

黑龙江大学

本科生毕业论文

论文题目：	个人空间系统的设计与实现
学 院：	软件学院
年 级：	2014
专 业：	软件工程
姓 名：	李放
学 号：	20145881
指导教师：	钟颖莉

2018 年 5 月 7 日

摘要

随着 Internet 的广泛应用，个人空间、微博以及博客等新兴事物随之出现，正式因为这些新兴事物的出现，才会推动互联网信息的传播，以及能够让人与人沟通更加简单。与此同时 web 的新技术也应运而生，从最开始的 JQuery，zepto 到现在前端的 Angular，react，vue 三大框架。通过查阅资料，各个技术的分析计较，前端采用 vue 框架以及 elementUI 框架来实现前台页面实现，后台采用 nodeJs 的 express 框架作为服务层，用来数据的处理，数据库采用 MongoDB，使用此技术栈进行开发。

个人空间是国内的腾讯公司推出的相关产品，它具有说说功能，文章，相册，音乐等功能，而且该系统使用方便，操作简单。此次设计正是参考它的为原型，来实现此次设计。经过系统的需求分析，个人空间系统有个人说说，个人相册，个人博客，留言板，好友管理，个人信息管理等模块组成。其中用户可以访问好友的个人空间，以及对其动态完成评论，也可对评论进行回复，以此达到沟通为目的。个人空间可以有多种形式达到信息的获取，交流，管理的目的，建立一个完全属于自己的个人网络空间。能够加深网友对自己的了解，也可方便自己的日常信息的管理。

关键词

个人空间；博客；交流；VUE；MongoDB；nodeJS；

Abstract

With the wide application of Internet, the emergence of new things such as personal space, micro-blog and blogs will promote the spread of Internet information and make it easier for people to communicate with people. At the same time, web's new technology also came into being, from the very beginning of JQuery, zepto to the front end of Angular, react, Vue three major frameworks. Through consulting the data, the analysis of each technology, the front end uses the Vue framework and the elementUI framework to realize the implementation of the front page. The background uses the express framework of nodeJs as the service layer, used for data processing, the database uses MongoDB, and uses this technology stack to develop.

Personal space is the related product launched by the domestic Tencent Inc. It has functions, articles, albums, music and other functions, and the system is easy to use and easy to operate. The design is based on its prototype, to achieve this design. After the system needs analysis, personal space system has personal talk, personal album, personal blog, message board, good friend management, personal information management and other modules. The users can visit their friends' personal space, complete their comments dynamically, and respond to the comments in order to achieve the goal of communication. Personal space can have various forms to achieve the purpose of information acquisition, communication and management, and establish a completely personal network space. It can deepen users' understanding of themselves and facilitate their daily information management.

Key words

Personal Space; Blog; Communication; VUE; MongoDB; nodeJS;

目录

摘要.....	I
Abstract.....	II
第一章 引言.....	1
1.1 项目背景以及意义.....	1
1.1.1 项目背景.....	1
1.1.2 研究意义.....	1
1.2 国内外研究现状.....	2
1.2.1 国内现状.....	2
1.2.2 国外现状.....	2
1.3 主要结构.....	3
第二章 系统相关技术介绍.....	4
2.1 B/S 结构技术介绍.....	4
2.1.1 B/S 架构简介.....	4
2.1.2 B/S 结构的优点.....	4
2.1.3 B/S 结构缺点.....	5
2.2 MongoDB 数据库介绍.....	5
2.3 JavaScript 和 ES6 介绍.....	6
2.3.1 JavaScript 简介.....	6
2.3.2 ES6 简介.....	6
2.4 NodeJs 介绍.....	6
2.4 MVVM 模式.....	7
2.4.1 MVVM 简介.....	7
2.4.2 MVVM 优点.....	7
2.5 相关工具介绍.....	8
2.5.1 VUE 框架介绍.....	8
2.5.2 Express 框架介绍.....	8
2.5.3 Less 介绍.....	8

第三章 系统分析.....	9
3.1 系统可行性分析.....	9
3.2 功能需求分析.....	9
3.2.1 识别系统相关者功能需求模型建立.....	9
3.2.2 用例建模.....	10
3.2.3 用例描述.....	12
3.2.4 数据字典.....	15
3.2.5 运行环境需求.....	16
3.2.6 界面需求.....	16
3.3 遇到问题.....	16
3.3.1 请求数据跨域问题.....	16
3.3.2 登录权限.....	17
3.3.3 图片上传.....	17
3.4 本章小结.....	18
第四章 个人空间系统软件设计.....	19
4.1 设计模型的建立.....	19
4.1.1 系统结构图的建立.....	19
4.1.2 时序图的建立.....	19
4.1.3 活动图.....	21
4.3 数据库设计.....	25
4.3.1 数据库概念设计.....	25
4.3.2 数据库逻辑设计.....	26
第五章 个人空间详细设计.....	30
5.1 登录模块详细设计.....	30
5.1.1 注册流程.....	30
5.1.2 登录流程.....	31
5.2 文章模块详细设计.....	32
5.2.1 发表文章流程.....	32
5.2.2 编辑文章流程.....	32

5.2.3 查看文章流程.....	33
5.2.4 删除文章流程.....	34
第六章 个人空间系统测试.....	36
6.1 功能模块测试.....	36
6.1.1 注册登录模块的等价类分类法测试.....	36
6.1.2 文章模块的等价类划分.....	37
6.1.3 相册模块的等价类划分.....	38
6.1.4 说说模块等价类划分.....	38
6.1.5 好友模块等价类划分.....	39
结论.....	40
参考文献.....	41
致谢.....	42

第一章 引言

截止到 2018，我过上网人数已近 7.7 亿，成为网民人数已经成世界最多的国家。在现在，人们或取信息的方式有很多，比如通过查找相关书籍、询问了解相关信息的人等等，然而，互联网是人们最先想到的获取的方式。网站最大吸引人的地方就是可以将文本、图像、声音等多媒体信息相结合，使之具有强大的多媒体表现力以及为用户互动的特点，不得不否认的是，在现代，最有效的信息传递方式就是网络。

网络的发展越来越迅速，对人们的日常生活影响也越来越大，网络极大的方便了人们的生活。而且网络已经成为新的传播媒体中的一员，而将个人的社交与网络的完美的结合，将会方便人与人之间的交流，减少人与人之间的距离。

个人空间已经成为一种时尚，它能够使人们拥有一片属于自己的天地，提供一个展现自己风采的舞台。

1.1 项目背景以及意义

1.1.1 项目背景

如今，互联网发展越来越迅速，仅仅浏览页面已经不能够满足我们的需求，我们更希望能够参与进去，互联网所有的强大开放性，让人拥有一个自己的个人网站，也使之成为了一种可能，让互联网用户拥有一个能够充分展现自我的舞台。

在互联网中我们彼此之间应该进行的联系，相信彼此交流能够让我们进步更快，让以后的前行的道路走的更远，不断的超越自我，让我们聆听不同人站在不同立场的想法和看法。设想一下，互联网的每一个用户都一个属于自己的个人空间，因为这个网页中拥有我们的见解，知识以及我们的思想。那么，用户无论在哪或者是否在线，我们随时随地都可以跨越时间和空间，彼此之间进行交流。

1.1.2 研究意义

在互联网飞速发展的今天，设计一个结构清晰，功能健全，技术先进的个人空间系统成为了具有重要意义的工作，主要以现在以下几个方面：

- 1、能够明显的降低信息传递的成本，能够减少大量的人力，物力以及资源的不必

要开销。运用计算机网络的优势，把先进的网络技术和传统的信息传递方式融合起来，是信息的传播和人们获取相应信息的方式提供了一个简单而且实用的系统。

2、对程序的技术者而言，可以更加深刻的理解使用的相关技术，提高知识的积累，而且能够扩展自己的知识面，不仅仅知识将自己技术能力体现在前台页面或者后台逻辑处理上，而是所有技术面都会接触，包括设计方面。

3、对于使用者而言，在互联网飞速发展的今天，能够是自己在互联网有一席之地，一个只专属于自己的空间。

1.2 国内外研究现状

1.2.1 国内现状

在互联网发达的今天，人们的生活水平不断提高，人们记录信息也不仅仅局限在使用笔写在纸上，而是通过朋友圈、微博、博客、或者腾讯推出的个人空间。这就是使用户在有互联网的地方，就能够通过手机或者电脑来使用这些软件或者应用，大大的方便用户记录信息，而且保证了存储的长久性。

当然通过互联网技术的发展，这几年的前端技术发展和后台技术发展，使这些软件或者应用的技术发生转型，从最初的神器 JQuery 到现在的前端三大框架，对页面的性能提高在提高，极大的优化了用户的使用体验，而且编写更方便，更迅速，使程序员有更多的经历来实现更加复杂和更加绚丽而且实用的功能。

1.2.2 国外现状

国外对于网络信息的传播方面与我国相比，要比我国较为先进，无论从技术还是空间系统的优良性，都要远远的强于我国。而且国外的技术选型上与我国也远远不同，在国外，前端的技术选型除了 React 就是 Angular，可以说 JQuery 的已经很少了，而国内大多数的网站还是使用 JQuery 为主，这就导致了系统性能远远的要弱于国外系统性能，而且大部分的性能都是用于对 DOM 查找以及更新，而国外使用的新技术都是基于 MVVM 模式的框架，大部分的注意力都放在数据上，对 DOM 的操作几乎为零，这样就大大的提升系统的速度，自然而然比国内的速度要快很多。

1.3 主要结构

本文的内容共分 6 个章节，具体如下：

第一章为引言部分，介绍项目的背景、研究意义、国内研究现状以及国外现状。最后对本文结构进行基本描述。

第二章为系统相关技术。介绍实现该系统所运用的一些相关技术，其中包括编程语言以及数据库。

第三章为系统需求分析部分。包括可行性分析，用例图用例描述以及数据字典。

第四章为系统概要设计部分，包括时序图，活动图，数据库设计等。

第五章为系统详细设计部分。该部分主要包括每个功能模块部分的流程图。

第六章为测试部分，简述一些测试方法以及该系统的测试用例部分。

第二章 系统相关技术介绍

2.1 B/S 结构技术介绍

2.1.1 B/S 架构简介

此次系统采用的架构模式是 B/S 架构模式。这种架构是 web 流行后，出现的一种架构模式，B/S 架构是由 C/S 架构经过改变后得到的一种架构。在这种结构下，用户通过操作浏览器进行对系统的操作和使用，前端只是实现了极少数的业务逻辑，服务端来实现主要的业务逻辑，形成所谓的三级架构。这种架构统一了客户端，在客户机上只要有诸如 IE，谷歌浏览器，火狐浏览器等主流浏览器即可，服务端上实现系统的核心功能，服务器须要安装数据库，例如 Oracle、Sybase、Infomix 或者 SQLserver 等。用户可以通过 Web 浏览器去访问 Internet 上的各种资源，包括：文本、图形、图像、flash、视频点播和声音信息等^[1]。

2.1.2 B/S 结构的优点

传统的 C/S 架构在 web 技术蓬勃发展中通过一些改动发展得到如今的 B/S 架构，而且当今的主要的网络架构就是 B/S 架构，目前大部分的网站都是采用这种结构，尤其是商务类型的网站。

B/S 结构优点主要有以下几点：

- 1、在客户端不需要对系统进行维护。不论身处什么地方，只要具备可以连接互联网的设备，则能够利用 B/S 模式在客户端运行成都旭，无地点制约^[2]。
- 2、扩充业务很简单而且方便，只需要增加 web 页面的内容和服务器的主要逻辑，可以实现业务的扩充。
- 3、维护起来容易，仅仅只是改变网页的内容，就可以完成所有用户的实时更新。
- 4、开发时间短，对客户端的设计相比 C/S 来说标准会更低，而且很大程度上降低设备维护和人员的使用培训^[3]。

2.1.3 B/S 结构缺点

虽然 B/S 架构有它的优势，但是也有不可避免的缺陷。比如：

- 1、个性化特点显著降低，无法实现一些拥有个性化的功能，所有用户的功能基本一致。
- 2、由于各个浏览器的差异较大，无法实现一套代码在各个浏览器上都能运行，就导致了编写代码时候需要对部分问题进行兼容性处理。
- 3、在速度上和安全性都须要我们花费庞大的成本来进行设计。
- 4、功能弱化，难以完成传统模式下的特殊功能需要。
- 5、交互式是客户端浏览器和服务端端的交互方式，需要动态客户端页面，这种交互方式会很大程度上降低响应速度（缓解这一问题方式可以使用 Ajax 技术）而且无法进行分页显示信息，这样给数据库访问造成较大的压力。

2.2 MongoDB 数据库介绍

MongoDB 由 C++语言编写的一个基于分布式文件存储介于关系数据库和非关系数据库之间的数据库。在所有非关系型数据库中，它是功能最丰富，而且最与关系型数据库最相近的。他扩展了关系型数据库中如辅助索引、范围查询和排序^[4]，而且，对于管理员来说，他提供给开发者驱动程序和 shell 命令提供的自然语言式的 API，使用起来特别方便，而且快捷。

每种数据库都会有自己的特点，当然，MongoDB 也是拥有自己的特点，比如它具有丰富的数据模型，它是一个面向文档的数据库，这就决定了它具有更加丰富的数据扩展性，将原来关系型数据的“行”概念改变成高灵活性的“文档”，也就是 JSON 的数据格式，而且对于使用面向对象语言的开发者来说，JSON 这种数据格式非常的友好，无论是添加还是对信息的获取，结构清晰而且便于操作。

前面提到过，它的功能丰富，其中最就具有一些 MongoDB 真正独特的功能：独具特色的索引机制，这个索引会提供唯一的、复合的和地理空间的索引能力，在查找时，通过索引，可以提高数据库的查询效率；Mongo DB 实现的分布式技术，主要在于数据的分片 (Shard) 以及副本集，分片是集群中负责某一子集的一台或多台服务器^[5]；不得不提到的一点就是 MongoDB 的性能，由于内存映射的机制，MongoDB 会将内存管理交给计算机的操作系统来解决，极大程度上简化了 MongoDB 的内存管理，提高了数据

库相应速度，但是这就造成了如果数据集过大，那么它会将所有可用的系统内存全部占用，所以，在不知道数据集大小的时候，要实时监测内存的使用情况。

2.3 JavaScript 和 ES6 介绍

2.3.1 JavaScript 简介

JavaScript 由 ECMAScript、DOM、BOM 三大模块组成。其中 ECMAScript 是由 ECMA-262 定义，ECMA-262 定义语法、类型、语句、关键字、保留字、操作符、对象，ECMAScript 就是实现了标准规定的各个方便内容的描述^[6]。DOM 即文本对象模型，是对 HTML 文档操作的 API，DOM 总共分为 DOM1，DOM2，DOM3 以及其他 DOM。其中 DOM1 的主要目的就是映射文档的结构，而 DOM2 目标就是完成对 DOM 视图、DOM 事件、DOM 样式以及更加容易的遍历和查找。DOM3 则就是在 DOM2 的基础上，完成统一的方式加载和保存文档的办法。BOM 即浏览器对象模型，其主要是对客户端浏览器进行操作，其中 navigator 对象的主要作用就是提供浏览器的详细信息，location 对象是提供浏览器加载加载页面的信息，screen 对象提供用户显示器分辨率的信息以及 BOM 提供了对 cookies 的支持，还提供了 ajax 所需要的自定义对象。

2.3.2 ES6 简介

ES6 实际上就是 ECMAScript 的第六个版本，它在原先的第五个版本上，扩展了很多的功能^[7]，比如：

新增的声明变量的两个关键字，声明的变量不在是 window 对象的属性，而是一个单独的个体。增加了块级作用域，当然，只有用 let 和 const 声明的变量才会有块级作用域。对异步操作进行了处理，为此新增的 Promise 对象以及 Generator 对象。增加了 for..of..循环，对数据的遍历更加容易。对对象的声明方式进行了扩展，通过 class 关键字来进行声明对象。以及等等一系列的扩展。

2.4 NodeJs 介绍

Node.js 它和 c，c++，java 等编程语言不同，它不是一种独立的语言，是一个开发实时 web 应用程序的一个平台，是让 JavaScript 运行在浏览器之外运行的一个平台。为

了能进一步的提高效率和大规模数据架构下的可扩展性，它的设计模式与以往不同，以往是通过多线程来实现高并发，而它使用的单线程、异步 I/O、事件驱动的设计模式^[8]。这样不但减少了多线程程序的复杂性，降低了开发难度，减少了开发成本，提高了开发效率，而且还提升了性能。

Npm(node package manager)是 NodeJS 平台内置的一个包管理器，这个包管理器强大而且非常灵活。目前 Npm 上已经拥有了上万个第三方模块，而且内容十分丰富，其中包括了网站的开发框架，有模板语言解析器、图形、Mysql、MongoDB 等数据库的接口，在 Npm 上甚至有图形用户界面和操作系统的 Api，这足以证明 Npm 的强大性。

2.4 MVVM 模式

2.4.1 MVVM 简介

MVVM 它本质上就是 MVC 的改进版，它由模型(Model)、视图(View)、视图模型(View Model)三部分组成[...], MVVM 会抽象化其中的视图层的状态以及行为，目的就是为了让我们将视图 UI 和业务逻辑进行分离，这些事情 ViewModel 来解决，他可以取出 Model 的数据同时帮忙处理 View 中需要展示内容而设计的业务逻辑。它的由来是 MVP 和 WPF 结合的应用方式发展演变过来的一种新兴的架构，基于 MVP 并且结合 WPF 的一些特点，达到可以应付日益复杂的客户需求。

2.4.2 MVVM 优点

MVVC 与 MVC 模式一样，其实就是为了能够让视图层和模型层进行分离，这么做的原因就是为了一下几点：

1、降低代码的耦合性，视图可以独立于模型的变化以及修改，而且一个 ViewModel 可以绑定到很多不用的视图上，当视图发生变化的时候，模型的值可以不发生改变，当模型的值发生变化时候，视图也已不发生变化。

2、提高代码的可重用性：可以把一些逻辑放在一 ViewModel 中，让很多视图重用这 ViewModel，可以减少带需要的代码量。

3、独立开发：开发人员可以专注于业务逻辑和数据的开发，设计人员可以专注于页面设计。

4、可测试

2.5 相关工具介绍

2.5.1 VUE 框架介绍

VUE 是前端三大框架之一，是一套用于构建用户界面的渐进式框架。不同于与其他大型框架，VUE 被设计为可以自底向上逐层应用，相比较 Angular，VUE 提供更简单更容易理解的 API，VUE 的核心库只关心 View 层，对于初学者来说特别容易上手，很多概念也很好理解，与第三方库或已有项目集成起来会特别方便，另一方面，当与现代化的工具链以及各种支持类库结合使用时，VUE 也能可以为复杂的单页应用提供强大的驱动。

2.5.2 Express 框架介绍

Express 是一个简洁而灵活的 NodeJS Web 应用框架，提供了一系列强大的特性来帮助开发者，创建各类 web 应用程序应用程序^[10]。可以任意组合的 Connect 中间件，提供了操作 Http 丰富快捷的方法，让开发者可以更加快速而且简单的创建出健壮友好的 API，开发者只需要在他的基础上扩展了相关 Web 应用所需要的基础功能。

2.5.3 Less 介绍

Less 在原有的 CSS 的语法基础上，添加了变量，Mixin，运算以及函数等功能，正式因为这些新增加的功能，能够让是 CSS 的编写更加简单容易，而且还降低了 CSS 的维护成本，与他的名字的意义一样，让我们花费很少的时间和成本，来完成更多的工作。

本质上，它包含一套自定义的语法以及一个解析器，用户根据 less 规定已经的语法定义自己的样式规则，开发者所写下规则最终都会通过解析器，将其编译成对应的 css 文件。Less 并没有砍掉 css 原有的特性，更不是用来取代 css 的，而是根据现有 css 语法的基础上，为 css 加入程序式语言的特性。

第三章 系统分析

3.1 系统可行性分析

本章就是对于技术可行性的研究，通过以项目实践的技术角度进行分析，合理的设计所用的技术方案，并对采用的技术进行评估。前端三大框架 VUE 它相比较 Angular 和 React，学习曲线比较低，容易上手，降低人们的学习成本。对模块友好，可以通过 NPM，Yarn 等包管理工具进行安装所需要的依赖。支持组件化开发，可以用解耦的、可复用的组件来组合，降低开发所需要时间，文件小，而且页面渲染迅速，用户体验特别好。Express 框架，可以创建 server 服务器，中间件，Session 以及 Partial 支持非常全面。在终端中输入在终端中输入 yarn add mongoose，在 Express 中即可安装对 MongoDB 的操作所需要的依赖包，而且 MongoDB 对于数据的 JSON 格式，与前端存储方式相同，方便数据的存储与获取。综合的所选的技术的优点，对个人空间系统的开发完全是可行的。

3.2 功能需求分析

3.2.1 识别系统相关者功能需求模型建立

系统相关者词汇有两个，一个用户，另外一个就是好友访客，二者可以相互转换，相对空间的拥有者，该身份就是用户，拥有空间的全部权限，相对自己的好友列表来说，自己是好友访客，可以访问自己好友列表中的所有用户的空间，可以对其说说、相册、文章等进行评论，但是不能够进行编辑、删除等操作。情见表 3-1。

表 3-1 相关者词汇表

序号	词汇	同义词	描述
1	用户	系统的使用的者	系统的使用者，权限最高，可以使用个人空间中的全部功能
2	好友访客	用户好友列表的用户	用户好友列表里的用户，可以使用个人空间系统中的部分功能

3.2.2 用例建模

文章模块的具体功能主要有：发表文章，编辑文章，删除文章，查看文章以及评论功能，其中查看文章，以及评论功能好友访客也可使用。用例图如图 3-1 所示。

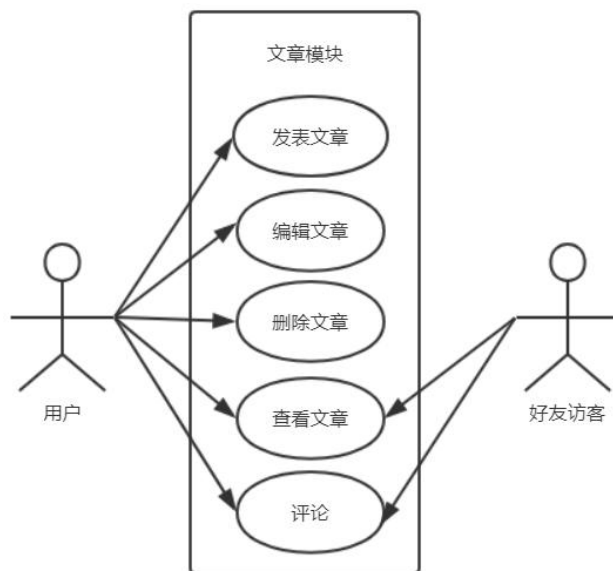


图 3-1 文章模块用例图

相册模块的具体功能主要有：创建相册，上传相片，编辑相册，查看相册，评论，其中好友访客有查看相册功能以及评论功能。用例图如图 3-2 所示。

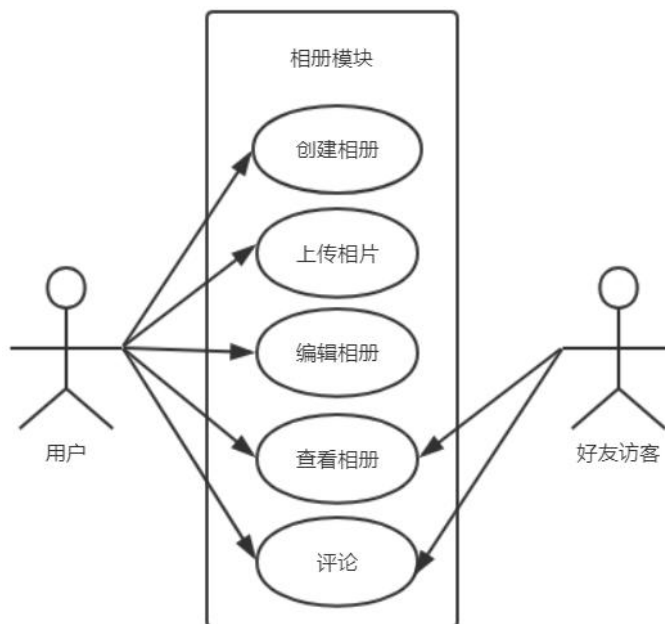


图 3-2 相册模块用例图

说说模块具体功能主要有：发表说说，删除说说，评论以及查看说说，好友访客只能够进行查看说说以及对说说进行评论。用例图如 3-3 所示。

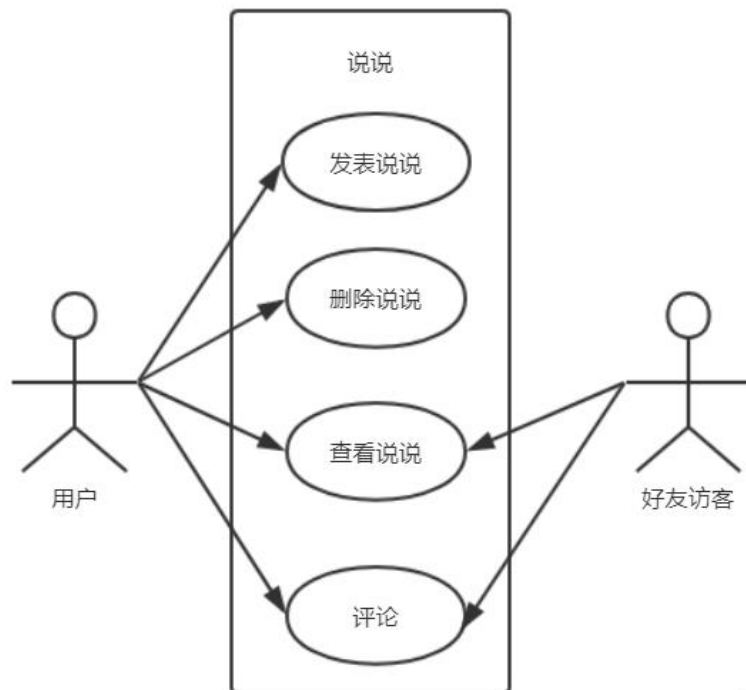


图 3-3 说说模块用例图

好友模块具体功能主要有：添加好友，删除好友，查看好友。用例图如 3-4 所示。

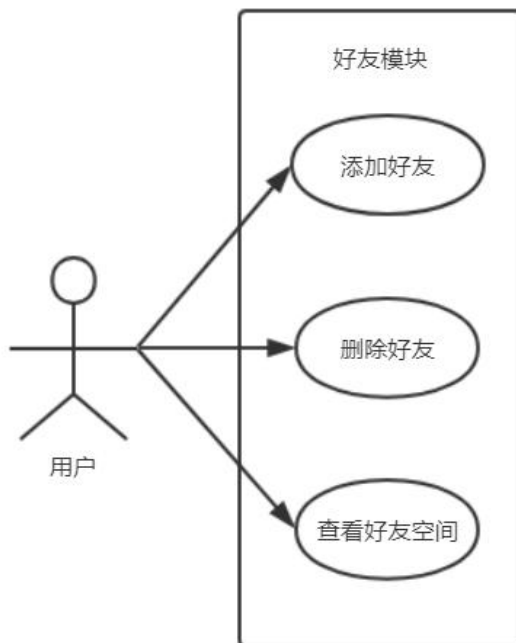


图 3-4 好友模块用例图

留言板模块具体功能主要有：发表留言，删除留言，查看留言，回复留言。其中发表留言，查看留言，回复留言三个功能，好友访客也有权限进行访问。用例图如 3-5 所示。

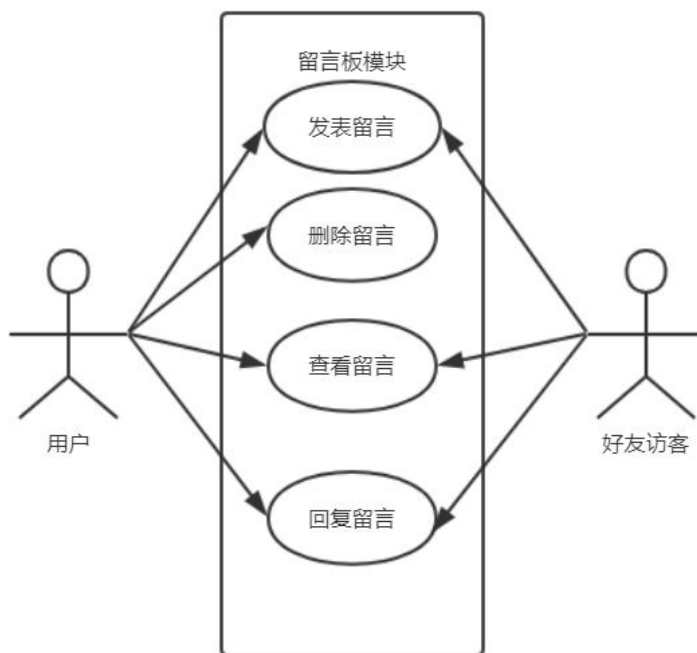


图 3-5 留言板用例图

3.2.3 用例描述

UC1 登录系统

用例描述：用户登录个人空间。

参与者：用户

前置条件：用户正确输入用户名密码。

后置条件：用户登录成功并且能对系统进行使用。

基本路径：

1. 进入主页面输入用户名及密码。
2. 点击登录按钮。
3. 前台通过 javascript 验证输入是否为空，不为空则将数据发送给后台进行数据处理。
4. 在数据库匹配是否有当前用户，有的话匹配密码是否正确。
5. 成功后获取用户的相关信息。
6. 用户对其系统进行操作。

补充说明：

- 3a. 用户名或密码输入为空。

3a1.页面弹出错误信息，不能进行登录。

4a.用户名或密码输入错误。

4a1.后台返回错误信息，并且抛出错误。

UC2 发表文章

用例描述：用户发表文章。

参与者：用户

前置条件：用户已经登录个人空间系统，并且点击新建文章。

后置条件：用户成功发表新的文章并且在列表中现在是。

基本路径：

1. 用户点击新建文章，并且进入相关页面。
2. 输入文章标题和文章内容。
3. 点击发表按钮。
4. 前台会通过 JavaScript 方法判断文章标题是否为空，不为空则向后台发送数据。
5. 后台将前端发送的数据保存到数据库。
6. 后台向前端发送保存完成。

补充说明：

4a.文章标题为空。

4a1.前端发送错误提醒，并且不向后台发送数据。

UC3 编辑文章

用例描述：用户编辑文章。

参与者：用户

前置条件：用户已经登录个人空间系统，并且点击编辑文章。

后置条件：用户成功修改新的文章。

基本路径：

1. 用户点击编辑文章。
2. 后台从数据库中获取要被修改文章的数据并且返回给前端。

3. 进入相关页面。
4. 用户修改相关内容，点击提交。
5. 将数据发送给后台。
6. 修改完成

补充说明：

2a.网络不畅通

2a1.前台进行提示请求超时。

UC4 查看文章

用例描述：用户查看文章。

参与者：用户

前置条件：用户已经登录个人空间系统，文章列表中的任意文章。

后置条件：用户成功查看想看的文章。

基本路径：

1. 用户点击编辑文章。
2. 向后台发送请求，后台从数据库中查找相关的数据，并且将数据返回给前台。
3. 前台获取相关数据，并且在页面中进行展示。

补充说明：

2a.网络不畅通

2a1.前台进行提示请求超时。

UC5 删除文章

用例描述：用户删除文章。

参与者：用户

前置条件：用户已经登录个人空间系统，并且点击删除文章。

后置条件：用户成功删除文章。

基本路径：

1. 用户点击删除按钮。
2. 前台向后台发送请求，并且后台对数据库进行操作。

3. 数据库删除完成, 后台将向前台发送一个删除完成标志
4. 前台刷信列表, 并且提示删除完成

补充说明:

2a.网络不通畅

2a1.前台进行提示请求超时。

4a.在文章内容页面进行删除

4a1.返回文章列表, 并提示删除完成。

其中说说、留言板、相册等功能的用例图和用例描述想类似, 在这里就不在进行赘述。

3.2.4 数据字典

该部分是对数据库的一个概要设计, 其中从数据项, 数据结构以及数据流三个方面来描述个人空间系统的实体之间的关系, 其中数据项各个字段意义, 具体如下表 3-2 所示。

表 3-1 数据字典

数据项名	数据项含义	别名	数据类型
Username	用户用来进行登录的账号, 且每个用户的值不同	用户名	String
Password	用户用来登录的密码。	密码	String
Nick_name	每个用户的昵称	用户昵称	String
Article_name	用户发表文章的标题, 且每个用户发表过得文章的标题不会重复	文章标题	String
Article_content	用户发表文章的内容	文章内容	String
Comment	用户或者好友对文章, 照片等进行评论	评论	Array
Create_time	注册新用户, 发表文章、等一些列增加数据库数据的时间	创建时间	Date
Moods_content	用户发表说说内容	说说内容	String
Good_num	用户发表说说获得的点赞数量	获赞数量	Number

续表 3-1 数据项表

数据项名	数据项含义	别名	数据类型
Photo_list_name	用户所有的相册不能够重复	相册名称	String
Photo_name	相片名称	相片名称	String
Photo_path		相片名称	String
Message_content		留言内容	String

3.2.5 运行环境需求

前端运行环境：需要 NodeJS 环境

后台运行环境：需要 NodeJS 环境

数据库环境：需要 MongoDB 环境

硬件环境：宏碁 VN7

系统环境：Windows 7

3.2.6 界面需求

整体页面简约大方，采用 elementUI 开源框架，所有样式遵循 elementUI，除个别需要，否则不会对其进行调整。

3.3 遇到问题

该小节主要列出遇到的问题，以及解决方案。

3.3.1 请求数据跨域问题

由于浏览器的同源策略（出于安全问题），只有在同一域名，同一服务器，同一端口下才能够进行数据的前后端传递，此次开发由于前端端口（8080）和后台端口（3000）不一致，到时请求数据时候会导致跨域问题，具体解决如下：

使用 vue-cli 中 webpack 的代理插件 proxyTable 来解决跨域问题，达到使访问的一致。具体如图所示：

```
proxyTable: {  
  '/api': {  
    target: 'http://localhost:3000',  
    changeOrigin: true,  
    pathRewrite: {  
      '^/api': '/'  
    }  
  }  
}
```

图 3-11 解决跨域代码图

3.3.2 登录权限

只有登录了才会访问相关页面，所以在没登录时是无法访问相关页面。前端路由对此进行拦截，具体实现方式如下：

```
var userInfo = sessionStorage.getItem('token');//获取浏览  
if (userInfo) { //如果有就直接到首页咯  
  next();  
} else {  
  if (to.path == '/login' || to.path == '/register') {/  
    next();  
  } else { //不然就跳转到登录;  
    next('/login');  
    Message.error({  
      message: '请先登录'  
    });  
  }  
}
```

图 3-12 前端登录权限控制图

3.3.3 图片上传

解决此问题的办法是使用一个叫做 `multipart` 的依赖包，使用内容类型解析 `http` 请求 `multipart/form-data`，然后通过相关方法来实现文件的读写，从而达到实现上传目的。

```
var form = new multiparty.Form({
  uploadDir: './public/photos/'
});
form.parse(req, function (err, fields, files) {
  if (err) {
    console.log('parse error: ' + err);
  } else {
    var inputFile = files.img[0];
    var uploadedPath = inputFile.path;
    var filename = uploadedPath.split('/')
    inputFile.originalFilename = uploadedPath.split('\\')[2]
    var dstPath = '/photos/' + inputFile.originalFilename;
    res.json({
      path: dstPath
    })
  }
})
```

图 3-12 图片上传后台代码

3.4 本章小结

本章主要对系统进行了需求分析，从系统可行性分析，功能需求分析以及数据的概要设计入手，完成对个人空间的需求分析，明确需求的功能，对数据库的进行了简单的概要设计。

第四章 个人空间系统软件设计

该模块采用面向对象的设计思路对个人空间系统进行概要设计，同样是此次设计的重要部分之一，包括设计模型的建立，。在面向对象的基础上，可以使个人空间系统产品通过持久化设计达到逻辑语言与现实生活的一一映射，将设计最终以系统的形式呈现出来。

4.1 设计模型的建立

设计模型的建立可以将各个模块连接起来，组成一个完成的系统，它主要包括系统结构图，时序图，软件结构图，活动图以及包图。

4.1.1 系统结构图的建立

通过之前的系统分析，明确了系统的主要功能，其中总共分为登录模块，文章模块，相册模块，说说模块，好友模块。得到如图 4-1 所示的系统结构图。

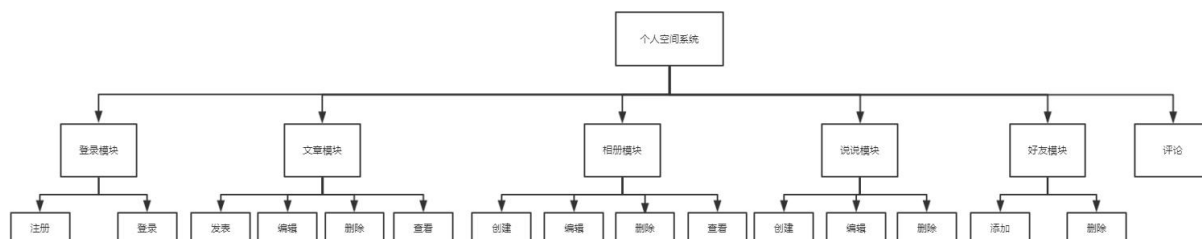


图 4-1 个人空间系统结构图

4.1.2 时序图的建立

新用户没有账号和密码，从而需要进行注册，然后进行登陆，到最后进入系统的的全过程，用户点击注册，输入相关信息，前台会对数据进行一个比对，判断输入的信息是否合法，比如，输入账号是否符合规则，是否重复，两次密码输入是否相同等。验证成功后将数据传给后台，然后对数据库进行操作，完成注册信息的操作，然后用户输入账号，对账号和密码进行比对，如果合法，将数据传给后台，后台匹配数据中相关数据，返回登录结果，登录成功跳转到系统主页面。如图 4-2 所示。

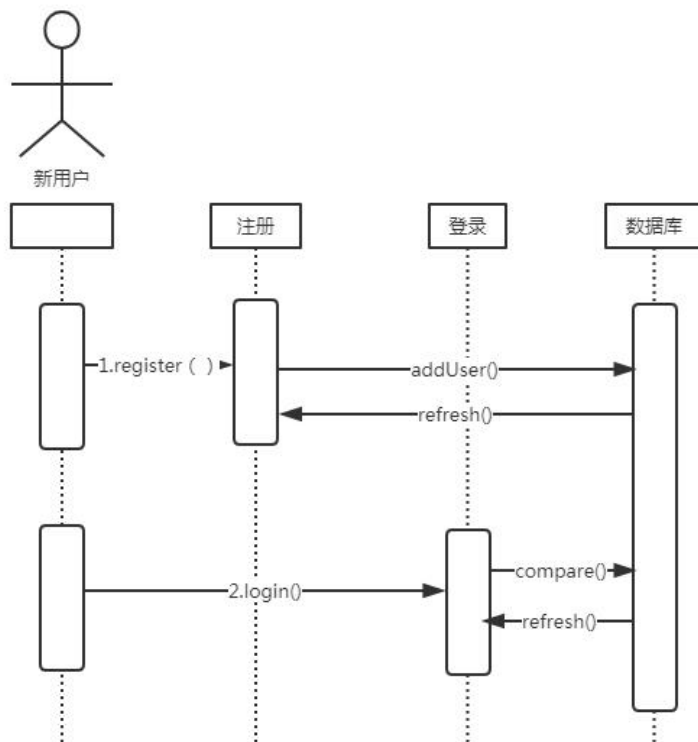


图 4-2 新用户注册登录时序图

用户登录成功后，可以使用不同的功能，进入文章功能页面，首先会调用后台数据接口，获取该用户所有的文章列表，点击发表新文章，会进入相应页面，用户输入相应的文章标题，以及文章内容，提交时，前台会对数据进行检验，如果标题为空，则不能进行提交。后台接收到数据会将数据写入到数据库中，查看文章，想后台发送请求，后台获取数据库中数据，然后返回给前台，编辑文章时，先向后台发送请求，后台获取相关数据，返回给前台，前台修改后，提交，提交时，依旧会对数据进行校验，只有通过校验才能将数据发送给后台，后台对数据库进行操作。时序图如图 4-3 所示。

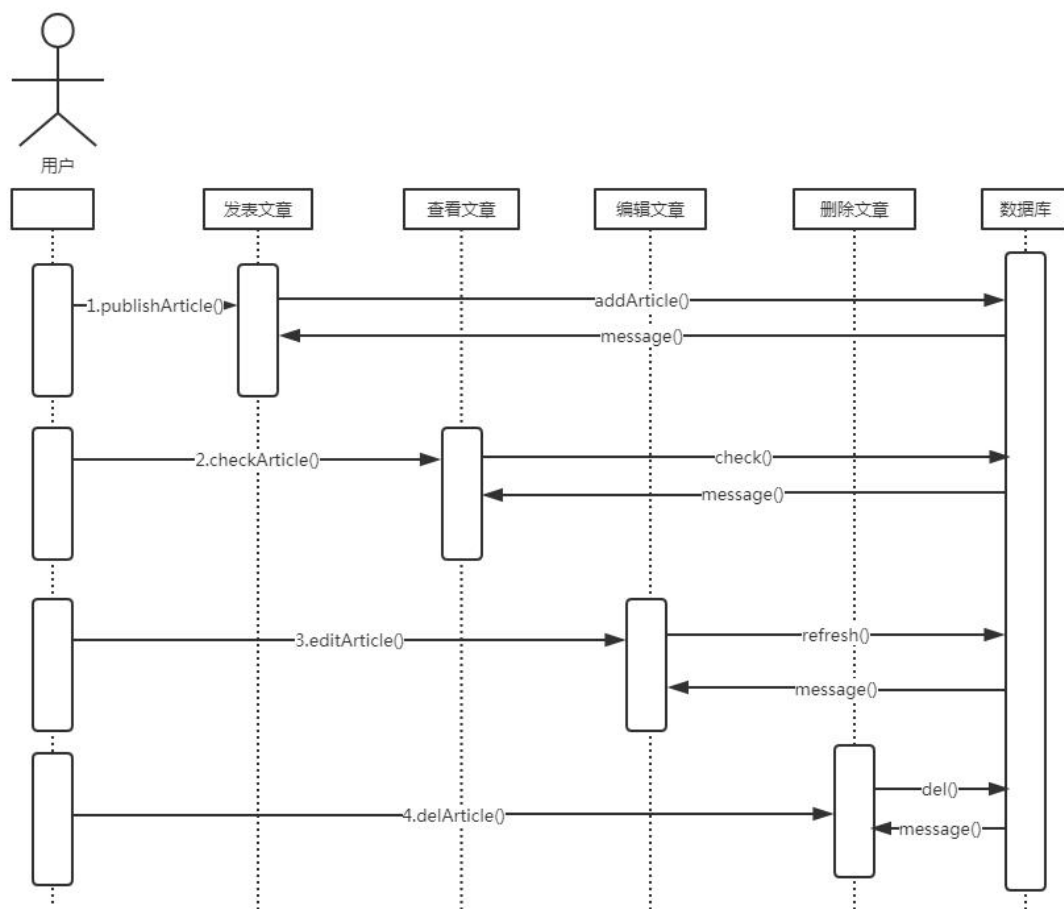


图 4-3 新用户文章模块时序图

该系统的其他模块的功能大体相似，在这里就不在进行赘述，其他的以文章模块的时序图为例。

4.1.3 活动图

当新用户第一次使用系统时候，会进入相关注册页面，页面会让用户输入相关的信息，比如用户名，密码，昵称以及个人介绍等相关的个人信息，当用户点击提交时，会在客户端进行信息合法性验证，如果信息合法的情况下，则调用后台的数据接口，信息不合法会在客户端提示相关的错误信息，并且让用户进行重新输入相关的信息。具体的活动图如 4-4 所示。

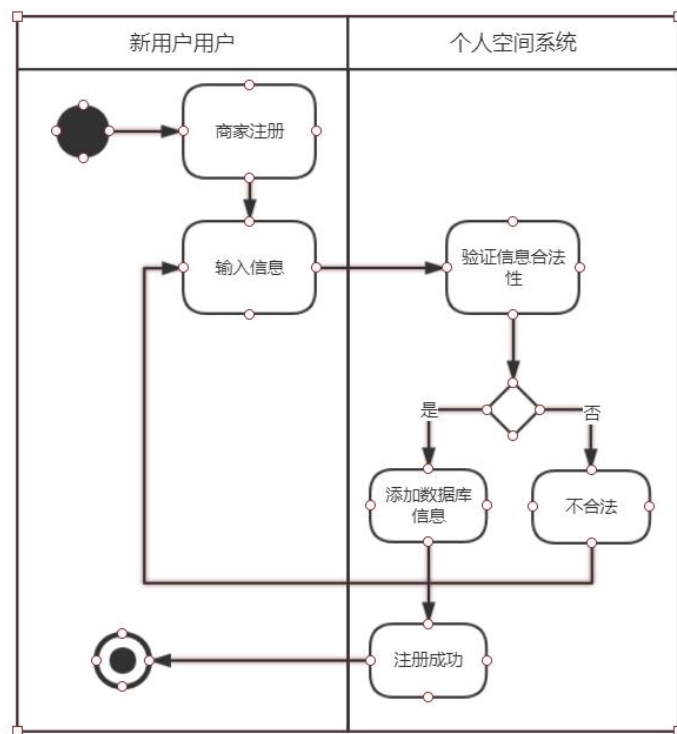


图 4-4 新用户注册活动图

用户登录系统时，会输入用户的账号，和密码。点击登录时候，调用后台的接口，此时会调用后台接口，查找相关的账号信息，查找成功后进入相关的主页面，如果没有相关信息，则会给前台返回查找失败的标志，并且提示用户的输入信息的有误。如图 4-5 所示。

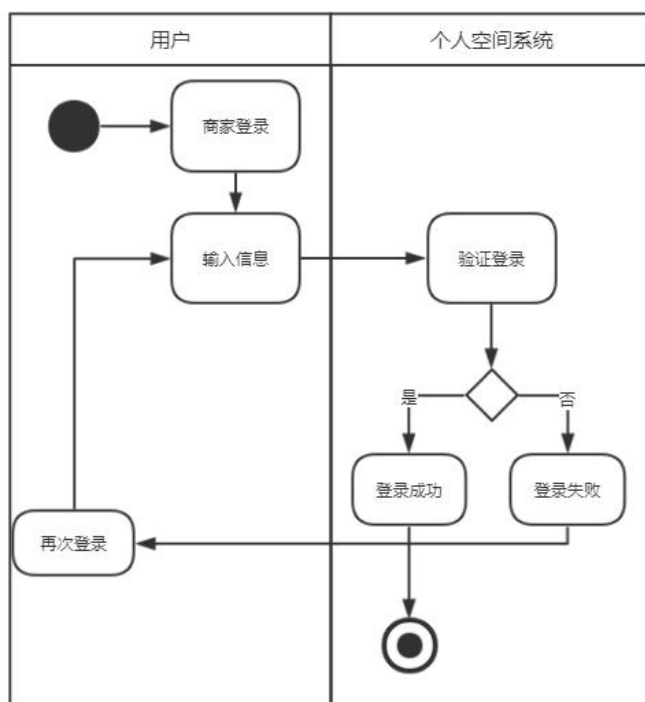


图 4-5 新用户登录活动图

用户点击发表文章，会进入相关的页面，要求用户输入文章标题，输入文章的相关内容，首先会在前台验证信息是不是合法，如果合法，将会调用后台数据接口，将数据更新到个人空间系统。如图 4-6 所示。

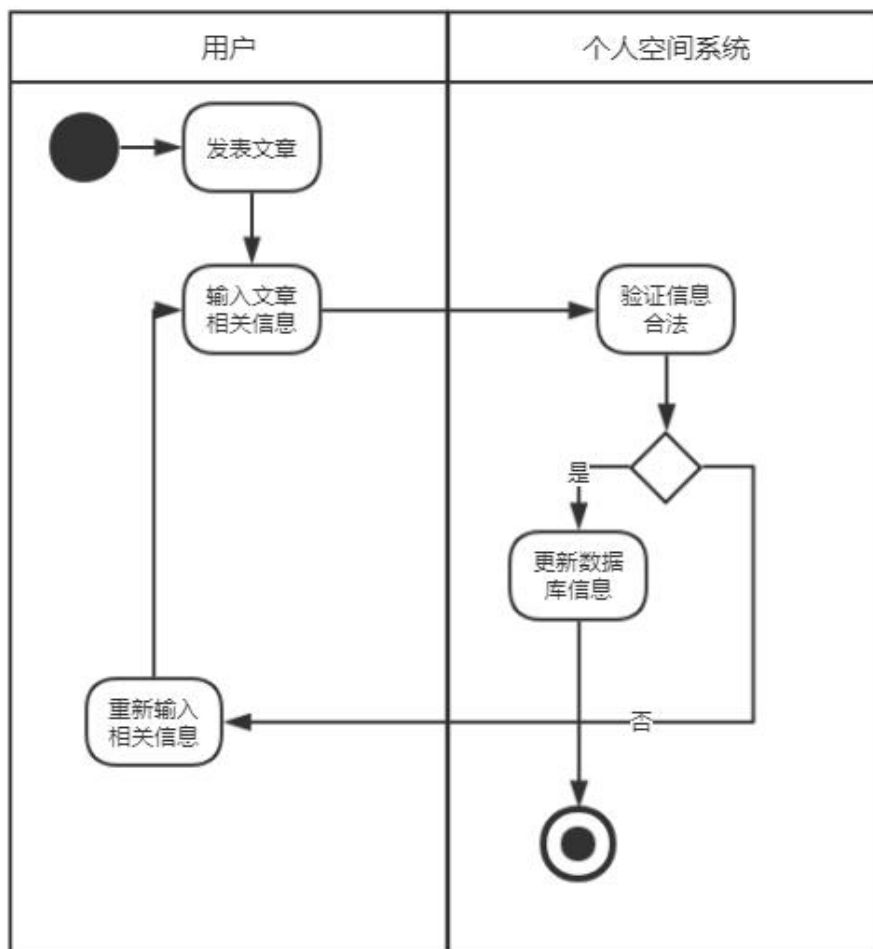


图 4-6 用户查看文章活动图

用户的查看文章只需要点击文章列表中的标题，这个时候会向服务端请求相关的数据，后台接口将数据返回给前台。用户进入文章面。如图 4-7 所示。

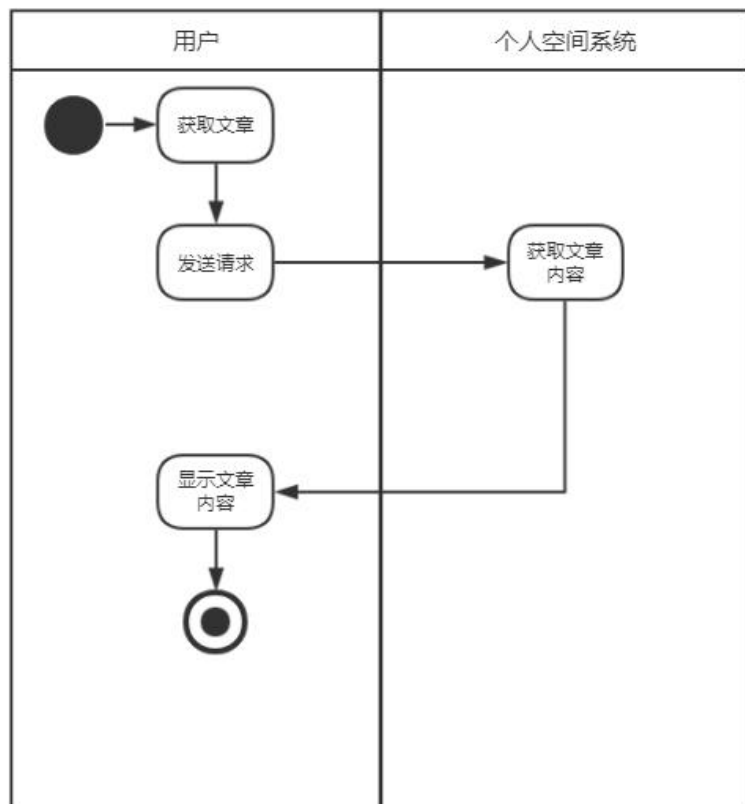


图 4-7 用户查看文章活动图

用户点击列表中的操作按钮，会有一个删除的选项，点击删除，会将文章的 id 传递给后台接口，并且将后台的中的该条数据状态改成删除。如图 4-8 所示。

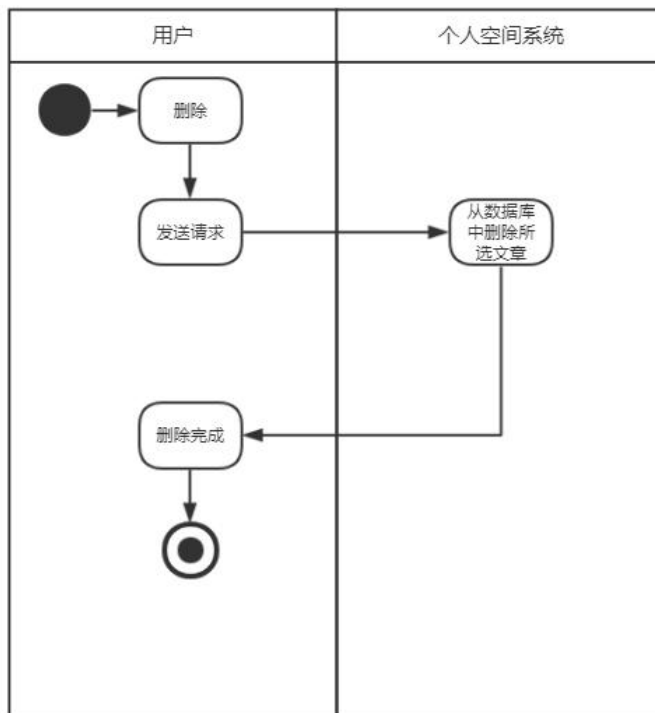


图 4-8 用户删除文章活动图

用户点击操作，会有一个编辑的选项，点击这个选项，会将该文章的信息传递给后台，后台获取相关数据，并且返回给编辑页面，用户对相关的内容进行修改，完成修改后将在一次调用数据接口，将该条的数据在文章表中进行更新。如图 4-9 所示

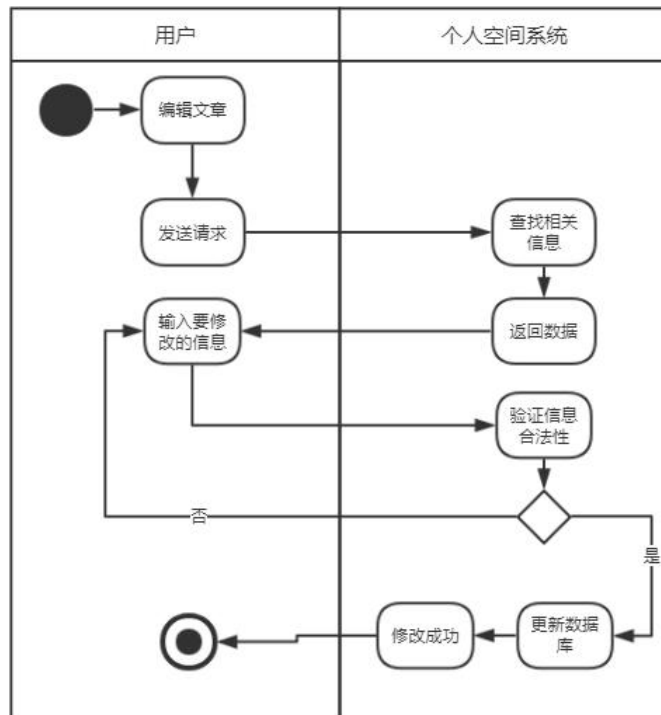


图 4-9 用户编辑文章活动图

4.3 数据库设计

4.3.1 数据库概念设计

根据之前的数据的需求分析，其中用户可以对文章、说说、留言、以及相册进行一系列的操作（增删改查），而且还可以管理好友，对好友进行添加，删除以及查看个人空间。可以确定的实体有：用户，说说，文章，相册，相片，留言，好友。他们之间的关系如图 4-10 所示。

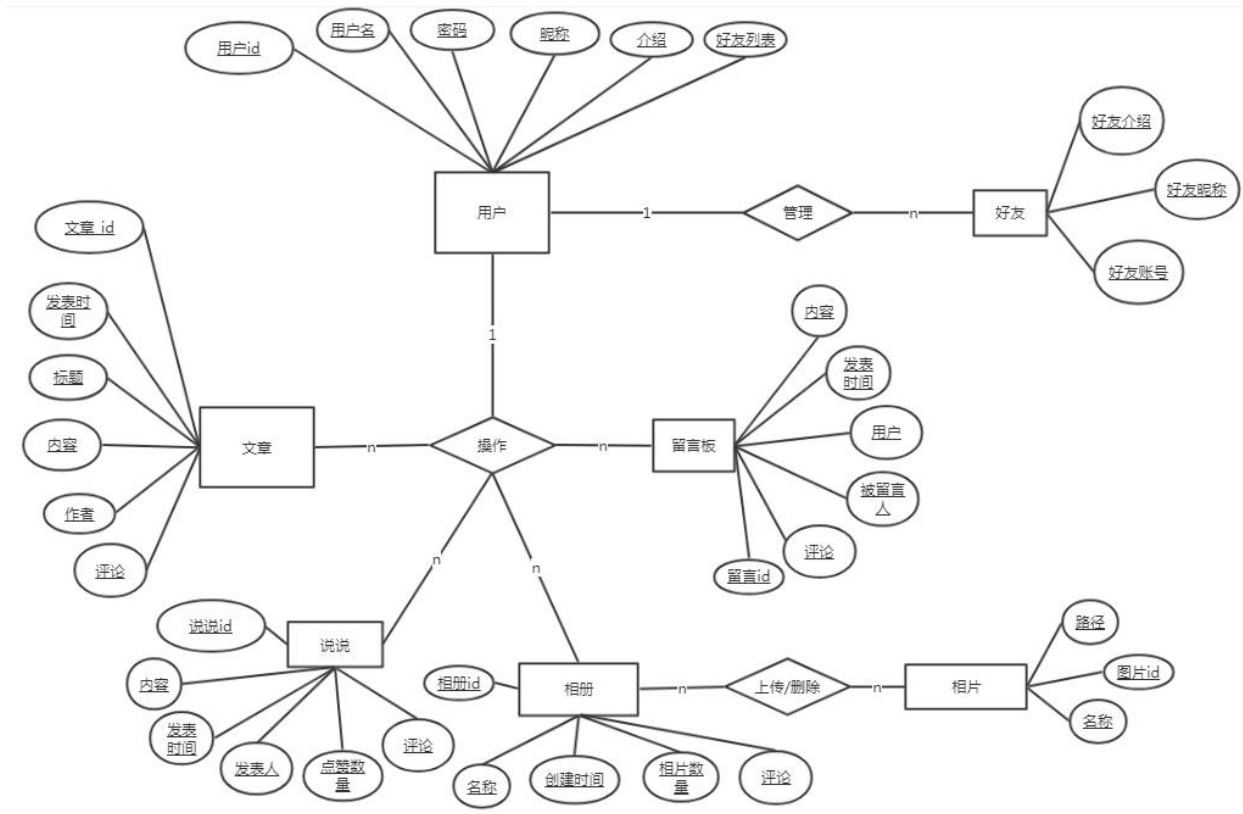


图 4-9 用户编辑文章活动图

4.3.2 数据库逻辑设计

根据图 4-10 所示的用户实体，应该有的键值为用户 id、用户名、密码、昵称、个人介绍以及好友列表，可以得到如表 4-1 所示的用户表。

表 4-1 user 表

字段名	字段类型	字段含义	备注
_id	ObjectID	用户 id	主键
Username	String	用户名	
Password	String	用户密码	
Nickname	String	昵称	
Introduction	String	个人介绍	

文章实体具备的键值是文章 id，标题，内容，作者，发表时间以及评论。具体看表 4-2 所示的文章列表。

表 4-2 article 表

字段名	字段类型	字段含义	备注
_id	ObjectID	文章 id	主键
Title	String	文章标题	
Content	String	文章内容	
Author	String	作者	
Comment	Array	评论	
Created_time	Date	文章创建时间	

说说的键值分别为：说说 id，说说内容，说说的作者，点赞数量以及对说说的评论，具体如表 4-3 所示。

表 4-3 moods 表

字段名	字段类型	字段含义	备注
_id	ObjectID	说说 id	主键
Contnet	String	说说内容	
Good_num	Array	点赞人	
Author	String	作者	
Comment	Array	评论	

相册实体的键值有：相册 id、相册名称、相片数量以及创建的时间，具体如下表 4-4 所示。

表 4-4 photoList 表

字段名	字段类型	字段含义	备注
_id	ObjectID	相册 id	主键
User_id	ObjectID	用户 id	
Title	String	相册名称	
Photo_num	Number	相册数量	
Comment	Array	评论	

相片实体的键值有：图片的 id，图片的名称以及图片的路径。具体详情如表 4-5 所示。

表 4-5 photo 表

字段名	字段类型	字段含义	备注
_id	ObjectID	相片 id	主键
Photo_name	String	相片名称	
Path	String	相册路径	

留言板，很容易理解，就是大家互相留言的地方，图 4-10 所示，得到如表 4-6 所示的留言板表。

表 4-6 message_board 表

字段名	字段类型	字段含义	备注
_id	ObjectID	留言 id	主键
User_id	ObjectID	用户 id	外键
Content	String	留言内容	
Friend_id	ObjectID	留言人 id	

最后，用户可以管理自己的好友，可以对其进行添加或者进行删除。用户主要是根据好友账号进行添加，可以获得到好友的昵称，好友介绍。得到如表 4-7 所示数据库表。

表 4-7 friend_list 表

字段名	字段类型	字段含义	备注
_id	ObjectID	好友 id	主键
Friend_username	String	好友账号	
Friend_nickname	String	好友昵称	
Friend_Introduction	String	好友介绍	

根据图 4-9 所示用户表与文章，说说，相册，留言板以及好友列表都有一定的关联，所以需要生成用户表与文章表进行联动的数据库表，由于说说表，相册表，以及留言板表的与用户与文章联动的表一样，在这就就不在赘述了，具体如表 4-8 所示。

表 4-8 user-article 表

字段名	字段类型	字段含义	备注
_id	ObjectID	好友 id	主键
User_id	ObjectID	用户 id	
Article_id	ObjectId	文章 id	

第五章 个人空间详细设计

5.1 登录模块详细设计

5.1.1 注册流程

当新用户点击注册新用户时候，客户端会跳转到用户注册的页面，页面提示用户需要输入相关的信息，比如用户名，密码，个人介绍以及昵称。当用户输入完用户名时候，会进行账号重复的检测，如果重复就不会完成注册。信息合法判断结束后，会将输入的数据传给后台的数据接口，并且将数据加入到用户表中，添加完成后会给前台发送一个注册完成的标志，并且返回登录页面。具体流程如 5-1 所示

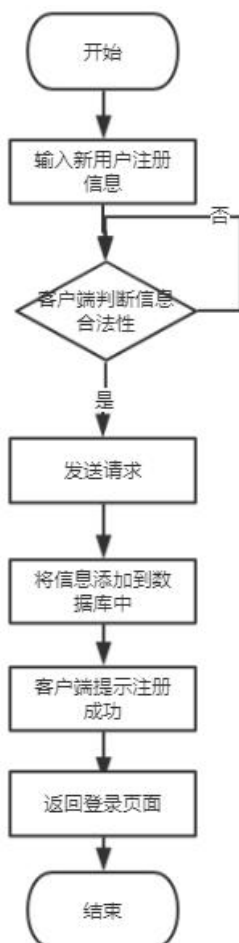


图 5-1 用户注册流程图

5.1.2 登录流程

用户每次进入系统时候都需要进行登录，登录时候需要用户输入登录账号以及密码。客户端会对输入数据进行检测，验证数据的合法性。如果输入不合法会给用户进行提示并且重新输入信息。输入合法会将数据传给后台，并且对数据中数据进行对比，对比不成功会给前提提示账号或者密码错误，如果查找相关数据，会通过 Jsonwebtoken 依赖包产生一个唯一的 token，用于后续的安全验证。并且将其返回给前台。然后进入用户的首页。具体流程如表 5-2 所示。

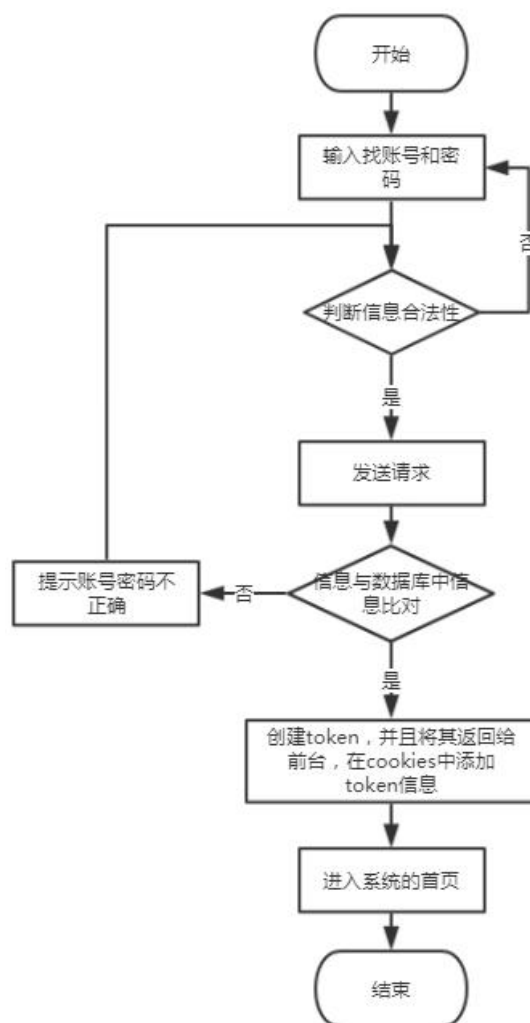


图 5-2 用户登录流程图

5.2 文章模块详细设计

5.2.1 发表文章流程

用户登录系统后可以相应的功能进行操作，用户发表文章时。需要输入文章的内容以及标题，而且系统会对文章的标题进行验证，验证成功后，会将数据传给后台，后台对数据库进行数据添加，并且返回文章的列表页，进行页面的重新渲染。如图 5-3 所示。

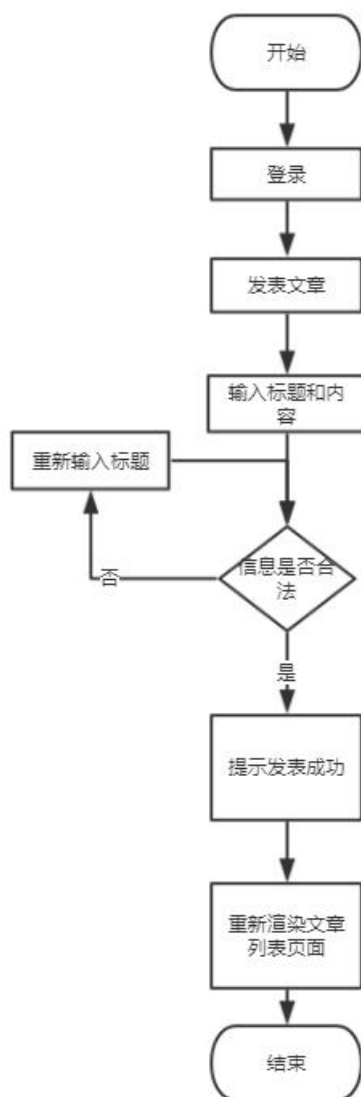


图 5-3 用户发表文章流程图

5.2.2 编辑文章流程

用户登录系统后，选择自己的文章列表，在每条信息的最后，有一个操作按钮。点

击操作按钮会出现，编辑选项，点击编辑选项进入编辑页面，或者相关的文章信息，并返回给前台，用户对返回的信息进行修改，修改完成点击完成按钮，会对文章的标题的合法性进行检验，验证通过后会对该条数据在数据库中的信息进行更新，并且返回文章列表，对列表重新渲染，提示用户修改信息成功。如图 5-4 所示。

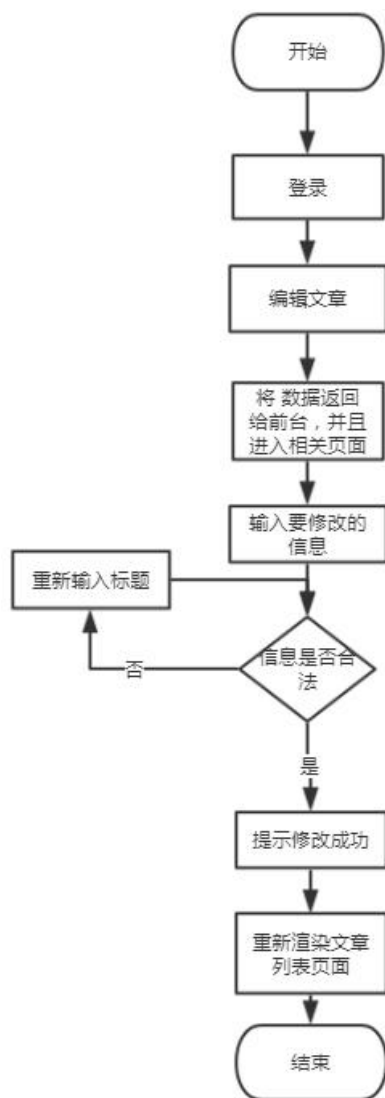


图 5-4 用户编辑文章流程图

5.2.3 查看文章流程

用户登录后，进入文章列表页面，在文章列表页面，会渲染出用户所有的发表过得文章或者用户范文好友页面也会看好友所有发表过的文章。用户点击任意一个文章标题，即可进入文章的列表页面，可以查看相关的文章内容以及评论的内容。查看文章只能查看用用户的自己的以及自己好友的。具体流程图如图 5-5 所示。



图 5-5 用户查看文章流程图

5.2.4 删除文章流程

用户登录系统后，选择自己的文章列表，在每条信息的最后，有一个操作按钮。点击操作按钮会出现，删除选项，点击后会在页面弹出提示框，是否删除，点击是，会调用后台的数据接口，对数据的该条信息进行删除操作，删除完成后会重新渲染文章列表的页面。具体的流程图如图 5-6 所示。

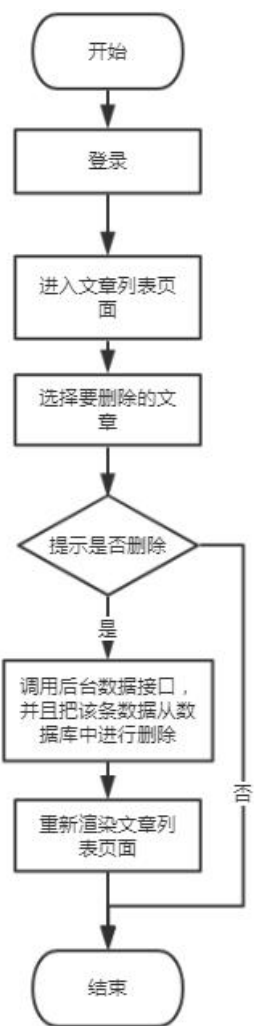


图 5-6 用户删除文章流程图

其他模块功能具体流程具体与该模块相差不多，在这里不再进行赘述了，具体参考文章模块相关的流程图。

第六章 个人空间系统测试

该章主要是对系统进行测试，以保证系统的正常运行。该部分分为功能模块测试，系统功能测试和性能测试三个部分。具体如下

6.1 功能模块测试

6.1.1 注册登录模块的等价类分类法测试

用户名和密码输入的情况主要有输入为空，输入不为空，输入合法，输入长度不合法几大类。昵称以及输入可能输入为空，或者不为空，为空信息不合法。个人介绍也是可以有两种情况，为空或者不为空，具体如表 6-1 所示。

表 6-1 基于个人空间系统注册登录模块等价类划分表

	有效等价类	标号	无效等价类	标号
用户名	输入合法	1	输入为空	6
			输入长度不符	7
密码	输入不为空	2	输入为空	8
	输入长度符合规则	3	输入长度不符合规则	9
昵称	输入不为空	4	输入为空	10
个人介绍	输入不为空	5	输入为空	11

注册登录模块测试用例及结果分析是将等价类划分的所有情况进行探讨，具体如表 6-2 所示：

表 6-2 基于个人空间系统注册登录模块测试用例以及结果分析表

	测试用例	结果分析
登录名	123456	符合要求
密码	123456	符合要求
用户名		用户名为空，不符合要求
密码		密码为空，不符合要求

(续) 表 5-2 基于个人空间系统注册登录模块测试用例以及结果分析表

	测试用例	结果分析
确认密码	111111	与密码不一致, 不符合要求
昵称	Sflk	符合要求
昵称		昵称无空, 不符合要求
个人介绍	Sjklfdkjl	符合要求
个人介绍		个人介绍为空, 不符合要求
用户名	12345	长度不符合
密码	12345	长度不符合

6.1.2 文章模块的等价类划分

用户发表文章时候, 需要输入的必须项是标题, 只有标题输入后才能进行下一步操作。划分的等价类如表 6-3 所示。

表 6-3 基于个人空间系统文章模块等价类划分表

	有效等价类	标号	无效等价类	标号
标题名称	输入不为空	1	输入为空	3
内容	输入不为空	2		

该部分主要针对文章的标题名称进行测试, 具体详情如表 6-4 所示。

表 6-4 基于个人空间系统文章模块测试用例以及结果分析表

	测试用例	结果分析
标题名称	输入不为空	符合要求
内容	输入不为空	符合要求
标题名称		标题为空不符合要求

6.1.3 相册模块的等价类划分

相册是与相片关联性较高的模块，其等价类只有相册名称，相册名称不为空则通过验证。具体详情如表 6-5 所示。

表 6-5 基于个人空间系统相册模块等价类划分表

有效等价类		标号	无效等价类		标号
相册名称	输入不为空	1	输入为空		2

该部分主要针对文章的标题名称进行测试，具体详情如表 6-6 所示。

表 6-6 基于个人空间系统相册模块测试用例以及结果分析表

测试用例		结果分析
相册名称	输入不为空	符合要求
相册名称	标题为空	不符合要求

6.1.4 说说模块等价类划分

说说部分里面的内容是主要部分，所以该部分主要对说说的内容进行了等价类划分，具体如表 6-7 所示。

表 6-7 基于个人空间系统说说模块等价类划分表

有效等价类		标号	无效等价类		标号
说说内容	输入不为空	1	输入为空		2

该部分主要对说说的内容进行测试，当用户输入数据时候，才会验证通过，否则样子会不能通过。具体如表 6-8 所示。

表 6-8 基于个人空间系统说说模块测试用例以及结果分析表

测试用例		结果分析
说说内容	输入不为空	符合要求
说说内容	标题为空	不符合要求

留言模块和评论模块与此类似，在这里就不在进行赘述了。

6.1.5 好友模块等价类划分

由于用户注册信息时候会对账号进行相关的验证，所以该部分的对好友账号的验证与用户名的等价类划分一致。具体详情如表 6-9 所示。

表 6-9 基于个人空间系统好友模块等价类划分表

	有效等价类	标号	无效等价类	标号
好友账号	输入不为空	1	输入为空	3
	输入长度符合规则	2	输入长度不符合规则	4

该部分主要针对好友账号进行测试。具体详情如表 6-10 所示

表 6-10 基于个人空间系统好友模块测试用例以及结果分析表

	测试用例	结果分析
好友账号	123456	符合要求
好友账号		输入为空不符合规则
好友账号	12345	输入长度不符合规则

结论

本篇论文的题目为个人空间系统，功能上实现了登录注册，文章的发表、编辑、查看、删除和评论，说说的发表、点赞等系列功能。程序的实现上使用了vue+express+mongodb，实现了前后端的通信。

本论文的的优点在于大胆尝试各种新兴技术，使自己对新型技术的了解进一步加深，能够使自己对后端的了解进一步加深。此次设计与传统的 JSP 页面相比，这次设计使用了组件化的思想，大大的提升了代码的重用性以及很大程度上降低了代码的耦合性。在性能发面，更多的使用 css3 的 translate 来减少回流，提高页面的加载速度。存在的问题就是本次论文编写时间过短，在一些功能上，不够完善，以及实现方式不是特别友好，我会在日后对此系统进行完善，并且能够在网上实现上线，让更多的人来使用此系统。

参考文献

- [1] 余久久, 章家卢. 基于 B/S 的社区幼儿园网站的设计与实现 [J]. 微型电脑应用, 2017, 33(10):13-16.
- [2] Aaron Tavistoek, DamonHougland. Essential. Net for web Professionals. Prentiee Hall Ptr, 2008.
- [3] Abraham Silberschatz, Henry F. Korth S. Sudarshn. Database Ststem Concepts[J]. Mc Graw-Hill ComPanies. 1999(4)13-20.
- [4] Kristina Cbodorow. MongoDB 权威指南[M]. 北京:人民邮电出版社. 2011. 05.
- [5] Dede E, Govindaraju M, Gunter D, et al. Performance evalua-tion of a MongoDB and Hadoop platfrm for scientific dataanalysis[C], Proceeding of the 4th ACM workshop on Scientif-ic cloud Computing. ACM, 2013:13-20.
- [6] Nicholas C, Zasks. JavaScript 高级编程 (第三版) [M]. 北京:人民邮电出版社. 2012. 03.
- [7] 阮一峰. ECMAScript6 入门[M]. 北京:电子工业出版社, 2017. 09.
- [8] BYVoid. Node. js 开发指南[M]. 北京: 人民邮电出版社. 2012. 07.
- [9] 陈涛. MVVM 设计模式及研究应用及其应用研究[J]. 计算机与数字工程, 2014(10):1982-1985
- [10] 程桂花, 沈炜, 何松林, 张珂杰. Node. js 中 Express 框架路由机制的研究[J]. 工业控制计算机, 2016, 29(08):101-102.

致谢

感谢钟颖莉老师在这段时间的精心指导，使用能够在规定时间之前完成毕业设计的相关功能，在对钟颖莉老师以及其他帮助过我的老师和同学提出感谢，帮助我答疑解惑，使我能够解决相关技术的难点。最后感谢学院以及学校对我的培养，让我成长进步。