遼寧大學

毕业论文（设计）

题　　目：基于 SpringBoot 的线上聊天网站的设计与实现

学　　院：信息学院

专　　业：计算机科学与技术

姓　　名：常笑男

指导教师：薛军

完成日期：2022/5/20

摘　要

人与人之间的交流是每个人发展必不可少的要素，目前由于新冠疫情影响，线上聊天工作变的很普遍，聊天软件也应该要有新的功能和特色。当前国内最常用的聊天软件就是 QQ 和微信。然而在手机上使用这写软件，时间一长就会导致 App 运行卡顿，缓存数据过多，占用的存储空间越来越大，用户的体验感也逐渐下滑。

针对现有的聊天软件在手机移动端出现的各种各样的问题，设计一个不占用过多用户存储空间和缓存空间的网站应用，使之可以更流畅的运行在各种手机移动端，变的越发迫切。开发一个 web 软件通常需要的时间是很多的，但是，随着技术的发展，现在开发 web 软件变得很快速。通过利用 SpringBoot 的开发框架，我们可以快速的开发一个网站的后台服务。前端利用 Vue 框架的特性，可以将前端的页面视图、渲染和逻辑模块快速的完成，与此同时，也需要使用原生 JavaScript 来保证前端页面在执行复杂运算时也有较为良好的性能。合理的利用 CSS3、HTML5 的优点，快速的开发一个能让用户有良好体验感，有着更强大的性能和功能的应用。

这种开发模式对于开发一个功能较为全面的网站来说，性价比相当的高。也可以通过这种快速开发模式来解决国内互联网的 web 组件不够丰富的现状，使 web 和移动端结合的更加紧密，使大众的在手机移动端的视野从手机 App 扩展到网站，而不是只局限于手机应用市场当中的应用。

关键词：SpringBoot;Vue;CSS;HTML;JavaScript

Abstract

Communication between people is an essential factor for everyone's development. At present, because of the influence of COVID-19, online chatting has become very common, and chat software should also have new functions and features. At present, the most commonly used chat software in China is QQ and wechat. However, using this writing software on the mobile phone will lead to app running stuck over time, too much cached data, taking up more and more storage space, and the user's sense of experience will gradually decline.

In view of the various problems of the existing chat software on the mobile terminal, it is more and more urgent to design a website application that does not occupy too much user storage space and cache space, so that it can run more smoothly on all kinds of mobile terminals. Developing a web software usually takes a lot of time, but with the development of technology, developing web software has become very fast. By using the development framework of springboot, we can quickly develop the background service of a website. Using the characteristics of Vue framework, the front-end can quickly complete the page view, rendering and logic modules of the front-end. At the same time, it also needs to use native JavaScript to ensure that the front-end page also has good performance when performing complex operations. Make rational use of the advantages of CSS3 and HTML5 to quickly develop an application that can make users have a good sense of experience and have more powerful performance and functions.

This development mode is quite cost-effective for developing a website with more comprehensive functions. This rapid development mode can also be used to solve the current situation that the web components of the domestic Internet are not rich enough, make the combination of the web and the mobile terminal closer, and expand the public's vision of the mobile terminal from the mobile app to the website, rather than limited to the applications in the mobile application market.

Key words: SpringBoot;Vue;CSS;HTML;JavaScript

目　录

序　言 1

第 1 章　系统概述 2

1.1 系统介绍 2

1.2　开发环境 2

1.3　开发工具及技术说明 3

1.4　系统框架 4

第 2 章　系统分析 5

2.1　功能性需求 5

2.2　非功能性需求 6

第 3 章　系统设计 8

3.1　模块设计 8

3.2　数据库设计 9

3.3　接口设计 13

第 4 章　系统实现 14

4.1　后端实现 14

4.1.1　模式 14

4.1.2　架构设计 14

4.1.3　关键技术设计 15

4.2　前端实现 17

4.2.1　模式 17

4.2.2　关键技术设计 19

4.3　项目实现 20

第 5 章　系统测试 24

5.1　功能测试 24

5.2　安全性测试 24

5.3　访问测试 25

5.4　异常测试 25

第 6 章　总结与建议 27

6.1 总结 27
6.2 建议 27
参考文献 28
致　谢 29

序　言

在人类出现的时候，人类传递消息是通过口口相传；在发明了文字后，人们传递消息可以通过文字；在中国发明了造纸术后，人们可以通过信件传递消息；在人类发明了电报机后，人们传递信息的距离大幅度增加；在 1969 年 10 月以后，第一封电子邮件面世，人类传递信息的途径又多了一个；1996 年，四名以色列人发明了 IM 的鼻祖——ICQ“坏小子”，至此人类可以实现——即便两人相隔万里也能相互对话，这一此前相当天方夜谭一样的想法；1998 年，QQ 这一在中国风靡一时的软件诞生，人与人之间的对话变的更为简单，人与人的社交变的更为快速、简洁、高效……

随着互联网时代的到来，在新冠疫情之下，人们越来越多的办公转到了线上——线上办公。人与人的社交距离在新冠疫情之下被拉的比正常情况下更远，人们大部分时间都是通过社交媒体进行沟通，他们可以语音聊天、视频通话，也可以通过文字、图片、表情等方式进行沟通。而实现这种功能的软件有很多，国内的比如说 QQ、微信等软件。但是随着使用时间的延长，应用所存储的内容越来越多，占用的空间越来越大。这种情况对于一部手机来说会使手机在运行这个软件的时候运算更多的数据，从而会使程序运行卡顿，用户的体验感就会变得很差，有时还会导致出现一些错误操作。所以做一款新的聊天软件来解决这种问题变的很有必要。

但是软件大部分都是需要安装的，也就会出现上面所述的情况。所以要做出一个不需要在设备上进行安装的应用。由此，设计一个在线聊天网站成为了符合各种要求，并且还能解决由使用时间变长所导致的运行卡顿的问题，只要浏览器能够流畅运行，那么这个在线聊天网站就能流畅的运行。所以，问题就变成了设计一个怎样的 web 应用来实现一个线上聊天的功能，并且使它能够实现像已经出现的那些社交软件一样的功能。这个新设计的聊天软件界面也应该如它们一样符合大家往常的使用习惯，使用户可以快速的适应一个新的软件界面。

第 1 章 系统概述

1.1 系统介绍

随着新冠疫情的到来，线下办公部分转为线上办公，对社交软件提出了新的要求。目前国内市场上使用最为广泛的就是 QQ 和微信，但是随着这些软件运行时间的加长，软件的存储所占用的空间也随之增加。而用户却不会删除平时缓存的数据，就造成了一种现象：用户下载软件，使用软件，手机运行变慢，买新手机。就这种问题设计了在线聊天网站来解决。

本项目使用的是敏捷过程开发模型，使用 B/S（browser/server）的开发方法。

实现了发送文件、聊天信息（表情）、用户注册登录功能（含有人脸识别）、自定义用户信息、用户自定义联系人分组、用户搜索功能和好友之间的增加、删除、修改等功能。

1.2 开发环境

为了进一步精简项目开发过程，使项目开发速度保持一个相对较快的速度，所以采用前后端独立开发的模式。考虑到项目开发所需要用到的软件，因为电脑配置原因，所以采用了体积较小的 VScode 来开发前端与后端，使用 Maven 来控制所需依赖。项目开发完毕后，为了测试应用的兼容性、稳定性、安全性和流畅性。分别在 Edge 和 Chrome 浏览器上进行测试。项目环境见下表（表 1.1）。

表 1.1 环境

环境类型	软件	插件
前端环境	VScode	Live Server(v5.7.5) HTML Snippets(v0.2.1) HTML CSS Support(v1.11.0) JavaScript (ES6) code snippets(v1.8.0)
后端环境	VScode	Spring Boot Extension Pack(v0.1.0) Extension Pack for Java(v0.22.3)
	Navicat Premium 15(MySQL v8)	无
测试环境	Edge、Chrome	无

1.3 开发工具及技术说明

（1）VS code，全称为Visual Studio Code，是微软2015年推出的一款免费开源的现代化轻量级代码编辑器，能够在windows、Linux、IOS等平台上运行，通过安装一些插件可以让这个编辑器摇身一变成为一个编译器。VS Code支持C++、Python、Java、C#、Go等多种语言，功能强大、插件丰富并且启动速度极快，比较适合电脑配置不高的开发者。

（2）Navicat 是一款图形化数据库管理及发展软件，支持多重连线到本地和远程数据库，设计符合从数据库管理员和程序员各种需求。

（3）MySQL 是一个关系型数据库，该数据库服务是一个完全托管的数据库服务，可使用世界上最受欢迎的开源数据库来部署云原生应用程序。原先因为成本低且支持快速开发，所以多用于互联网上的中小型网站，后随着技术的发展逐渐应用于更多大规模的网站。

（4）Java 是一种被广泛使用的计算机编程语言，拥有面向对象、泛型编程、跨平台等特性，被广泛应用在互联网网站后端开发。

（5）HTML5 是HTML最新的修订版本，由万维网联盟（W3C）完成标准的制定，新版本的标准重新规定了一些标准，目的是为了在移动端上支持多媒体，以及更容易地在网页中添加和处理多媒体。

（6）ES6(ECMAScript 6) 是JavaScript的一种新的语言标准，主要是为了解决ES5的先天不足，比如JavaScript里面没有类的概念。为了能够更好的实现应用的功能，其添加了许多的语法特性，并使代码实现更简洁

（7）SprintBoot2.0框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架致力于快速开发微服务[1]，该框架使用的配置全部都有默认配置，解决了传统的SSM下手动配置过于复杂的问题[2]，从而使开发人员不再需要定义繁琐的初始配置。其内嵌了Tomcat服务器，同时配合Maven可以将项目打包成jar或war包方便部署[3]，Maven可以用来快速且方便的管理项目所需要的依赖等一系列工作，大大的提高了项目开发效率。

（8）Vue是一个用于创建用户界面的开源JavaScript框架，当前比较流行[4]。其采用了Mvvm的设计模式，在本项目中主要是使用它将数据渲染时的响应式行为动态的更改界面上的数据。

（9）OpenCV 是一个跨平台的计算机视觉库，在本项目中用作人脸识别,准确度高、速度快。[5]

1.4 系统框架

图 1.1 系统框架图

第 2 章 系统分析

2.1 功能性需求

由于这是一个在线聊天网站，所以只需要有普通用户端界面。普通用户功能需求主要如下。

（1） 注册功能。对于没有注册或没有登录的用户是不允许进入聊天网站的主页面，防止未登录或注册的用户对其他用户的数据进行更改。注册分为邮件注册、手机号码注册和刷脸注册，为用户提供多种注册方式，给用户带来新鲜的体验感的同时，为用户提供安全的账号信息保障。并且如果注册失败要返回错误原因。

（2） 登录功能。如果没有登录或者长时间没有登录，网页会自动跳转到登录，用户需要进行一次登录。与注册功能的三种方式对应，登录也有三种对应的登录方式，为用户带来新鲜的科技感，如果登录失败返回登录失败的原因，如果登录成功则在给出提示后，直接跳转到网站的主页面，并且更新数据库中最新的登录时间。

（3） 添加好友功能。添加好友功能无疑对一个聊天网站来说是最重要的功能，这个功能要通过搜索功能搜索用户，再发出好友申请。若已发送，则返回对应提示信息。若用户同意申请，则返回对应提示信息。为用户提供友好的提示信息，从而帮助用户快速的知晓添加好友已被同意

（4） 查看好友信息功能。与一个好友聊天是最主要的功能，但是好友信息会为我们提供更多的信息，比如联系方式等。可以通过查看好友信息来查看所发送的文件，好友的头像。

（5） 删除、拉黑好友功能。与添加好友功能对应，也要有删除好友和拉黑好友功能，并且要有友好、明显的提示信息。好友之间要有双向拉黑机制（对方拉黑我之后，我也可以拉黑对方），只有双方都不是拉黑状态，才能聊天、发送文件等。

（6） 更改用户信息功能，用户可能对自己的头像不满意，或者自己上传的信息有错误。需要更改自己的信息。

（7） 发送文件功能，与世面上的大部分软件一样，要有文件上传和下载功能，并且能够长时间保留，为用户相互发送文件提供了便利。

（8） 在线查看功能，一些文件，如音频、视频、图片等文件是用户看一遍就基本不会下载的文件，所以提供在线查看功能是一个必要的选择，并且提供给用户下

载选项。

（9） 搜索功能，是搜索联系人功能的一种扩展，能够搜索用户与所有联系人相互发送的文件，可以快速的提供给用户想要查找的信息，并且提供下载选项，方便用户使用。

（10） 修改联系人昵称功能，即用户可以自定义对联系人的昵称（备注），给用户提供了一定的自定义能力。

（11） 用户分类功能，用户有时会对自己的联系人进行分组以便快速的锁定要查找的联系人，对用户的自定义能力进一步加强。为用户对好友的分组管理提供增加、删除、修改功能。每个用户可自定义多个用户分类。

（12） 压缩功能。对于一些用户上传的大文件，如图片文件，如果不进行压缩操作会对网页的反应速度造成很大的影响，所以要像保证用户的体验感就需要对大文件进行压缩

（13） 聊天功能，是网站中最重要的功能，完成两人之间的对话。

系统用例图见图 2.1。

2.2 非功能性需求

（1）稳定性，充分的考虑到网站运行的诸多异常（会导致服务器崩溃），抛出错误。

（2）简洁性，页面非必须的渲染不显示，返回的数据要简洁，代码要足够简单。

（3）美观性，用户界面需要让用户感觉眼前一亮，很舒适，为用户提供人性化提示帮助用户使用网站。

（4）高效性，在数据压缩、服务器运行缓存、前后端数据处理等方面优化性能，使得在线聊天网站能够对用户的使用提供高效的响应。

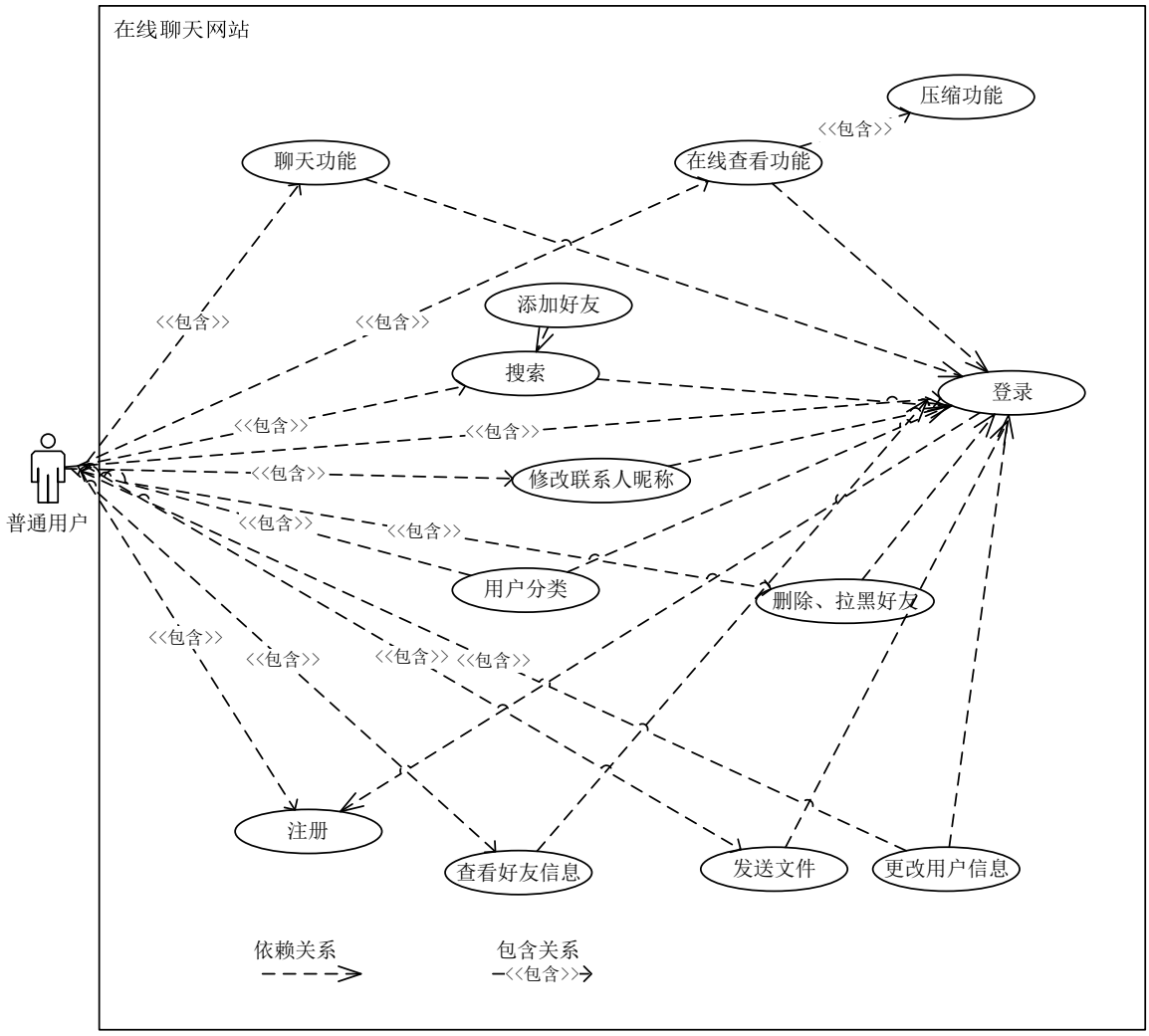
在线聊天网站
压缩功能
<<包含>>
聊天功能
在线查看功能
添加好友
<<包含>>
<<包含>>
搜索
登录
<<包含>>
<<包含>>
<<包含>>
修改联系人昵称
普通用户
<<包含>>
<<包含>>
用户分类
删除、拉黑好友
<<包含>>
<<包含>>
<<包含>>
<<包含>>
注册
查看好友信息
发送文件
更改用户信息
依赖关系
包含关系
<<包含>>

图 2.1 系统用例图

第 3 章 系统设计

3.1 模块设计

根据第 2 章的系统功能分析，得出系统模块的设计图如图 3.1 所示。

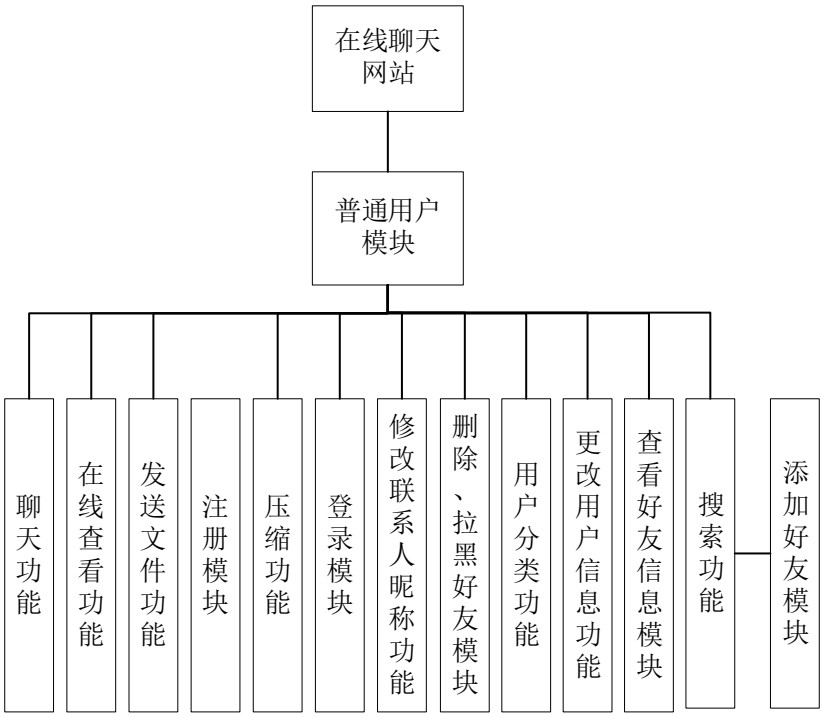

图 3.1 系统模块设计图

聊天功能是在线聊天网站中最重要的模块和功能，与发送文件功能一起几乎占据了聊天网站功能中的绝大部分。相比文件上传功能用户更在意的是完整的聊天功能（比如说发送表情），完整的聊天功能更能生动形象地表现用户此时此刻的心情状态。发送聊天信息有两种方式，一种是按 Enter 键发送，一种是按 ALT+Enter 的组合键发送聊天信息。之所以设计两种方式发送聊天信息，是因为有时用户需要在发送的聊天信息中发送换行符号。用户可以通过设置中的通用设置进行更改（也可以更改如字体大小的通用设置），之后网站会把用户的通用设置的信息存储到数据库中，方便用户下一次使用。网站中的大部分功能模块与聊天模块比较相像。下图是聊天模块的流程图。

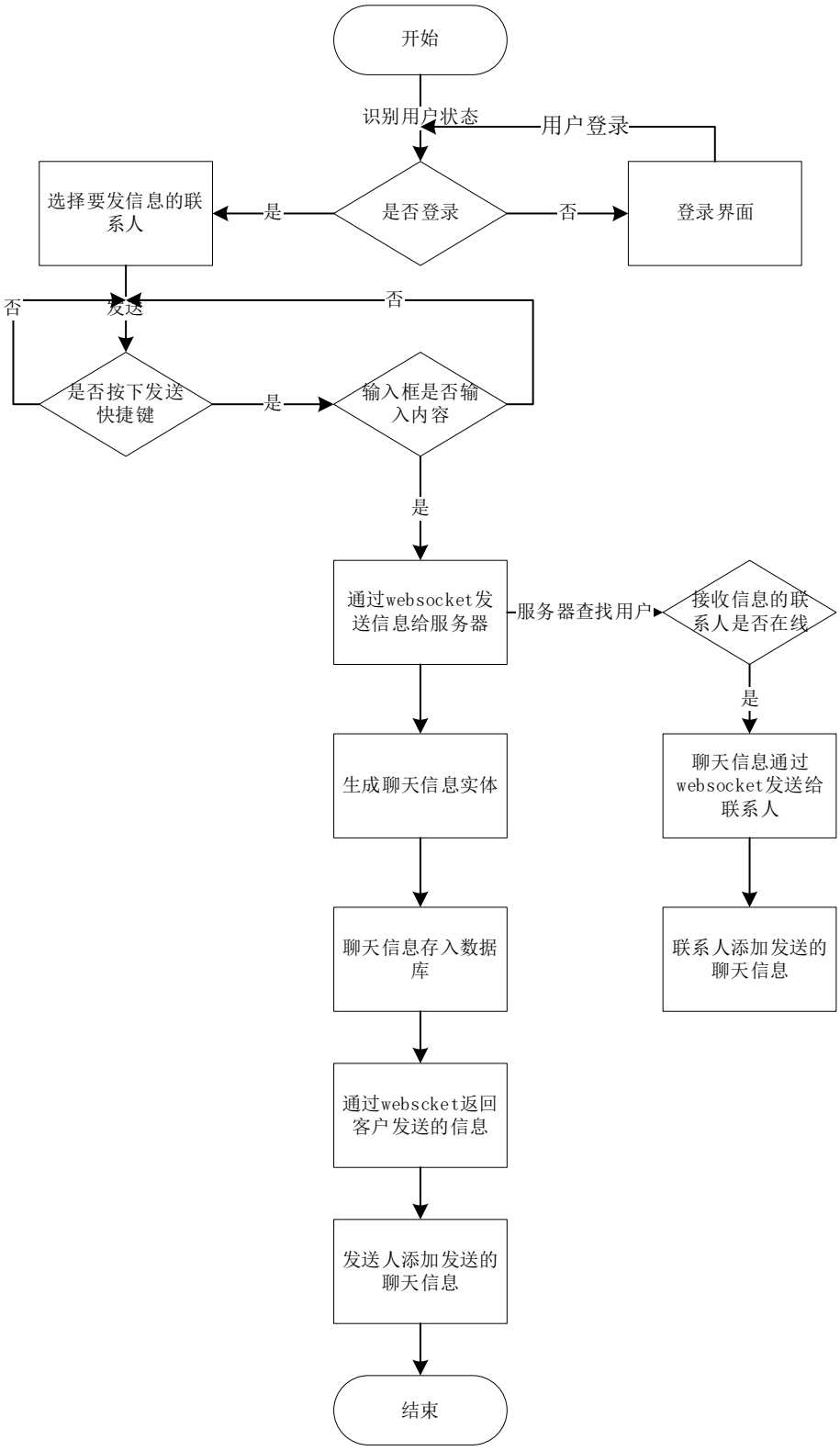

图 3.2 聊天模块流程图

3.2 数据库设计

为了节省更多的空间，不造成大量的冗余，尽量减少数据库数据的存储次数，从

而便于数据库数据的更新、维护、备份和迁移，在设计数据库时尽量遵循数据库设计的三范式。因此根据图 3.1 的功能模块设计，通过 Navicat 使用 MySQL 创建了 8 张数据库表，其中最主要的表是 user 表，它通过外键的形式与 chat_info 表、contact 表、file_storage 表、folder_table 表、friend_request 表、general_setting 表、video_thumbnail 表建立联系。E-R 如图 3.3 所示，数据库表详细信息见表 3.1，

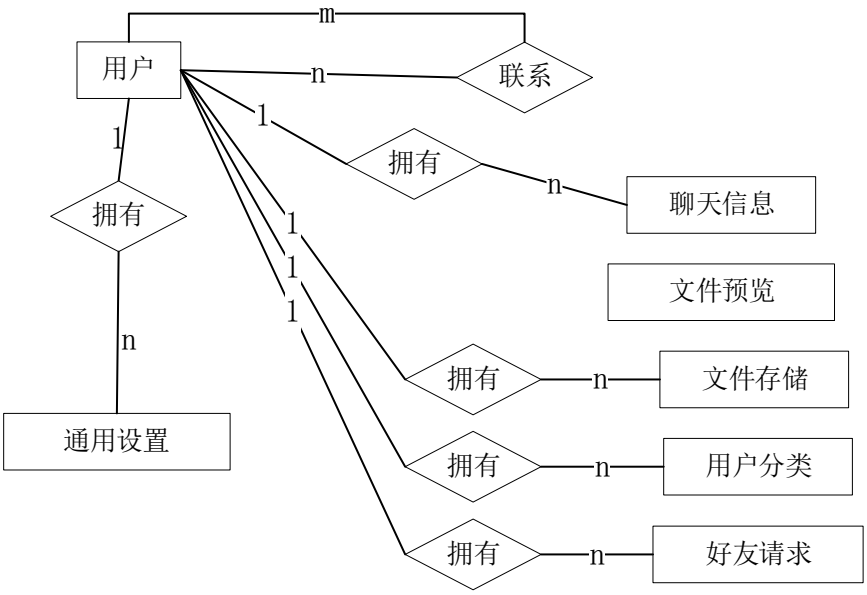

图 3.3 在线聊天 E-R 图

表 3.1 数据库详细信息表

序号	表	说明
1	chat_info	聊天信息表，用来存储用户所发送的所有信息（如是否已读、聊天内容、文件名、发送者、接收者等），同时文件、分享联系人也是通过聊天信息的形式存储的。使用两个信息区分聊天信息类型，file（使用 tinyint 类型减少浪费的空间）用来区分是否是正常聊天内容，share 用来区分是否是分享信息。
2	contact	联系人表，用来存储所有的联系人关系（如联系人所属分类，头像文件名称，是否打开免打扰、联系人的昵称和黑名单等），黑名单分为四种状态使用 tinyint 存储节约空间,用户双方均可以将对方拉入黑名单

表 3.1 数据库详细信息表（续）

序号	表	说明
3	file_storage	文件存储信息表，用来存储用户所发送的所有文件（如头像，文件，图片等）通过存储的发送人和接收人确定文件是谁的，通过上传后生成的 uuid 编码来去除文件重名的问题，而像头像和图片的区分是在项目的开始就新建了一个文件夹来存储头像信息。
4	folder_table	用户分类信息表，用来存储用户的所有分类名称，并且使用用户的唯一 id 来确认分类
5	friend_request	用户好友请求信息表，用来存储所有添加好友的信息，如果同意就删除，所有的好友都是存储的唯一 id
6	general_setting	通用设置表，该表用来存储用户的通用设置信息（如字体大小、发送方式等），为用户不论何时何地登录网站都能有相同的舒适体验感。
7	user	用户信息表，用来存储用户的所有信息（不包括头像），如用户的唯一 id、密码（使用 MD5 加密的密码）、用户的邮件地址、手机号、用户自定义的昵称、登录时间等信息。此表也存储用户用来进行人脸识别的图片文件名。用户的 id 是自动递增且无符号的数，通过这个 id 对其他 7 张表进行约束。
8	video_thumbnail	文件预览表，用来存储用户上传的视频、图片文件的预览图信息（如预览图的 uuid 文件名，对应的在聊天信息和文件存储中的对应信息和文件类型等），uuid 如有则是视频或者图片、否则就是文件、type 存储文件的 Content-Type 类型（用来区分文件）

以下是八个表的详细结构：（int 类型的字段长度为 0 自动默认为 11）

名	类型	长度	小数点	不是 null	虚拟	键	注释
chat_no	int	0	0	☑	☐	🔑 1	主键
read_flag	tinyint	1	0	☑	☐		是否未读 0未读 1已读
content	longtext	0	0	☑	☐		聊天信息
time	datetime	0	0	☑	☐		发送信息的时间
dest	int	0	0	☑	☐		目标用户
origin	int	0	0	☑	☐		发送用户
file	tinyint	1	0	☑	☐		0聊天 1文件
size	varchar	10	0	☐	☐		文件大小
share	int	0	0	☐	☐		分享的联系人id

图 3.4 chat_info 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
contact_no	int	0	0	☑	☐	1	主键
userid	int	0	0	☑	☐		用户id
contactid	int	0	0	☑	☐		联系人id此项不唯一
folder	varchar	10	0	☑	☐		联系人分类
headportrait	varchar	45	0	☐	☐		头像uuid
do_not_disturb	tinyint	1	0	☑	☐		是否开启免打扰
name	varchar	255	0	☐	☐		昵称
black_list	tinyint	0	0	☐	☐		黑名单 0正常 1user黑contact 2contact黑user 3互相黑

图 3.5 contact 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
no	int	0	0	☑	☐	1	
Id	int	0	0	☑	☐		发送者的id
uuid	varchar	45	0	☐	☐		唯一的id
originname	varchar	255	0	☐	☐		文件的原名
datetime	datetime	0	0	☑	☐		文件存储时间
folder	int	0	0	☐	☐		所属文件夹
receive_id	int	0	0	☐	☐		接收者id
path	varchar	20	0	☐	☐		为了存储除了folder的文件的路径

图 3.6 file_storage 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
no	int	0	0	☑	☐	1	
user_id	int	0	0	☑	☐		
folder	varchar	10	0	☑	☐		分类名称

图 3.7 folder_table 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
no	int	0	0	☑	☐	1	
request_id	int	0	0	☑	☐		发起请求的ID
receive_id	int	0	0	☑	☐		接收请求的ID

图 3.8 friend_request 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
no	int	0	0	☑	☐	1	主键
id	int	0	0	☑	☐		用户id
font_size	int	0	0	☑	☐		字体大小
send_style	tinyint	1	0	☑	☐		发送信息方式1enter 0ctrl+enter

图 3.9 general_setting 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
Id	int	0	0	☑	☐	1	用户的唯一id
name	varchar	50	0	☑	☐		用户姓名
password	varchar	32	0	☐	☐		用户密码
gender	varchar	1	0	☐	☐		性别
email	varchar	255	0	☐	☐		邮箱
phone	char	15	0	☐	☐		电话号码
nickname	varchar	50	0	☑	☐		昵称
active	tinyint	1	0	☑	☐		账户是否激活
login_time	datetime	0	0	☐	☐		登录时间
pre_login_time	datetime	0	0	☐	☐		上次登录时间
face_image_uuid	varchar	45	0	☐	☐		存储脸部数据的名称

图 3.10 user 表结构

名	类型	长度	小数点	不是 null	虚拟	键	注释
chat_no	int	0	0	☑	☐	1	外键
uuid	varchar	45	0	☐	☐		视频缩略图的uuid
file_storage_no	int	0	0	☑	☐		文件存储的主键
type	char	50	0	☐	☐		文件类型

图 3.11 video_thumbnail 表结构

3.3 接口设计

一个结构思路清晰的系统，接口设计一定是层次明确、一目了然的，接口为前端页面提供网站的数据。使用 Restful(Representational State Transfer)格式设计接口，可以使结构更加清晰，更方便获取数据。使用 HTTP 协议开发接口。将用户各种请求的信息作为接口的示例，如表 3.3 所示，使用的端口为 8080。其中所有的请求均有 http://localhost:8080/account/的前缀，这样有助于规范应用的接口设计和后期的管理和实现。

表 3.2 用户接口表

映射地址	接口说明
http://localhost:8080/account/resetName/2	用户 Id 为 2 的人重命名用户的昵称
http://localhost:8080/account/black/2	用户 Id 为 2 的人拉黑好友
http://localhost:8080/account/white/2	用户 Id 为 2 的人将好友移出黑名单
http://localhost:8080/account/contact/share/2/1	用户 Id 为 2 的人将 Id 为 1 的人分享到其他好友
http://localhost:8080/account/email/code	用户忘记密码向邮箱发送验证码

第 4 章 系统实现

4.1 后端实现

4.1.1 模式

项目的后台使用的是 MVC（Model-View-Controller）模式，即模型-视图-控制器模式。这是软件开发中用到的一种软件架构模式，这种架构模式可以使后续对项目的开发和维护变的更加简单、高效。这种架构模式通过对项目结构复杂度的简化，还能够使代码变的更加直观，甚至还能够重复利用某段代码。最典型的 MVC 就是 JSP + servlet + javabean 的模式。在本系统中使用的是 SpringBoot2 中集成的 SpringMVC 的项目模式。下图为 MVC 模式的关系。

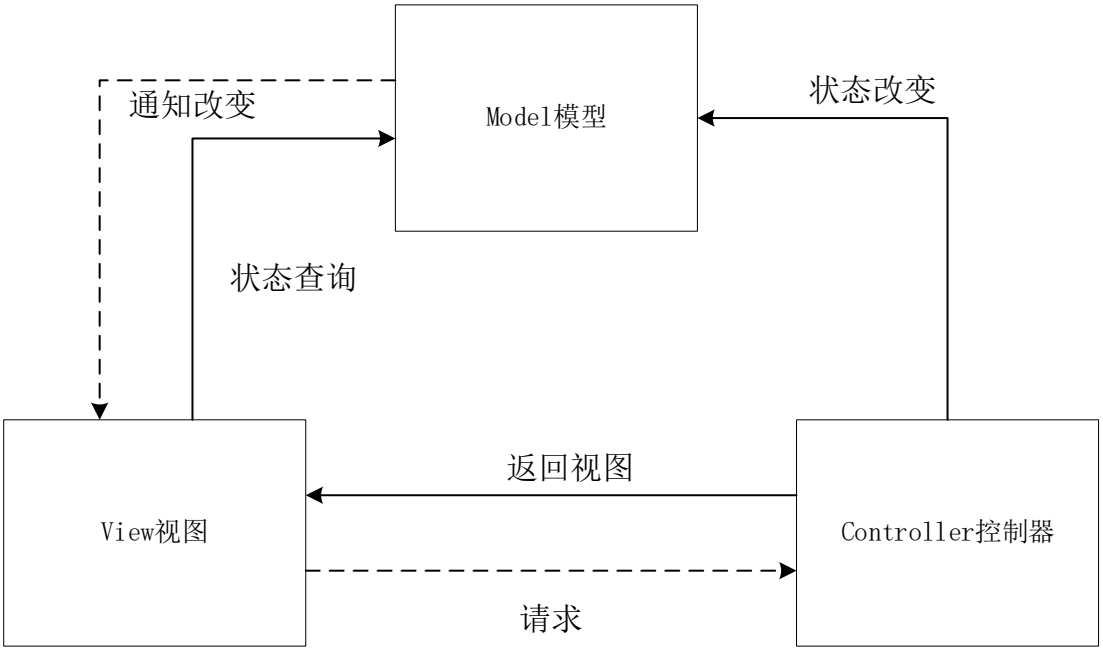

图 4.1 MVC 模式

4.1.2 架构设计

后端开发基于三层架构，即数据访问层（Dao）、业务逻辑层（Service）、控制层（Controller）。[6]

（1）数据访问层。是系统后端与数据库交互的层次，本系统使用的是 MyBatis Plus 框架来操作的数据库，并实现接口开发设计。与 MyBatis 相比 MyBatis Plus 在原来的

基础上做了增强，简化了开发、提高了开发效率，本项目没有使用映射文件的方式做数据库操作，而是使用映射器继承 IService 接口的形式，避免了书写大量的 SQL 映射文件。

（2）业务逻辑层，服务层在数据层和控制层之间进行逻辑处理。本项目使用的是先书写服务层的接口再实现，最后通过依赖注入的方式（注解注入的方式）让控制层调用。这种依赖注入的方式减少了模块间的耦合，使得项目开发和维护提高了很多效率。同时为了降低各个模块的耦合，每个种类的服务都有其各自的业务处理逻辑。

（3）控制层，通过调用业务逻辑层的接口来操作数据库，为前端页面提供所需要的信息，或者返回对应的视图信息。

4.1.3 关键技术设计

（1）生成随机的文件名。对于一个好的系统而言，用户可能会上传多个文件导致文件名重复，本项目使用生成 UUID 的方法解决这个问题，并且使用文件名的后缀作为分类依据分类存储使得后台管理起来更为方便。其代码实现如下图所示。

```
String[] fileNameSplit=file.getOriginalFilename().split( regex: "\\.");
String suffix=fileNameSplit[fileNameSplit.length-1];
String folderName=rootPath+suffix;
File folder=new File(folderName);
if(!folder.exists()){
    folder.mkdir();
}
//存储文件
String uuid= UUID.randomUUID().toString().toLowerCase();
try {
    file.transferTo(new File(folderName+"/"+uuid+"."+suffix));
} catch (IOException e) {
    e.printStackTrace();
}
```

图 4.2 生成 UUID 并存储文件

（2）文件上传。本项目里有较多的地方使用到了文件上传，用户可能会一次上传多个文件，需要存储在对应的位置。本系统使用用户上传的 content-type 类型和文件后缀进行分类存储。

（3）用户身份确定。对于一个网站系统来说拦截器能够保护网站免受大部分的

攻击，而且能够防止用户错误访问产生错误的结果

（4）密码加密。对于一个系统来说用户的个人信息安全是最重要的，尤其是用户的密码信息，为了更为安全的存储用户的个人密码，所以本项目采用了加密技术。现在的加密技术有很多比如 Base64、Sha256、MD5 等，本项目使用的是 MD5(Message-Digest Algorithm)加密技术[7]，获得密码后将密码转换为十六进制的字符串形式。安全性得以提升。其代码实现如下图所示。

```
public static String md5Encode(String inStr) throws Exception {
    MessageDigest md5 = null;
    try {
        md5 = MessageDigest.getInstance(algorithm: "MD5");
    } catch (Exception e) {
        System.out.println(e.toString());
        e.printStackTrace();
        return "";
    }

    byte[] byteArray = inStr.getBytes(charsetName: "UTF-8");
    byte[] md5Bytes = md5.digest(byteArray);
    StringBuffer hexValue = new StringBuffer();
    for (int i = 0; i < md5Bytes.length; i++) {
        int val = ((int) md5Bytes[i]) & 0xff;
        if (val < 16) {
            hexValue.append(str: "0");
        }
        hexValue.append(Integer.toHexString(val));
    }
    return hexValue.toString();
}
```

图 4.3 MD5 加密

（5）图片压缩。用户在使用过程中可能会发送很多大文件的图片，这就使得用户在加载这些图片时变的卡顿，降低了用户的体验感，为此本项目使用图片压缩技术来解决这个问题，在用户申请的时候会申请特定的接口得到压缩后的图片，使得用户页面快速响应。其代码如下所示。

```
@GetMapping("/photo/dim/{fileStorageNo}")
public void dimPhoto(@PathVariable("fileStorageNo") Integer fileStorageNo,
                     HttpServletResponse response) throws IOException {
```

```
        FileStorage fileStorage=fileDataService.getById(fileStorageNo);
        String root=FileUtil.getPathByFileStorageNo(fileStorage);
        response.setContentType("image/*");
        ServletOutputStream outputStream=response.getOutputStream();
        BufferedImage bi = ImageIO.read(new File(root));
        Thumbnails.of(root)
                        .sourceRegion(0, 0,bi.getWidth(),bi.getHeight())
                        .size(bi.getWidth(),bi.getHeight())
                        .keepAspectRatio(true) // 是否保持原来的长宽比
                        .toOutputStream(outputStream); // 返回缩率图的输出流
}
```

4.2 前端实现

4.2.1 模式

本项目使用的 Vue 框架使用了 MVVM（Model-View-ViewModel）模式，它也是一种软件开发中所用到的软件架构模式。这种开发模式是基于 MVC 和 MVP（Model-View-Presenter）的模型结构基础上衍生而来的[8]其关系如图 4.2 所示，这种软甲架构模式可以将数据实现一个双向绑定、同步刷新数据。这使得 Vue 框架整个都是响应式的，即只要数据一变，呈现的内容也随之改变，不在需要依赖后端所发送来的数据。由图 4.2 可知整个系统数据的变化是用户对 View 层进行更改，因为整个框架是响应式的，所以数据被 ViewModel 层所拿到，进而更改 Vue 框架的 Model 层[9]。反之，Model 层的数据发生变化，进而 ViewModel 响应 Model 层的变化发生改变，最后 View 层响应 ViewModel 层的变化将数据渲染后呈现给用户友好、美观的视图。

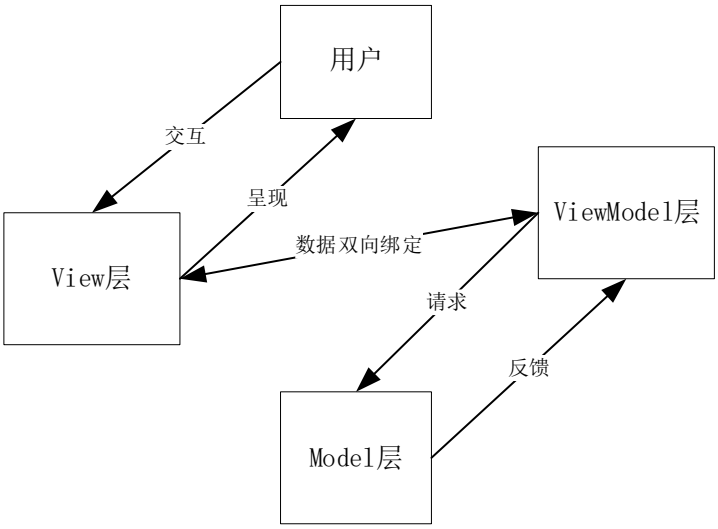

图 4.4 MVVM 模式

（1）View 层，视图层。这个是 Vue 中用户感受最为明显的地方，Vue 实现了内容和页面样式的分离。组件是 Vue 中最有特色的内容，可以大幅度的减少开发者的工作量，但是本系统是聊天系统并不需要很多种类不同的页面，所以本项目没有使用组件，而是使用了一个 Vue 的实例。

（2）ViewModel 层，也叫视图数据层。ViewModel 是 MVVM 中最重要的地方，它通过 Vue 中的 v-bind 和 v-model 命令影响 View 层。前者是数据的单向绑定（也就是说用户无法通过更改 View 层来影响 Model 层），后者是数据的双向绑定，用户可以通过更改 View 层来修改 View 层的数据。

（3）Model 层，也叫数据层。它通过接口的形式向后端发送请求，后端接收请求后向前端发送数据和状态码，通过状态码识别是否处理完毕。本系统使用的是 axios(ajax i/o system)，这不是一种新的技术，从本质上来看它还是对原生的 XMLHttpRequest 的封装。

本系统使用的是一整个 Vue 实例对象，实现 Vue 的组件功能是通过 v-if 等逻辑判断来实现模块的显示，进而在需要的位置得到相应的模块，以联系人分类为例，这样写能够节约较多的时间，也方便修改，代码如下

```
<template v-for="(classify,index) in contactClassify">
  <div  v-if="classify!=' 新 的 朋 友 '&&classify!=' 所 有 '"  class="folder-items" @click.self="deleteContactFromFolder(index,$event)">
    <div class="classify-name">
```

```
            <span>{{classify}}</span>
        </div>
        <div class="classify-statistics">
            <span>{{contact[index].length}}人</span>
        </div>
        <div class="folder-delete" @click.stop="deleteFolder(classify)">
            <div class="small-2 ion-icon-center">
                <ion-icon name="trash-outline"></ion-icon>
            </div>
        </div>
    </div>
</template>
```

代码中主要使用 v-for 和 v-if 来确定显示何种数据，@click.self，@click.stop 是为这个元素添加了一个点击事件，两者都是阻止点击事件之后的冒泡事件

4.2.2 关键技术设计

（1）发送聊天信息。因为本系统是在线聊天网站，所以本系统最关键的技术就是实时发送信息技术。如果不是实时的信息发送那么本项目就无法实现，为了实现该功能，本项目使用的是 HTML5 支持的 WebSocket[10]。实现代码如下：

```
const target = "ws://localhost:8080/chat";
if ('WebSocket' in window) {
    websocket = new WebSocket(target);
} else {
    alert('Not support websocket')
}
websocket.onerror = function () {
    setMessageInnerHTML("error");
}
websocket.onopen = function (event) {
    setMessageInnerHTML("Loc MSG:  建立连接");
}
websocket.onmessage = function (event) {
    let message=event.data;
```

```
    vue.putMessage(message);
}
websocket.onclose = function () {
    setMessageInnerHTML("Loc MSG:关闭连接");
}
window.onbeforeunload = function () {
    websocket.close();
}
```

（2）用户登录系统。用户和服务器每次连接都会产生一个 session，若用户每次只关闭页面不关闭浏览器，那么用户再次进入页面会直接登录。为了解决关闭浏览器就无法自动登录的问题，本系统使用 cookie 来解决这个问题，因为 cookie 在服务器为浏览器添加后会存在一段时间而不被删除。保证用户在一段时间内流畅登录系统。

4.3 项目实现

在本项目开发完毕后，在电脑端的 Edge 和 Chrome 浏览器经过测试并无异常（目前只开发到电脑端，安卓和 iPhone 有待开发响应式），呈现给用户的页面与需求一致。用户的功能模块如下列图片所示。

这是用户登录进入网站主页时，选择联系人聊天所显示的界面，其伪代码如下：

```
//向服务器发送请求返回用户的个人信息
axios({
        url: '/account/prepare/' + getCookie('id'),
        method: 'get',
    }).then(function (msg) {
    //对返回信息进行处理放入 vue 的 data 中
    putMessageToVue();
});
//点击联系人
 contactClick(item,index, e) {
     //更新所选中的联系人坐标
     updateSelectedContactLocal();
     //更新联系人的被点击状态
     updateSelectedContactView();
```

```
        //展开聊天主界面和联系人信息界面
        showChatMainAndContactInfo();
        //发送给服务器已读
        sendInfoToServer();
        //客户端变为已读
        clientChangeStatue();
    }
    //点击显示表情界面
    {
        //重置表情包宽度
        ResizeEmojiWidth();
        //重新加载表情包上面的聊天信息栏高度
        onLoadChatMain();
    }
```

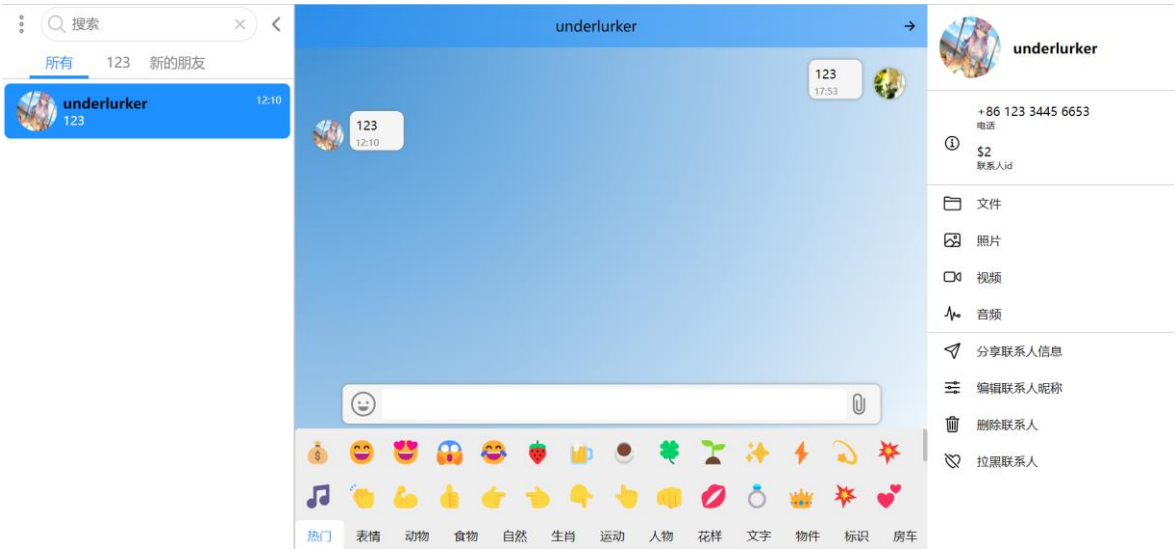

图 4.5 用户主要功能界面

下图是聊天主界面的黑暗模式，主要是使用 CSS 配合 JS 来实现的。CSS 设置:root 的 CSS 变量，再写一个 dark 类重新写一下与:root 对应的变量，更改模式只需要将 dark 类加到最外层标签即可，代码如下：

```
let dark=false;
darkModel.onclick=function(){
```

```
    if(!dark){
        最外层标签.addClass('dark');
        dark=true;
    }
    else{
        最外层标签.removeClass('dark');
        dark=false;
    }
}
```

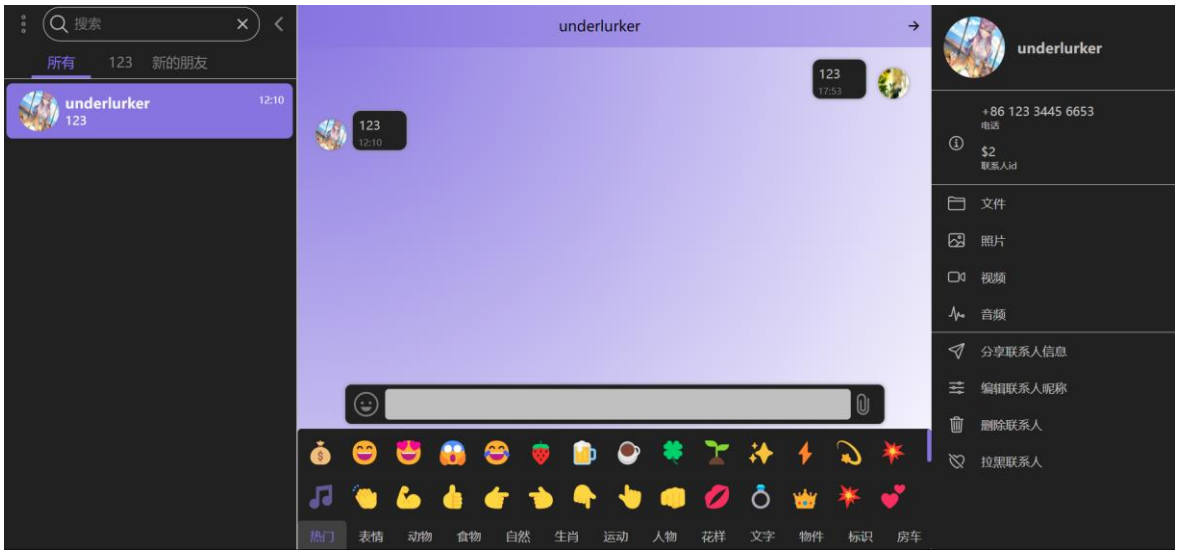

图 4.6 夜间模式

用户点击设置选项进入用户设置界面，伪代码如下：

```
While(所有设置界面按钮){
    按钮绑定点击事件{
        对应界面显示
    }
}
```

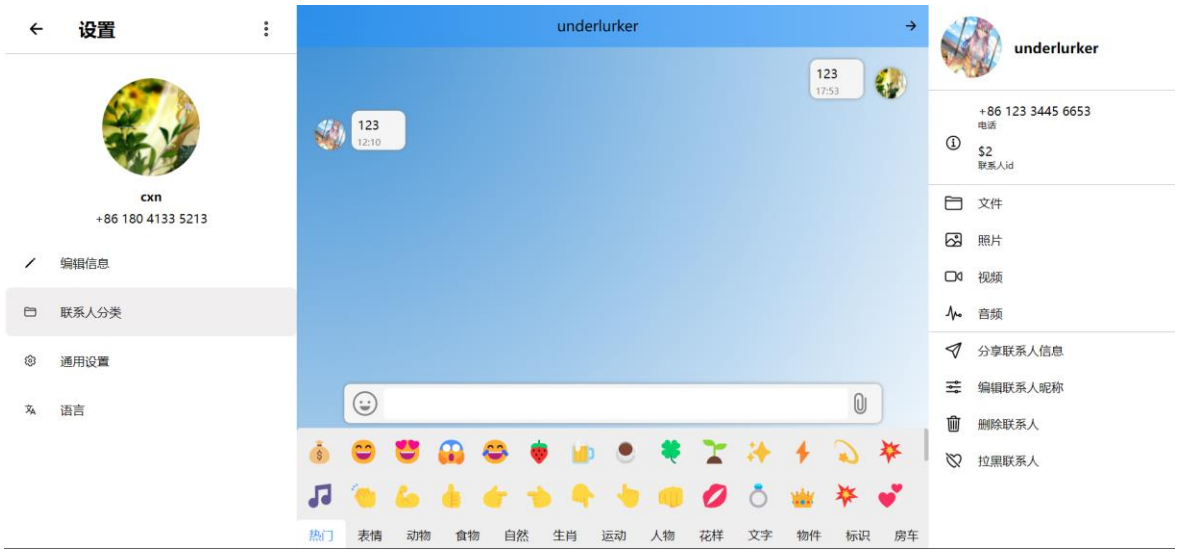

图 4.7 用户设置页面

用户上传头像伪代码如下：

```
//复制上传的图片文件到
CopyFileToForm();
Image.onload=function(){
    //使用图片剪切框架
    Image.Jcrop();//得到图片裁剪信息
}
```

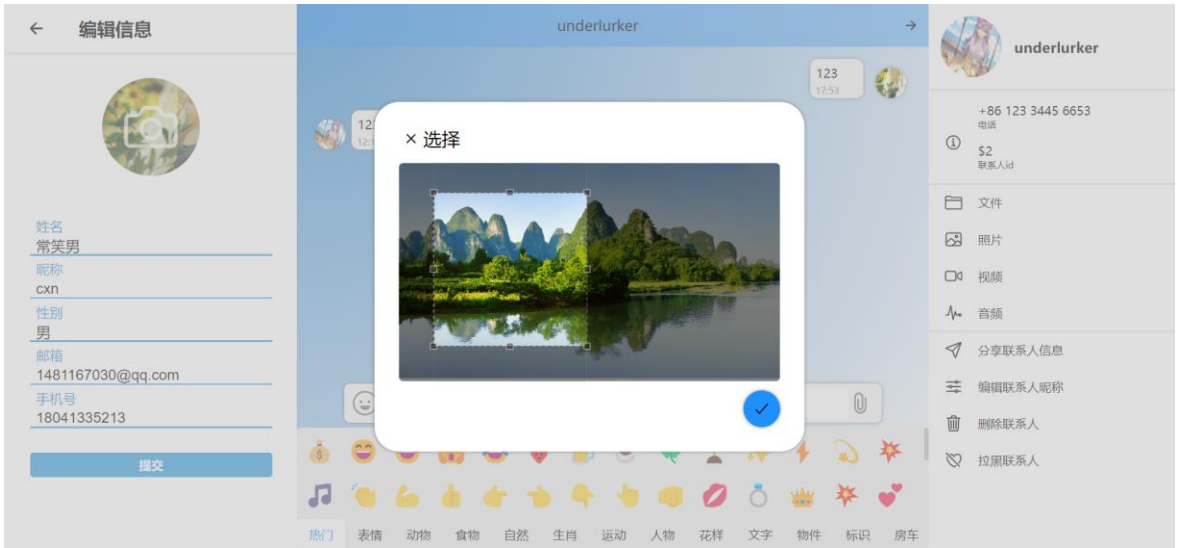

图 4.8 用户上传头像

第 5 章 系统测试

5.1 功能测试

功能测试（Functional Testing），这是测试中最基本的测试，不需要管项目的内部实现逻辑，主要看项目是否符合软件的需求分析，是否达到了软件的预期结果，验证系统是否符合产品的需求规格。由第 4 章的项目实现部分可以得知，项目的功能测试基本通过。下面是项目的其他测试。

表 5.1 其他测试

测试项目	测试结果
文件上传功能	图片上传成功、视频上传成功、文件上传成功、音频上传成功
图片预览功能	图片预览成功、图片放大缩小成功、图片移动成功、图片下载成功
视频播放功能	视频播放功能成功、视频全屏功能成功
音频播放功能	音频播放功能成功
头像上传功能	头像上传功能成功

5.2 安全性测试

安全性测试（Security Testing）是用来验证系统中所设置的保护机制是否能够在真实的环境中保护系统不受到非法的入侵。测试中使用向接口发送请求的方式企图访问服务器的数据来测试系统的安全性。本项目中采用的是拦截器机制，拦截除了登录、注册、初始页面和忘记密码页面外的所有接口，保证用户信息不被外界人士获取保证用户信息的安全。对于拦截器的测试如下表：

表 5.2 安全性测试

Cookie 有无	访问链接	测试结果
有	localhost:8080/main.html	登录成功
无	localhost:8080/main.html	跳转回登录页面
有	localhost:8080/prepare/1	返回 JSON 格式的用户 1 信息
无	localhost:8080/prepare/1	跳转回登录页面

5.3 访问测试

由于本项目是一个网站不需要安装，只要有浏览器就行。所以安装测试变为测试其他设备是否能够访问在线聊天网站主页。

在其他电脑上访问本网站就要将网站发布在公网上，本项目使用的是 natapp.exe。使用的是免费的隧道协议。配置如下：

authtoken=4e4c57ba53342a81 #对应一条隧道的 authtoken

clienttoken= #对应客户端的 clienttoken,将会忽略 authtoken,

log=none #log 日志文件,可指定本地文件

loglevel=ERROR #日志等级 默认为 DEBUG

http_proxy= #代理设置 非代理上网用户请务必留空

启动 natapp.exe 后得到的域名为 p695zz.natappfree.cc，在浏览器输入并登录网站，得到如图 5.10 所示结果。由此可得，其他电脑也能访问本项目。

表 5.3 访问测试

访问设备	访问地址	访问结果
本机	localhost:8080	成功
本机	p695zz.natappfree.cc	成功
其他设备	localhost:8080	失败
其他设备	p695zz.natappfree.cc	成功

5.4 异常测试

系统异常测试又叫做系统容错和可恢复性测试，它是通过人工介入的手段使系统产生异常（软件逻辑和硬件两方面），验证系统产生异常之前与产生异常之后的两种状态（包括了系统功能状态和系统的运行状态），以求达到检验系统容错、排错和恢

复的能力。

测试方法如下：将设备的网络断开，本机访问 localhost:8080/forgetCode.html，所得结果如下表所示。

表 5.4 异常测试

测试条件	测试结果
有网络连接	接收到信息“success”
无网络连接	接收到信息“fail”

第 6 章 总结与建议

6.1 总结

本文论述了基于 Springboot2、Vue 与 MySQL 数据库的在线聊天网站的设计与实现，主要介绍了系统开发环境、系统分析、系统设计、系统的前后端实现和系统的测试。通过接口设计、文件压缩和原生 JavaScript 等方式来提示系统的使用性能、安全性。通过这次实验验证了 Springboot2 和 Vue 框架相比于其他框架的优势。通过在不同平台的测试，也证明了在 PC 端跨平台的性质。但在移动端还需要进行改进。本项目达到了大部分的项目需求，下一步进行在手机移动端的响应式代码编写，使之适应移动端，并优化数据库存储结构。

6.2 建议

应在在线聊天网站中添加语音聊天和视频通话功能，使用户有更加多样的通信方式，而不是局限于文字聊天。

参考文献

[1] 巢晟盛.基于 SpringBoot 微服务架构下前后端分离的 MVVM 模型浅析[J].电脑知识与技术,2021,(23):128-129+141.

[2] 霍福华;韩慧.基于 SpringBoot 微服务架构下前后端分离的 MVVM 模型[J].电子技术与软件工程,2022,(01):73-76.

[3] 张峰.应用 SpringBoot 改变 web 应用开发模式[J].科技创新与应用,2017(23):193-194.

[4] 朱二华.基于 Vue.js 的 Web 前端应用研究[J].科技与创新,2017(20):119-121.

[5] 杨典;李小燕;刘培焱;刘丰硕.基于 OpenCV 的变电站仪表识别方法研究[J].自动化与仪表,2022,37(04):75-80.

[6] 翟宝峰;戴永彬;武志刚;王学志.“Java EE 应用开发框架”课程内容的优化[J].辽宁工业大学学报(社会科学版),2021,23(03):120-122.

[7] 王镇道;李妮.一种优化的 MD5 算法与硬件实现[J].湖南大学学报(自然科学版),2022,49(02):106-110.

[8] 刘亚茹;张军.Vue.js 框架在网站前端开发中的研究[J].电脑编程技巧与维护,2022,(01):18-19+39.

[9] 焦鹏珲.基于 SpringBoot 和 Vue 框架的电子招投标系统的设计与实现[D].南京:南京大学,2018.

[10] 吴绍卫.WebSocket 在实时消息推送中的应用设计与实现[J].福建电脑,2021,(11):80-83.

致　谢

实践是检验真理的唯一标准。自己的专业技术掌握的牢不牢固还是要自己动手实践一番才能真正的得出结果，要不然只称得上是理论上的巨人，行动上的矮子。我们的实践过程是我们学习中的宝贵的财富。学习固然是没有问题的，但实践页同样重要。我们要从实践中得到感性认识，在从感性认识中得到理性认识，这就是书本上的知识。但这种知识归根到底是要用到实践中去的。而像计算机这种学科，最新的知识永远不在书上，而是在劳动一线中，过了几年才会出一本几年前的知识。感谢这次的论文实践，让我认识到了实践与课本上知识的差距。

常笑男

2022 年 4 月 于沈阳