组长:魏志航

组员:兰天阳，彭尹垚，张世纪，叶翔昊，谢泽川

总体情况：

**魏志航**

今天主要学习了

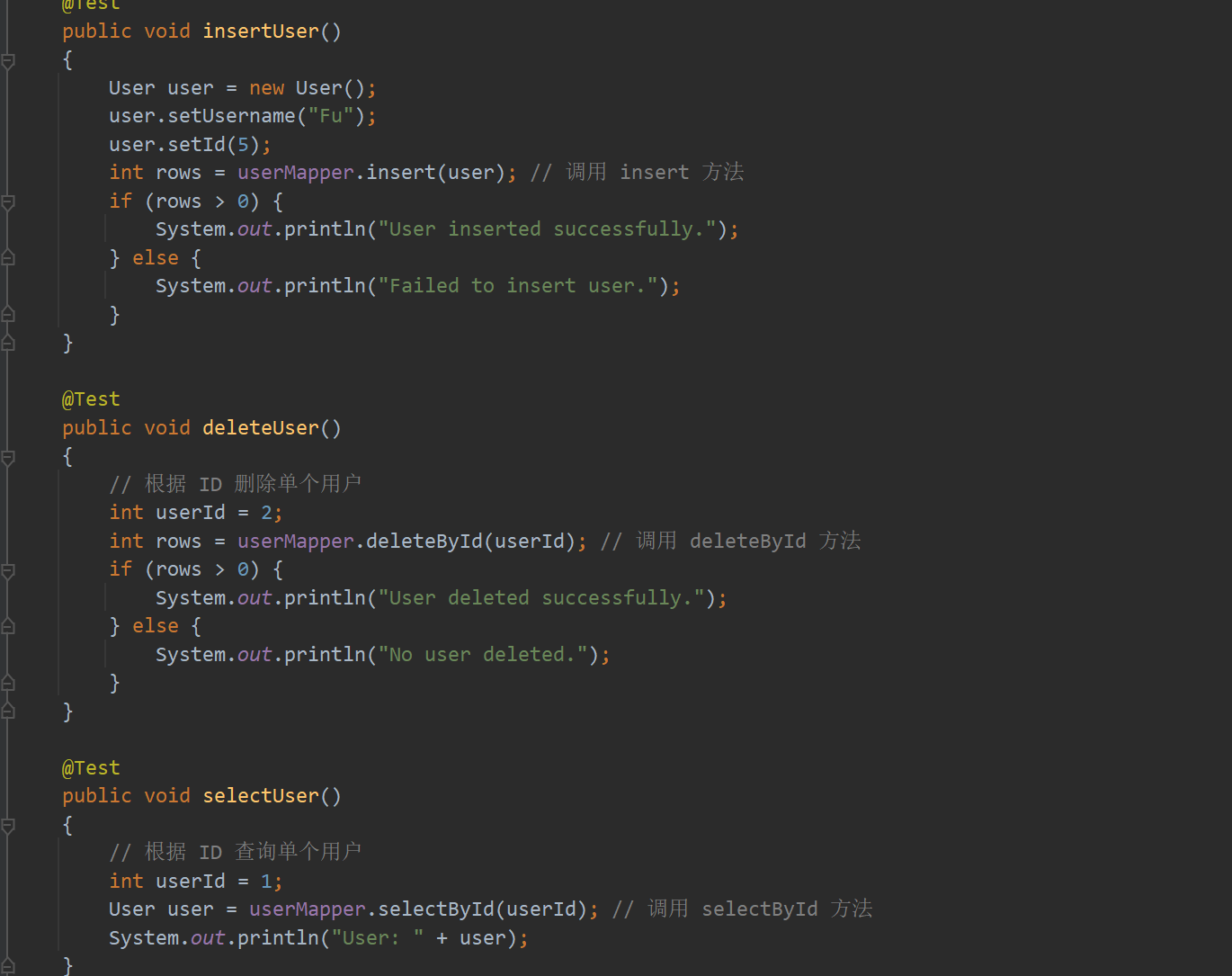
1.Mybatis-plus中的BaseMapper

2.Mybatis-plus中的Service

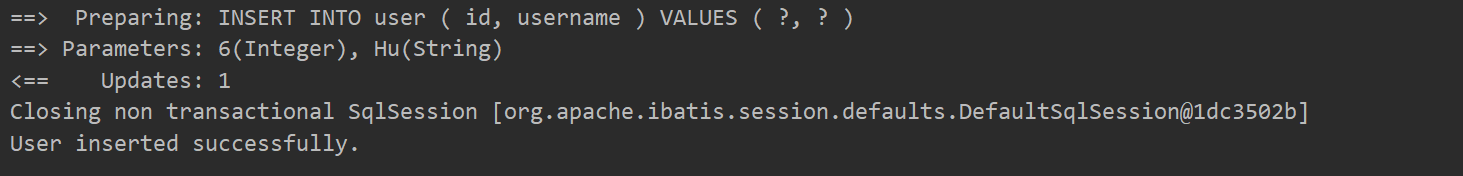
3.浏览器请求控制器层

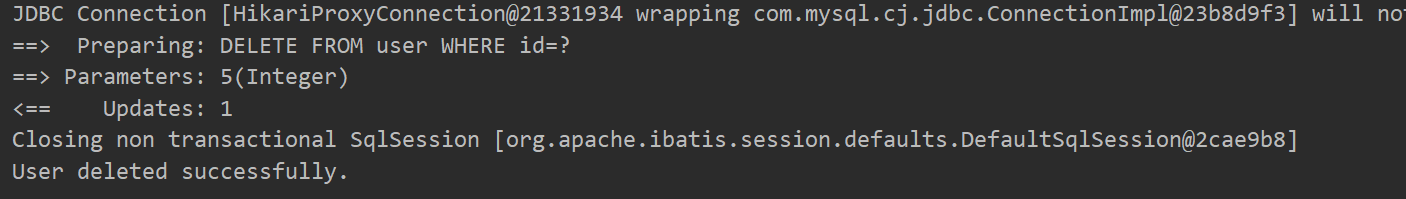
4.前端：用hbuilder中编写HTML、div、css、vue

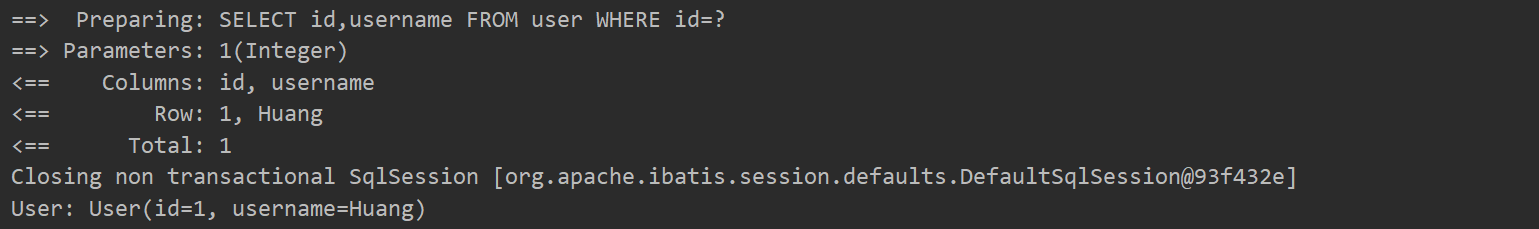
**1.测试Mapper方法中的增删查改均成功：**

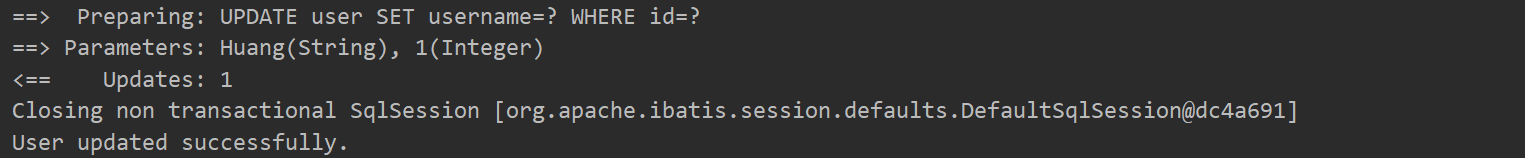




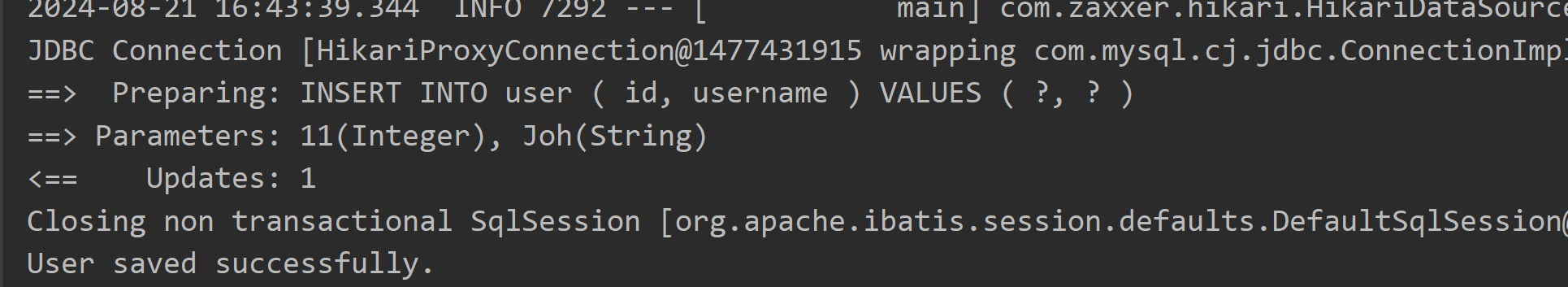


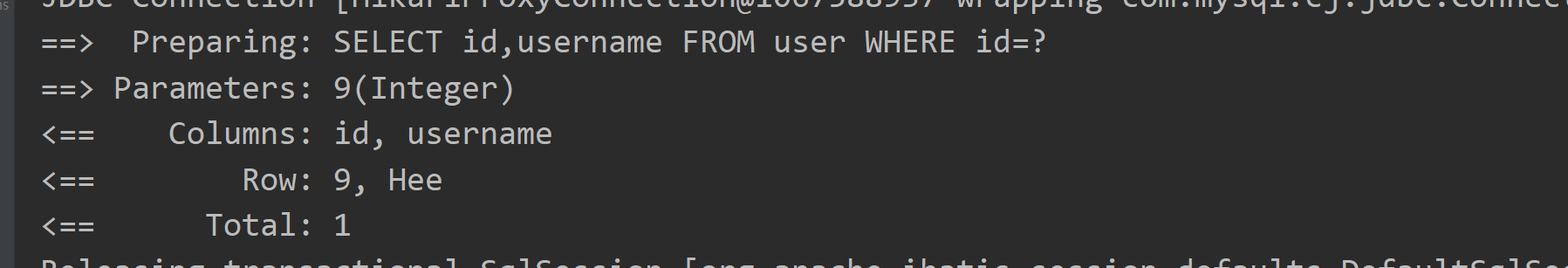


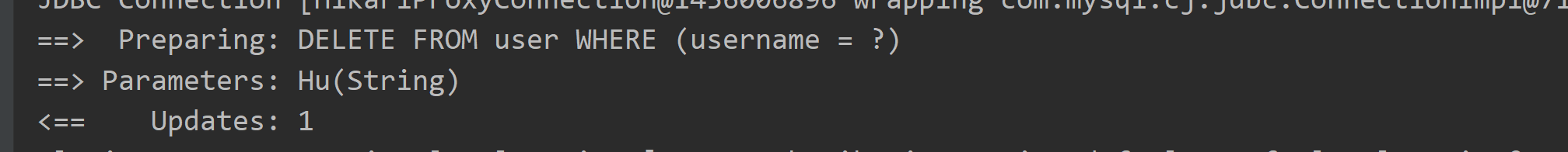


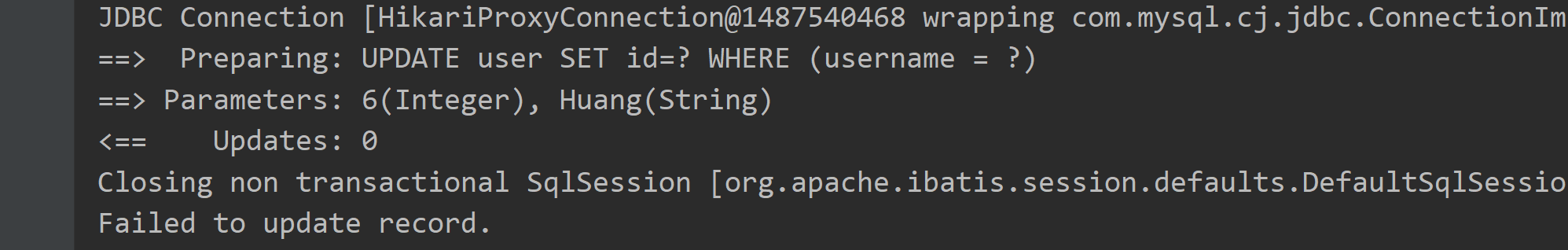


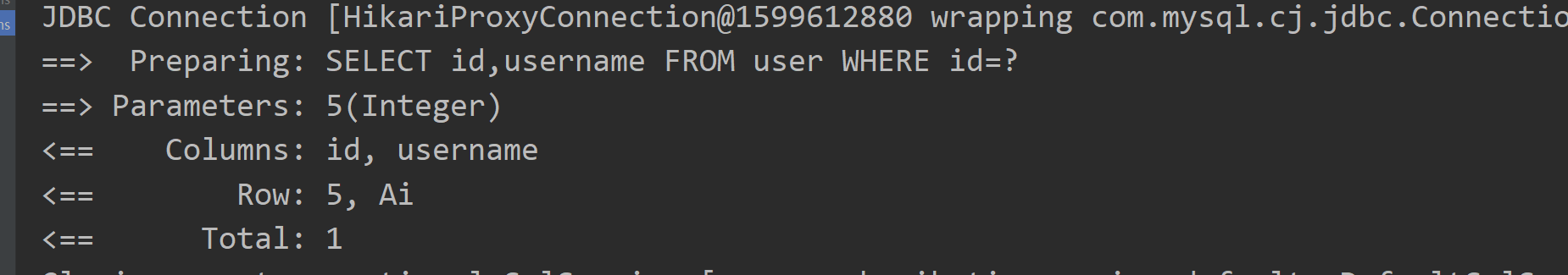
**2.Service方法测试成功**

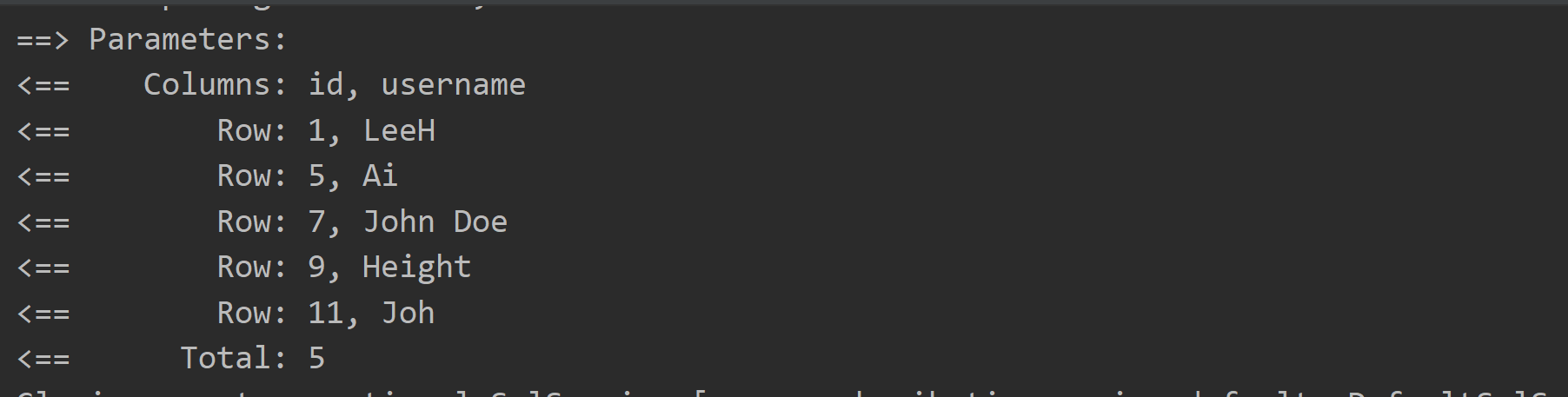


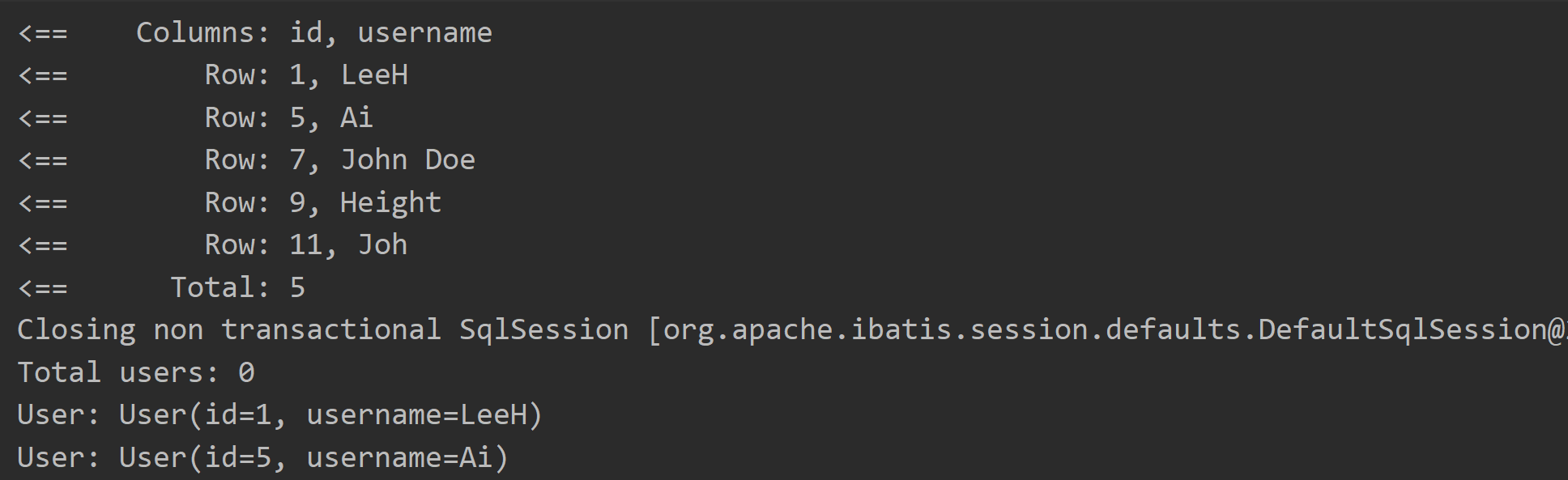


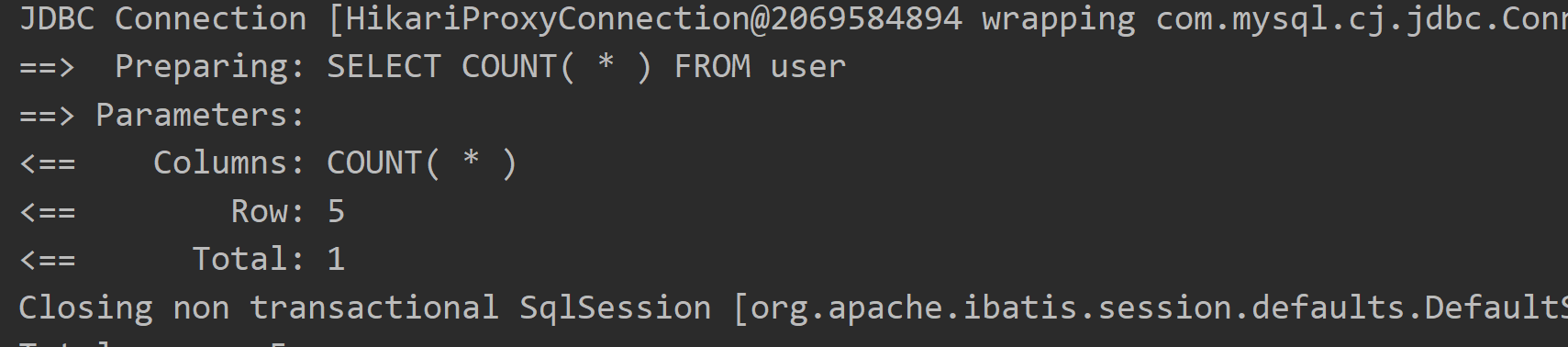












**代码：**

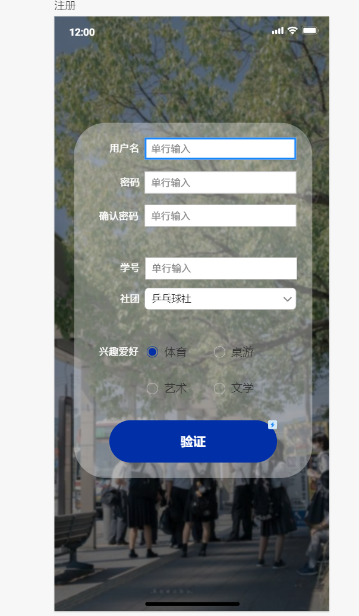
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.baomidou.mybatisplus.core.conditions.update.UpdateWrapper;  
import com.baomidou.mybatisplus.core.metadata.IPage;  
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;  
import com.cq.cd.pojo.User;  
  
import org.junit.jupiter.api.Test;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.SpringBootTest;  
import java.util.List;  
  
@SpringBootTest  
public class UserServiceTest {  
 @Autowired  
 public UserService userService;  
  
 @Test  
 public void saveUser()  
 {  
 // 假设有一个 User 实体对象  
 User user = new User();  
 user.setUsername("Joh");  
 user.setId(11);  
 boolean result = userService.save(user); // 调用 save 方法  
 if (result) {  
 System.*out*.println("User saved successfully.");  
 } else {  
 System.*out*.println("Failed to save user.");  
 }  
 }  
 @Test  
 public void saveOrUpdateUser()  
 {  
 // 假设有一个 User 实体对象，其中 id 是 TableId 注解的属性  
 User user = new User();  
 user.setId(9);  
 user.setUsername("Height");  
 boolean result = userService.saveOrUpdate(user); // 调用 saveOrUpdate 方法  
 if (result) {  
 System.*out*.println("User updated or saved successfully.");  
 } else {  
 System.*out*.println("Failed to update or save user.");  
 }  
 }  
  
 @Test  
 public void removeUser()  
 {  
 // 假设有一个 QueryWrapper 对象，设置删除条件为 name = 'John Doe'  
 QueryWrapper<User> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("username", "Hu");  
 boolean result = userService.remove(queryWrapper); // 调用 remove 方法  
 if (result) {  
 System.*out*.println("Record deleted successfully.");  
 } else {  
 System.*out*.println("Failed to delete record.");  
 }  
 }  
  
 @Test  
 public void updateUser()  
 {  
 // 假设有一个 UpdateWrapper 对象，设置更新条件为 name = 'John Doe'，更新字段为 email  
 UpdateWrapper<User> updateWrapper = new UpdateWrapper<>();  
 updateWrapper.eq("username", "Huang").set("id", 6);  
 boolean result = userService.update(updateWrapper); // 调用 update 方法  
 if (result) {  
 System.*out*.println("Record updated successfully.");  
 } else {  
 System.*out*.println("Failed to update record.");  
 }  
 }  
  
 @Test  
 public void getUser()  
 {  
 // 假设要查询 ID 为 1 的用户  
 User user = userService.getById(5); // 调用 getById 方法  
 if (user != null) {  
 System.*out*.println("User found: " + user);  
 } else {  
 System.*out*.println("User not found.");  
 }  
 }  
  
 @Test  
 public void listUser()  
 {  
 // 查询所有用户  
 List<User> users = userService.list(); // 调用 list 方法  
 for (User user : users) {  
 System.*out*.println("User: " + user);  
 }  
  
 }  
  
 @Test  
 public void pageUser()  
 {  
 // 假设要进行无条件的分页查询，每页显示10条记录，查询第1页  
 IPage<User> page = new Page<>(1, 2);  
 IPage<User> userPage = userService.page(page); // 调用 page 方法  
 List<User> userList = userPage.getRecords();  
 long total = userPage.getTotal();  
 System.*out*.println("Total users: " + total);  
 for (User user : userList) {  
 System.*out*.println("User: " + user);  
 }  
 }  
  
 @Test  
 public void countUser()  
 {  
 // 查询用户表中的总记录数  
 long totalUsers = userService.count(); // 调用 count 方法  
 System.*out*.println("Total users: " + totalUsers);  
 }  
}

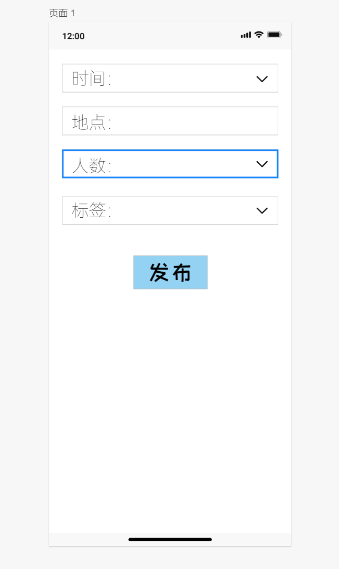
**3.浏览器显示“Hello Springboot”和List成功**





4.设计了原型中手机端的活动、登录注册、发布活动等页面



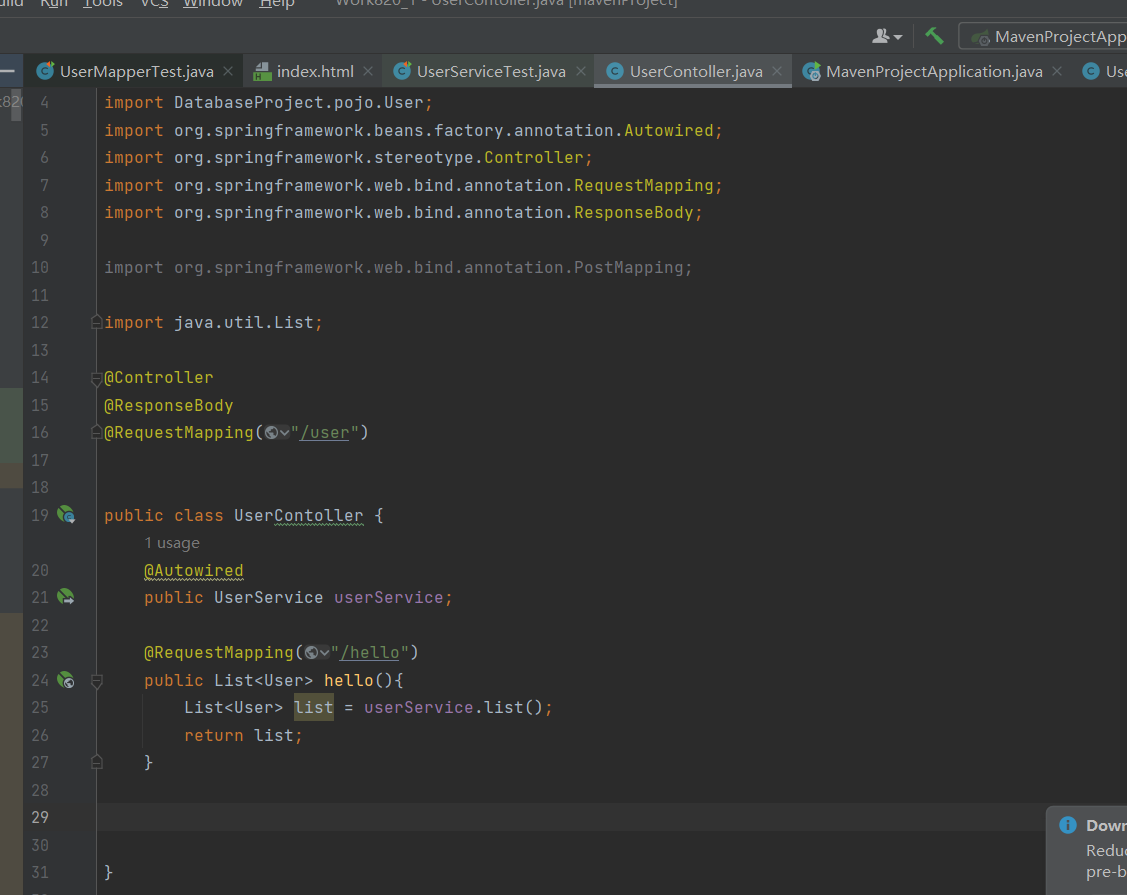


**彭尹垚**

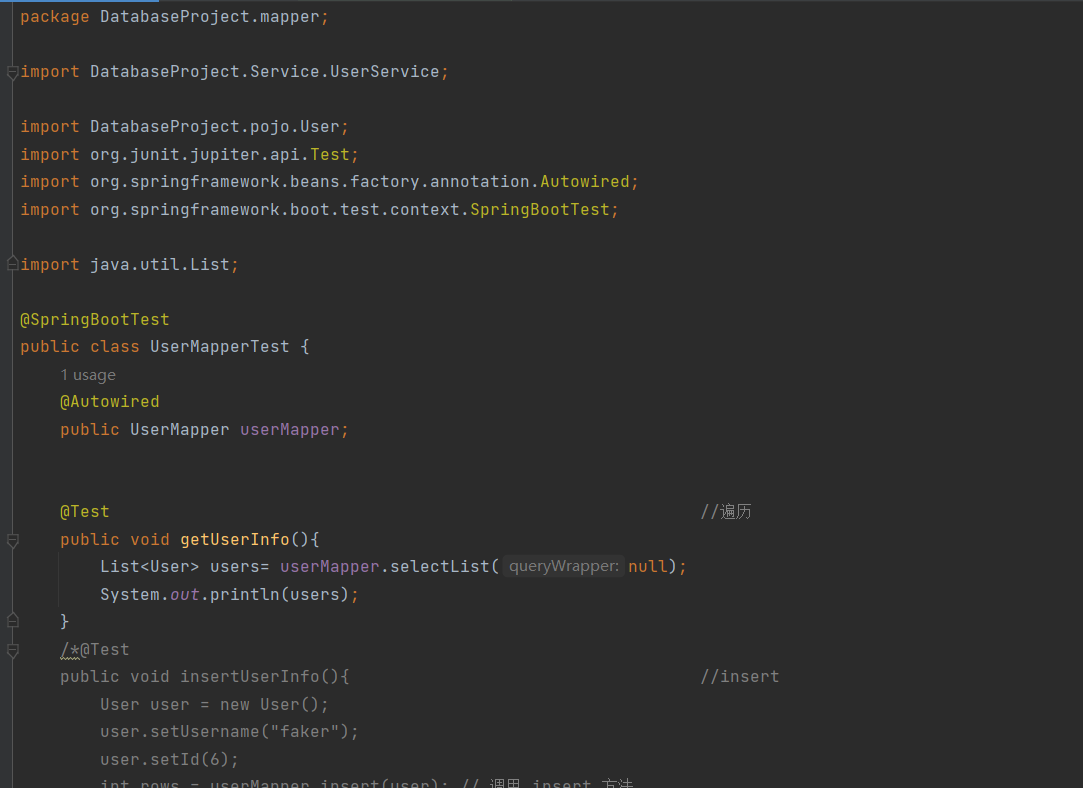
Java日志

今天主要学了html前端，配置了springboot启动，连接网页和本地数据库，将本地数据库的信息打印到<http://localhost>

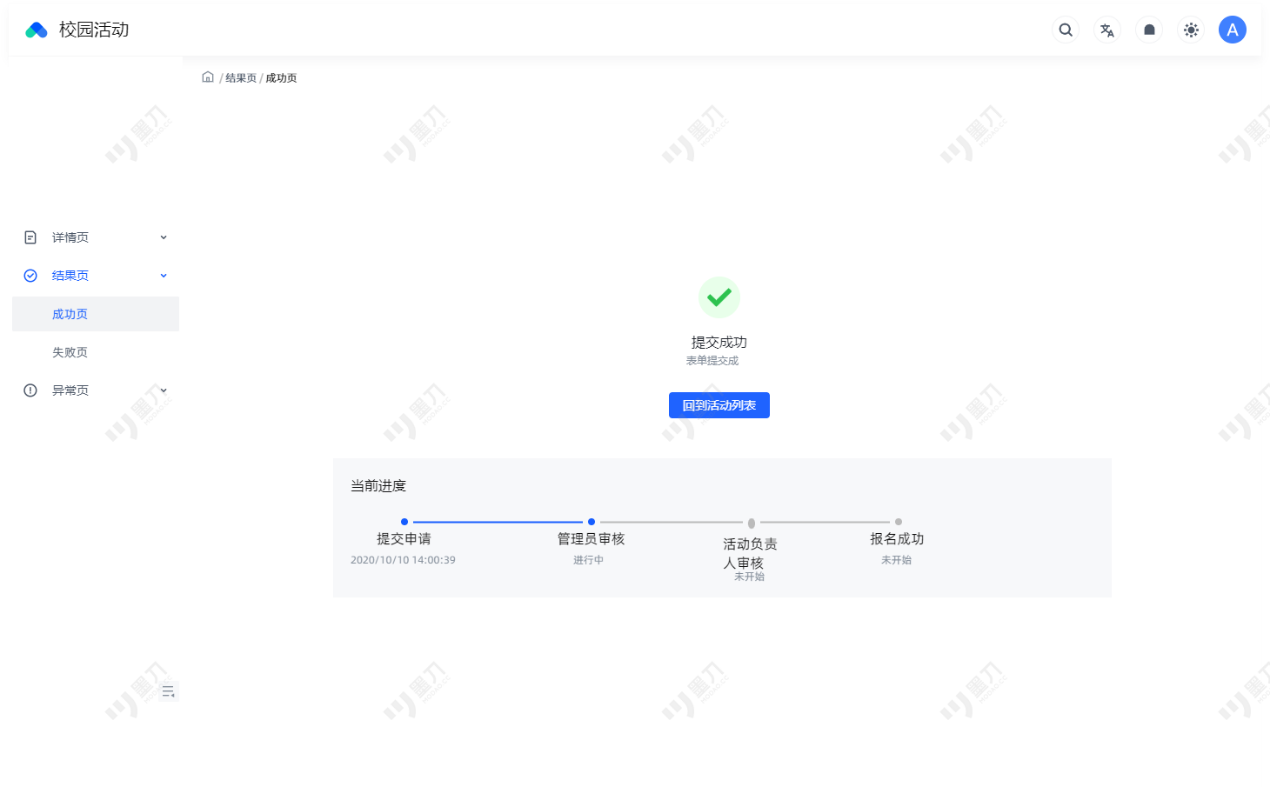
并且用mapper和service两种方法操作

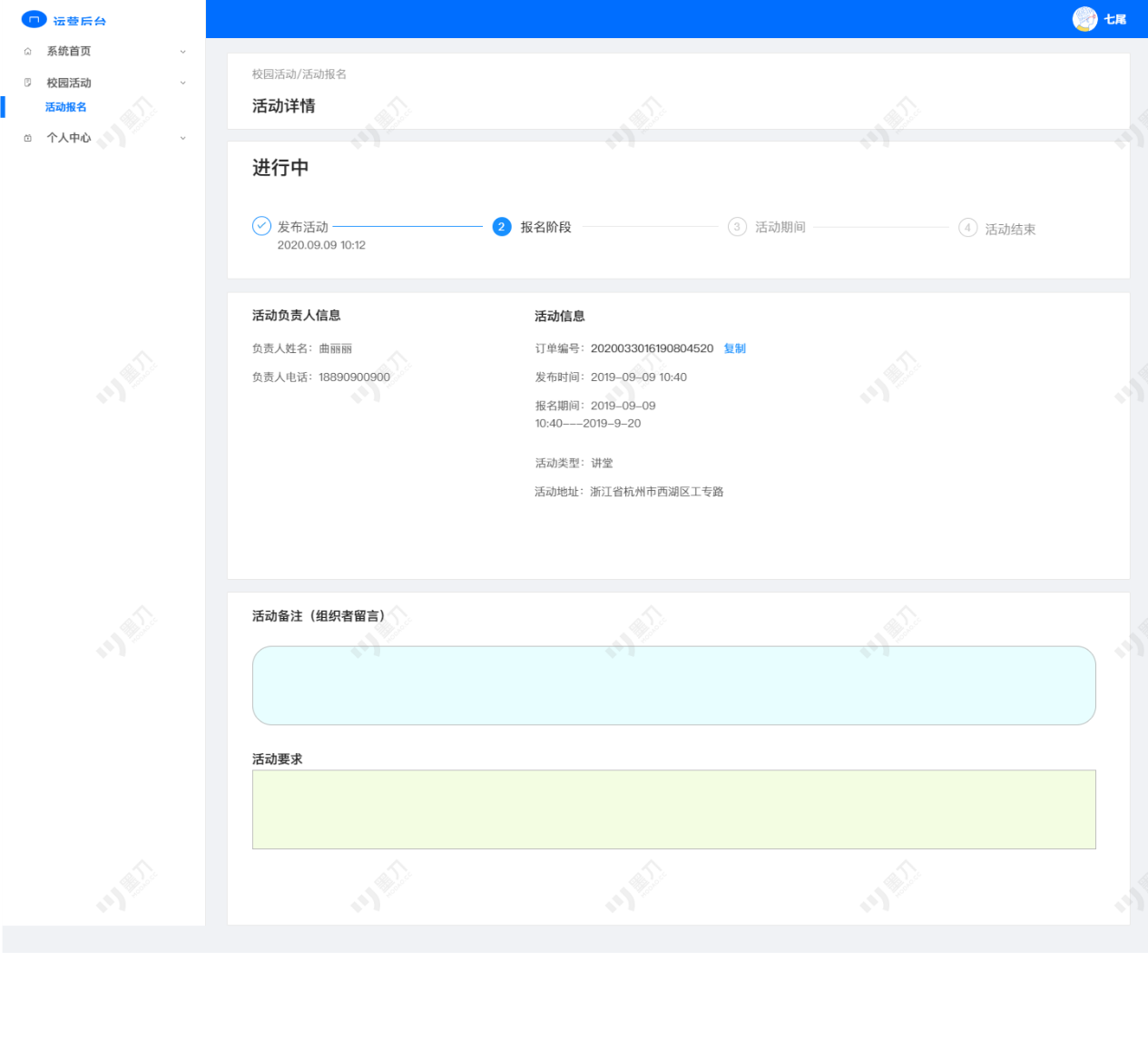
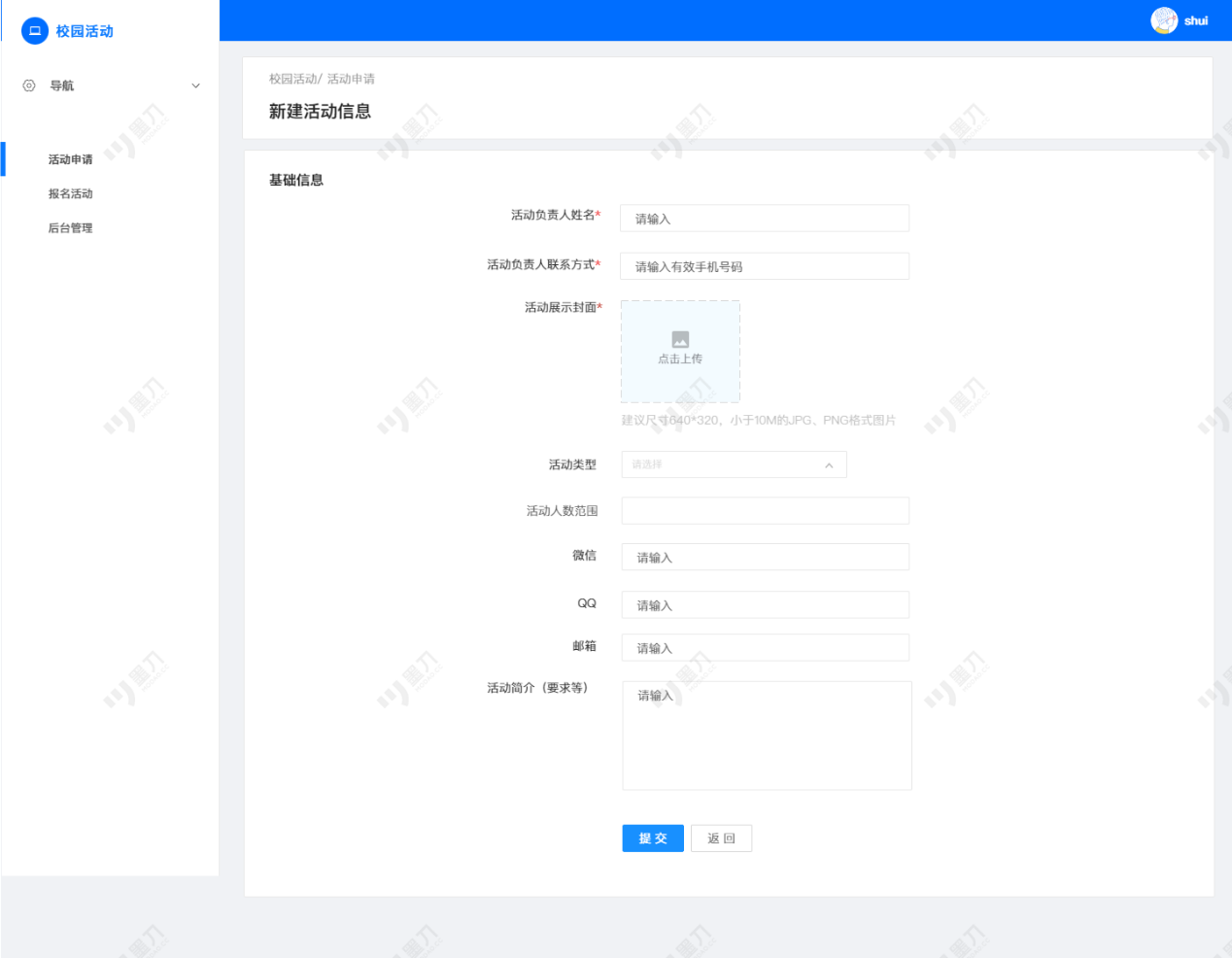






并且设计了网页端的原型





**谢泽川**

##### Java学习日志 - JAVASpringboot的框架持久层服务层和控制器，前端的编写和网页连接

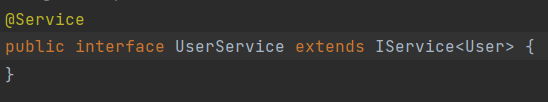
##### 日期：2024年8月21日

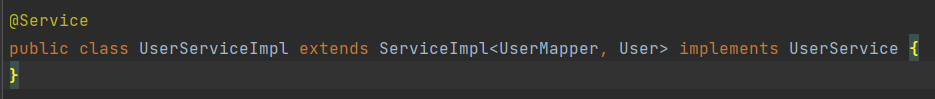
##### 学习内容：

控制层：



业务层：



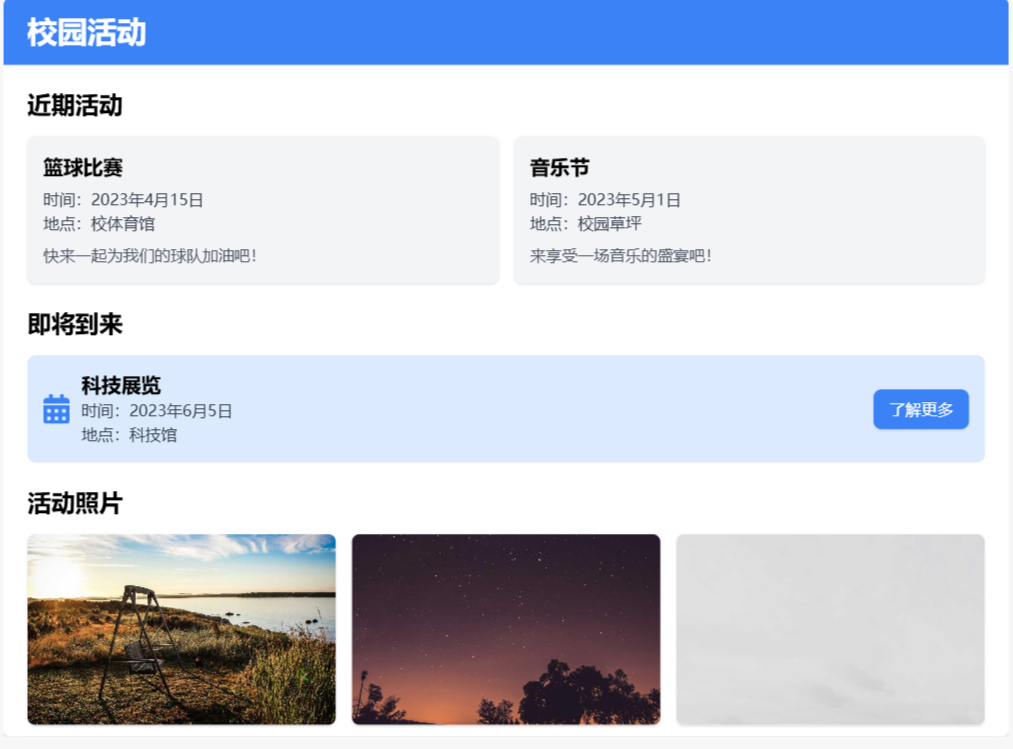


持久层：



原型的设计：

校园活动展示：



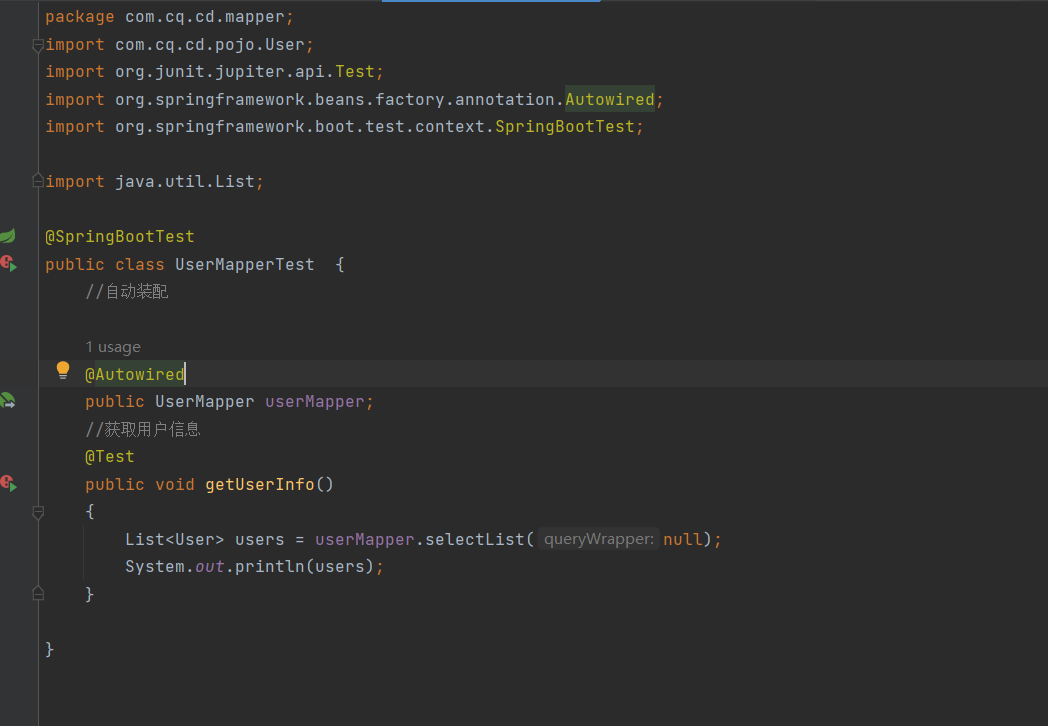
活动后台管理：



活动管理：



**张世纪**

1. 昨天的依然没调通，但发现了之前遗留的问题，今天的还没做。
2. 做了小组任务的基础设计





**叶翔昊**

8·21实训日志

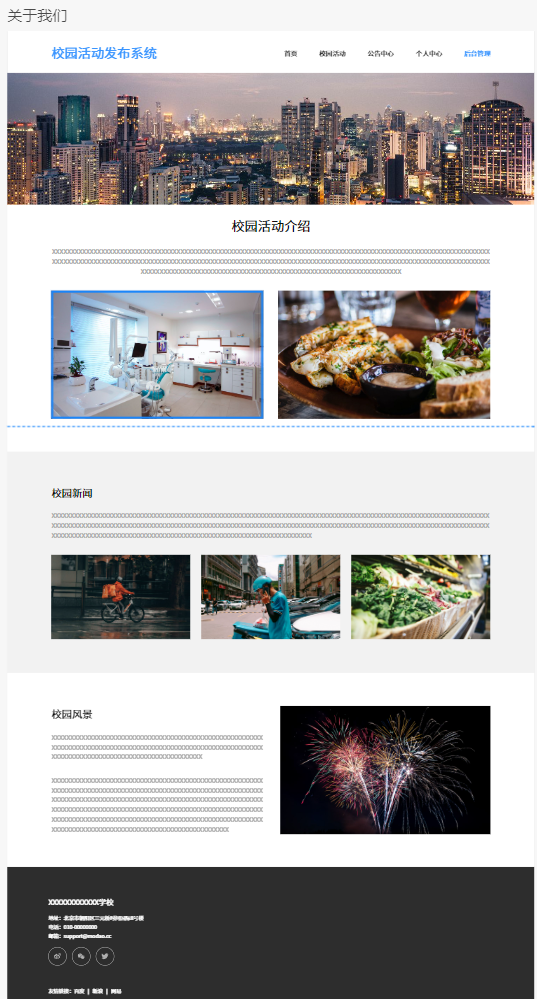
总体情况：任务完成良好，没有什么问题。

任务：

完成原型设计图。

完成情况：

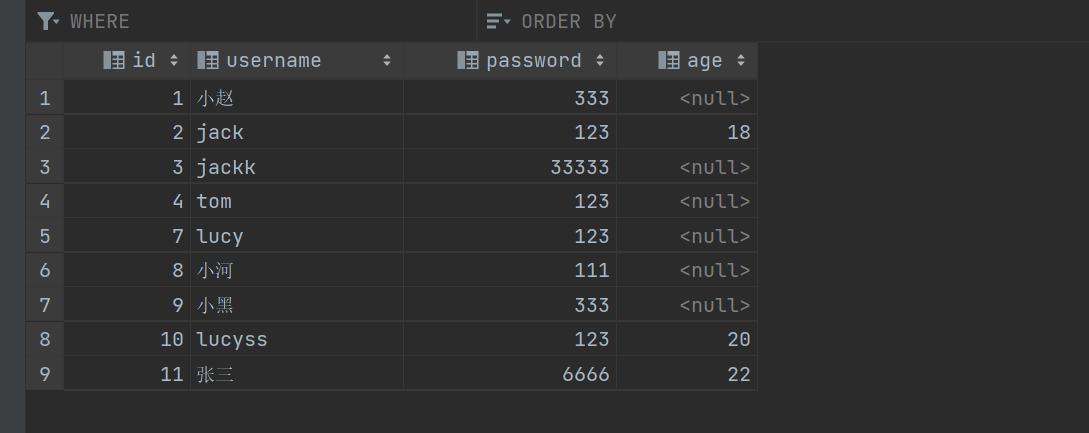




**兰天阳**

2024.8.21

数据库如下:



Mapper层：

package com.cq.cd.mapper;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import com.cq.cd.pojo.User;

import org.apache.ibatis.annotations.Mapper;

@Mapper

public interface UserMapper extends BaseMapper<User> {

}

package com.cq.cd.mapper;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.cq.cd.pojo.User;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import java.util.List;

@SpringBootTest

public class UserMapperTest {

//自动装配

@Autowired

public UserMapper userMapper;

//根据id获取用户信息

@Test

public void getUerInfoById(){

User user = userMapper.selectById(1);

System.out.println(user);

}

//获取全部用户信息

@Test

public void getUserInfo(){

List<User> user = userMapper.selectList(null);

System.out.println(user);

}

//插入用户数据

@Test

public void insertUserTest(){

User user = new User();

user.setUsername("张三");

user.setId(11);

user.setAge(22);

user.setPassword(6666);

int insert = userMapper.insert(user);

System.out.println(insert);

}

//根据id删除用户数据

@Test

public void deleteUserByIdTest(){

int rows = userMapper.deleteById(11);

System.out.println(rows);

}

//使用条件构造器删除用户数据

@Test

public void deleteUserByQueryWrapperTest(){

// 假设有一个 QueryWrapper 对象，设置查询条件为 age > 21，删除满足条件的用户

QueryWrapper<User> queryWrapper = new QueryWrapper<>();

queryWrapper.gt("age", 21);

System.out.println(queryWrapper);

}

//根据id更新数据

@Test

public void updateUserByIdTest(){

User updateUser = new User();

updateUser.setId(11);

updateUser.setUsername("哈哈");

int rows = userMapper.updateById(updateUser);

if (rows > 0) {

System.out.println("User updated successfully.");

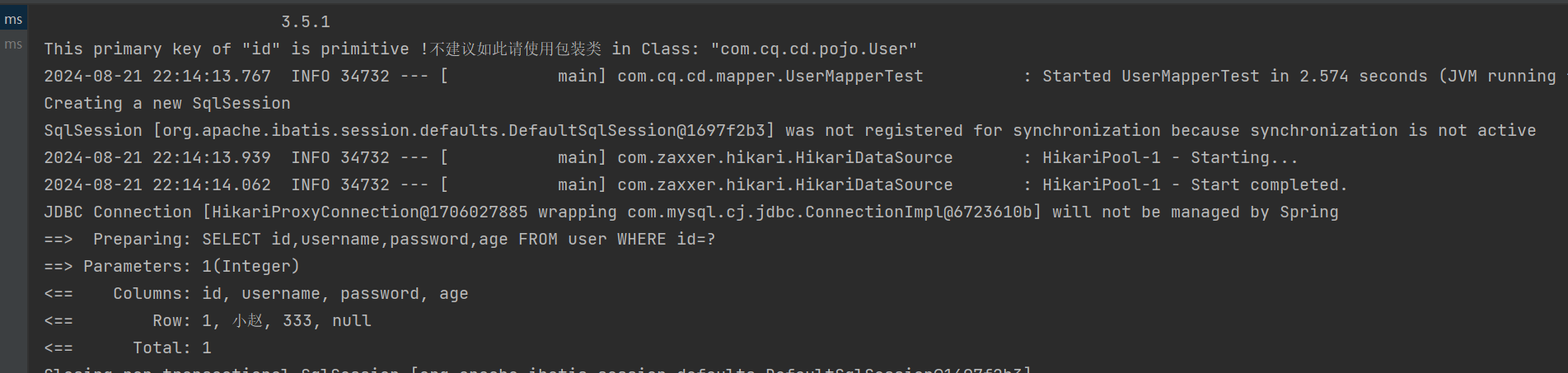
} else {

System.out.println("No user updated.");

}

}

}



Service层

package com.cq.cd.IService;

import com.baomidou.mybatisplus.extension.service.IService;

import com.cq.cd.pojo.User;

public interface UserIService extends IService<User> {

}

package com.cq.cd.IService;

import com.cq.cd.pojo.User;

import org.junit.jupiter.api.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import java.util.Arrays;

import java.util.Collection;

import java.util.List;

@SpringBootTest

public class UserIServiceTest {

@Autowired

public UserIService userIService;

//获取全部用户信息

@Test

public void getUserListTest(){

List<User> list = userIService.list();

System.out.println(list);

}

//获取或更新用户信息

@Test

public void getOrUpdateListTest(){

User user = new User();

user.setId(11);

user.setUsername("John Doe");

user.setAge(66);

boolean result = userIService.saveOrUpdate(user);

if (result) {

System.out.println("User updated or saved successfully.");

} else {

System.out.println("Failed to update or save user.");

}

}

//删除用户信息

@Test

public void removeByIdTest(){

boolean result = userIService.removeById(11); // 调用 removeById 方法

if (result) {

System.out.println("User deleted successfully.");

} else {

System.out.println("Failed to delete user.");

}

}

//获取给定条件信息

@Test

public void getOneTest(){

// 假设有一个 QueryWrapper 对象，设置查询条件为 name = 'John Doe'

QueryWrapper<User> queryWrapper = new QueryWrapper<>();

queryWrapper.eq("username", "Jack");

User user = userIService.getOne(queryWrapper);

if (user != null) {

System.out.println("User found: " + user);

} else {

System.out.println("User not found.");

}

}

@Test

public void listTest(){

// 假设有一组 ID 列表，批量查询用户

List<Integer> ids = Arrays.asList(1, 2, 3);

Collection<User> users = userIService.listByIds(ids); // 调用 listByIds 方法

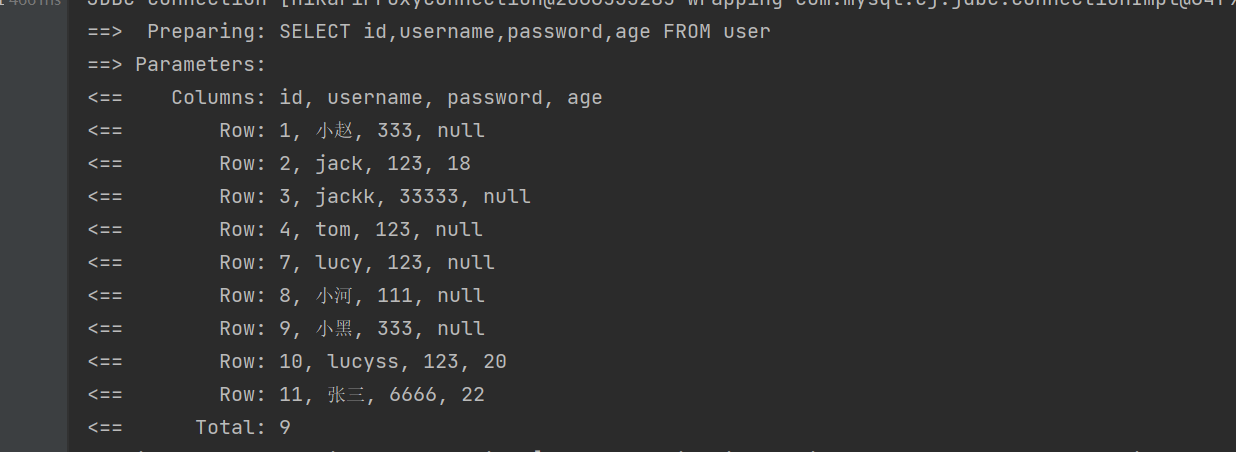
for (User user : users) {

System.out.println("User: " + user);

}

}

}



Controller层

package com.cq.cd.controller;

import com.cq.cd.IService.UserIService;

import com.cq.cd.pojo.User;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

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

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

import java.util.List;

@Controller

@ResponseBody

@RequestMapping("/user")

public class UserController {

@Autowired

public UserIService userIService;

@RequestMapping("/hello")

public String Hello(){

return "hello springboot";

}

@RequestMapping("/list")

public List<User> getUserList(){

List<User> list = userIService.list();

return list;

}

}

