**结题报告**

第四组

聂世豪 卫唯 余林峰 蒲思乔

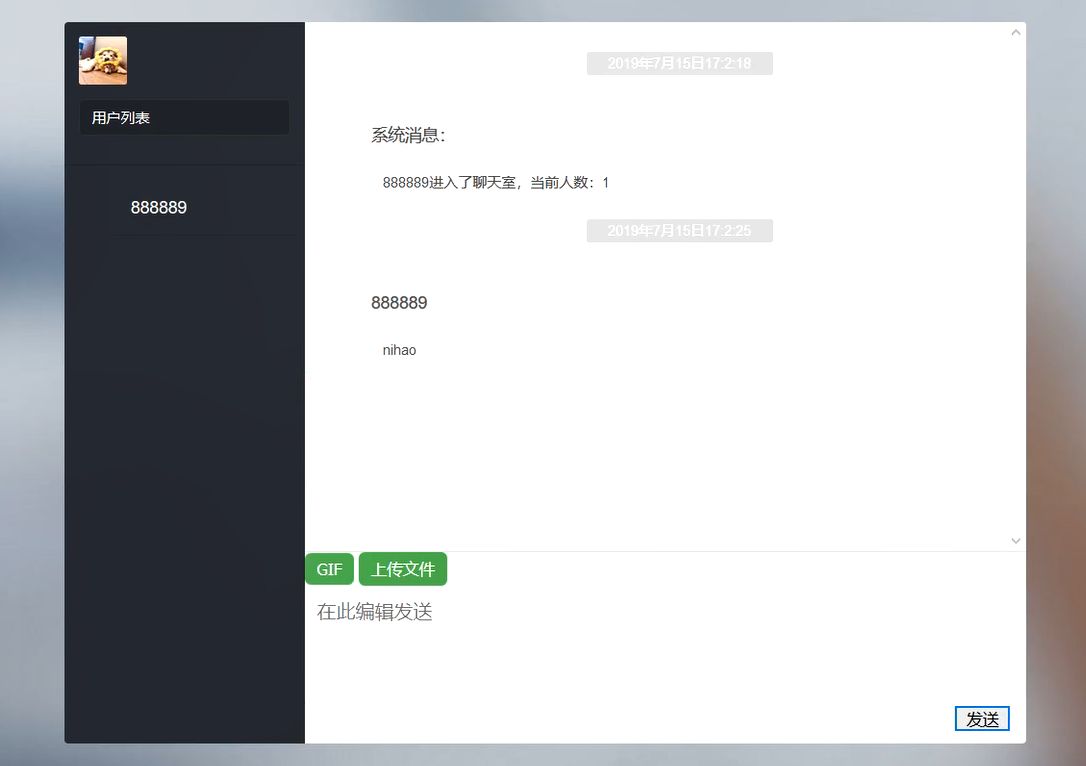
1. **软件使用手册**
2. **注册与登录**

已有账号的用户，可以输入账号及密码，通过验证后即可登录，点击确定按钮即可进入聊天界面。



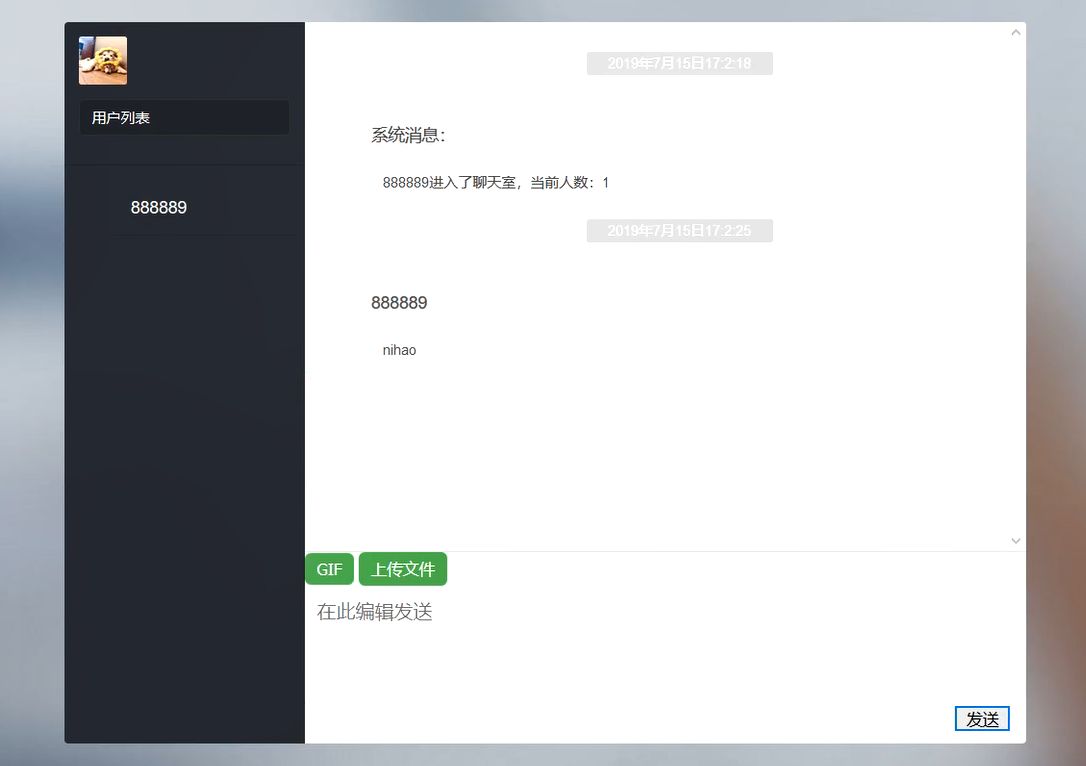
1. **群聊功能**

用户可以在消息框中输入会话内容，点击发送按钮发送。发送信息显示用户发送时间及发送者用户名。



1. **上传文件与图片**

用户点击下方“上传文件”按钮，即可上传本地文件，上传后发出消息，点击消息框中的下载地址即可下载并使用该文件。





用户可点击下方“GIF”标识按钮，选择本地的图片并发送。

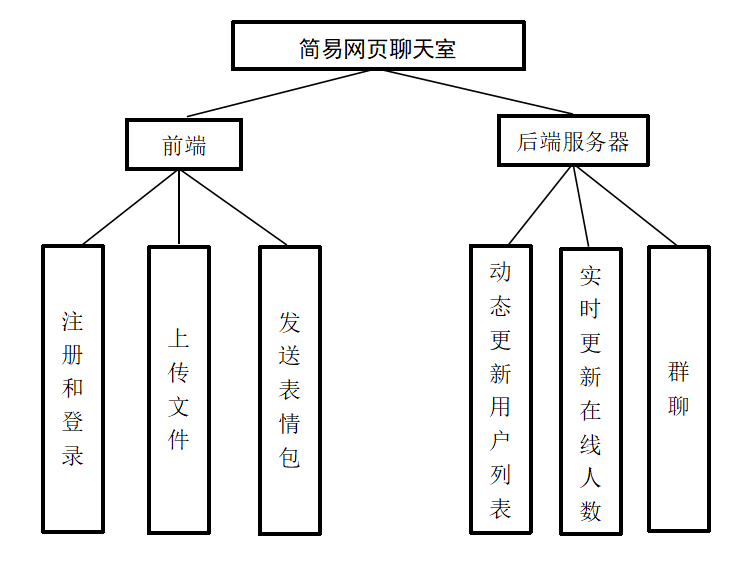
1. **动态更新用户列表并广播**





1. **软件开发文档**

聊天室分为前台和后台两个部分。前台分为用户注册和登录，群聊，上传文件，发送表情包四大模块。后台分为实时更新在线人数、动态更新用户列表，实时确定消息发送时间三大模块。



操作系统：Windows

开发平台：

\*后端开发工具：idea

\*前端开发工具：Hbuilder X & Visual Studio Code

\*数据库：Navicat for MySQL

浏览器:Chrome,IE

编程语言：JavaScript语言，HTML语言，CSS语言，Java语言等

1. **登录功能**

当用户准备进入聊天室，需输入账号和密码进行验证，验证通过时方可进入聊天界面。

核心代码：

@PostMapping("/login")

public Map login(@RequestParam(value = "username") String username,

@RequestParam(value = "password") String password) {

Map<String, String> map = new HashMap<>();

if (repository.findByUsernameAndPassword(username, password) != null) {

map.put("result", "success");

} else {

map.put("result", "fail");

}

return map;

}

}

1. **（2）群聊功能**

所有用户可以在同一聊天室进行聊天，每个用户可以同步显示其他在线用户发送的消息和发送者的用户名。

核心代码：

@ServerEndpoint("/websocket")

//注册到spring容器中

@Component

public class Websocket {

//当前连接

private Session session;

private String username;

//线程安全的无序集合,用一个集合保存当前所有连接

private static CopyOnWriteArraySet<Websocket> websockets = new CopyOnWriteArraySet<>();

//在线列表,列表必须绑定用户名和会话，前端点击就能进入私聊会话

//路由表,sessionid和用户名绑定，路由表即时会话列表

private static Map<String, String> routetab = new HashMap<>();

//对指定session发送消息，即对指定用户发送消息

public void send(String content,Session session) {

session.getAsyncRemote().sendText(content);

}

//广播

public void broadcast(String message){

for (Websocket websocket : websockets) {

websocket.session.getAsyncRemote().sendText(message);

}

}

@OnOpen

public void onOpen(Session session){

this.session=session;

websockets.add(this);

broadcast("大家好");

}

// @OnOpen

// public void onOpen(Session session, @PathParam("username") String username) {

// this.session = session;

// this.username = username;

// websockets.add(this);

// routetab.put(session.getId(),username);

// String content = username + "进入了聊天室，当前人数："+ websockets.size();

// broadcast(content);

// //broadcast(routetab.toString());

// }

1. **上传文件功能**

用户在聊天过程中 可以选择本地文件进行上传，后端会返回一个该文件的下载地址，用户点击该下载地址即可下载查看该文件。

核心代码：

function fileBtn() {

document.getElementById('file').click()

}

function upload(file) {

var img = document.getElementById('img')

var formData = new FormData()

var temp = file.files[0]

if (temp) {

formData.append('file', temp)

$.ajax({

url: "http://" + window.location.href.split('/')[2] + "/upload",

type: "POST",

data: formData,

processData: false, // 告诉jQuery不要去处理发送的数据

contentType: false,

success: function(result) {

//前端对result进行处理

//websocket.send("<li><span class=\"get-time\">" + get\_time() + "</span><br /><span class=\"get-message\">上传了文件:</span><a href=" + result.url + " download>" + result.name + '</a></li > ');

websocket.send("<li><span class=\"text\">上传了文件:</span><a href=" + result.url + " download class=\"text\">" + result.name + '</a></li > ')

// 不加download会报错，点击下载立即退出聊天室

}

})

}

}

1. **软件设计说明文档**
2. **需求分析**

目标：旨在开发一个新型便捷交友聊天平台，为大家网络交流提供便利。

功能要求：为用户提供登录注册功能以及群聊功能，并且在聊天中可以发送图片及文件。

性能要求：用户能及时获取其他用户聊天信息并且对客户端的新加入用户进行实时同步更新。

系统维护要求：整个系统管理员可以访问，可以通过访问数据库维护信息等。

需求规定：聊天室用户必须在网页版聊天室登录

1. **设计原理**

本次作业我们设计了一个简易网页聊天室，本次网页设计使用了idea、Visual studio Code、mySQL等软件，Java、JavaScript、html，css等语言，实现了用户登录、注册功能，群聊，上传文件，发送表情包、实时更新在线人数、动态更新用户列表，实时确定消息发送时间等功能。

1. **最终效果**

在PPT的视频中展示

1. **优缺点分析**

实现了最初简化社交软件的预期，有良好的交互界面，但由于组员对语言的学习有限，无法进行私聊或者添加好友的操作。

1. **小组分工**

聂世豪（组长）：负责前后端连接部分，答辩

卫唯：负责前端代码编写（登陆及聊天页面设计）、总结报告文档编写

余林峰：负责后端代码编写

蒲思乔：负责软件安装、环境配置、代码测试、ppt制作。