**2021年天津市大学生信息技术“新工科”**

**工程实践创新技术竞赛**

**参赛作品文档**

**参赛作品名称： 人际关系自动维护**

**学 校： 天津职业技术师范大学**

**学 院： 信息技术工程学院**

**团队名称： victory**

**指导教师： 谷兆麟**

**参赛成员： 刘庆成 况雪**

# 目 录

一、技术竞赛**登记表 ------------------------------ 3**

# 作品文档方案部分----------------------------- 4

2.1方案概述------------------------------------ 4

## 2.2作品的创新点-------------------------------- 4

## 2.3系统实现原理-------------------------------- 4

2.4软件设计------------------------------------- 4

## 2.5系统测试及结果------------------------------- 5

## 2.6程序源代码----------------------------------- 6

1. **照片及说明----------------------------------- 11**

# 一、2021年天津市大学生信息技术“新工科”工程实践创新技术竞赛**登记表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题目 | 人际关系自动维护 | | | | |
| 学校及学院 | 天津职业技术师范大学 信息技术工程学院 | | | | |
| 学生姓名1 | 刘伟怡 | 专业班级 | 软件1902 | 联系方式  （手机、邮箱） | 18222695630  l2158223895@qq.com |
| 学生姓名2 | 刘庆成 | 专业班级 | 软件2001 | 联系方式  （手机、邮箱） | 18790198945  2807652103@qq.com |
| 学生姓名3 | 况雪 | 专业班级 | 软件1901 | 联系方式  （手机、邮箱） | 18822290933  1765956112@qq.com |
| 指导教师 | 谷兆麟 | | | | |
| 作品关键词 | Netty、聊天提示 | | | | |
| 作品摘要 | 本项目是由人与人网上沟通为基础，提醒用户维护人际关系。平台采用springboot+netty框架异步通信提高了处理链接的并发性，客户端发送数据采用ajax接收，采用MySQL数据库存储客户端向服务端发来的数据，服务端未接收到消息时定时向客户端提醒发送消息，用户聊天次数由进度条显示。通过对此项目的研究我们了解到沟通是建立良好关系重要的一部分，沟通过程可能因沟通者本身的特质或沟通的方式而造成曲解，因此传送讯息者与接收者间必须借着不断的回馈，去澄清双方接收及了解到的是否一致。 | | | | |
| 主要参考  文献 | https://www.cnblogs.com/han-1034683568/p/7199168.html  https://www.cnblogs.com/msi-chen/p/10974009.html  https://blog.csdn.net/pbrlovejava/article/details/104186101  https://blog.csdn.net/u013144287/article/details/77099579/ | | | | |
| 备注 | 文档要求思路清晰、描述详细而明确、无错别字、图文并茂。 | | | | |
| * 特别   声明 | **提交内容均原创，作品属于参赛选手，大赛组委会有发布权和使用权。** | | | | |

# 二、作品文档方案部分

## 1.方案概述

沟通是人际关系中最重要的一部分，它是人与人之间傅递情感、态度、事实、信念和想法的过程，所以良好的沟通指的就是一种双向的沟通过程，不是你一个人在发表演说、对牛弹琴，或者是让对方唱独角戏，而是用心去听听对方在说什么？去了解对方在想什么？对方有什么感受？并且把自己的想法及时回馈给对方。建立良好的人际关系，不仅能让你在遇到困难时，获得帮助，更重要的是，这对自身能力的提高也有很大帮助。但是随着科技的发展生活质量的提高，人们也开始变得繁忙了起来，没有时间或忘记维护人际关系，所以我们团队设计和研究出了此平台帮助用户维护人际关系。

## 作品的创新点

本平台可以定期的维护和提醒用户维护人际关系，能够促进人们之间的信息交流和信息共享，人们可以从有好的人际关系中吸取力量，增强信心，在人生业绩的创造中得心应手，其次良好的人际关系是人身心健康的需要，一个人如果处在相互关心相互爱护，关心密切融洽的人际关系中，一定心情舒畅，有益于身体健康。良好的人际关系能使人保持轻松平稳，态度乐观。不良的人际关系，可干扰人的情绪，使人产生焦虑、不安和抑郁。良好的人际关系有利于营造使人在物质生产过程中充分发挥创造力的优化环境，人的积极性创造性的发挥能增加物质财富的生产，丰富人们的物质生活。

## 系统实现原理

系统由springboot+gradle框架搭建，用户登录注册的数据通过ajax转换成json 格式先存入hashmap中然后存入mysql数据库，用户聊天的数据通过websocket长链接传入netty的服务端进行处理，然后传入给用户好友，若用户好友长时间未和好友聊天，通过netty心跳机制向用户发送消息提醒用户维护人际关系。

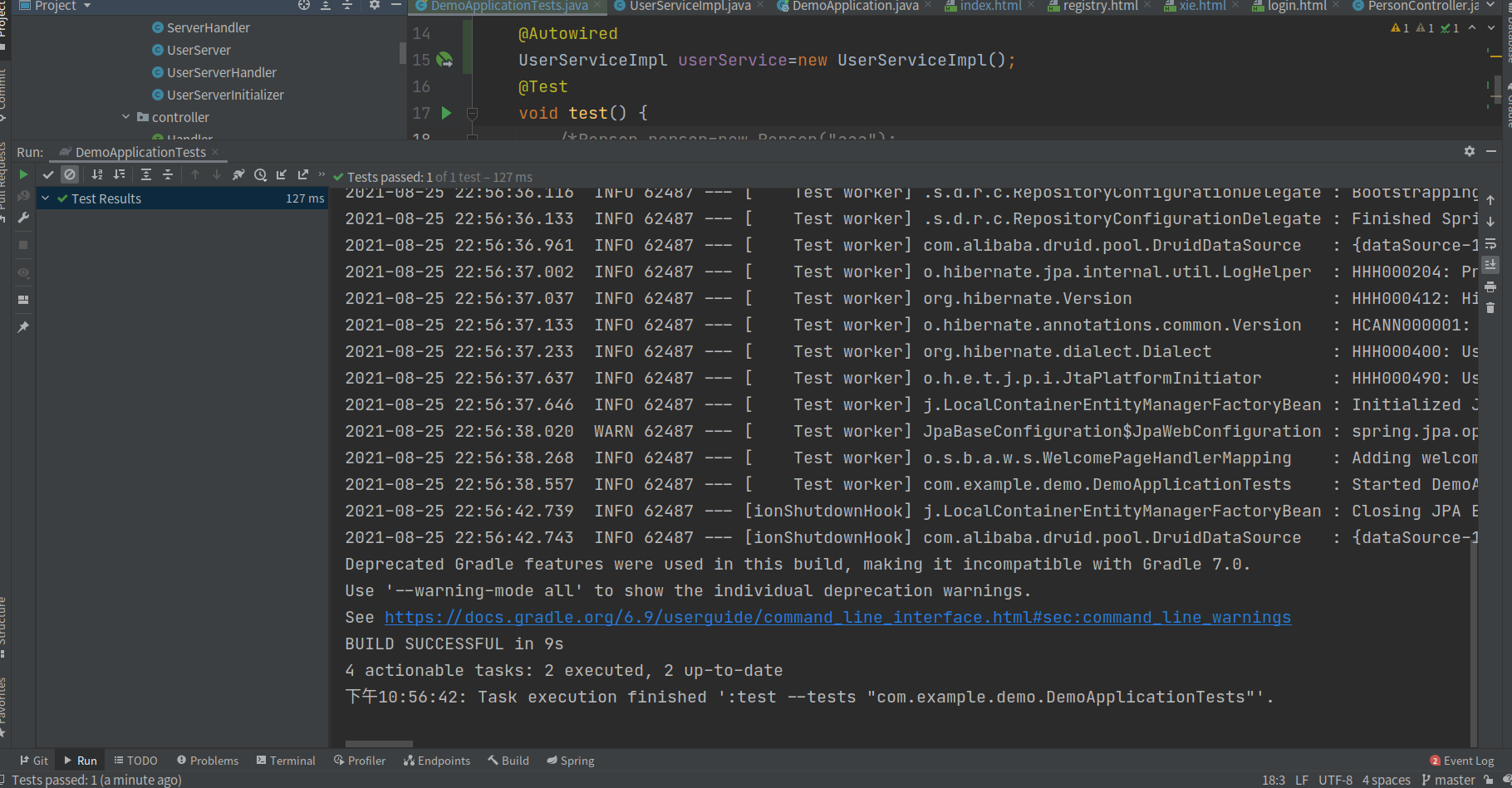
## 4.软件设计

本平台通过idea软件java语言编写，用户先进入主页面，若未注册先注册然后登陆，进入聊天页面若用户好友上线，显示用户好友名字，然后用户可以与好友聊天，若用户好久未和好友聊天，改平台可以提醒用户维护人际关系，用户可以进入进度条页面，该页面可以显示和好友关系程度。

## 系统测试及结果

### 系统测试：

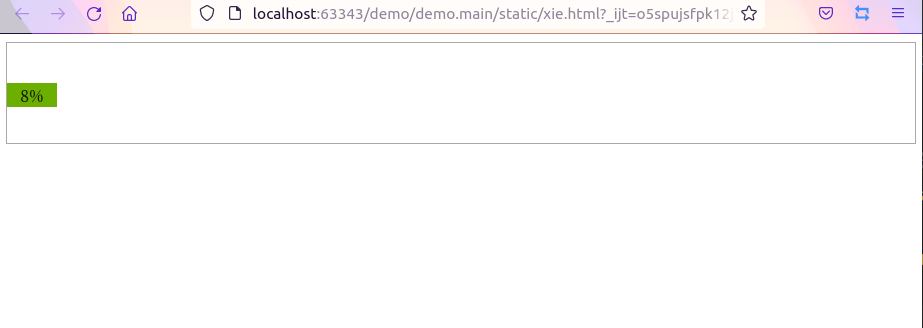
### 测试成功



## **测试结果：**

## 

## **[C7JCA4SSKQXNP1Y]E{8(WT**



## 6.程序源代码

github代码：https://github.com/lwy123-wq/personal-demo

@Component

public class ChatServer {

private EventLoopGroup bossGroup;

private EventLoopGroup workGroup;

private void run() throws Exception {

// log.info("开始启动聊天服务器");

bossGroup = new NioEventLoopGroup(1);

workGroup = new NioEventLoopGroup();

try {

ServerBootstrap serverBootstrap = new ServerBootstrap();

serverBootstrap.group(bossGroup, workGroup)

.channel(NioServerSocketChannel.class)

.childHandler(new ChatServerInitializer());

//启动服务器

ChannelFuture channelFuture = serverBootstrap.bind(8848).sync();

//log.info("开始启动聊天服务器结束");

channelFuture.channel().closeFuture().sync();

} finally {

bossGroup.shutdownGracefully();

workGroup.shutdownGracefully();

}

}

/\*\*

\* 初始化服务器

\*/

@PostConstruct()

public void init() {

new Thread(() -> {

try {

run();

} catch (Exception e) {

e.printStackTrace();

}

}).start();

}

@PreDestroy

public void destroy() throws InterruptedException {

if (bossGroup != null) {

bossGroup.shutdownGracefully().sync();

}

if (workGroup != null) {

workGroup.shutdownGracefully().sync();

}

}

}

public class ServerHandler extends SimpleChannelInboundHandler<TextWebSocketFrame> {

public static ChannelGroup channelGroup = new DefaultChannelGroup(GlobalEventExecutor.INSTANCE);

public static AtomicInteger online = new AtomicInteger();

Message message=null;

public static int i=0;

@Override

protected void channelRead0(ChannelHandlerContext ctx, TextWebSocketFrame msg) throws Exception {

message = new Gson().fromJson(msg.text(), Message.class);

XieHandler xieHandler=new XieHandler();

xieHandler.handler(message);

if(message==null){

sendMessageByChannel(ctx.channel(), new Message(message.getName(),"消息错误", System.currentTimeMillis()));

}else {

ChannelId channelId = ChatConfig.map.get(message.getName());

System.out.println(message.getName());

if(message.getName()==null){

System.out.println(message.getName()+"xxxxxxxxxxxxx");

sendMessageByChannel(ctx.channel(),new Message(message.getName(),"对方已下线", System.currentTimeMillis()));

}else {

sendMessageByChannel(channelGroup.find(channelId),message);

sendMessageByChannel(channelGroup.find(ctx.channel().id()),message);

}

}

}

@Override

public void handlerAdded(ChannelHandlerContext ctx) throws Exception {

i++;

channelGroup.add(ctx.channel());

//ChatConfig..put(ctx.channel().name().asShortText(), ctx.channel().id());

online.set(channelGroup.size());

sendMessageForAll(new Message(ChatConfig.name.get(i),ChatConfig.name.get(i), System.currentTimeMillis()));

System.out.println(ctx.channel().remoteAddress() + "上线！" + "--->" + ChatConfig.name.get(i));

ChatConfig.map.put(ChatConfig.name.get(i),ctx.channel().id());

System.out.println(i);

}

@Override

public void handlerRemoved(ChannelHandlerContext ctx) throws Exception {

channelGroup.remove(ctx.channel());

ChatConfig.name.remove(i);

online.set(channelGroup.size());

sendMessageForAll(new Message(ChatConfig.name.get(i), ChatConfig.name.get(i)+"下线！", System.currentTimeMillis()));

System.out.println(ctx.channel().remoteAddress() + "下线！");

}

@Override

public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {

if(evt instanceof IdleStateEvent){

IdleStateEvent event=(IdleStateEvent) evt;

if(event.state()== IdleState.ALL\_IDLE){

System.out.println(ctx.channel().remoteAddress()+"读写空闲\n");

sendMessageByChannel(ctx.channel(),new Message("","您已好久未和您的好友聊天了",System.currentTimeMillis()));

}

}

}

@Override

public void exceptionCaught(ChannelHandlerContext ctx, Throwable cause) throws Exception {

cause.printStackTrace();

}

private void sendMessageByChannel(Channel channel, Message message) {

channel.writeAndFlush(new TextWebSocketFrame(new Gson().toJson(message)));

}

private void sendMessageForAll(Message message) {

for (Channel channel : channelGroup) {

channel.writeAndFlush(new TextWebSocketFrame(new Gson().toJson(message)));

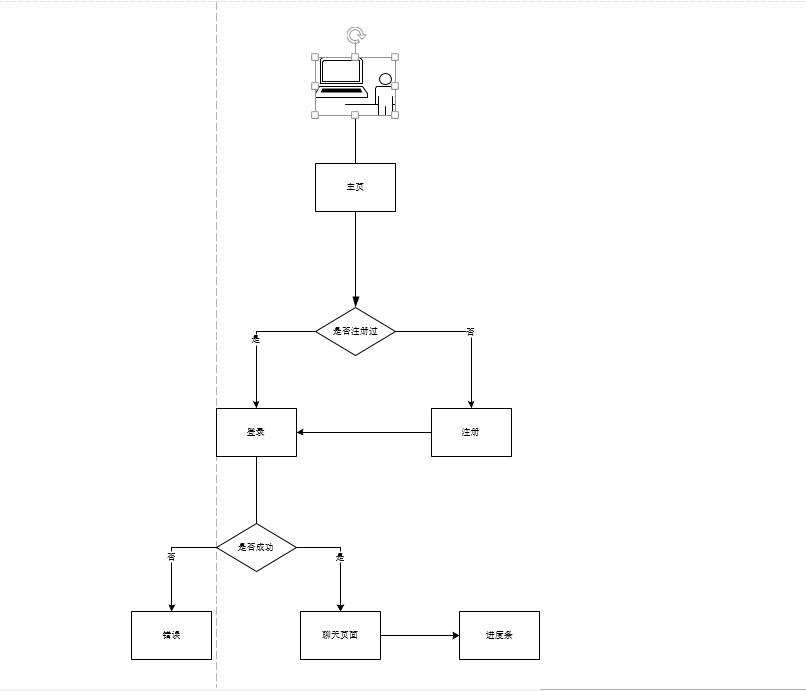
}

}

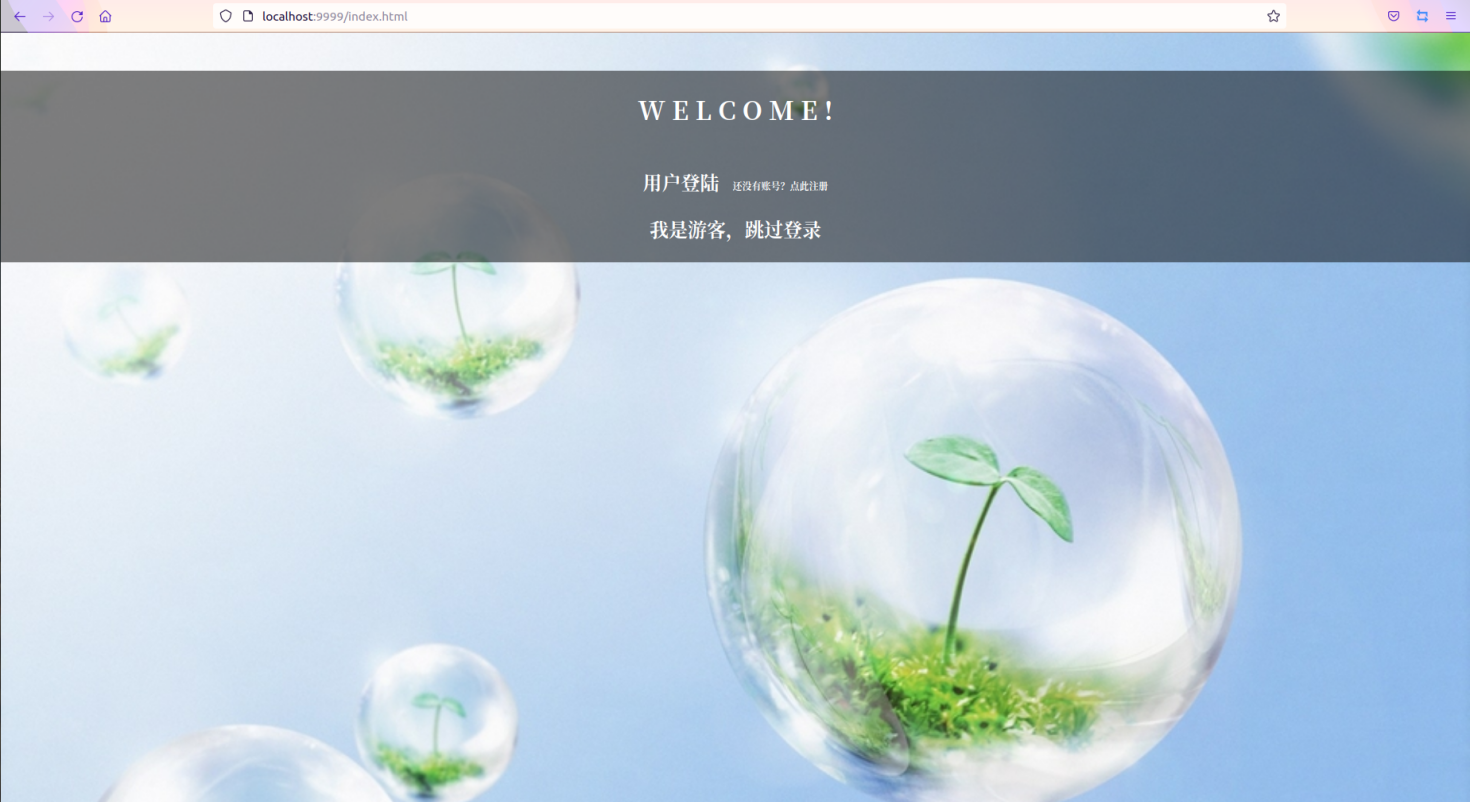
}

1. **照片及说明**

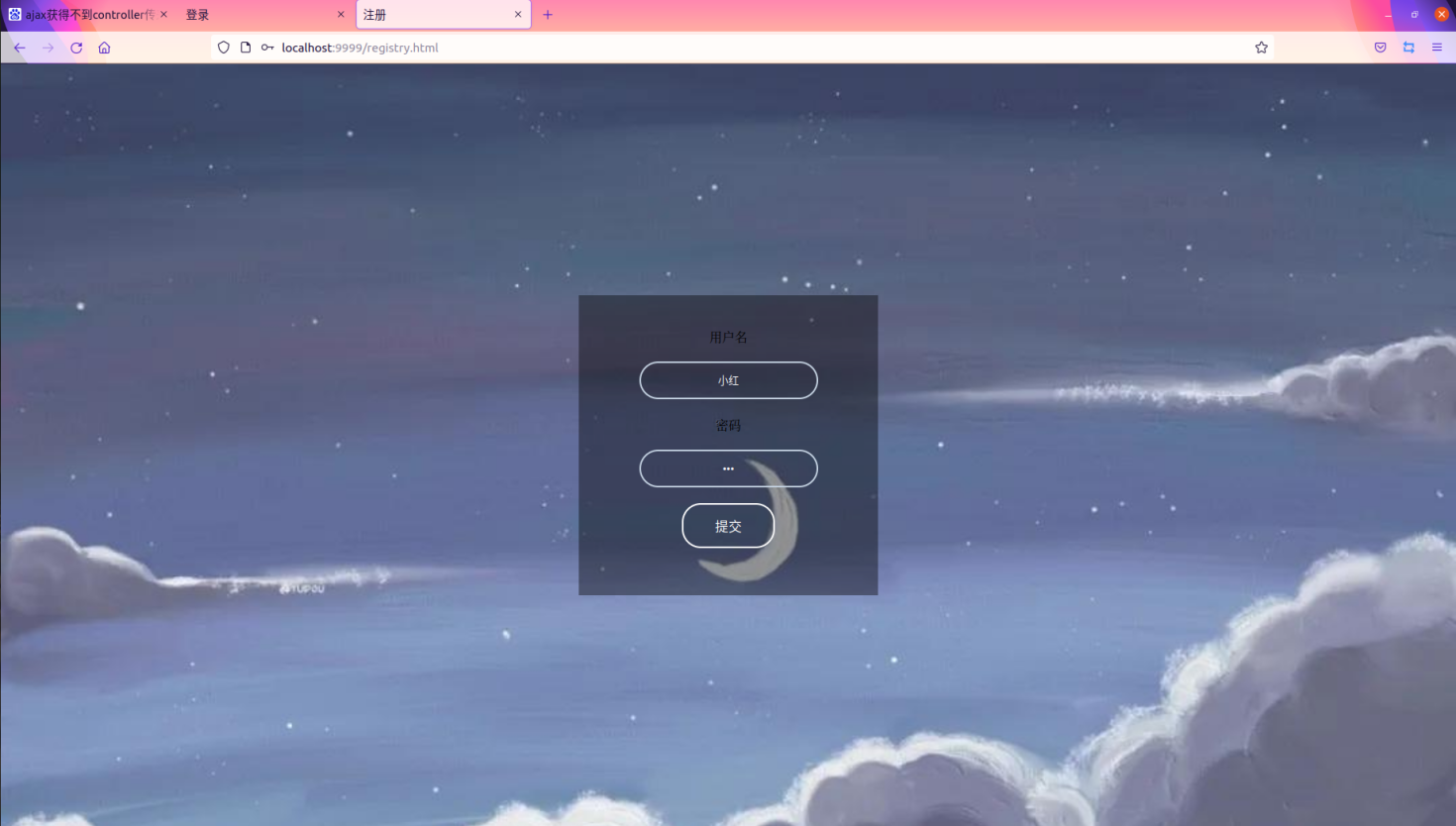
**流程图：**

****

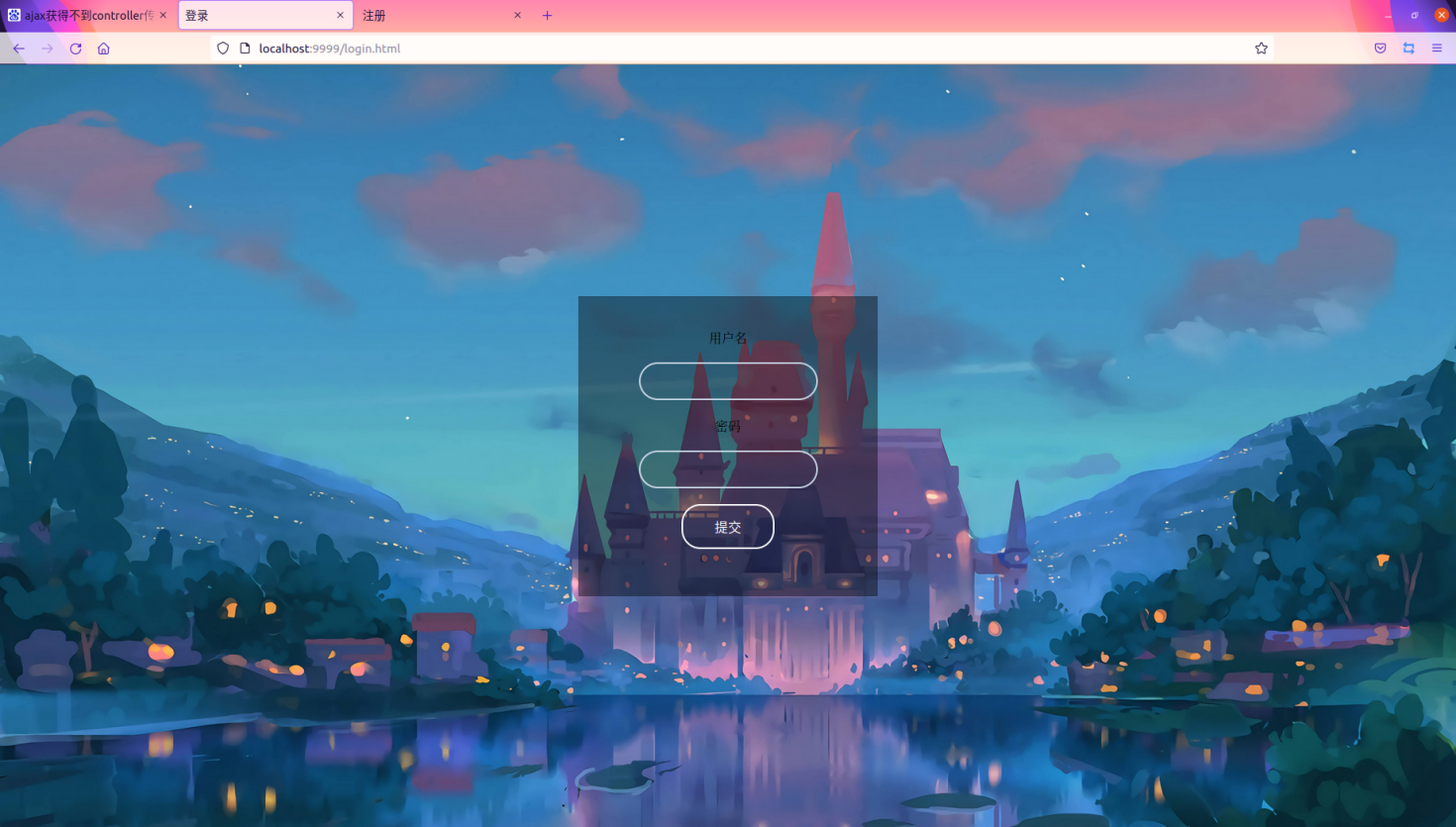
**该平台主页：**



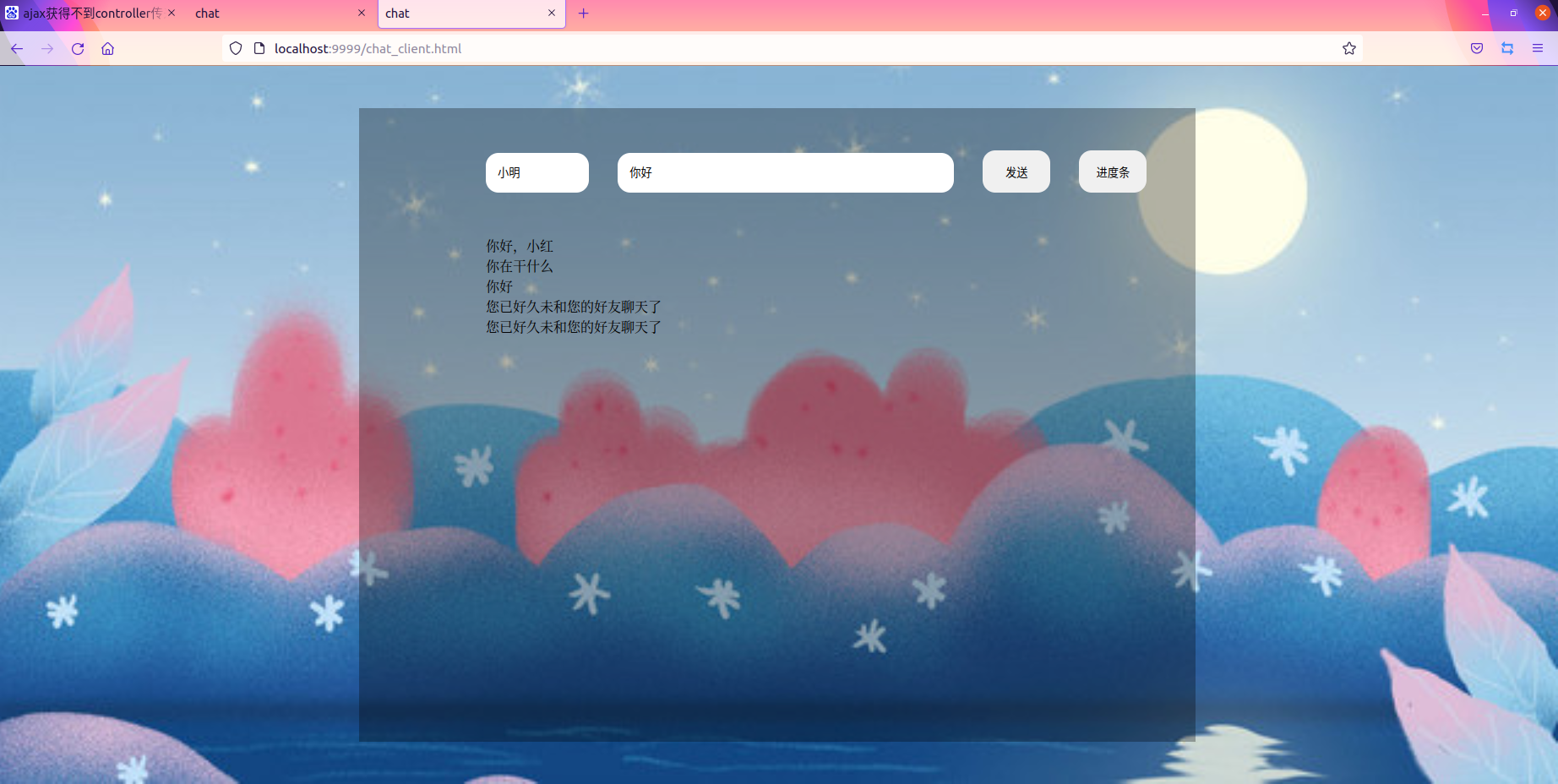
用户注册：



用户登录：



用户聊天界面：

****