# 摘要

随着大学生的生活不断丰富多彩，之间的交流也愈发的密切，无形当中在校园生活中就形成了校园社交。从一个寝室，到一个专业，再到同一所学校，校园社交的规模不断扩大，学生的见识和思想也不断提高，沟通交流得到了锻炼，还找到了志同道合的朋友。现在当今互联网时代，互联网中越来越多的社交软件出现人们的生活当中。校园社交网站是一个服务于校园人群的社交网站，为人们创造了共同爱好的交流和碰撞，与其他社交软件相比，还提供校园的服务，如查成绩，考试时间，课程表，为用户提供了方便与便捷，真正的做到了社交学习两不误。

本文描述了校园社交网站的设计与实现。实现了用户之间进行聊天，发文件，发动态等等，管理员可以发布通知，查看用户的分布图和每天在线人数的实时数据等。网站页面的UI采用了独特的设计，清晰明了，轻快明朗。

该网站采用前后端分离模式，前端技术以Vue.js进行开发，以Node.js进行连接数据库等后端开发，UI组件库用ElementUI组件库，数据库为Mysql，为用户提供美观的聊天、学习环境和良好的使用体验。

关键词

校园社交；前后端分离；Vue.js；Node.js

# **Abstract**

With the colorful life of college students, the communication between the increasingly close, invisible in the campus life in the formation of campus social. From a dormitory to a major to the same school, the scale of campus social contact has been constantly expanded, students' knowledge and ideas have been constantly improved, communication has been exercised, but also find like-minded friends. Now in the Internet era, more and more social software appears in people's life. The campus social network website is a social network website that serves the campus crowd, creating the communication and collision of common interests for people. Compared with other social software, it also provides campus services, such as checking scores, examination time and class schedule, providing convenience and convenience for users, and truly achieving the goal of making friends and learning at the same time.

This paper describes the design and implementation of campus social network website. Realized the chat between users, send files, send dynamic and so on, the administrator can issue notifications, view the distribution of users and the number of online real-time data every day. The UI of the website page adopted unique design, clear and clear, light and bright.

The website adopts the front and rear end separation mode, the front end technology is developed with vue. js, the UI component library is developed with ElementUI component library, the database is connected with node. js and other back-end development, the database is Mysql, providing users with beautiful chat, learning environment and good use experience.

**Key words**

campus social; front-end separation; Vue.js; Node.js

目录

[摘要 I](#_Toc39670809)

[**Abstract** II](#_Toc39670810)

[第一章 绪论 1](#_Toc39670811)

[1.1系统的研究背景 1](#_Toc39670812)

[1.2研究的目的和意义 1](#_Toc39670813)

[1.3国内外的研究现状 1](#_Toc39670814)

[1.3.1 国内研究现状 1](#_Toc39670815)

[1.3.2国外研究现状 2](#_Toc39670816)

[1.4 相关应用技术的介绍 2](#_Toc39670817)

[1.4.1 Vue.js框架 2](#_Toc39670818)

[1.4.2 Node.js框架 2](#_Toc39670819)

[1.5 论文研究内容 3](#_Toc39670820)

[1.6 研究的目的和意义 3](#_Toc39670821)

[1.7 论文组织结构 3](#_Toc39670822)

[第二章 校园社交网站系统需求分析 4](#_Toc39670823)

[2.1用户需求 4](#_Toc39670824)

[2.2可行性分析 4](#_Toc39670825)

[2.2.1 技术可行性分析 4](#_Toc39670826)

[2.2.2 操作可行性分析 4](#_Toc39670827)

[2.2.3 经济可行性分析 5](#_Toc39670828)

[2.3 系统功能需求 5](#_Toc39670829)

[2.3.1 系统功能参与者 5](#_Toc39670830)

[2.3.2 系统用例模型 5](#_Toc39670831)

[2.4各模块用例描述 6](#_Toc39670832)

[2.4.1登录注册模块用例描述 6](#_Toc39670833)

[2.4.2动态模块用例描述 8](#_Toc39670834)

[2.4.3好友模块用例描述 10](#_Toc39670835)

[2.4.4个人中心模块用例描述 12](#_Toc39670836)

[2.4.5 管理员模块用例描述 13](#_Toc39670837)

[2.5 非功能性需求 16](#_Toc39670838)

[2.5.1环境需求 16](#_Toc39670839)

[2.5.2完整性需求 16](#_Toc39670840)

[2.5.3 适应性需求 16](#_Toc39670841)

[2.6本章小结 17](#_Toc39670842)

[第三章 校园社交网站系统设计 18](#_Toc39670843)

[3.1 系统设计原则 18](#_Toc39670844)

[3.2 系统总体架构设计 18](#_Toc39670845)

[3.3 系统功能模块结构设计 19](#_Toc39670846)

[3.4 系统类图设计 19](#_Toc39670847)

[3.5 数据库设计 20](#_Toc39670848)

[3.5.1 数据库概念模型设计 20](#_Toc39670849)

[3.5.2 数据库逻辑结构设计 24](#_Toc39670850)

[3.6 系统界面设计 29](#_Toc39670851)

[3.7 本章小结 29](#_Toc39670852)

[第四章 系统详细设计与实现 30](#_Toc39670853)

[4.1 用系统开发与运行环境 30](#_Toc39670854)

[4.2 发布动态功能详细设计与实现 30](#_Toc39670855)

[4.2.1 界面详情 30](#_Toc39670856)

[4.2.2 实现细节 31](#_Toc39670857)

[4.2.3 功能流程 31](#_Toc39670858)

[4.3 选择动态背景主题功能详细设计与实现 32](#_Toc39670859)

[4.3.1 界面详情 32](#_Toc39670860)

[4.3.2 实现细节 33](#_Toc39670861)

[4.3.3 功能流程 33](#_Toc39670862)

[4.4 添加好友功能详细设计与实现 34](#_Toc39670863)

[4.4.1 界面详情 34](#_Toc39670864)

[4.4.2 实现细节 35](#_Toc39670865)

[4.4.3 功能流程 36](#_Toc39670866)

[4.5 聊天功能详细设计与实现 36](#_Toc39670867)

[4.5.1 界面详情 37](#_Toc39670868)

[4.5.2 实现细节 37](#_Toc39670869)

[4.5.3 功能流程 38](#_Toc39670870)

[4.6 管理员授予用户权限/删除用户功能详细设计与实现 39](#_Toc39670871)

[4.6.1 界面详情 39](#_Toc39670872)

[4.6.2 实现细节 39](#_Toc39670873)

[4.6.3 功能流程 40](#_Toc39670874)

[4.7本章小结 40](#_Toc39670875)

[第五章 系统测试 41](#_Toc39670876)

[5.1 系统测试的目的 41](#_Toc39670877)

[5.2 系统测试的方法 41](#_Toc39670878)

[5.3 系统测试用例 41](#_Toc39670879)

[5.3.1 登录注册模块测试用例 41](#_Toc39670880)

[5.3.2 动态模块测试用例 42](#_Toc39670881)

[5.3.3 好友模块测试用例 43](#_Toc39670882)

[5.3.4 个人中心模块测试用例 44](#_Toc39670883)

[5.3.5 管理员模块测试用例 44](#_Toc39670884)

[5.4 本章小结 45](#_Toc39670885)

[结论 46](#_Toc39670886)

[参考文献 47](#_Toc39670887)

[致谢 48](#_Toc39670888)

# 第一章 绪论

## 1.1系统的研究背景

近几年，即时通讯产业发展迅猛，随着Facebook在短时间内风靡全球，社交网络(又称社交网络系统或社交网络服务)成为近年来最为流行的互联网应用之一,并被认为是继搜索和Web2.0之后互联网的又一次革命[[[1]](#endnote-1)]，例如微信、微博等，但其中专属于大学生的校园社交平台数量不多、规模不大，曾经在大学生社交中起到重要作用的人人网，近几年也渐渐退出了历史舞台[[[2]](#endnote-2)]。

说起校园社交软件，一代校园社交霸主人人网匆忙落幕，现校园社交充斥着浮躁不平静的生活气氛，同学们需要一个纯净的、包容的、自由的社交环境。给同学们在校园互联网上的一片净土，立足于和谐的校园生活环境，以兴趣为起点宣扬个性的校园社交网站，把快乐扩散分享给周围同学。因此我们拟构建一个专门为大学生设计的社交平台[[[3]](#endnote-3)]。

## 1.2研究的目的和意义

校园社交网站是一个服务于校园人群通过网站就可以结识校园内志同道合的朋友的平台。在网上进行校园社交是一种全新的社交环境和独具特色的社交新体验。目前校园社交网站加入了课程表，考试时间安排，查成绩等当下流行功能来贴近校园生活。校园社交网站使用前后端分离后，前端负责View和Controller层，后端只负责Model层，业务处理/数据等。前端也可以进行数据处理工作，对服务器的压力减少到最小。

## 1.3国内外的研究现状

### 1.3.1 国内研究现状

国内：对于中国社交网络的发展，我们看到很多新的平台。比如豆瓣、糗百、猫扑、马蜂窝、唱吧等，聚合一大批有共兴趣或者有共同理想的人，整个平台内UGC的质量和数量都是有目共睹，用户粘性更高，所以要以兴趣为导向聚合人脉。目前也有很多网站以论坛+博客形式，也同样产生了很多优质、富有见解内容，满足个人深度思考的需求，又以兴趣和知识性为依托聚合一大批志同道合的人。所以要以强关系人脉为基础，产生优质内容为目标[[[4]](#endnote-4)]。

### 1.3.2 国外研究现状

国外：其实Facebook的起源来自于校园。最早的时候，根据“六度理论”，通过6个这样的结点，你就能认识到世界上任何想认识的陌生人。这个理论的源头，来自1967年哈佛大学的心理学教授Stanley Milgram创立的六度分割理论(Six Degrees of Separation) ，即“你和任何一个陌生人之间所间隔的人不会超过六个”[[[5]](#endnote-5)]。1997年，这个理论最早的实践者--SixDegrees,上线，其将这一理论用互联网的方式体现出来，巅峰时期也有近百万用户。1999年，这个网站的创始人以1.25亿美元的价格把它卖给青年潮流传媒网络。约一年后，SixDegrees 由于不具备赢利能力，被迫关闭。到2004年的2月，美国的实名制网站Facebook上线，创始人马克.扎克伯格是哈佛大学的学生。Facebook 的创意其实来自扎克伯格中学时的同学录，扎克伯格只是将其搬上网络。最初，网站的注册仅限于哈佛学院的学生，后来拓展到更多的高校[[[6]](#endnote-6)]。

## 1.4 相关应用技术的介绍

### 1.4.1 Vue.js框架

Vue.js是一套构建用户界面的渐进式框架，它只关注视图层，与其他大型框架不同的是，Vue被设计为可以自底向上逐层应用，不仅易于上手，还便于与第三方库或既有项目整合[[[7]](#endnote-7)]。Vue.js的特性如下: (1)轻量级的框架。(2)双向数据绑定。(3)指令。(4)组件化。(5)基于依赖追踪的观察并且使用异步队列更新。所有的数据都是独立触发的。

### 1.4.2 Node.js框架

Node.js是一个基于Chrome JavaScript运行时建立的一个平台。它是一个事件驱动I/O服务端JavaScript环境，基于Google的V8引擎，V8引擎执行JavaScript的速度非常快，性能非常好[[[8]](#endnote-8)]。Node.js的用户不必担心死锁过程，Node中几乎没有函数直接执行I/O操作，因此进程从不阻塞。由于没有任何阻塞，可伸缩系统在Node中开发是非常合理的。

## 1.5 论文研究内容

本网站的设计与实现是采用前后端分离模式的网站。本网站将包括管理员、用户两种身份用户。将包含的模块功能主要有登录注册模块、动态模块、好友模块、个人中心模块、管理员模块。

## 1.6 研究的目的和意义

校园社交网站是一个服务于校园人群通过网站就可以结识校园内志同道合的朋友的平台。在网上进行校园社交是一种全新的社交环境和独具特色的社交新体验。目前校园社交网站加入了课程表，考试时间安排，查成绩等当下流行功能来贴近校园生活。校园社交网站使用前后端分离后，前端负责View和Controller层，后端只负责Model层，业务处理/数据等。前端也可以进行数据处理工作，对服务器的压力减少到最小。

## 1.7 论文组织结构

本次论文是由六个大章节构成，介绍如下：

第一章：绪论，本章通过系统的研究与背景来介绍了校园社交网站的优势以及网站开发技术的选择。

第二章：系统的需求分析，本章通过需求分析、功能需求分析其实现的可能性。

第三章：系统的设计原则和总体架构，本章包括数据库概念模型设计和数据库表设计。

第四章：系统的功能实现。本章为前端界面以及功能展示和后台功能实现，系统的配置文件和部署。

第五章：系统测试。本章提供为系统进行黑盒测试。

第六章：结论。研发系统时遇到的困难和总结。

# 第二章 校园社交网站系统需求分析

## 2.1用户需求

对于当今的校园大学生，大部分大学生一天使用手机电脑的时间占全天的50%，基本离不开手机电脑，然而通过软件进行社交聊天的时间占使用手机电脑时间的50%，可见现在社交聊天较受欢迎。以前社交的方式基本通过书信进行交流，回复时间过长。如今互联网的发展，大学生则通过网上进行社交聊天。在此环境下，诸多互联网企业开始创造校园社交软件，同学们之间可以通过网上校园进行分享趣事或者找到志同道合的朋友，为社交创造一个良好的氛围[[[9]](#endnote-9)]，在校园中将会比较受欢迎。

## 2.2可行性分析

### 2.2.1 技术可行性分析

Vue.js是当今前端三大主流框架之一，不仅易上手，而且是轻量级、组件化的。它还具有生命周期函数，其中包括beforeCreate、created、beforeMount、mounted、beforeUpdate、updated、beforeDestroy、destroyed八大生命周期函数。这些生命周期函数被赋予不同的功能，而且相较于以前的HTML、JS、CSS技术，对前端有了更好的规划和分类[[[10]](#endnote-10)]。后端采用Node.js框架，它是基于JavaScript语言的框架，不但具有统一类库的功能，而且在代码设计时具有标准化，可以通过npm命令引入自己要用到的封装好的包，还保证了此网站的性能和稳定性。因此，考虑到本次网站的设计与实现，最终选择了前端Vue.js框架、后端Node.js框架来作为本次网站的技术框架。通过以上分析，本网站的设计与实现在技术上是完全可行的。

### 2.2.2 操作可行性分析

当今互联网飞速发展，社交软件较受欢迎，校园社交软件更为安全，几乎大学生都会通过软件来进行社交聊天，同时此网站还增添了选择主题功能，更加吸引用户的注意。老师也可通过此网站来给同学们发学习任务及文件，同学还可以发送动态分享自己的博客，真正的做到了学习社交两不误。用户直接通过电脑就可进行使用，操作起来比较方便，易于理解。因此，本网站在操作方面是完全可行的[[[11]](#endnote-11)]。

### 2.2.3 经济可行性分析

该网站成本很低，设计与实现只需要一台计算机和相关软件，软件可以免费下载。该网站对计算机没有太高要求，用户只需要通过浏览器就可访问该网站，使用人数预计很多，将来通过此平台增加一些消费功能，能够给网站带来更高的经济效益以及维持运作。所以，从经济可行性分析的角度上进行分析，有较高的可行性。

## 2.3 系统功能需求

### 2.3.1 系统功能参与者

参与者如表2-1参与者词汇表所示。

表2-1 参与者词汇表

|  |  |  |
| --- | --- | --- |
| 序号 | 词汇 | 描述 |
| 1 | 用户 | 用户通过首页进入登录界面进行登录/注册。登录后的用户可以进行聊天(群聊、聊天记录收藏、文件记录显示)功能、发布/点赞/评论动态、还可查看学校内部信息(课程表、考试信息、成绩)等等 |
| 2 | 管理员 | 管理员登录之后不光和普通用户所具备的功能相同，还可以发布通知，也可以授予用户权限以及删除用户，还有具有查看数据分析图(用户分布图、学校数据柱状图、在线人数折线图)功能 |

### 2.3.2 系统用例模型

本系统的总体用例模型如图2-2所示。

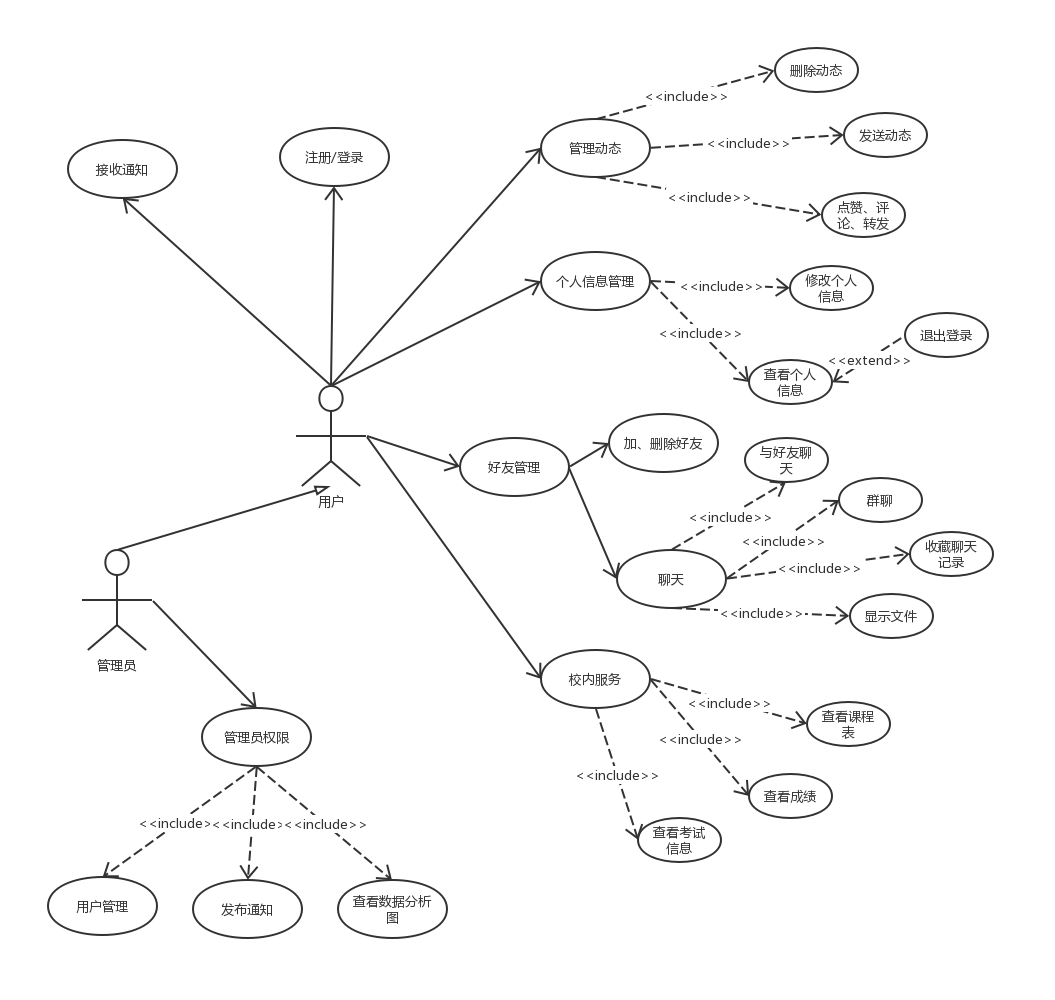


图2-2 系统总体用例图

## 2.4各模块用例描述

本系统主要分为以下模块：登录注册模块、动态模块、好友模块、个人中心模块、管理员模块。

### 2.4.1登录注册模块用例描述

登录注册模块主要功能有登录、注册。用户进入首页时，通过点击登录进入登录界面，用户通过账号和密码方可登录，系统会判断账号密码是否为空。若都不为空，会交给后台进行查询用户是否存在，密码是否正确。若其中一项为空，会提示警告。若没有注册的用户，在登录界面点击注册按钮进行注册。

用户登录的用例描述表如表2-3所示。

表2-3 用户登录用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户登录 |
| 前置条件 | 用户已注册 |
| 后置条件 | 登录成功直接跳转到首页 |
| 主要流程 | 1、用户进入登录界面 |
|  | 2、用户输入账号和密码 |
|  | 3、前端检测账号、密码是否为空 |
|  | 4、后台检测账号是否匹配 |
|  | 5、界面提示登录成功并跳转到首页 |
| 实现约束 | 用户未登录 |
| 结束 | 提示登录成功，跳转到首页 |

用户注册的用例描述表如表2-4所示。

表2-4 用户注册用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户注册 |
| 前置条件 | 未注册的用户 |
| 后置条件 | 注册成功直接自动登录 |
| 主要流程 | 1、用户进入登录界面 |
|  | 2、点击注册按钮进入注册界面 |
|  | 3、输入相关注册信息 |
|  | 4、检测注册信息的正确性和完整性 |
|  | 5、注册成功直接登录 |
| 实现约束 | 用户未注册 |
| 结束 | 提示注册成功，跳转到首页 |

### 2.4.2动态模块用例描述

动态模块主要功能有发布动态，点赞/评论动态，转发动态、删除动态。用户登录后进入首页时，通过点击动态进入动态页面，动态界面显示好友和自己的动态，用户可进行发布动态，点赞动态，评论动态和转发动态，以及可以删除自己的动态用户。

动态模块用例图如图2-5所示。

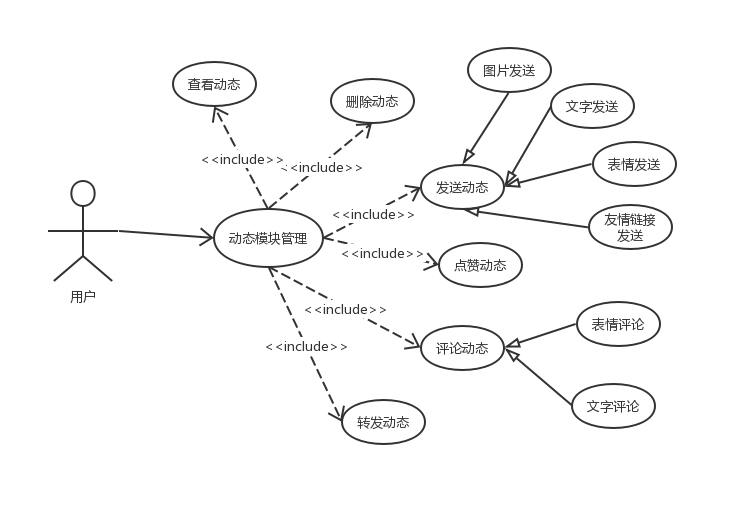


图2-5 动态模块用例图

发布动态的用例描述表如表2-6所示。

表2-6 发布动态用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户发布动态 |
| 前置条件 | 用户已登录 |
| 后置条件 | 发布成功，动态列表刷新 |
| 主要流程 | 1、用户点击发布动态按钮 |
|  | 2、用户输入动态内容 |
|  | 3、检测动态内容的完整性 |
|  | 4、点击发布按钮 |

续表2-6 发布动态用例描述表

|  |  |
| --- | --- |
|  | 5、界面提示发布成功并刷新动态列表 |
| 实现约束 | 动态内容为空 |
| 结束 | 动态发布成功 |

点赞/评论动态的用例描述表如表2-7所示。

表2-7 点赞/评论动态用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户点赞/评论动态 |
| 前置条件 | 用户已登录，动态列表存在动态 |
| 后置条件 | 点赞/评论成功，动态点赞/评论区域刷新 |
| 主要流程 | 1、用户进入动态页面 |
|  | 2、用户在动态列表中选择某动态进行点赞/评论 |
|  | 3、提示点赞/评论成功，点赞/评论区域刷新 |
| 实现约束 | 动态列表为空 |
| 结束 | 提示点赞/评论成功 |

转发动态的用例描述表如表2-8所示。

表2-8 转发动态用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户转发动态 |
| 前置条件 | 用户已登录，动态列表存在动态 |
| 后置条件 | 转发成功，动态列表刷新 |
| 主要流程 | 1、用户进入动态页面 |
|  | 2、用户选择某个动态点击转发按钮 |
|  | 3、提示转发成功并刷新动态列表 |

续表2-8 转发动态用例描述表

|  |  |
| --- | --- |
| 实现约束 | 动态内容为空 |
| 结束 | 动态转发成功 |

删除动态的用例描述表如表2-9所示。

表2-9 删除动态用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户删除动态 |
| 前置条件 | 用户已登录，并且只能删除自己发过的动态 |
| 后置条件 | 转发成功，动态列表刷新 |
| 主要流程 | 1、用户进入动态页面 |
|  | 2、用户在自己的动态上点击删除按钮 |
|  | 3、提示删除成功并刷新动态列表 |
| 实现约束 | 动态为自己发布 |
| 结束 | 动态删除成功 |

### 2.4.3好友模块用例描述

好友模块主要功能有添加好友、删除好友、好友聊天。用户登录后进入首页时，通过进入动态页面进行添加好友和删除好友，点击聊天进入聊天界面进行聊天。

添加好友的用例描述表如表2-10所示。

表2-10 添加好友用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户添加好友 |
| 前置条件 | 用户已登录 |
| 后置条件 | 对方接收到消息后同意 |

续表2-10 添加好友用例描述表

|  |  |
| --- | --- |
| 主要流程 | 1、用户点击添加好友 |
|  | 2、用户输入对方账号或从列表中直接添加 |
|  | 3、等待对方同意或拒绝 |
|  | 4、若对方同意，则提示对方已添加为好友。若对方拒绝，则提示对方已拒绝。 |
| 实现约束 | 对方账号不存在或对方拒绝 |
| 结束 | 添加好友成功 |

删除好友的用例描述表如表2-11所示。

表2-11 删除好友用例描述表

|  |  |  |
| --- | --- | --- |
| 用例条例 | 描述 | |
| 参与者 | 用户 | |
| 描述 | 用户删除好友 | |
| 前置条件 | 对方存在用户好友列表中 | |
| 后置条件 | 删除好友成功 | |
| 主要流程 | 1、用户选择好友 | |
|  | 2、点击删除按钮 | |
|  | 3、提示删除成功，对方和自己好友列表都刷新 | |
| 实现约束 | 对方已删除自己 | |
| 结束 | | 删除好友成功 |

好友聊天的用例描述表如表2-12所示。

表2-12 好友聊天用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户与好友聊天 |
| 前置条件 | 用户已登录，有好友 |
| 后置条件 | 双方都能互发消息 |

续表2-12 好友聊天用例描述表

|  |  |
| --- | --- |
| 主要流程 | 1、用户点击聊天，进入聊天页面 |
|  | 2、用户选择其中好友 |
|  | 3、用户和对方可以互发送文字、表情、图片和文件 |
|  | 4、对方收到用户发送的消息息 |
| 实现约束 | 对方已删除好友 |
| 结束 | 不进行聊天 |

### 2.4.4个人中心模块用例描述

个人中心模块主要功能有查看个人信息、修改个人信息、查看校内信息(课程表、成绩、考试信息)。用户登录后点击头像，进入个人中心界面，显示个人信息，可进行修改，点击校内信息可查看课程表、成绩、考试信息。

查看个人信息的用例描述表如表2-13所示。

表2-13 查看个人信息用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户查看个人信息 |
| 前置条件 | 用户已登录 |
| 后置条件 | 用户需进入个人中心界面 |
| 主要流程 | 1、用户点击头像进入个人中心页面 |
|  | 2、用户个人信息展示 |
|  | 3、用户查看个人信息 |
| 实现约束 | 用户未登录 |
| 结束 | 用户查看到个人信息 |

修改个人信息的用例描述表如表2-14所示。

表2-14 修改个人信息用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |

续表2-14 修改个人信息用例描述表

|  |  |
| --- | --- |
| 参与者 | 用户 |
| 描述 | 用户修改个人信息 |
| 前置条件 | 用户已登录 |
| 后置条件 | 用户修改个人信息成功 |
| 主要流程 | 1、用户点击头像进入个人中心页面 |
|  | 2、用户点击修改信息 |
|  | 3、用户进行修改，保存 |
|  | 4、提示修改信息成功 |
| 实现约束 | 用户未登录 |
| 结束 | 用户修改个人信息成功 |

查看校内信息的用例描述表如表2-15所示。

表2-15 查看校内信息用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 用户 |
| 描述 | 用户查看校内信息 |
| 前置条件 | 用户已登录 |
| 后置条件 | 校内信息在页面显示 |
| 主要流程 | 1、用户点击校内信息，进入校内信息界面 |
|  | 2、用户选择查看某一校内信息(课程表、成绩、考试信息) |
| 实现约束 | 用户未登录 |
| 结束 | 用户查看到校内信息 |

### 2.4.5 管理员模块用例描述

管理员模块主要功能有发布通知、管理用户、查看数据分析图。用户登录后，若用户为管理员，就可通过菜单栏点击管理员权限进入管理员页面。管理员可以发布通知，可查看所有用户，进行用户管理(授予管理员权限、删除用户)，也可查看数据分析图(用户分布图、学校数据柱状图、在线人数折线图)。

管理员模块用例图如图2-16所示。

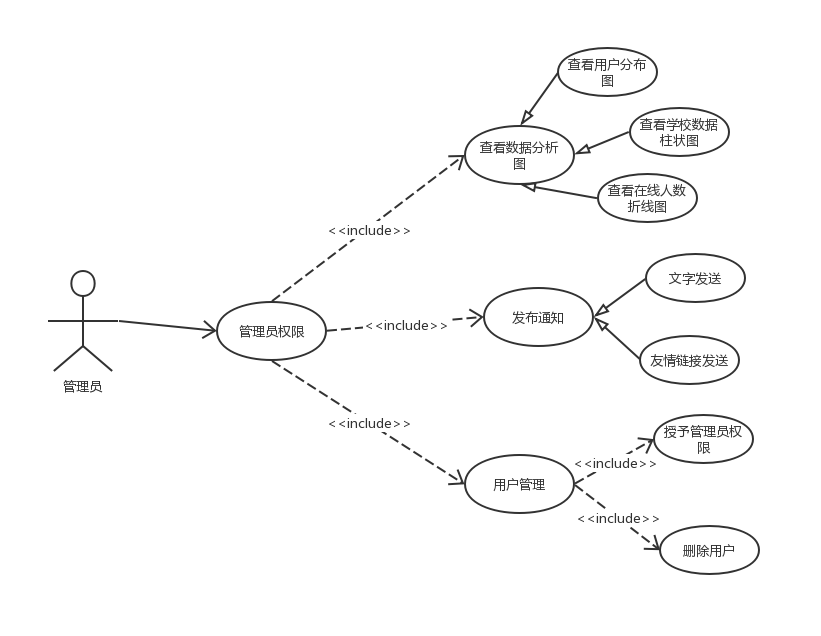


表2-16 管理员模块用例图

发布通知的用例描述表如表2-17所示。

表2-17 发布通知用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 管理员 |
| 描述 | 管理员发布通知 |
| 前置条件 | 用户为管理员 |
| 后置条件 | 管理员发布通知成功 |
| 主要流程 | 1、用户为管理员，点击管理员权限 |
|  | 2、管理员进入管理员页面，点击发布通知 |
|  | 3、管理员输入标题和内容 |
|  | 4、点击提交，通知发布成功 |
| 实现约束 | 用户不为管理员 |
| 结束 | 管理员发布通知成功 |

管理用户的用例描述表如表2-18所示。

表2-18 管理用户用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 管理员 |
| 描述 | 管理员管理用户(授予管理员权限、删除用户) |
| 前置条件 | 用户为管理员 |
| 后置条件 | 管理员授予权限/删除用户成功 |
| 主要流程 | 1、用户为管理员，点击管理员权限 |
|  | 2、管理员进入管理员页面，点击授予权限 |
|  | 3、管理员选择授予权限或删除的用户 |
|  | 4、点击按钮，授予权限/删除用户成功 |
| 实现约束 | 用户不为管理员 |
| 结束 | 管理员授予权限/删除用户成功 |

查看数据分析图的用例描述表如表2-19所示。

表2-19 查看数据分析图用例描述表

|  |  |
| --- | --- |
| 用例条例 | 描述 |
| 参与者 | 管理员 |
| 描述 | 管理员查看数据分析图 |
| 前置条件 | 用户为管理员 |
| 后置条件 | 数据分析图在页面展示 |
| 主要流程 | 1、用户为管理员，点击管理员权限 |
|  | 2、管理员进入管理员页面，点击数据分析 |
|  | 3、用户分析图、学校数据柱状图、在线人数折线图分别进行展示 |
| 实现约束 | 用户不为管理员 |
| 结束 | 数据分析图在页面展示 |

## 2.5 非功能性需求

在需求分析时，功能性需求是人们普遍关注的，但也不能忽视非功能性需求的分析，因为它所涉及的方面比较广泛。非功能性需求也可以说是“软件开发的界限”。还有其他限制和限制其他可能解决方案的要求。软件的非功能性需求主要是软件系统的性能、可靠性、运行限制等多个方面。

### 2.5.1环境需求

软件方面和硬件方面需求如表2-20所示。

表2-20运行环境需求表

|  |  |
| --- | --- |
| 内容 | 环境需求 |
| 服务器 | Express |
| 编译软件 | Visual Studio Code |
| 数据库 | MySQL |
| 操作系统 | windows系统 |
| 运行内存 | 1GB以上 |

### 2.5.2完整性需求

整个网站的良好开发代码的情况下取决于网站的完整性，当我们需要添加某功能的情况下我们不需要再修改源代码，就能完成该功能的实现，这样就能体现了系统的完整性。这样为后期的维护和修改功能节约了更多的时间，提高效率。

### 2.5.3 适应性需求

校园社交网站基本是由javascript语言来实现的，依靠vue.js和node.js这两大框架，编译器我选择的是前端开发人员常用的Visual Studio Code，VS Code利用各种插件来支持各种功能，完全适用vue.js和node.js来开发校园社交网站，网站采用兼容性最好的Chrome浏览器，能在各浏览器上进行完美运行，让用户有更好的使用体验，大大提高了适应性。

## 2.6本章小结

本章通过系统用例图和用例描述表相结合，对该系统的功能进行了全面的了解，又对该功能的非功能性需求进行了分析，其中包括环境需求、完整性需求、适应性需求，让我们对此系统更加熟悉与了解。

# 第三章 校园社交网站系统设计

## 3.1 系统设计原则

本系统是为了满足用户可以通过校园进行社交所开发的，用户可以通过校园社交网站结识志同道合的朋友，也可进入校内服务进行查看信息，老师可通过此系统可以向学生发送学习类文件，发布学习任务等等，真正做到了学习社交两不误。通过系统数据分析图的展示，管理员可以更加了解用户，更好的操作管理。

## 3.2 系统总体架构设计

校园社交网站由动态管理、好友管理、通知管理、聊天管理、用户管理为主要功能来设计，通过这些功能点具体实现了一个网站应具备的基本条件特点。

系统架构图如下图3-1所示：

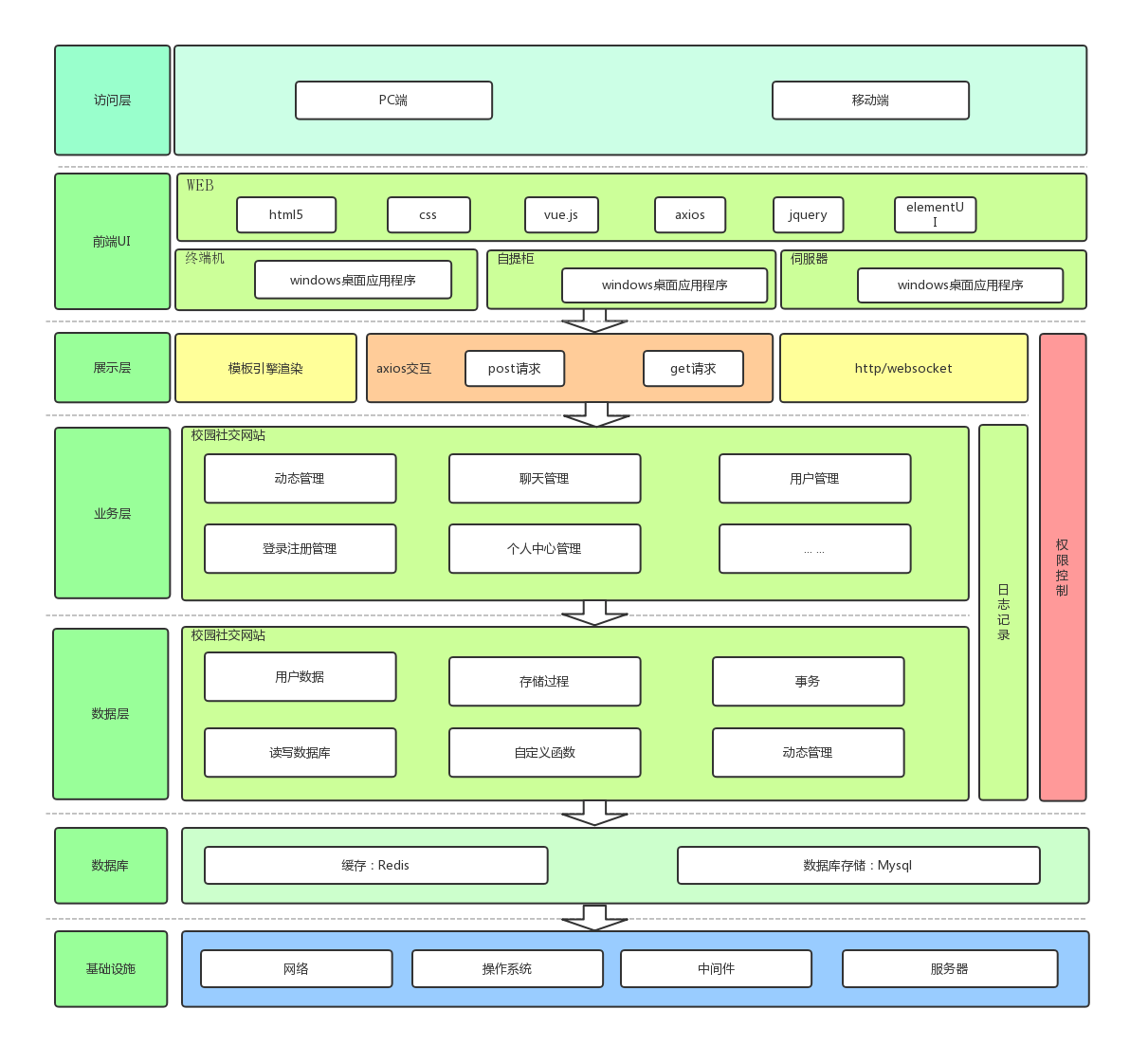


图3-1 系统架构图

## 3.3 系统功能模块结构设计

校园社交网站详细的描述了用户、管理员的各种功能。 本系统由用户登录，用户若为管理员，会具有管理员权限功能。本系统分为以下五个功能模块：登录注册功能模块、动态功能模块、好友功能模块、个人中心功能模块、管理员功能模块。管理员可以对以上的所有功能模块进行操作，但用户不能操作管理员功能模块。

该网站的各项功能模块内容分析，下图为校园社交网站的总功能结构图，如图3-2所示：

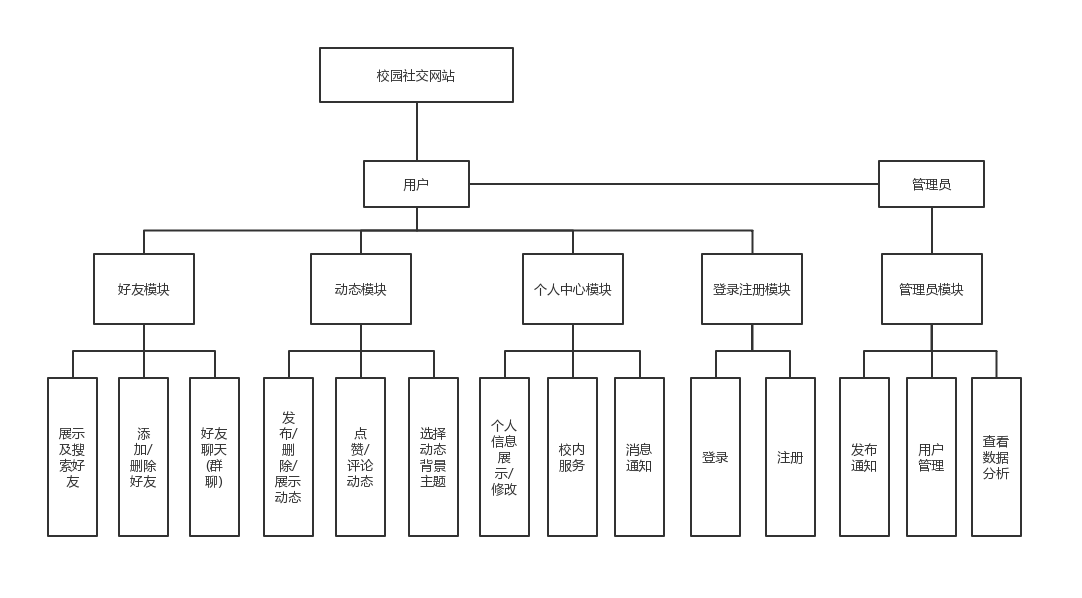


图3-2 系统总体功能结构图

## 3.4 系统类图设计

该系统类图将它分为九个类，分别是用户类、管理员类、动态类、点赞类、评论类、通知类、文件类、聊天类、群组类。用户类与管理员类是泛化关系，管理员是用户的一种，用户的有的功能，管理员也有。用户类与文件类、聊天类、群组类以及动态类是一种关联关系。动态类与点赞类以及评论类是一种组合关系，它们是整体和部分的关系，点赞类和评论类不能离开动态类而单独存在。下图为系统类图，如图3-3所示：

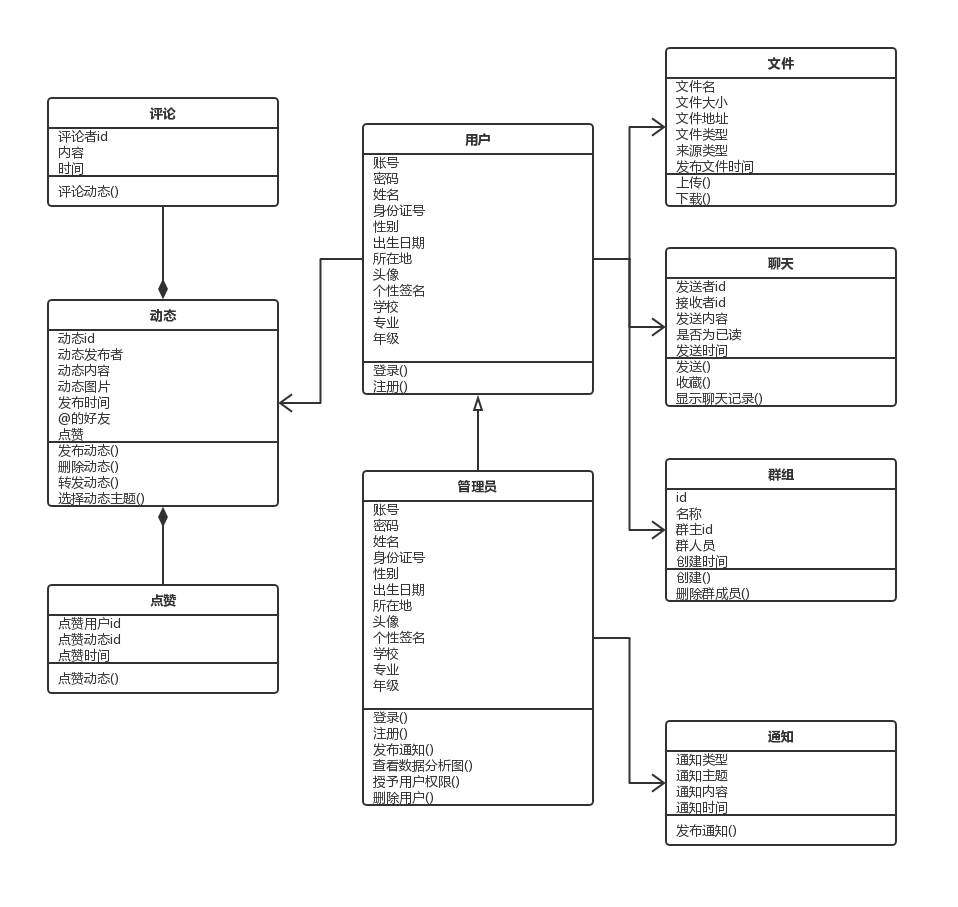


图3-3 系统类图

## 3.5 数据库设计

数据库的设计是为了在指定的应用环境中建立一个最高的数据库模型，并建立一个数据库，有效地存储应用系统，满足不同用户对应用程序的需求。在数据库领域内，常常把使用数据库的各类系统统称为[数据库应用系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E5%BA%94%E7%94%A8%E7%B3%BB%E7%BB%9F/10440620)[[[12]](#endnote-12)]。

### 3.5.1 数据库概念模型设计

从需求分析中获得的需求，先抽象为信息世界的一个结构，然后通过数据库管理系统进行更准确的修改和实现。

概念模型的主要特点：

1. 整个真实世界，包括不同事物之间的关系，以及用户在使用真实世界模型进行数据处理时的需求。

2. 容易理解并与用户交换意见，用户的积极参与是成功打造数据库的关键。

3. 如果应用环境和应用程序的需要发生变化，概念模型可以很容易地修改和扩大。

4. 易于向关系、网状、层次等各种数据模型转换[[[13]](#endnote-13)]。

概念模型是各种数据模型的共同基础，与数据模型相比，这种模型具有机器的可读性、抽象性和稳定性。

系统整体E-R图如图3-4所示。

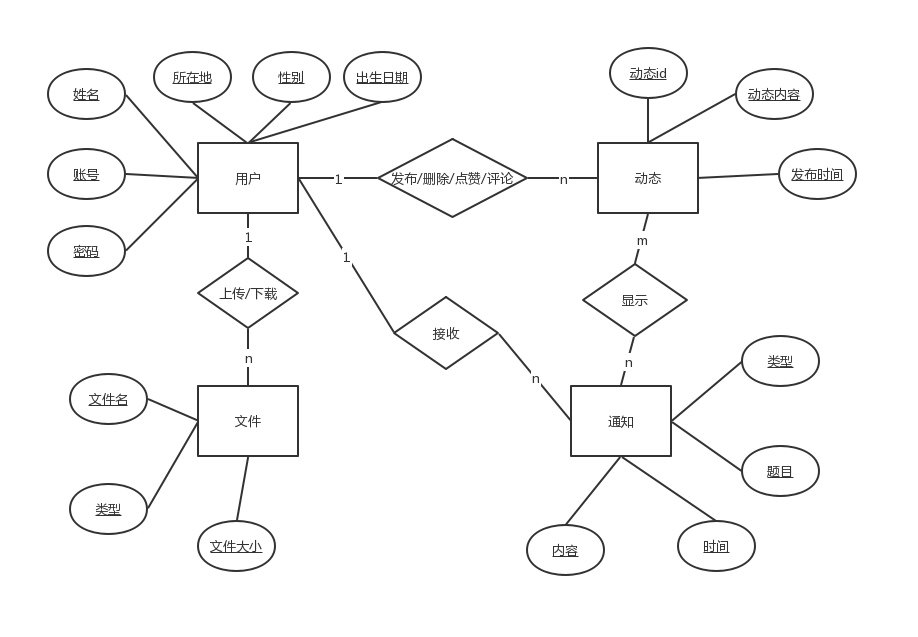


图3-4 系统总体E-R图

（1）用户实体的属性包括账号、密码、姓名、性别、出生日期、所在地、头像、个性签名、是否为管理员、学校、专业、年级、身份证。如图3-5所示。

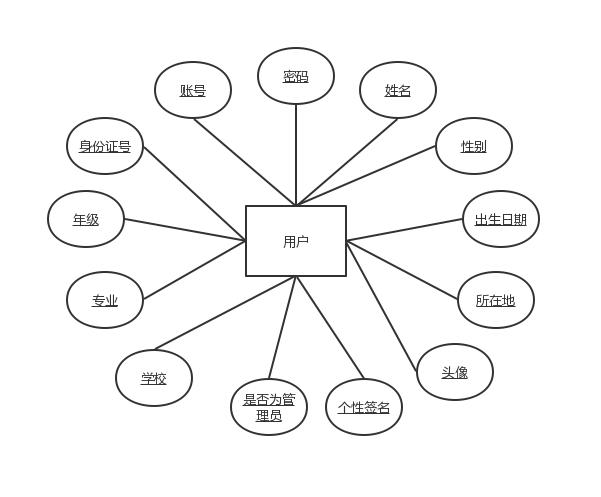


图3-5 用户实体属性图

（2）动态实体的属性包括动态id、动态内容、动态图片、点赞、@的好友、是否为转发、发布时间。如图3-6所示。

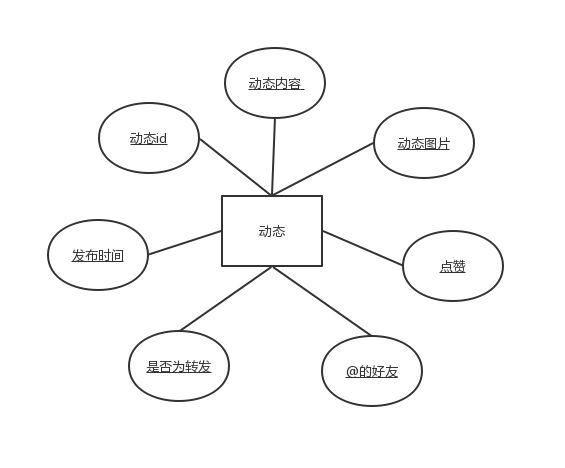


图3-6动态实体属性图

（3）文件实体的属性包括文件名、文件大小、文件存放地址、文件类型、文件发布时间。如图3-7所示。

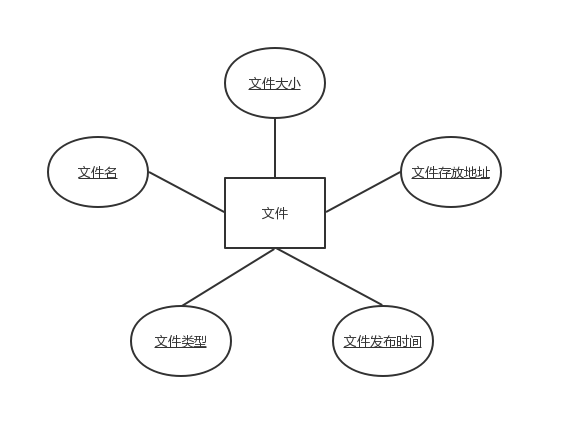


图3-7文件实体属性图

（4）通知实体的属性包括题目、内容、类型、时间。如图3-8所示。

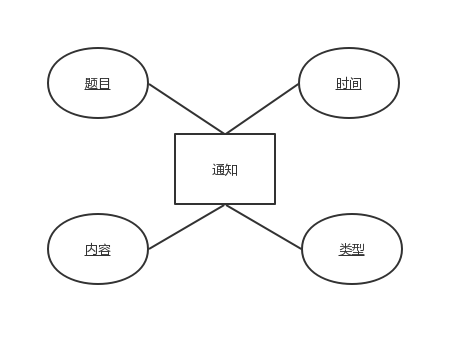


图3-8通知实体属性图

（5）聊天记录实体的属性包括接收者id、发送者id、发送时间、发送内容、是否为已读。如图3-9所示。

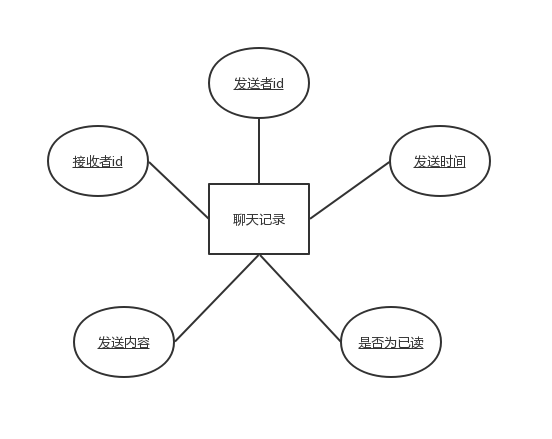


图3-9聊天记录实体属性图

（6）群组的属性包括群id、群名称、群人员、群主id、创建群时间。如图3-10所示。

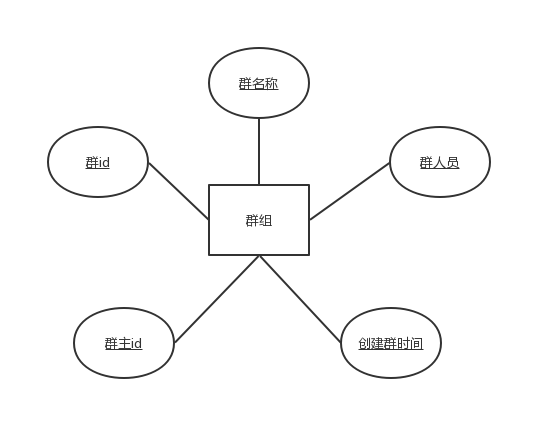


图3-10群组实体属性图

（7）动态评论实体的属性包括动态id、评论者id、评论内容、动态发布者id、评论时间。如图3-11所示。

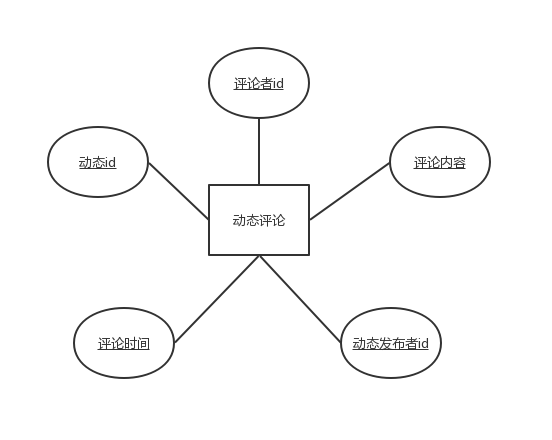


图3-11动态评论实体属性图

### 3.5.2 数据库逻辑结构设计

MySQL数据库中的用户person表，其中用户账号是主键，用来标识一条唯一记录。其他字段包括密码、姓名、身份证号、性别、出生日期、所在地、头像、个性签名、学校、专业、年级、登录时间、是否为管理员、动态背景主题url。

用户person表如表3-12所示：

表3-12 person用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| person\_account | varchar | 255 | 否 | 是 | 用户账号 |
| person\_password | varchar | 255 | 否 | 否 | 用户密码 |
| person\_name | varchar | 255 | 否 | 否 | 用户姓名 |
| person\_id\_card | varchar | 14 | 否 | 否 | 用户身份证号 |
| person\_sex | varchar | 2 | 否 | 否 | 用户性别 |
| person\_birthday | date |  | 否 | 否 | 用户出生日期 |
| person\_place | varchar | 255 | 否 | 否 | 用户所在地 |
| person\_picture | varchar | 255 | 否 | 否 | 用户头像 |

续表3-12 person用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| person\_signature | varchar | 255 | 否 | 否 | 用户个性签名 |
| person\_school | varchar | 255 | 否 | 否 | 学校 |
| person\_professional | varchar | 255 | 否 | 否 | 专业 |
| person\_grade | varchar | 4 | 否 | 否 | 年级 |
| person\_time | varchar |  | 否 | 否 | 登录时间 |
| person\_ismanage | varchar | 1 | 否 | 否 | 是否为管理员 |
| blog\_theme | varchar | 255 | 否 | 否 | 动态背景url |

MySQL数据库中的动态blog表，其中动态id是主键，用来标识一条唯一记录。其他字段包括动态发布者、动态内容、动态图片、发布时间、@的好友、点赞、是否为转发、动态原创作者。

动态blog表如表3-13所示：

表3-13 blog动态信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| blog\_id | varchar | 32 | 否 | 是 | 动态id |
| blog\_sender | varchar | 255 | 否 | 否 | 动态发布者 |
| blog\_content | varchar | 10000 | 否 | 否 | 动态内容 |
| blog\_picture | varchar | 500 | 否 | 否 | 动态图片 |
| blog\_time | timestamp |  | 否 | 否 | 发布时间 |
| blog\_friend | varchar | 500 | 是 | 否 | @的好友 |
| blog\_like | varchar | 500 | 是 | 否 | 点赞 |
| is\_blog\_forwarding | tinyint | 1 | 是 | 否 | 是否是转发 |
| blog\_original | varchar | 255 | 是 | 否 | 动态原创作者 |

MySQL数据库中的聊天记录chat表，其中字段包括发送者、接收者、发送内容、是否为已读、发送时间。

聊天记录chat信息数据库表如表3-14所示：

表3-14 chat聊天记录信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| from\_person\_id | varchar | 255 | 否 | 否 | 发送者 |

续表3-14 chat聊天记录信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| to\_person\_id | varchar | 255 | 否 | 否 | 接收者 |
| message | varchar | 10000 | 否 | 否 | 发送内容 |
| status | int | 1 | 否 | 否 | 是否为已读 |
| send\_time | datetime |  | 否 | 否 | 发送时间 |

MySQL数据库中的群组chatgroup表，其中群id是主键，用来标识一条唯一记录。其中字段包括群名称、群人员、群主id、创建群时间。

chatgroup群组信息数据库表如表3-15所示：

表3-15 chatgroup群组信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| group\_id | varchar | 255 | 否 | 是 | 群id |
| group\_name | varchar | 255 | 否 | 否 | 群名称 |
| group\_person | varchar | 10000 | 否 | 否 | 群人员 |
| group\_creater | varchar | 255 | 否 | 否 | 群主id |
| group\_time | datetime |  | 否 | 否 | 创建群时间 |

MySQL数据库中的群聊收藏collection表，其中字段包括收藏者id、收藏内容、来源者id、来源群名、时间。

collection群聊收藏信息数据库表如表3-16所示：

表3-16 collection群聊收藏信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| collection\_person\_id | varchar | 255 | 否 | 否 | 收藏者id |
| collection\_content | varchar | 10000 | 否 | 否 | 收藏内容 |
| collection\_from\_person\_id | varchar | 255 | 是 | 否 | 来源者id |
| collection\_from\_group\_name | varchar | 255 | 是 | 否 | 来源群名 |
| collection\_time | datetime |  | 否 | 否 | 时间 |

MySQL数据库中的文件file表，其中字段包括发布者id、发布者头像、接收者id、来源类型(来自群、来自好友)、文件所在群id、文件名、文件大小、文件地址、文件类型、发布文件时间。

文件file信息数据库表如表3-17所示

表3-17 file文件信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| file\_from\_person\_id | varchar | 255 | 否 | 否 | 发布者 |
| file\_from\_person\_picture | varchar | 255 | 否 | 否 | 发布者头像 |
| file\_person\_id | varchar | 255 | 是 | 否 | 接收者id |
| from\_type | varchar | 255 | 否 | 否 | 来源类型 |
| file\_group\_id | varchar | 255 | 是 | 否 | 文件所在群id |
| file\_name | varchar | 255 | 否 | 否 | 文件名 |
| file\_size | varchar | 255 | 否 | 否 | 文件大小 |
| file\_url | varchar | 20 | 否 | 否 | 文件地址 |
| file\_type | varchar | 255 | 是 | 否 | 文件类型 |
| send\_file\_time | varchar | 255 | 否 | 否 | 发布文件时间 |

MySQL数据库中的好友friends表，其中用户id是主键，用来标识一条唯一记录。其中字段包括好友列表、发送好友申请列表、接收好友申请列表。

好友friends信息数据库表如表3-18所示：

表3-18 friends好友信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| person\_id | varchar | 255 | 否 | 是 | 订单号 |
| friend\_list | varchar | 7000 | 是 | 否 | 订单编号 |
| friend\_apply | varchar | 7000 | 是 | 否 | 订单日期 |
| friend\_accept | varchar | 7000 | 是 | 否 | 收件人 |

MySQL数据库中的群聊天记录groupchatrecords表，其中字段包括发送者id、群id、发送内容、发送者头像url、发送时间。

群聊天记录groupchatrecords信息数据库表如表3-19所示

表3-19 groupchatrecord群聊天记录信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| from\_person\_id | varchar | 255 | 否 | 否 | 发送者id |
| to\_group\_id | varchar | 255 | 否 | 否 | 群id |

续表3-19 groupchatrecord群聊天记录信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| message | varchar | 10000 | 是 | 否 | 发送内容 |
| from\_person\_picture | varchar | 255 | 否 | 否 | 发送者头像url |
| send\_time | datetime | 255 | 否 | 否 | 发送时间 |

MySQL数据库中的点赞likes表，其中字段包括点赞动态id、点赞动态发布者、点赞用户、点赞时间。

点赞likes信息数据库表如表3-20所示

表3-20 likes点赞信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| blog\_id | varchar | 32 | 否 | 否 | 点赞动态id |
| blog\_sender | varchar | 255 | 否 | 否 | 点赞动态发布者 |
| like\_account | varchar | 255 | 否 | 否 | 点赞用户 |
| blog\_time | timestamp |  | 否 | 否 | 点赞时间 |

MySQL数据库中的通知notice表，其中字段包括点赞用户账号、通知类型、通知时间、通知主题、通知内容。

通知notice信息数据库表如表3-21所示

表3-21 notice点赞信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| person\_id | varchar | 255 | 是 | 否 | 用户账号 |
| notice\_type | varchar | 255 | 否 | 否 | 通知类型 |
| notice\_time | timestamp |  | 否 | 否 | 通知时间 |
| notice\_title | varchar | 100 | 否 | 否 | 通知主题 |
| notice\_content | varchar | 20000 | 是 | 否 | 通知内容 |

MySQL数据库中的评论remark表，其中字段包括动态id、评论者id、评论内容、动态发布者id、评论时间。

评论remark信息数据库表如表3-22所示

表3-22 remark评论信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 空 | 主键 | 注释 |
| blog\_id | varchar | 32 | 否 | 否 | 动态id |

续表3-22 remark评论信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| remark\_account | varchar | 255 | 否 | 否 | 评论者 |
| remark\_content | varchar | 5000 | 否 | 否 | 评论内容 |
| blog\_sender | varchar | 255 | 否 | 否 | 动态发布者 |
| remark\_time | timestamp | 255 | 否 | 否 | 评论时间 |

## 3.6 系统界面设计

页面采用了多色布局，让用户在页面上功能辨识度更强。个性化设计建立品牌认知度，大圆角、精致的图标让设计更有品质。采用了组件化设计，保持了页面的统一性和可复用性，投影融入设计中明确表达层级关系，准确的产品结构和层级关系，卡片式设计、模块差异化设计，使结构清晰、功能明确。

## 3.7 本章小结

本章介绍了校园社交网站的系统总体功能数据库概念模型设计，并且展示了此系统的系统总体功能结构图和系统总体E-R图以及实体属性图，又列出MySQL数据库中的所有的信息表，阐述了此系统的界面设计，对此系统更加充分的了解。

# 第四章 系统详细设计与实现

本系统的主要框架为Vue.js框架和Node.js框架，前端运用Vue.js框架，它提供了 MVVM 风格的双向数据绑定的 Javascript 库，专注于View 层。后端使用Node.js是因为前端界面更加了解需要的数据，每一个前端页面对应一些后端接口，每个接口需要前端界面来传递。再加上实际业务开发中，前端页面需求经常会发生变化，需要修改字段或者数据结构，所以对接页面的这部分接口由前端直接开发非常合适，可以显著的减少沟通成本。

主要界面在本系统中含有：系统首页，用户登录/注册，动态页面，选择动态背景主题页面，通知页面，聊天页面，管理员权限等。

## 4.1 用系统开发与运行环境

系统开发与运行环境表如表4-1所示：

表4-1 系统开发与运行环境表

|  |  |
| --- | --- |
| 系统开发 | 内容 |
| 服务器 | Express |
| 编译软件 | Visual Studio Code |
| 操作系统 | Windows 10 |
| 数据库 | MySQL |
| 运行内存 | 1GB及以上 |
| 编程语言 | JavaScript |

## 4.2 发布动态功能详细设计与实现

本小节主要介绍发布动态功能详细设计与实现的界面详情、细节以及功能流程图。

### 4.2.1 界面详情

发布动态界面如图4-2所示：



图4-2 发布动态界面

### 4.2.2 实现细节

用户进入动态页面，点击左下角发布动态图标，用户输入动态内容和动态图片，动态也可输入表情、友情链接以及@的好友，动态内容不得为空，前端将进行判断是否为空。输入完成后，点击发布按钮，前端向后台发送数据，后台将数据插入数据库中，如果插入失败，则向前端返回失败信息，如果插入成功，则向前端返回提示发布成功。

### 4.2.3 功能流程

此功能流程为发布动态功能流程图，如图4-3所示：

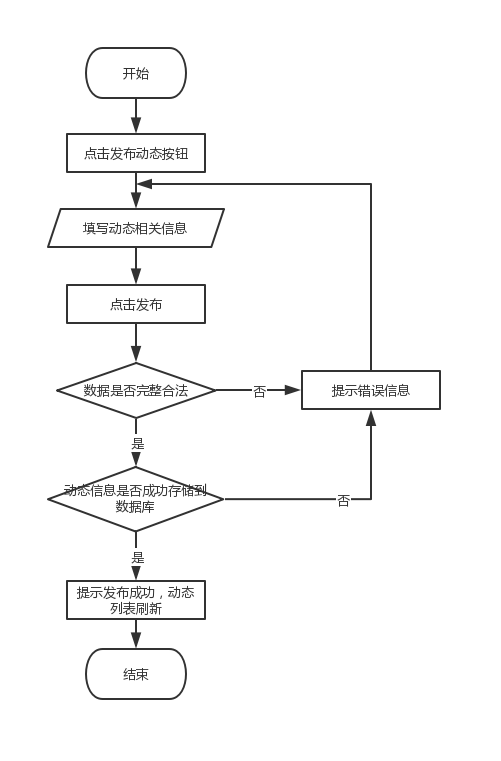


图4-3 发布动态功能流程图

## 4.3 选择动态背景主题功能详细设计与实现

本小节主要介绍选择动态背景主题功能详细设计与实现的界面、细节以及功能流程图。

### 4.3.1 界面详情

选择动态背景主题界面如图4-4所示：

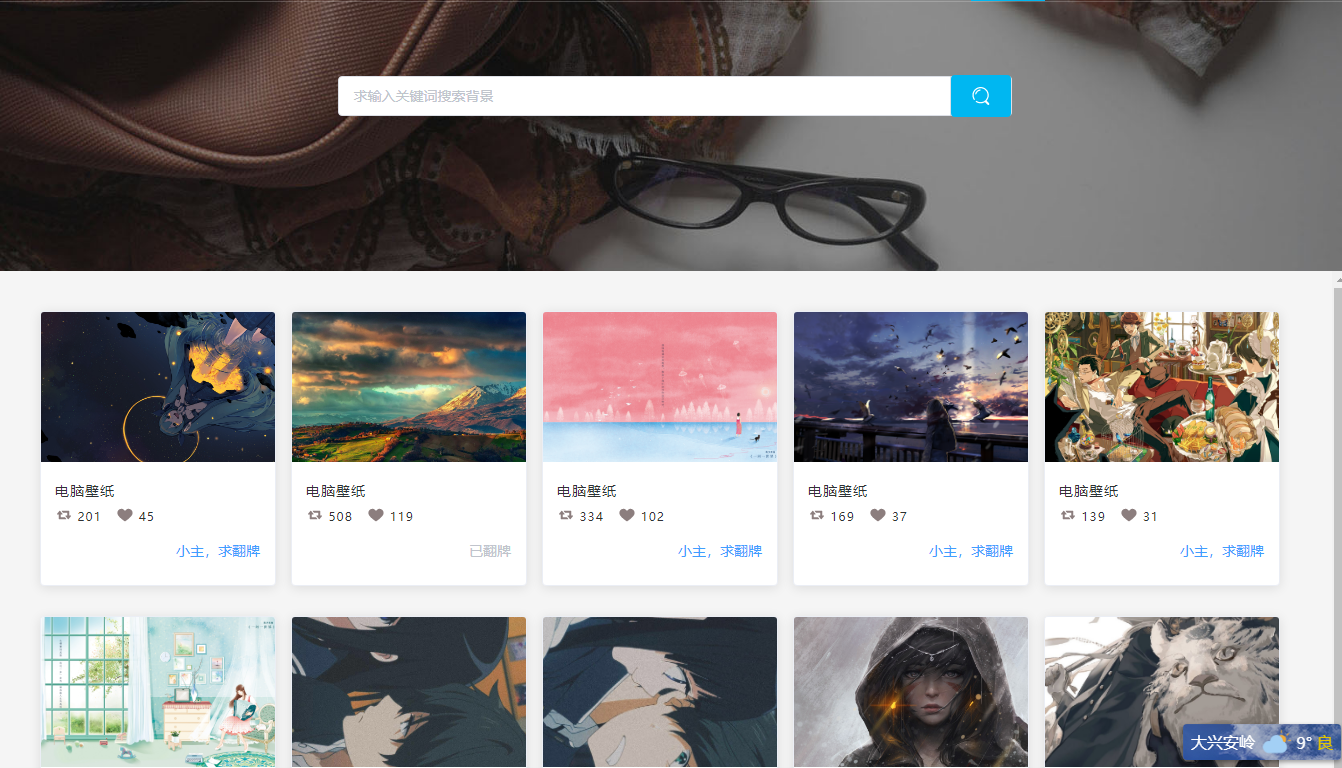


图4-4 选择动态背景主题界面

### 4.3.2 实现细节

用户点击动态界面中的动态主题，进入动态主题页面，该界面是通过Web爬虫实现，爬取花瓣网上的优质图片。

该界面首先判断哪一个图片是用户现在选择的，则选择按钮为禁选状态，则其他图片的选择按钮不为禁选状态。同时该界面的图片列表默认搜索电脑壁纸背景图片，同时用户也可通过输入框搜索自己喜欢的主题背景图片，然后在列表中选择背景图片，然后点击选择按钮进行选择图片，前端向后台传递主题图片的相关信息，后台将数据保存到数据库个人表中。若保存失败，则返回给前端报错信息。若保存成功，则返回选择背景主题图片成功信息提示，并返回跳转到动态页面中。

### 4.3.3 功能流程

此功能流程为选择动态背景主题功能流程图，如图4-5所示：

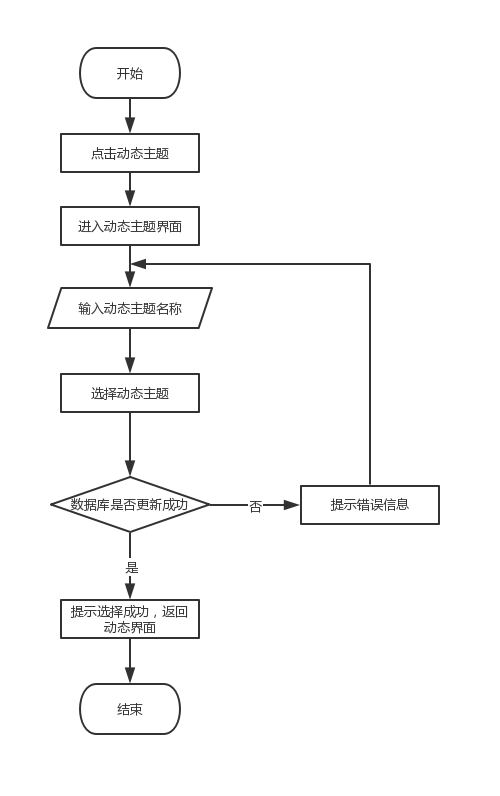


图4-5 选择动态背景主题功能流程图

## 4.4 添加好友功能详细设计与实现

本小节主要介绍添加好友功能的界面、细节以及功能流程图。

### 4.4.1 界面详情

添加好友界面如图4-6所示：

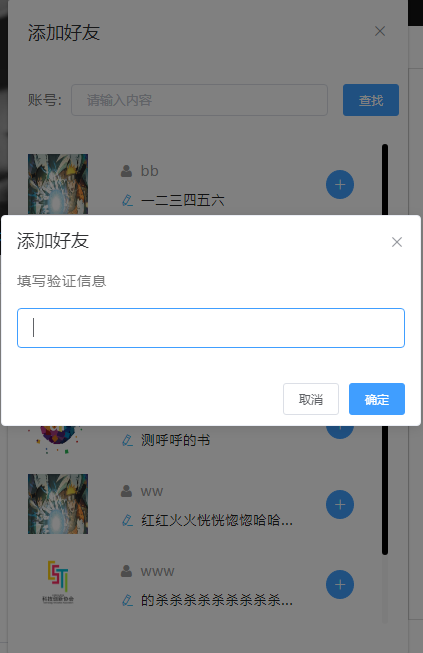


图4-6 添加好友界面

### 4.4.2 实现细节

此功能运用websocket技术，用户点击添加好友小图标，用户可通过输入框输入对方账号进行搜索，从列表中选择要添加的好友，点击加号按钮，然后列表中会有等待验证中的提示，同时更新后台数据库表中的信息。对方界面会立即出现请求添加好友验证信息弹窗，对方可点击同意或拒绝按钮，然后通过websocket来发送验证信息，用户界面会立即收到弹窗，同时更新后台数据库中相应数据。

### 4.4.3 功能流程

此功能流程为添加好友功能流程图，如图4-7所示：

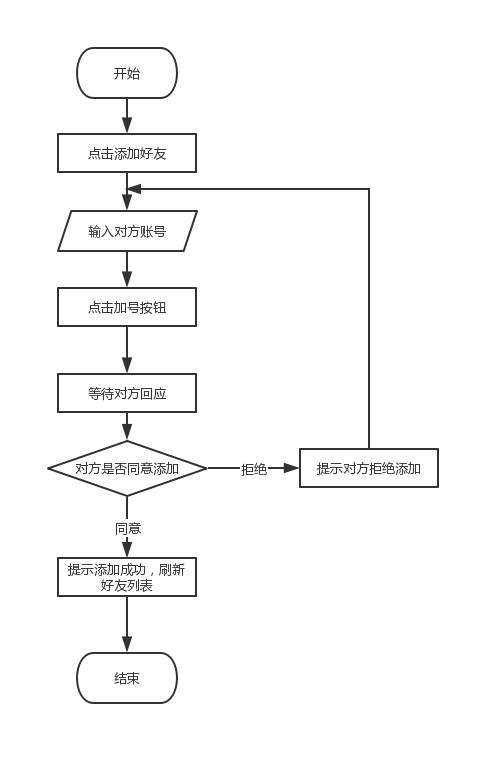


图4-7添加好友功能流程图

## 4.5 聊天功能详细设计与实现

本小节主要介绍聊天功能的界面、细节以及功能流程图。

### 4.5.1 界面详情

聊天界面如图4-8所示：

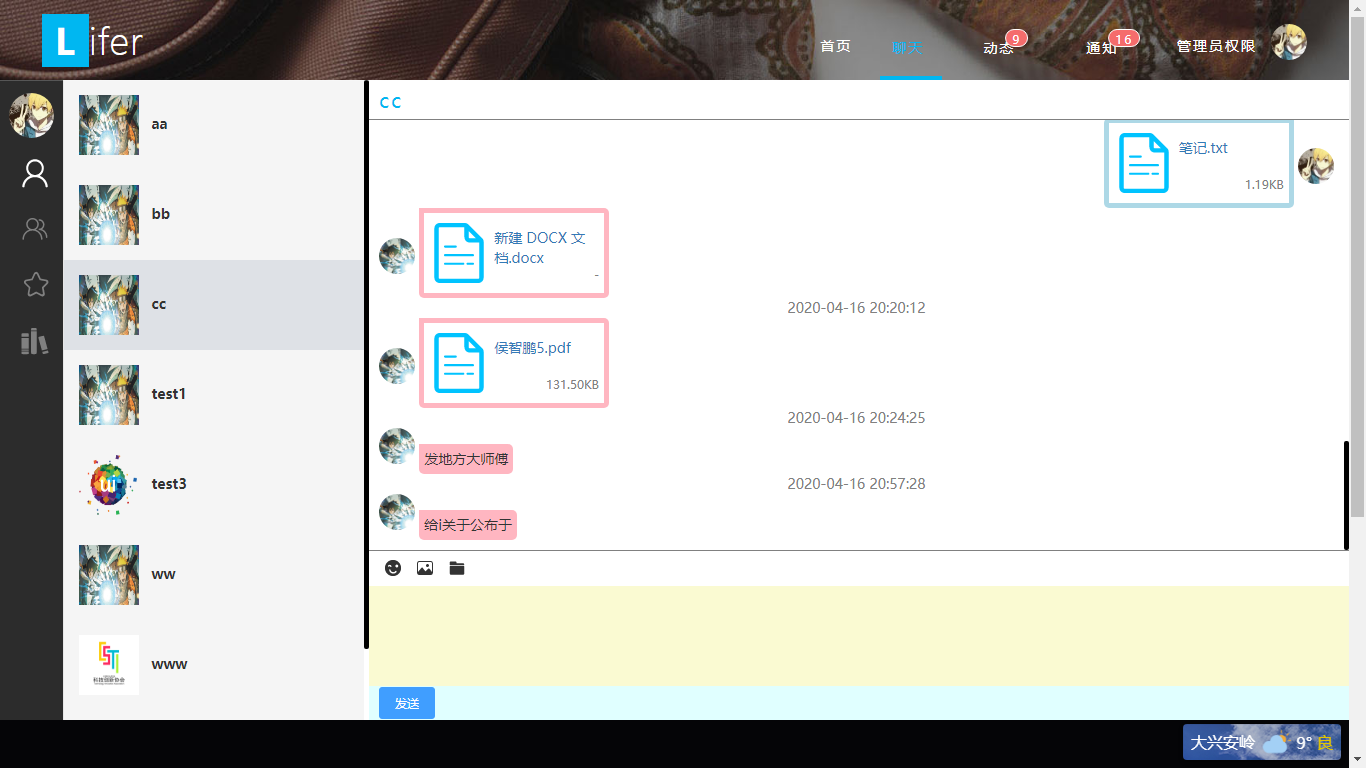


图4-8 聊天界面

### 4.5.2 实现细节

此功能运用websocket技术，前后端利用npm引入socket.io的包实现websocket。用户在聊天界面选择其中好友进行聊天。用户可发送文字、表情、图片或文件等。当输入内容点击发送按钮时，前端向websocket发送相应信息，后台查找用户是否已经登录，又判断用户是否在聊天界面上，若在聊天界面上，websocket直接将信息发送给对方，同时数据库新增一条聊天信息，并标记为已读(status:1)。若不在聊天界面上，数据库新增一条聊天信息的同时，将此信息标记为未读(status:0)。

若对方已经登录，websocket向对方发送提示信息，菜单栏上的聊天按钮显示小红点。对方聊天列表进行更新未读提示。若对方没有登录，则等到对方登录时，再更新对方聊天列表[[[14]](#endnote-14)]。

### 4.5.3 功能流程

此功能流程为聊天功能流程图，如图4-9所示：

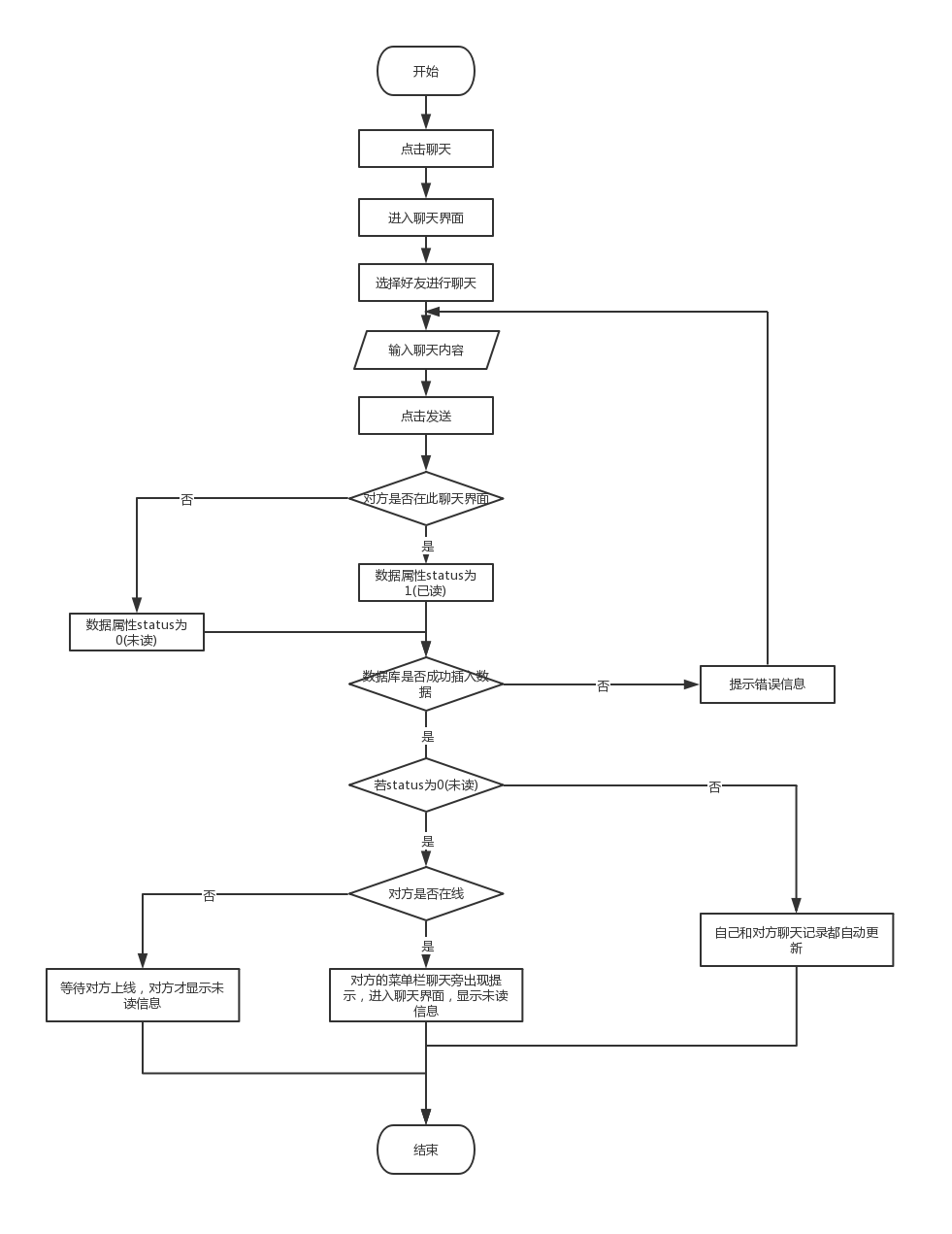


图4-9 聊天功能流程图

## 4.6 管理员授予用户权限/删除用户功能详细设计与实现

本小节主要介绍管理员授予用户权限/删除用户功能的界面、细节以及功能流程图。

### 4.6.1 界面详情

管理员授予用户权限/删除用户界面如图4-10所示：

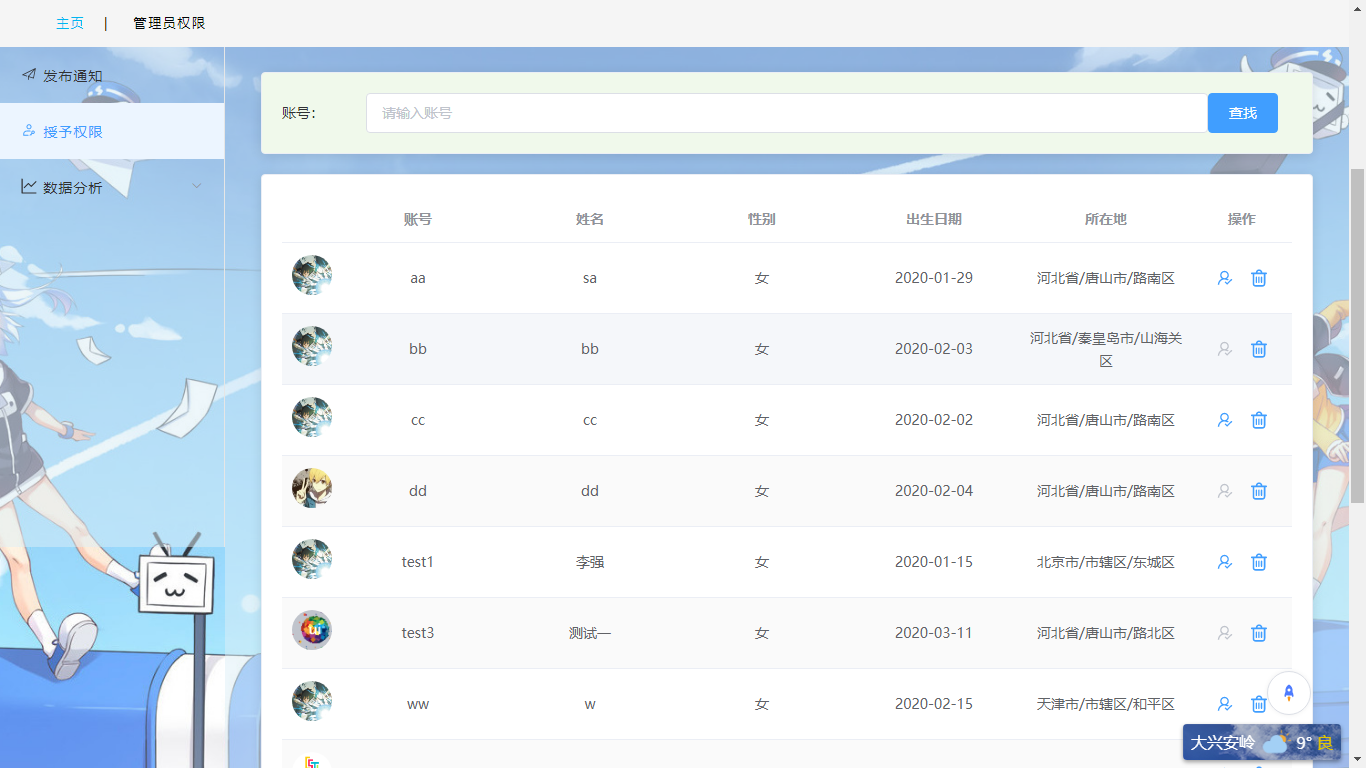


图4-10管理员授予用户权限/删除用户界面

### 4.6.2 实现细节

用户为管理员可通过点击管理员权限进入管理员界面。点击授予权限，出现所有用户相关信息列表，管理员可选择某一用户点击授予权限/删除按钮进行授予用户权限/删除用户。前端向后台发送相关用户信息，若为授予权限，后台数据库则进行相关数据更新。若为删除用户，后台数据库则进行删除相关数据。若数据库更新成功，后台向前端返回成功提示。若更新失败，后台向前端返回报错信息。

### 4.6.3 功能流程

此功能流程为管理员授予用户权限/删除用户功能流程图，详情如图4-11所示：

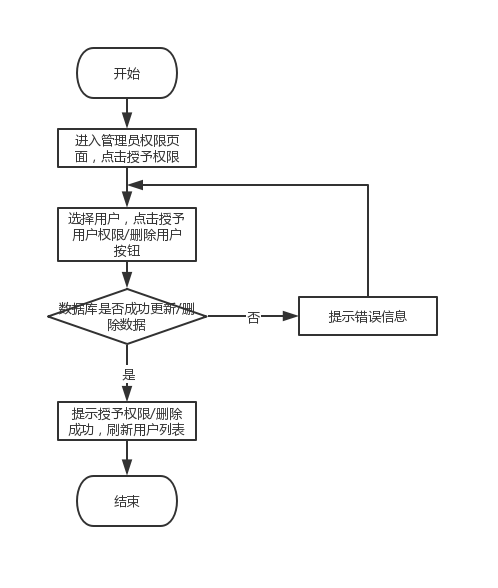


图4-11 管理员授予用户权限/删除用户功能流程图

## 4.7本章小结

本章描述了校园社交网站开发及运行环境以及部分功能详细设计，通过部分功能的流程图、相关界面和功能细节，对该系统有了更深入的了解。让用户更加熟悉对该系统的使用。

# 第五章 系统测试

## 5.1 系统测试的目的

软件测试是用来检测错误的过程。这是为了证明错误，但要证明没有错误。正确的例子是迄今尚未发现的错误。

系统测试的目的是找出校园社交网站中可能出现的bug，以便及时修改系统。测试各项功能是否达到预期效果，还会帮助发现当前所采用的软件过程的缺陷，以便改进。

## 5.2 系统测试的方法

软件测试方法的目的是测试软件程序的错误，软件是否满足设计要求，是否满足合同的技术要求，评估软件的质量，并为用户实现软件的基本测试过程，系统和软件主要包括静态和动态测试、功能测试、性能测试、黑盒测试和白盒测试。

该系统的测试方法我采用的是黑盒测试，黑盒测试把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，针对“软件界面”和“软件功能”进行测试，只检查功能是否符合需求规格说明书能正常使用[[[15]](#endnote-15)]。因此黑盒测试又叫功能测试或数据驱动测试。

## 5.3 系统测试用例

### 5.3.1 登录注册模块测试用例

用户点击登录进入登录页面，没有账号可点击左下角发送动态图标，出现发送动态界面弹窗，可进行测试登录注册模块。具体测试用例如表5-1 所示。

表5-1 登录注册模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | 用例名称 | 操作步骤 | 预期结果 | 实际结果 |
| 1 | 用户登录 | 用户进入登录页面，输入账号和密码，点击登录 | 登录成功，并跳转至首页，可进行界面上的所有功能 | 与预期结果一致 |

续表5-1 登录注册模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 用户注册 | 用户点击注册进入注册界面，按步骤填写信息，点击注册 | 数据库成功新增数据，注册成功，并自动登录跳转至首页，可进行界面上的所有功能 | 与预期结果一致 |

### 5.3.2 动态模块测试用例

用户点击动态进入动态页面，可进行测试动态模块，其中包括发布动态、删除动态、点赞动态、评论动态以及选择动态主题。具体测试用例如表5-2 所示。

表5-2 动态模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | 用例名称 | 操作步骤 | 预期结果 | 实际结果 |
| 1 | 发布动态 | 用户进入动态页面，点击发布动态图标。输入相关内容，点击发布 | 数据库成功新增数据，提示发布成功，并刷新动态列表 | 与预期结果一致 |
| 2 | 删除动态 | 用户进入动态页面，选择自己的动态点击删除图标 | 数据库成功删除数据，提示删除成功，并刷新动态列表 | 与预期结果一致 |
| 3 | 点赞动态 | 用户进入动态页面，选择动态点击点赞图标 | 数据库成功新增数据，提示点赞成功，并更新点赞列表 | 与预期结果一致 |
| 4 | 评论动态 | 用户进入动态页面，选择动态点击评论图标，输入评论内容，点击评论按钮 | 数据库成功新增数据，提示评论成功，并更新评论列表 | 与预期结果一致 |
| 5 | 选择动态主题 | 点击动态主题进入动态主题页面，输入主题名称，选择动态主题 | 数据库成功更新数据，提示选择成功，跳转会动态界面，背景更换成功 | 与预期结果一致 |

### 5.3.3 好友模块测试用例

用户点击动态进入动态页面，通过好友列表可进行测试好友模块，其中包括添加好友、查找好友、删除好友、好友聊天、收藏好友聊天记录、显示聊天文件、创建群聊以及删除群聊中群成员。具体测试用例如表5-3 所示。

表5-3 好友模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | 用例名称 | 操作步骤 | 预期结果 | 实际结果 |
| 1 | 添加好友 | 点击添加好友图标。输入验证内容，点击添加 | 等待对方同意或拒绝，提示添加成功或失败，若成功，刷新好友列表 | 与预期结果一致 |
| 2 | 查找好友 | 点击查找好友图标，输入查找的好友账号，点击搜索按钮 | 好友列表成功更新 | 与预期结果一致 |
| 3 | 删除好友 | 在好友列表中选择好友点击删除 | 数据库成功更新数据，提示删除成功，并更新好友列表，同时更新对方好友列表 | 与预期结果一致 |
| 4 | 好友聊天 | 用户进入聊天页面，选择好友进行聊天，输入内容或文件，点击发送按钮 | 对方成功接收到信息  数据库成功新增数据，并更新聊天记录 | 与预期结果一致 |
| 5 | 收藏好友聊天记录 | 用户进入聊天页面，选择好友显示聊天记录，选择聊天内容点击收藏 | 数据库成功新增数据，提示收藏成功，进入收藏界面，列表显示收藏记录 | 与预期结果一致 |
| 6 | 显示聊天文件 | 用户与好友之间进行聊天，发送文件 | 数据库成功新增数据，进入文件界面，列表显示所有发送过的文件 | 与预期结果一致 |
| 7 | 创建群聊 | 点击群聊，进入群聊界面，点击创建群聊，输入群名和选择群成员，点击创建 | 数据库成功新增数据，更新群聊列表 | 与预期结果一致 |

续表5-3 好友模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | 删除群聊中群成员 | 点击群聊，进入群聊界面，选择群聊，此群必须为自己创建，选择群成员点击删除 | 数据库成功更新数据，更新群成员列表 | 与预期结果一致 |

### 5.3.4 个人中心模块测试用例

用户点击头像进入个人中心页面，可进行测试个人中心模块，其中包括查看个人信息、修改个人信息以及查看校内服务。具体测试用例如表5-4 所示。

表5-4 个人中心模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | 用例名称 | 操作步骤 | 预期结果 | 实际结果 |
| 1 | 查看个人信息 | 用户点击头像，进入个人中心界面，查看个人信息 | 成功显示个人信息 | 与预期结果一致 |
| 2 | 修改个人信息 | 用户点击头像，进入个人中心界面，点击修改按钮，修改信息后，点击保存按钮 | 数据库成功更新数据，提示修改成功，并刷新个人信息 | 与预期结果一致 |
| 3 | 查看校内服务 | 用户点击头像，进入个人中心界面，点击校内服务，查看校内信息 | 成功显示校内信息 | 与预期结果一致 |

### 5.3.5 管理员模块测试用例

管理员点击管理员权限进入管理员页面，可进行测试管理员模块，其中包括发布通知、授予用户权限、删除用户、显示用户分布图、显示学校数据柱状图以及显示在线人数折线图。具体测试用例如表5-5 所示。

表5-5 管理员模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | 用例名称 | 操作步骤 | 预期结果 | 实际结果 |
| 1 | 发布通知 | 点击发布通知。输入相关标题和内容，点击发布 | 数据库成功新增数据，提示发布成功，并刷新通知列表 | 与预期结果一致 |

续表5-5 管理员模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 授予用户权限 | 点击授予权限，选择用户点击授权图标 | 数据库成功更新数据，提示授权成功，用户菜单栏出现管理员权限，可点击进入管理员界面 | 与预期结果一致 |
| 3 | 删除用户 | 点击授予权限，选择用户点击删除图标 | 数据库成功删除数据，提示删除成功，并更新用户列表 | 与预期结果一致 |
| 4 | 显示用户分布图 | 点击用户分布图 | 界面成功显示用户分布图 | 与预期结果一致 |
| 5 | 显示学校数据柱状图 | 点击学校数据柱状图 | 界面成功显示学校数据柱状图 | 与预期结果一致 |
| 6 | 显示在线人数折线图 | 点击在线人数折线图 | 界面成功显示在线人数折线图 | 与预期结果一致 |

## 5.4 本章小结

本章介绍了该系统测试的目的、系统测试的方法以及系统测试的用例。从而提高该系统的稳定性与准确性。经过测试，整体的结果与预期基本一致，性能上也可以达到要求。

# 结论

随着互联网的普及,社交已经渗透到人们生活中,然而在校园中，社交在大学生中更加的受欢迎。现校园社交充斥着浮躁不平静的生活气氛，同学们需要一个纯净的、包容的、自由的社交环境，同时老师可以在此系统上给同学们进行学习辅导，该系统可以给同学们在校园互联网上提供一片净土，立足于和谐的校园生活环境，以兴趣为起点宣扬个性的校园社交网站。校园社交网站针对学校学生和老师为目标群体，帮助学生建立一个好的社交形态，添加校内服务功能为学生提供帮助的同时丰富社交形态。

本系统是校园社交网站的设计与实现，是一款以全栈开发的PC端网页。前端开发运用Vue.js框架技术，利用vue create命令行搭建前端项目。后端开发运用Node.js框架技术，利用express搭建后端项目，使得整个项目采用前后端分离模式，这样能快速找出问题的所在，对于一个前端开发者而言，是一个不错的选择。同时在本次项目开发时将项目放到Github上进行托管，方便管理，保证代码不丢失，提高开发效率。通过这次系统的开发，使我对vue.js和node.js有了更加深刻的了解，然后通过视频教学学会了web爬虫，并且应用到了该系统当中。

从总体上来说，这个校园社交网站的项目最终开发与实现较为成功，此系统的亮点在于即时消息通知、web爬取图片、聊天功能较为全面等等，美中不足还是存在的，如密码没有加密、代码冗余性较高、性能优化不足等等。随着将来工作上提高自己的专业技能，会对该系统进行完善和优化。

# 参考文献

# 致谢

经过20周左右的不懈努力下，我终于完成了毕业设计。虽然毕业设计需要20周左右来完成，可是由于疫情的原因，我已经在一月份就开始着手研发系统了。从大一开始，我还没有一次独立开发一个项目，所以对此次毕业设计非常重视。在选课题前，我思考了很久，纠结选一个什么样的课题，后来与指导教师郭兴凯老师经过一番商量后，最后决定了这个课题-校园社交网站的设计与实现。首先我要感谢我的指导教师郭兴凯老师，在设计与实现过程中，遇到了很多困难，在老师的指导和建议下，使我克服了设计与实现过程中的所有困难，为我大学最后一门课程画上圆满的句号。

在大学四年的生活中，我还要感谢身边的同学、老师这么关心我，帮助我，其实我更要感谢科协这个组织。大一时对编程很迷茫，甚至想要放弃，但是进了UI组后，经过学长学姐的培训下，使我对前端起了浓厚的兴趣，对我现在的工作有了很大的帮助，而且还结识了很多的朋友，扩充了我的人脉。短短四年一眨眼就过去，由于疫情无法回到学校拍毕业照、与室友相聚、再吃一次食堂的饭，现在仔细一回想，都是满满的遗憾。

校园生活即将结束，即将步入社会的我，对校园生活还是有些不舍，人生第一阶段结束，即将迎来的是人生第二阶段，对社会即憧憬即好奇。

最后，对审核和负责毕业答辩的老师们说一声感谢，还请老师们多多批评和指正。

1. [] 原丽娜,鲍再伦,王克强,et al.基于“互联网+校园”的高校学生社交平台“格物宝”的设计与实现[J]. 科技经济导刊, 2018, v.26；No.645(19):178. [↑](#endnote-ref-1)
2. [] 刘国东,李文杰.基于Javascript的校园社交网络系统的研究与开发[J]. 电脑知识与技术, 2014(11) :2689-2692. [↑](#endnote-ref-2)
3. [] 袁川晔,刘江.校园SNS社交网站传播特性[J].宜春学院学报,2013,35(02):152-154. [↑](#endnote-ref-3)
4. [] 骆宝丞,黄琼.引导大学生正确使用校园SNS网站——对杭州10所高校的调查和分析[J].经营与管理,2013(02):141-143. [↑](#endnote-ref-4)
5. [] Ayodele, Kolawole Olanrewaju;;Ezeokoli, Rita Nkiruka. Research on Humanities and Social Sciences [↑](#endnote-ref-5)
6. [] Daniel W. Salter,Reynol Junco,Summer D. Irvin Journal of Student Affairs Research and Practice，10.2202/1949-6605.1397 [↑](#endnote-ref-6)
7. [] 吕英华.渐进式JavaScript框架Vue.js的全家桶应用[J].电子技术与软件工程,2019(22):39-40. [↑](#endnote-ref-7)
8. [] 张鹏飞,王乾,胡晓冬,杨明浩,崔明旺.基于Node.js和JS的前后端分离实现[J].软件,2019,40(04):11-17. [↑](#endnote-ref-8)
9. [] 韩雪. 面向科研社交网络的混合好友推荐方法研究[D].合肥工业大学,2019. [↑](#endnote-ref-9)
10. [] 王鹏强.基于vue的MVVM框架的研究与分析[J].电脑知识与技术,2019,15(11):97-98+100. [↑](#endnote-ref-10)
11. [] 谭舒月,侯玲.校园社交网站客户分类分析[J].市场观察,2019(02):59. [↑](#endnote-ref-11)
12. [] 杨雨成,任利峰.MySQL数据库性能优化技术研究[J].科技经济导刊,2020,28(03):32. [↑](#endnote-ref-12)
13. [] 王芃孜. 陕煤运销信息管理系统的设计与开发[D]. 西安科技大学, 2013. [↑](#endnote-ref-13)
14. [] 茆玉庭. 基于Node.js和WebSocket的即时通信系统的设计与实现[D].南京邮电大学,2018. [↑](#endnote-ref-14)
15. [] 邢瑞芬. 基于ASP的教务处管理系统的设计与实现[D]. 2011. [↑](#endnote-ref-15)