60\*3);

response.getWriter().write("200");

}else {

response.getWriter().write(phoneCode.getCode()+"");

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

用于获取验证码，返回验证码类并将所发送的验证码保存到session中。

## 3.6 CheckSumBuilder.java

**package** dao;

**import** java.security.MessageDigest;

**public** **class** CheckSumBuilder {

**public** **static** String getCheckSum(String appSecret, String nonce, String curTime) {

**return** *encode*("sha1", appSecret + nonce + curTime);

}

**public** **static** String getMD5(String requestBody) {

**return** *encode*("md5", requestBody);

}

**private** **static** String encode(String algorithm, String value) {

**if** (value == **null**) {

**return** **null**;

}

**try** {

MessageDigest messageDigest

= MessageDigest.*getInstance*(algorithm);

messageDigest.update(value.getBytes());

**return** *getFormattedText*(messageDigest.digest());

} **catch** (Exception e) {

**throw** **new** RuntimeException(e);

}

}

**private** **static** String getFormattedText(**byte**[] bytes) {

**int** len = bytes.length;

StringBuilder buf = **new** StringBuilder(len \* 2);

**for** (**int** j = 0; j < len; j++) {

buf.append(*HEX\_DIGITS*[(bytes[j] >> 4) & 0x0f]);

buf.append(*HEX\_DIGITS*[bytes[j] & 0x0f]);

}

**return** buf.toString();

}

**private** **static** **final** **char**[] *HEX\_DIGITS* = { '0', '1', '2', '3', '4', '5',

'6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

}

用于计算所发送的手机验证码。

## 3.7 SendCheckCode.java

package dao;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import org.apache.http.client.entity.UrlEncodedFormEntity;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.message.BasicNameValuePair;

import org.apache.http.util.EntityUtils;

public class SendCheckCode {

private static final String SERVER\_URL="https://api.netease.im/sms/sendcode.action";

private static final String APP\_KEY="67adb1fd66bb9e485ae2be20f19ae5eb";

private static final String APP\_SECRET="d1d8df7b380c";

private static final String NONCE="123456";

private static final String CODELEN="6";

public static String sendCode(String mobile,String templateId) throws Exception {

DefaultHttpClient httpClient = new DefaultHttpClient();

HttpPost httpPost = new HttpPost(SERVER\_URL);

String curTime = String.valueOf((new Date()).getTime() / 1000L);

String checkSum = CheckSumBuilder.getCheckSum(APP\_SECRET, NONCE, curTime);

httpPost.addHeader("AppKey", APP\_KEY);

httpPost.addHeader("Nonce", NONCE);

httpPost.addHeader("CurTime", curTime);

httpPost.addHeader("CheckSum", checkSum);

httpPost.addHeader("Content-Type", "application/x-www-form-urlencoded;charset=utf-8");

List<NameValuePair> nvps = new ArrayList<NameValuePair>();

nvps.add(new BasicNameValuePair("templateid", templateId));

nvps.add(new BasicNameValuePair("mobile", mobile));

nvps.add(new BasicNameValuePair("codeLen", CODELEN));

httpPost.setEntity(new UrlEncodedFormEntity(nvps, "utf-8"));

HttpResponse response = httpClient.execute(httpPost);

return EntityUtils.toString(response.getEntity(), "utf-8");

}

}

调用网易云信的验证码发送接口，进行手机验证码的发送行为。

## 3.8 SaveToSql.java

package dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.Statement;

import vo.User;

public class SaveToSql {

public boolean save(User user) {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection(

"jdbc:mysql://127.0.0.1:3306/excise?useunicode=true&character=utf-8",

"root", "root");

String sql = "select \* from t\_user1 ";

PreparedStatement pst = con.prepareStatement(sql);

java.sql.ResultSet rs = pst.executeQuery();

while (rs.next()) {

if (rs.getString("phone").equals(user.getPhone())) {

return false;

}

}

String insertSql = "insert into t\_user1 values ('"+user.getPhone()+"','"+user.getPassWord()+"')";

Statement sm = con.createStatement();

sm.executeUpdate(insertSql);

} catch (Exception e) {

e.printStackTrace();

}

return true;

}

}

连接数据库，并将user的手机号和密码存入数据库中。

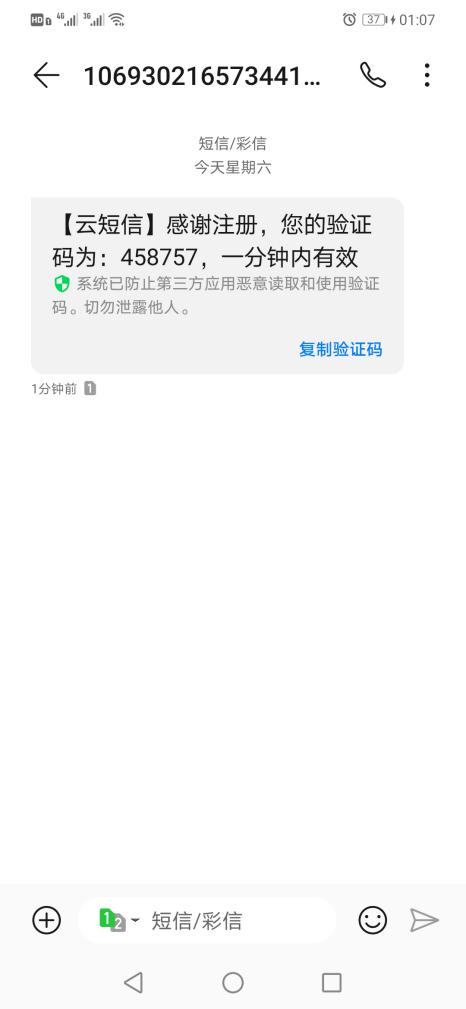
# **4 系统测试**



注册页面



点击获取验证码后页面



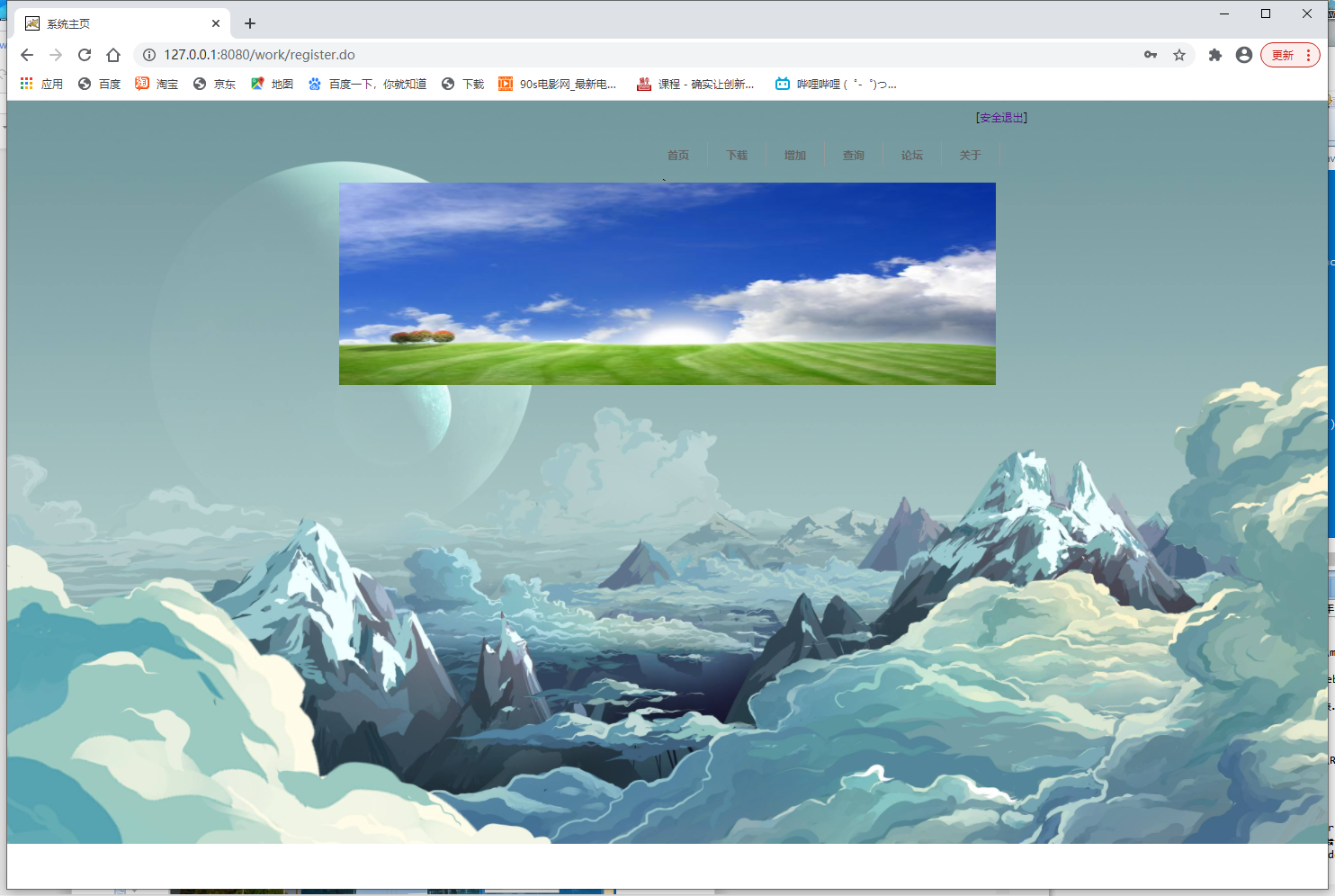
手机收到验证码



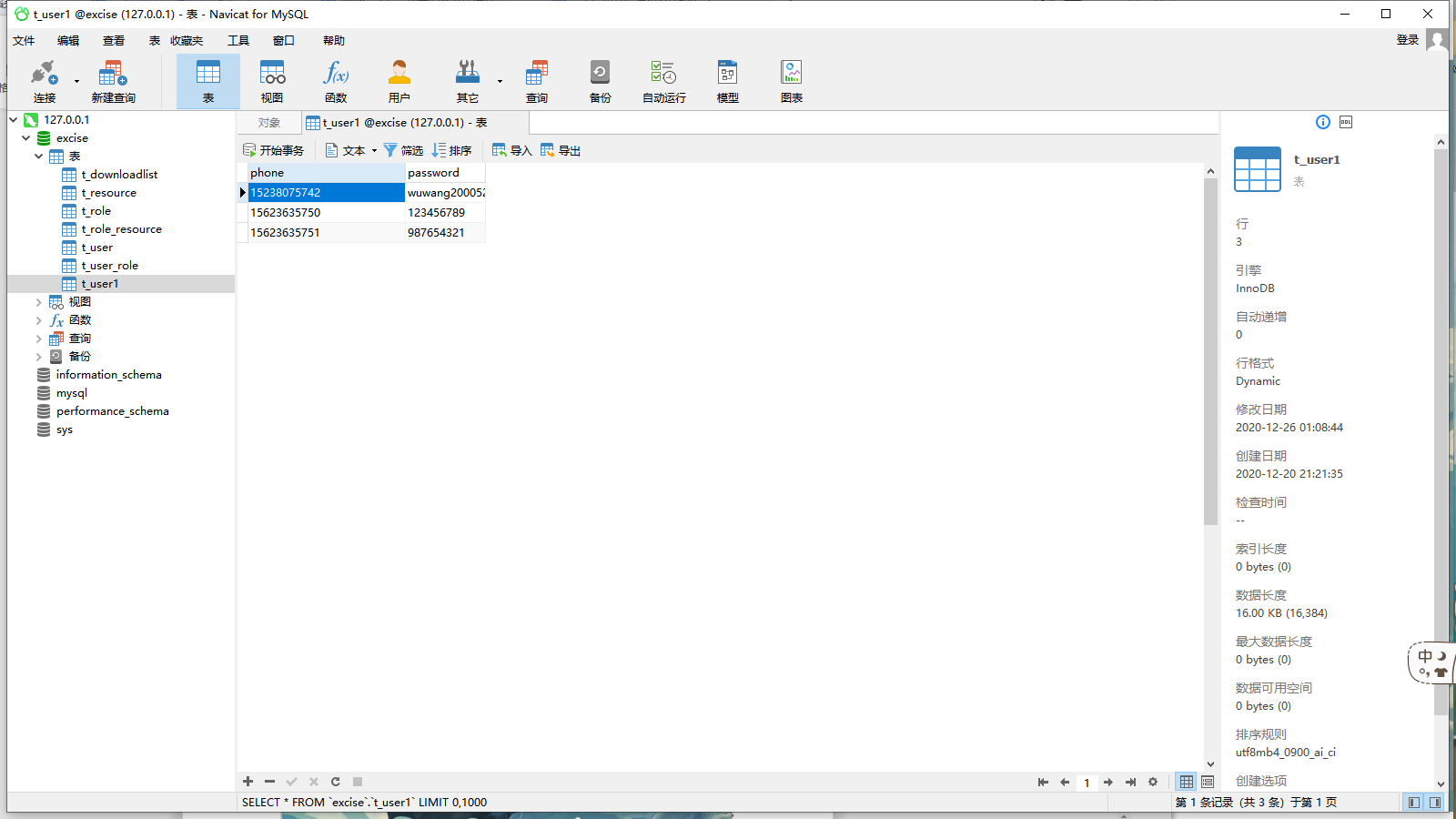
确认密码错误页面



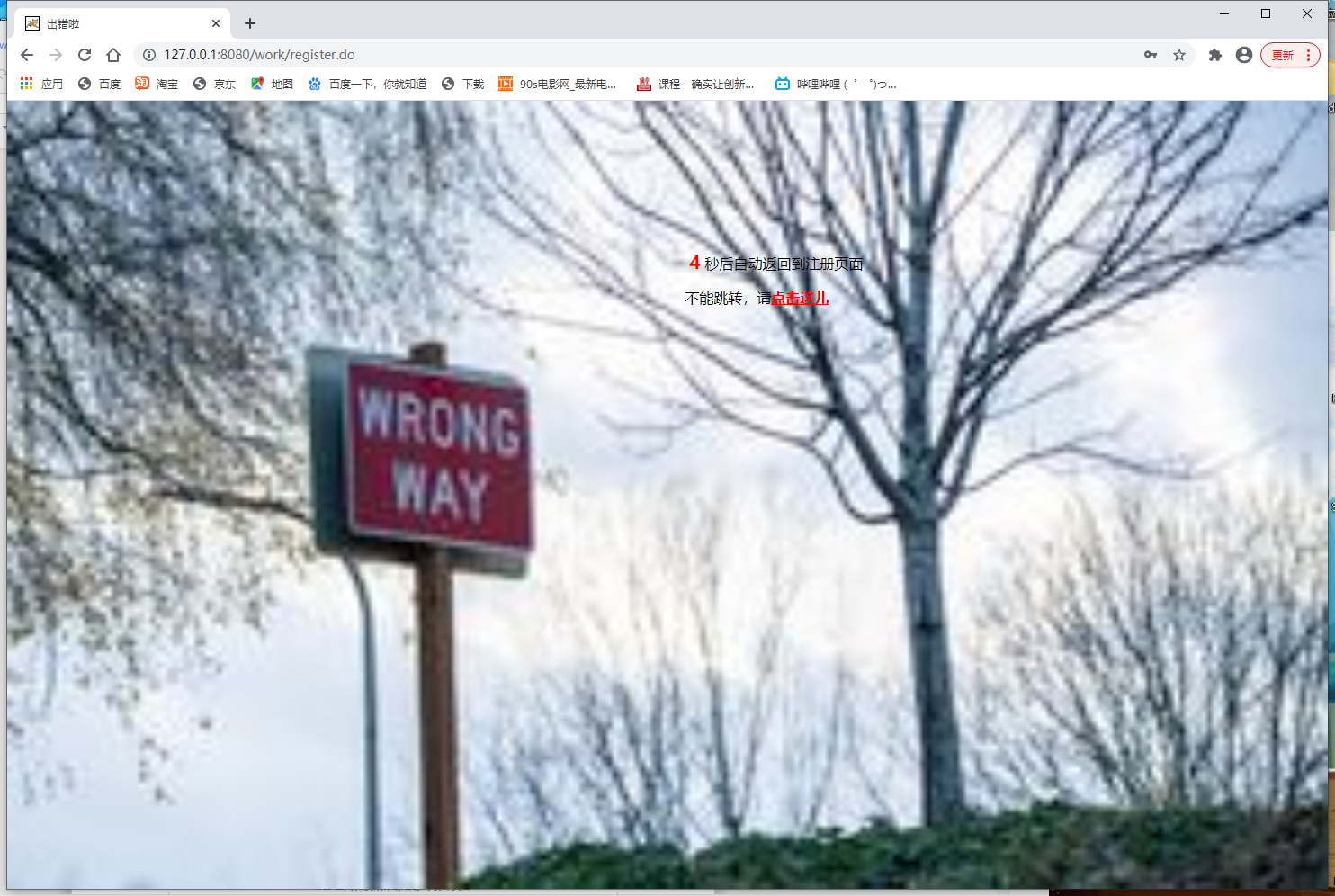
全部正确页面



点击注册后成功进入主页面



成功存入数据库



相同手机号再次注册后报错页面

# **5 系统总结**

验证码发送接口的使用一开始比较难，参考了很多网上的案例后终于明白如何使用，后来就比较顺利了，构思好用户输入信息如何校验，之后再写存入数据库的操作就可以了。