武汉纺织大学

Web应用开发课程设计

**短信验证实现用户注册**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 赵盼盼**

**学 号： 1809380108**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月19日**

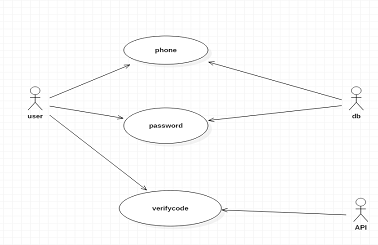
**1 需求分析**

实现通过向注册手机号发送验证码短信进行用户注册的功能，并将用户注册数据存进数据库内。

* 1. 用户注册：



1. **系统设计**
   1. 用例图

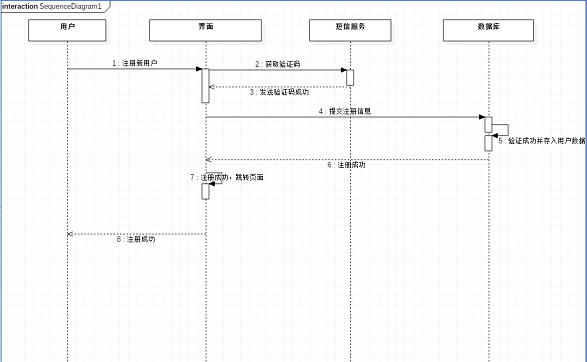


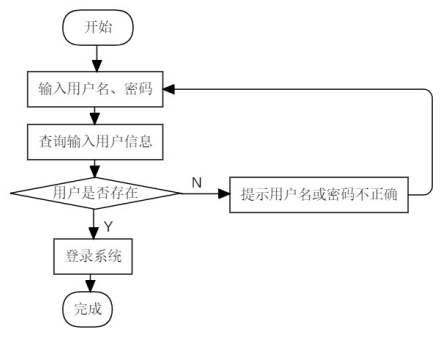
这个功能有三个角色参与，一是用户，用户进行收哦记号注册；而是数据库，注册手机号时可以通过数据库连接来识别该手机号有没有呗注册；三是榛子云短信服务API，调用它来使输入的手机号接收到验证码。

* 1. 时序图

用户注册：

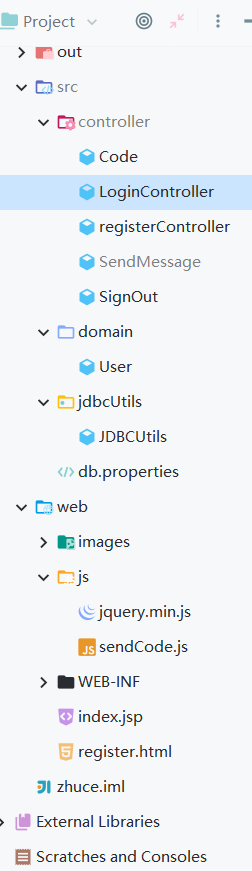
用户进行注册时获取验证码，获取成功并填写后提交注册信息，进行数据库查重验证，如果是新用户并且其他信息合法，则将用户数据存进数据库，返回注册成功信息。



登录活动图

**3.系统实现：**

3.1 项目结构



# 3.2 配置文件：

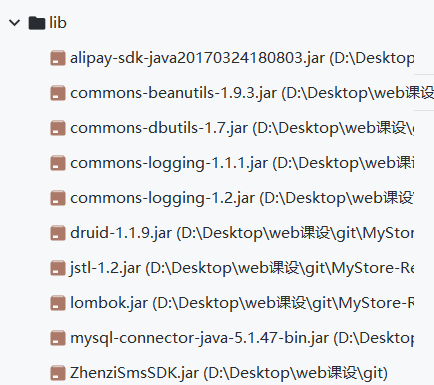
db.properties:

db.driverClassName=com.mysql.jdbc.Driver  
db.url=jdbc:mysql://localhost:3306/excise?useSSL=false  
db.username=root  
db.password=root

这里面是数据库配置信息，填上你的username和password

要点：mysql如果是8版本以上，只用改成：db.driverClassName=com.mysql.cj.jdbc.Driver

jar包：



新建的web项目没有采用maven管理；榛子云短信服务ZhenziSmsSDK.jar和连接池druid-1.1.9.jar

3.3 domain User.java

package domain;  
  
public class User {  
 private String password;  
 private String phone;  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public String getPhone() {  
 return phone;  
 }  
  
 public void setPhone(String phone) {  
 this.phone = phone;  
 }  
  
 @Override  
 public String toString() {  
 return "user [phone=" + phone + ", password=" + password + "]";  
 }  
}

controller类：

LoginController.java:

package controller;  
  
import java.io.IOException;  
import java.sql.SQLException;  
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.apache.commons.dbutils.QueryRunner;  
import org.apache.commons.dbutils.handlers.BeanHandler;  
import domain.User;  
import jdbcUtils.JDBCUtils;  
  
*/\*\*  
 \* Servlet implementation class LoginServlet  
 \*/*@WebServlet("/LoginController")  
public class LoginController extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 protected void service(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 //获取用户名和密码  
 String name = request.getParameter("username");  
 String pwd = request.getParameter("password");  
 System.*out*.println(name+"|密码："+pwd);  
 //到数据库中查询有没有该用户  
 QueryRunner qr = new QueryRunner(JDBCUtils.*getDataSource*());  
 String sql = "select \* from user where username=? and password=?";  
 User u = null;  
 try {  
 u = qr.query(sql, new BeanHandler<>(User.class),name,pwd);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 response.setContentType("text/html;charset=utf-8");  
 //判断有没有值  
 if (u != null) {  
 response.getWriter().write("登陆成功");  
 //保存用户（四个域 用session域）  
 *HttpSession* session = request.getSession();  
 session.setAttribute("user",u);  
 //跳转到主页面  
 response.setHeader("refresh", "3,url=/web/index.jsp");  
 }else {  
 response.getWriter().write("登录失败");  
 //跳转回注册页面  
 response.setHeader("refresh", "3,url=/web/register.html");  
 }  
 }  
}

regeisterController.java:

package controller;  
  
import java.io.IOException;  
import java.lang.reflect.InvocationTargetException;  
import java.sql.SQLException;  
import java.util.Arrays;  
import java.util.*Map*;  
import java.util.UUID;  
  
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 org.apache.commons.beanutils.BeanUtils;  
import org.apache.commons.dbutils.QueryRunner;  
  
import domain.User;  
import jdbcUtils.JDBCUtils;  
  
  
@WebServlet("/registerController")  
public class registerController extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 protected void service(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 System.*out*.println("RegisterController");  
 */\*\*  
 \* 判断输入的验证码是否正确  
 \* 接收所有参数  
 \* 把参数封装成user对象  
 \* 设置uid  
 \* 写入到数据库  
 \* \*/* //解决乱码问题  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("text/html;charset=utf-8");  
  
 //获取参数 验证码  
 String code = request.getParameter("code");  
 System.*out*.println("code="+code);  
 //获取服务器生成的验证码  
 String word = (String) this.getServletContext().getAttribute("checkCode");  
 //判断输入的验证  
 if (code.equals(word)) {  
 //如果正确  
 //1.接收所有参数  
 *Map*<String, String[]> parameterMap = request.getParameterMap();  
  
 for(Map.Entry<String, String[]> entry : parameterMap.entrySet()) {  
 System.*out*.println("----------");  
 System.*out*.println(entry.getKey()+":"+Arrays.*toString*(entry.getValue()));  
 }  
  
 User u = new User();  
 //2.把接收的参数封装成user对象  
 try {  
 BeanUtils.*populate*(u, parameterMap);  
 } catch (IllegalAccessException | InvocationTargetException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println(u);  
  
 //3.设置uid  
 u.setPhone(UUID.*randomUUID*().toString());  
 //4.写入数据库  
 QueryRunner qr = new QueryRunner(JDBCUtils.*getDataSource*());  
 String sql="insert into user value(?,?,?,?)";  
 try {  
 qr.update(sql,u.getPhone(),u.getPassword());  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 //跳转到登录页面  
 response.getWriter().write("注册成功");  
 response.setHeader("refresh", "3,url=/web/index.jsp");  
 } else {  
 //不正确，用户验证码错误，跳转回注册页  
 response.getWriter().write("验证码错误");  
 response.setHeader("refresh", "3;url=/web/register.jsp");  
 }  
 }  
  
}

SignOUt.java:

package controller;  
  
import java.io.IOException;  
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*;  
  
@WebServlet("/SignOutController")  
public class SignOut extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 protected void service(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 *HttpSession* session = request.getSession(false);  
 if(session == null) {  
 response.sendRedirect("/WEB-INF/index.jsp");  
 return;  
 }  
 session.invalidate();  
 response.sendRedirect("/WEB-INF/index.jsp");  
 }  
}

3.4 jdbcUtils:

JDBCUtils.java:

package jdbcUtils;  
  
import java.io.FileInputStream;  
import java.sql.*Connection*;  
import java.sql.*ResultSet*;  
import java.sql.*Statement*;  
import java.util.Properties;  
  
import javax.sql.*DataSource*;  
  
import com.alibaba.druid.pool.DruidDataSourceFactory;  
  
public class JDBCUtils {  
 private JDBCUtils(){}  
 public static *DataSource ds* = null;  
 static {// 只加载一次驱动就够了，为了保证代码只执行一次，使用静态代码块：当类加载的时候就会执行里面的内容  
 try {  
 //1.加载配置文件  
 Properties p = new Properties();  
 String path = JDBCUtils.class.getClassLoader().getResource("db.properties").getPath();  
 FileInputStream in = new FileInputStream(path);  
 p.load(in);  
 *ds* = DruidDataSourceFactory.*createDataSource*(p);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 //获取数据源  
 public static *DataSource* getDataSource() {  
 return *ds*;  
 }  
 public static *Connection* getConn() {  
 try {  
 // 2.连接数据库  
 return *ds*.getConnection();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
 public static void close(*Connection* conn, *Statement* st, *ResultSet* rs) {  
 // 5.释放资源【释放资源时先释放ResultSet,再释放Statement,最后释放Connection】  
 if (rs != null) {  
 try {  
 rs.close();  
 } catch (Exception e1) {  
 e1.printStackTrace();  
 }  
 }  
 if (st != null) {  
 try {  
 st.close();  
 } catch (Exception e2) {  
 e2.printStackTrace();  
 }  
 }  
 if (conn != null) {  
 try {  
 conn.close();  
 } catch (Exception e3) {  
 e3.printStackTrace();  
 }  
 }  
 }  
}

3.5 css文件

#loginForm{  
 width: 400px;  
 height: 500px;  
 background-color: #399cec80;  
 margin-left: 70%;  
 padding-top: 0px;  
 border: 1px solid #ccc;  
 border-radius: 1em;  
}  
#phone,  
#password1,  
#password2  
{  
 width: 260px;  
 height: 23px;  
}  
  
#checkCode  
{  
 width: 160px;  
 height: 23px;  
}  
  
#getCheckCode  
{  
 width: 95px;  
 height: 28px;  
 vertical-align: middle;  
}  
  
#register  
{  
 width: 270px;  
 height: 30px;  
 background-color: orange;  
}  
  
#login  
{  
 text-indent: 210px;  
}

Register.html:

<!DOCTYPE html>  
<html>  
 <head>  
 <meta charset="utf-8" />  
 <title>用户注册</title>  
 <link href="css/register.css" rel="stylesheet"/>  
 <style type="text/css">  
 .body{  
 background-image: url(images/log3.jpg);  
 background-repeat: no-repeat ;  
 background-size: 100%;  
 }  
 img{  
 margin: 30px;  
 cursor: pointer;  
 }  
 </style>  
 </head>   
 <body class="body">  
 <div id="loginForm">  
 <form>  
 <p style="font-size: larger; color: blue; margin-left: 30%;">账号注册</p>  
 <p>  
 <label>&nbsp;&nbsp;  
 <input type="text" id="phone" name="phone" placeholder="手机号码"/></label>  
 </p>  
 <p>  
 <label>&nbsp;&nbsp;  
 <input type="text" id="checkCode" name="checkCode" placeholder="校验码"/>  
 <button id="getCheckCode" name="getCheckCode" value="获取验证码" οnclick="sendCode(this)"></button>  
 </label>  
 </p>  
 <p>  
 <label>&nbsp;&nbsp;  
 <input type="password" id="password1" name="password1" placeholder="登陆密码"/>  
 </label>  
 </p>  
 <p>  
 <label>&nbsp;&nbsp;  
 <input type="password" id="password2" name="password2" placeholder="确认密码"/>  
 </label>  
 </p>  
 <p>&nbsp;&nbsp;  
 <label>  
 <input type="checkbox" />已阅读并同意《用户协议》  
 </label>  
 </p>  
 <p>&nbsp;&nbsp;  
 <label>  
 <input type="submit" id="register" value="立即注册" />  
 </label>  
 </p>  
 <p id="login">  
 <font >立即登陆>></font>  
 </p>  
 </form>  
 </div>  
  
 </body>  
</html>

3.6:jsp

成功界面：

<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>success</title>  
</head>  
<body>  
成功注册/登录，下面开始服务吧！  
</body>  
</html>

表单验证：

<script type="text/javascript">  
 function *change*(obj) {  
 obj.src = "${ctx}/Code?time=" + new Date().getTime();  
 }  
 function *checkForm*() {  
 var password = *checkPassword*();  
 var pwd2 = *ConfirmPassword*();  
 var phone = *checkPhone*();  
 return password && pwd2 && phone;  
// alert(nametip.value);  
 }  
 //验证密码  
 function *checkPassword*() {  
 var password = document.getElementById('password');  
 var errpass = document.getElementById('errpass');  
 var pattern = /^\w{8,16}$/; //密码要在8-16位  
 if (!pattern.test(password.value)) {  
 errpass.innerHTML = "密码不合规范"  
 errpass.className = "error"  
 return false;  
 } else {  
 errpass.innerHTML = "OK"  
 errpass.className = "success";  
 return true;  
 }  
 }  
 //确认密码  
 function *ConfirmPassword*() {  
 var userpasswd = document.getElementById('userPasword');  
 var userConPassword = document.getElementById('userConfirmPasword');  
 var errConPasswd = document.getElementById('conPasswordErr');  
 if ((userpasswd.value) != (userConPassword.value)  
 || userConPassword.value.length == 0) {  
 errConPasswd.innerHTML = "上下密码不一致"  
 errConPasswd.className = "error"  
 return false;  
 } else {  
 errConPasswd.innerHTML = "OK"  
 errConPasswd.className = "success";  
 return true;  
 }  
 }  
 //验证手机号  
 function *checkPhone*() {  
 var userphone = document.getElementById('userPhone');  
 var phonrErr = document.getElementById('phoneErr');  
 var pattern = /^1[34578]\d{9}$/; //验证手机号正则表达式  
 if (!pattern.test(userphone.value)) {  
 phonrErr.innerHTML = "手机号码不合规范"  
 phonrErr.className = "error"  
 return false;  
 } else {  
 phonrErr.innerHTML = "OK"  
 phonrErr.className = "success";  
 return true;  
 }  
 }  
 </script>

# **4 系统总结**

这次课设没有很好的安排时间，导致时间不足；后来由于SSM框架还没有掌握好，做的不是很好；跟程胜同学换了课题，做手机验证登录课题。主要是两方面，一个是将数据存进数据库，一个是短信API服务。在设计后者时，用到的是榛子云短信服务，可能是代码有点问题，在获取验证码的时候手机号一直收不到消息，点击的时候也会出现刷新操作；感觉非常缺少经验所以花了很多时间在跟后端交互这一块；经过这次课程设计，清晰的认知在java web学习中的不足，以及对框架的把握不准，所以这次采用的设计非常的普通，希望能在后面的学习中能多运用框架的知识，进行工程的轻便开发。经过这次课设，感觉学习web还是要花很多心思获得收获，给实战经验带来写提升。