湖南科技大学计算机科学与工程学院

综合实践实训 课程设计报告

**专业班级：** 计算机科学与技术二班

**姓 名：** 贺政涛

**学 号：** 2105010212

**指导教师：** 朱海燕

**时 间**： 2023年5月22号

**地 点**： 逸夫楼418

|  |
| --- |
| **指导教师评语：**  **成绩： 等级：**  **签名：**  **年 月 日** |

一、实验题目

**仿QQWeb即时聊天系统**

二、实验目的

* 实现Web的点对点即时的文本消息聊天功能。
* 实现Web的表情的发送、接收和显示功能。
* 实现Web的图片的发送、接收和显示功能。
* 实现本地消息的存储，在离线的时候也能加载和查看历史消息；
* 要求使用WebSocket;

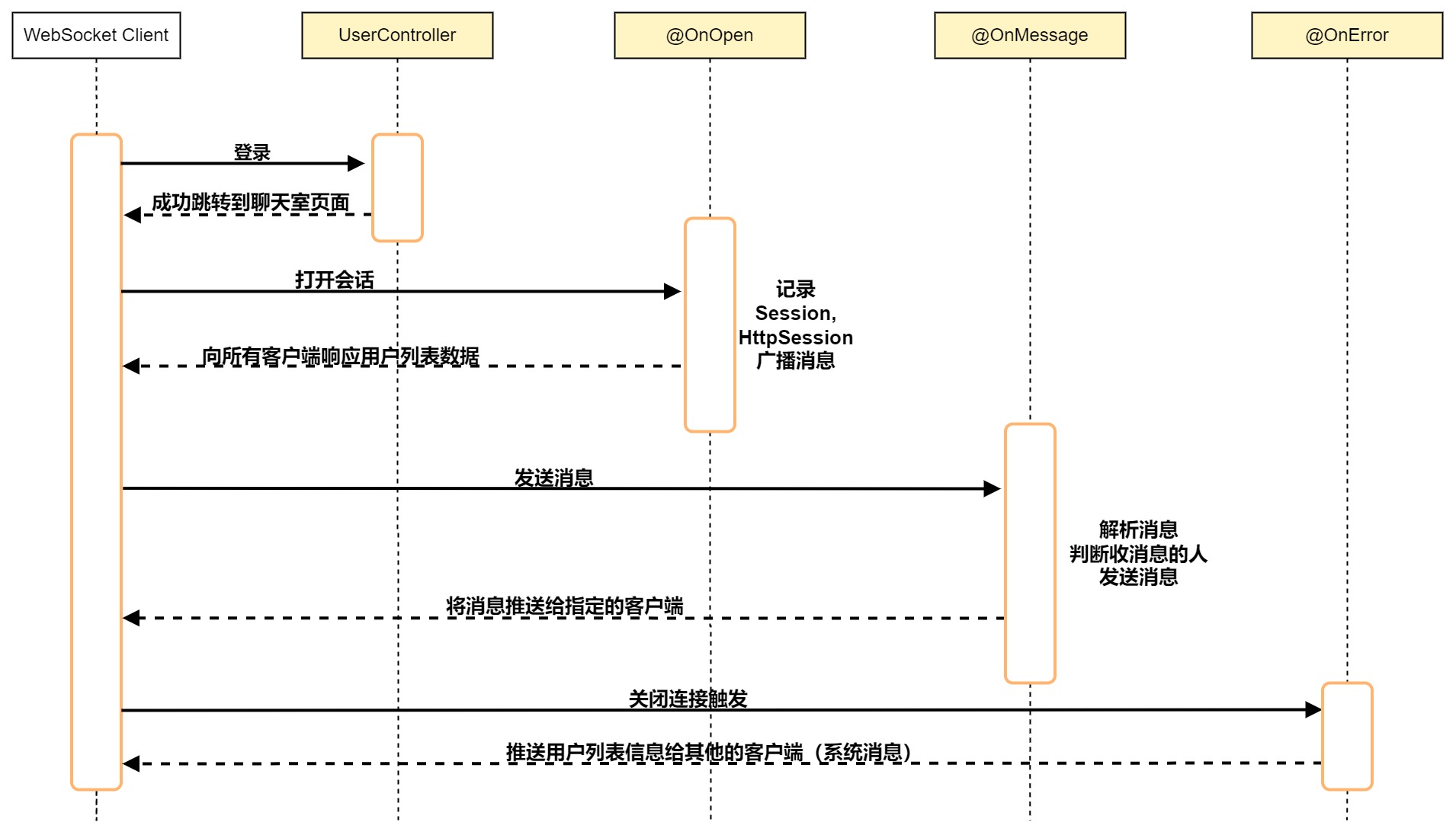
三、总体设计（含背景知识或基本原理与算法、或模块介绍、设

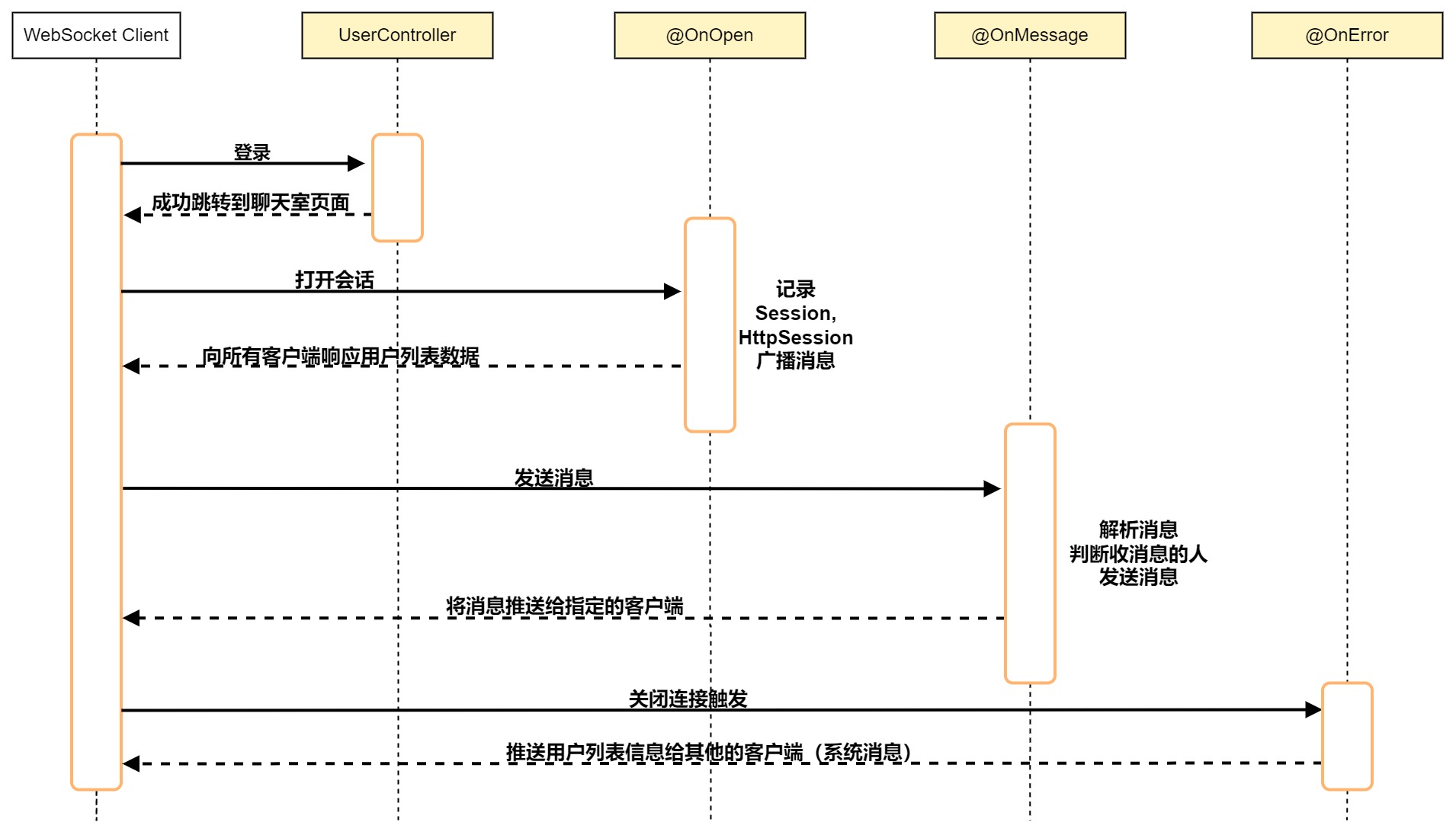
计步骤等）

背景及条件：该项目需要使用到的语言主要有kotlin，html，js，css等。采用的框架为springboot+vue。通过http来完成前后端数据的互相传输。采用json作为传输工具。通过websocket来实现实时通信功能。

基本原理：WebSocket是一个在单TCP连接上进行全双工通信的协议。通过websocket实现点对点即时的消息聊天功能。

本地存储则采用localStorage来完成历史记录的存储。

主要模块设计：



**设计步骤：**

**先完成最基础的文本消息聊天功能**

**然后完善前端框架，增加表情包，图片发送功能。**

**最后实现本地存储，将消息记录保存到本地。可以查阅**

**登录模块：**

例图：

接口规范：

{username:””}

前后端只需要传递用户名即可。

**聊天模块：**

例图：

**会话模块：**

连接时：

会与后端建立连接，并获取所有在线的人数

接口规范：

{isSystem:true,users:[{userId:””,userName:””,avatar:””,status:””}]}

**发送消息:**

接口规范:

{from:””,to:””,message:””}

**接收消息：**

接口规范：

{isSystem:false,message:{from:””,to:””,message:””}}

以上是完成了基本的文本消息发送接收功能。

至此已经大致完成前端页面的设计。

**表情模块：**

例图：





**表情管理和发送：**

为了减少服务器的负担，因此选择在前端存储所有的表情图片，然后为每张表情设置一个nano编码作为key值，或者直接用文件名作为key值。用以对应每一个表情。然后发送过程选择将key发送到对方。（原先设计为该方法）。

后来根据实际情况确定，表情包作为一个图片大小只有10几kb，很小，可以直接通过服务器在前后端传输，并不会带来不好的体验。（暂不考虑优化问题）

**大改动：**因为设计时，要求能够让表情包可以和文字内容共存。所以之前所有关于聊天框及其内容的设计出现了很大的缺陷。不能使用input和textarea，而转变为使用可编辑的div标签。在样式，结构已经调用的方法上出现了很大的改动。最后实现的成果即表情可以与文件共存，并且像后端发送消息时，会直接将整个img标签发送过去。

Bug：

不能解决在文本中粘贴图片的问题。

产生了关于ctrl+z异常的错误，可能需要重新设置撤销重做的栈等关系。

**图片模块：**

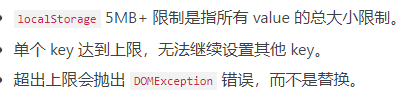
拟定为通过上传按钮实现图片的上传，然后带着发送者和接收者的信息发送到后端，再发送回去。

已经完成了图片模块的实现功能。

例图：



主要设计思路：本打算通过post发送到后端然后再通过websocket发送回前端，但是考虑到图片并不需要进行后端存储，只是需要完成传输功能。因此修改了websocket的最大缓存区的机制，让他最大可以接收10MB以内的图片。前端实现预览只需要通过FileReader即可完成。

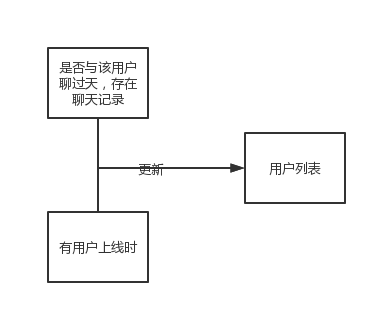
至此，所有的主要功能基本完成，包括文本消息，表情，图片的发送，接下来需要实现的只有本地存储或者再新增其他功能。而作为一个web在线应用程序，他实现本地存储时，几乎无法做到完整的存储图片的功能。

因此实现本地存储的时候，会忽视掉图片的后续保存，而只保存表情和文本信息。

**本地存储模块：**

考虑到要求能够离线都查看到本地聊天记录，而实际上，在浏览器中，若直接断网的话，将会无法查看到任何内容。而且本项目设计时并没有考虑到注册，登录和存储用户的功能。因此他的本地存储和理想上的会有差别。

大致实现过程:



至此本地存储已经完全完成

项目基本完成，以上是主要设计。

项目采用了前后端分离的方法，前端通过vue来实现，后端通过kotlin+springboot完成。

1. 详细设计（含主要的数据结构、程序流程图、关键代码等）

后端两个模块：

登录模块：

五、实验结果与分析

六、小结与心得体会