

JIU JIANG UNIVERSITY

**毕 业 论 文**

题 目 云收藏的设计与实现

英文题目The Design and Implementation of

Cloud Collect

院 系 信息科学与技术学院

专 业 计算机科学与技术

姓 名 施诚

班级学号 A161341

指导教师 吕慧

二Ｏ一九年十二月

摘 要

随着信息时代的快速发展，人们日常接触到的信息愈来愈多，如何从这些繁乱的信息中选出自己所关注的信息，已经日益成为一大难题，人们迫切需要更现代、更便捷的方式来解决这一需求。本应用即是为了解决以上需求。

本应用专注于收藏的查询和管理，用户可以方便地对收藏进行创建、编辑、分类、搜索、备注、评论、查询等操作，另外，还提供了登录、注册、点赞、关注、链接复制等功能。

本应用由网页端和后台系统两部分构成，采用前后端分离的思想。前端采用TypeScript语言开发，使用到了Vue框架和Element UI框架，结合Vue Router、Vuex、Vue Class Component、Axios等技术。后台采用Kotlin语言开发，使用到了Spring Boot、Spring Web Mvc、Spring Data Jpa、Spring Security等框架，以及Swagger、Jwt等技术。本项目使用Maven作为构建工具，使用MySQL作为数据库。

本应用具有轻量、便捷等特点，能够满足收藏爱好者的一般需求。

**关键词：**收藏管理；书签管理

Abstract

With the rapid development of information age, people access to information more and more daily, how to choose their own focus from this messy information, has increasingly become a big problem. In addition, the traditional paper notebooks have been unable to meet the needs of collection of notes, recording knowledge, there is an urgent need to more modern, more convenient way to solve this demand. This application is in order to solve the above requirements.

This application focuses on query and management of collects. Users can easily create, edit, classify, remark, comment, search and do other operations on collects. In addition, it also provides functions such as login, register, thumb up, attention, links, etc.

This application is composed by a web client and a background system, following the thought of separation of frontend and backend. The frontend uses TypeScript as development language, along with Vue framework and Element UI framework, and in combination with technologies such as Vue Router, Vuex, Vue Class Component, and Axios. The backend uses Kotlin as development language, along with frameworks including Spring Boot, Spring Web Mvc, Spring Data Jpa and Spring Security framework, and technologies such as Swagger, etc. This project is developed by IntelliJ IDEA, and using Maven as the build tool, using Mysql as the database.

This application is light weight and convenient. It can satisfy the general needs of collection lovers.

**Keywords:** Collect management; Bookmark management

# 目 录

[摘 要 I](#_Toc26121849)

[Abstract II](#_Toc26121850)

[目 录 III](#_Toc26121851)

[1 绪论 1](#_Toc26121852)

[1.1 研究背景 2](#_Toc26121853)

[1.2 研究的目的和内容 3](#_Toc26121854)

[1.3 国内外应用发展现状 4](#_Toc26121855)

[1.4 论文结构 5](#_Toc26121856)

[1.5 本章小结 6](#_Toc26121857)

[2 系统分析 7](#_Toc26121858)

[2.1 需求分析 7](#_Toc26121859)

[2.2 可行性分析 7](#_Toc26121860)

[2.3 系统功能分析 8](#_Toc26121861)

[2.4 数据库需求分析 9](#_Toc26121862)

[2.5 开发环境及硬件需求 10](#_Toc26121863)

[2.6 本章小结 11](#_Toc26121864)

[3 系统设计 12](#_Toc26121865)

[3.1 系统总体设计 12](#_Toc26121866)

[3.2 系统详细设计 14](#_Toc26121867)

[3.3 系统数据库设计 15](#_Toc26121868)

[3.4 用户界面设计 21](#_Toc26121869)

[3.5 本章小结 24](#_Toc26121870)

[4 系统实现 25](#_Toc26121871)

[4.1 登录模块 25](#_Toc26121872)

[4.2 注册模块 26](#_Toc26121873)

[4.3 查询收藏 27](#_Toc26121874)

[4.4 评论收藏 28](#_Toc26121875)

[4.5 复制收藏链接 29](#_Toc26121876)

[4.6 创建收藏 29](#_Toc26121877)

[4.7 编辑收藏 30](#_Toc26121878)

[4.8 删除收藏 31](#_Toc26121879)

[4.9 用户档案系统 32](#_Toc26121880)

[4.1 搜索系统 33](#_Toc26121881)

[4.12 本章小结 34](#_Toc26121882)

[5 系统测试 35](#_Toc26121883)

[5.1 系统功能测试 35](#_Toc26121884)

[5.2本章小结 1](#_Toc26121885)

[总结与展望 2](#_Toc26121886)

[致 谢 3](#_Toc26121887)

[参考文献 4](#_Toc26121888)

# 1 绪论

如今，我们已经进入高速发展的信息时代。对于重视学习和记录的人们来说，妥善管理网络上的各种收藏，如网页、文章、笔记、教程、攻略等，已经成为一种迫切的需求。尽管浏览器附带的书签管理器提供了收藏管理的功能，且在部分功能上有所欠缺。至于云笔记应用，虽然同样满足了人们的需求，但它存在的目的并非是为了管理网络收藏。

传统的书签管理器主要提供了收藏的保存、编辑和分类功能，虽然这些功能已经相当完善，在收藏搜索、多用户互动等功能上，仍然存在许多不足之处。笔者认为，一款强大的网络书签工具，应该可以在不同设备之间自动同步稍后阅读列表，不仅要可以轻松分类整理，还要可以快速搜索查找内容[1]。在不同的用户之间，也应拥有一定的交互性。另外，无论是收藏分类还是收藏查询，都应提供多种方式，方便用户灵活使用。

基于网络信息的网络收藏，也可称为网络标签。其实网络书签又叫社会化书签, 是一种超链接网址的收藏和分享的社交性软件或网站[2]。用户可以收集、分类、编辑和查询自己专心的、感兴趣的网络信息，如网页、文章、图片等。在此之外，用户之间也能通过关注、点赞、评论、拷贝等方式，互相分享各自的网络收藏。网络收藏应当是公开的，所有用户都能对其进行检索。

网络收藏不同于一般的浏览器书签（收藏夹），除了基础的管理和查询功能之外，同样注重用户之间的交互性，如同博客一样。用户可以作为一个创作者而存在，也可以仅仅作为一个浏览者而存在。

分享网络收藏，分享见闻，分享知识。

## 1.1 研究背景

### 1.1.1 Vue

Vue[3]是一套用于构建用户界面的渐进式框架。Vue与其他大型框架不同，它被设计为能够自底向上逐层应用，其核心库只关注视图层，易于上手以及与第三方库（或者既有项目）整合。另外，当与现代化的工具链和各种支持类库结合使用时，Vue也完全能为复杂的单页面应用提供驱动。Vue利用MVVM设计模式和组件化的思想组织代码，拥有轻量级、简单易学、运行速度快等特点，得到了广大开发者的认可。

MVVM[4]全名Model-View-ViewModel，是一种改进自MVC的设计模式。在MVVM设计模式中，前端开发者只需关注数据，通过数据的变化改变DOM,，即视图的变化。在以往的开发模式中，开发者在改变数据时，往往需要手动操作DOM，不仅费时费力，同样也难以维护。MVVM通过数据的双向绑定，以及数据更新时界面的自动重新渲染，降低了前端的开发难度。

### 1.1.2 Spring Boot

Spring Boot是由Pivotal团队提供的一套崭新框架，它的设计目的是简化Spring应用的初始搭建和开发过程。Spring Boot让开发者不再需要定义样板化的配置，因为它使用一种特定的方式进行配置。可以说，Spring Boot其实并不是什么新的框架，它相当于Spring的脚手架，整合了Spring框架以及其他诸多框架。

Spring Boot开启了各种自动装配，开发者不需要编写各种配置文件，仅仅只是引入相关的依赖，就能快速搭建起一个web项目。

Spring Boot提出了“习惯优于配置”的观点，允许开发者极少地进行配置或者根本不用进行配置[5]。它对第三方框架的集成难度相对较低，深受后台开发人员的喜爱[6]。

### 1.1.3 TypeScript

TypeScript是由微软开发的一种自由的、开源的编程语言。TypeScript可以视为JavaScript的一个超集，支持ECMAScript 6标准。另外，TypeScript还从本质上添加了可选的静态类型以及基于类的面向对象编程。

TypeScript扩展了JavaScript的语法，因此任何现有的JavaScript程序，都可以不加改变地在TypeScript下工作。

TypeScript的设计目标是开发大型应用，它可以编译成纯JavaScript，且编译出来的JavaScript可以运行在任何浏览器上。

### 1.1.4 Kotlin

Kotlin是一种针对Java平台的新编程语言。Kotlin简洁、高效，并且专注于与Java代码的互操作性[7]。Kotlin与Java高度兼容，任何用到Java代码的地方，几乎都可以用Kotlin代码替代，例如：服务器开发、安卓开发，等等。Kotlin是一种跨平台的语言，因此也能替代JavaScript进行前端开发。Kotlin另外还提供了原生版本。

Kotlin相比Java更加灵活，提供了扩展方法、中缀方法、协程、数据类、函数式编程、领域特定语言等诸多特性，是一门有趣而实用的编程语言。

Kotlin是Java的一种替代语言。其主要设计目标如下：兼容Java。比Java更安全，能够静态检测常见的陷阱，如：引用空指针[8]。

## 1.2 研究的目的和内容

### 1.2.1 研究的目的

此课题要达到设计并实现“云收藏”应用，用户可以通过网页端进入“云收藏”应用，搜索和查看所有用户的收藏、分类、标签和个人资料等。当用户登录后，还可以对自己的个人信息进行管理，包括个人资料、收藏、喜爱的收藏、分类、关注用户、粉丝用户、通知、浏览记录等。其中，收藏管理是该应用最主要的功能。

该系统让用户在功能过多的云笔记应用和功能较少的书签管理应用之外，存在第三种选择，来收藏和管理自己所关注的网页和文章。该系统不仅提供了基本的收藏管理功能，还提供了诸如分类、标签、概述、评论、通知、浏览记录、链接复制等常见且常用的附加功能。对于那些拥有记录和收藏重要内容的习惯的用户来说，该系统已经足以满足他们的一般需求。

该系统的核心设计思想是简单化和轻量化，因此，该系统只专注于收藏管理以及必要的拓展功能，而不会添加那些冗余的、不必要的功能。在此基础上，为收藏爱好者们提供一个轻量、灵活而精致的收藏管理应用。

### 1.2.2 研究的主要内容

“云收藏”App，主要是通过基于Vue的前端系统和基于Spring Boot的后台系统，实现收藏管理功能。这是它的核心功能，另外还提供了分类管理、标签管理、用户个人资料管理、多用户互动方面的一些功能。该系统可以看作是云笔记应用的精简版，或是收藏管理器的扩展版。

收藏查询：用户可以根据数种方式，包括名字、分类、标签、类型等，对收藏进行全局查询，或是仅查询自己的收藏。

收藏管理：用户可以创建、分类、编辑和删除自己的收藏。另外，用户也可以评论收藏、点赞收藏、以多种方式复制收藏的链接等。用户也能拷贝他人的收藏。

分类、标签、通知、浏览记录的查询和管理：用户可以查询分类和标签，查询自己的通知和浏览记录，并管理自己的分类、标签、通知和浏览记录。

除此之外，本系统也拥有绝大部分系统都具有的登录注册系统和个人资料管理系统。

## 1.3 国内外应用发展现状

### 1.3.1 收藏管理应用现状

目前，市面上可用于收藏管理的应用主要包括多功能笔记类应用以及书签管理器两大类。

多功能笔记类应用功能强大、全面而丰富。目前，市面上比较流行的多功能笔记类应用包括印象笔记、有道云笔记、为知笔记等。其中最具代表性的印象笔记，于2008年正式发布，于2012年5月宣布进入中国市场，并在随后数年内不断发展，推陈出新。印象笔记的主要功能包括保持同步、剪辑网页、深度搜索、存储重要资料、支持第三方。相对于本应用要实现的需求来说，多功能笔记类应用虽然能够完全满足，但却显得过于臃肿，且操作复杂。

书签管理器由各种网页浏览器自带，用于对用户收藏的网页进行分类管理。谷歌的Chrome书签管理器便是其中的优秀代表，于2008年9月2日随Chrome浏览器一同发布，具有书签的添加、修改、删除、分类、自由排序、搜索以及导入导出等多种功能。书签管理器已能满足个人用户的一般需求，但其依赖于特定的浏览器，并非独立的应用，且在多用户互动方面功能缺乏，难以对其进行自定义和扩展。

## 1.4 论文结构

第1章 绪论：介绍了“云收藏”应用的研究背景和国内外的研究现状，并介绍了“云收藏”应用研究的目的和主要内容。

第2章 系统分析：主要说明了“云收藏”应用的需求分析、可行性分析以及数据库分析，同时也说明了开发环境的分析。

第3章 系统设计：主要说明了“云收藏”应用的总体设计，根据之前的需求分析绘制总体的功能图、流程图和ER图，罗列对应的属性图和数据表。最后，还描述了界面设计的风格，并给出了效果截图。

第4章 系统实现：主要说明了“云收藏”应用的主要功能的实现，并给出了对应的流程图。

第5章 系统测试：主要说明了“云收藏”应用的系统测试，并给出了对应的测试用例表。

余下部分：包含了此次毕业设计的总结与展望、致谢以及参考文献。

## 1.5 本章小结

本章主要介绍了“云收藏”应用的研究背景，阐述了该系统的研究背景和研究目的，明确了该系统的研究前景，并总结了该系统的研究内容。

# 2 系统分析

软件需求分析的任务是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求[9]。

## 2.1 需求分析

需求分析的阶段，我们需要了解的便是用户的需求，确定系统必须完成哪些工作，对目标系统做出完整、准确、清晰而具体的要求。

首先，该系统的核心功能即是收藏管理，用户应当能且仅能创建、编辑和删除自己的收藏。当用户创建和编辑收藏时，应当进行必要的参数验证；当用户删除收藏时，应当再次确认用户的操作。

其次，除了收藏管理之外，另一个主要功能即是收藏查询。用户应当能通过多种方式查询收藏，也应当能仅限查询属于自己的收藏。

除了收藏管理和收藏查询这两个主要功能之外，为了让该系统更加便捷和完善，也需要提供类似目录、标签、点赞、评论、关注等常见的功能。当然，登录注册功能和个人资料管理功能也是必不可少的。

考虑到用户的收藏本质上是一个带有额外信息的网站链接，本系统还提供了链接复制和链接跳转的功能。

## 2.2 可行性分析

可行性分析是项目开始阶段必须进行的一项工作。可行性分析是指通过一套准则，在项目正式开发之前，投入一定的时间和精力，从经济、技术、社会等方面，对项目的必要性、合理性、可能性以及开发过程中可能遇到的风险做出分析和评价，得出项目是否可行的结论[10]。可行性分析是项目开发中必不可少的准备工作之一，它能帮助我们更好地开发项目，减少问题和麻烦的发生。可行性分析主要包括以下三个方面。

### 2.2.1经济可行性

“云收藏”应用是一款Vue单页面应用，后台采用Spring Boot开发。由于使用到了许多较新的技术，开发这款应用需要一定的学习成本和时间成本。不过，这款应用的基本功能需求并不复杂，在不考虑一些比较难以实现的扩展功能的前提上，无论是前端页面还是后台系统，在时间和精力的开支上，仍然是可以接受的。

### 2.2.2技术可行性

本系统虽然与书签管理器有颇多相似之处，但由于附带了一些多用户互动功能，在实现上其实更加贴近博客、论坛之类的系统。尽管在开发时可以参考网络上的个人博客系统的实现，但是考虑到本系统使用到了许多新技术，这些参考大部分并无多大价值。总的来说，对于本系统而言，仍然需要独立完成相当一部分的代码。

### 2.2.3操作可行性

本系统的界面设计虽然较为简单，但是胜在页面齐全、导航完善、层次分明。本系统中包含诸多链接和按钮，用户可通过它们进行路由跳转、页面显示格式切换和后台操作。因此，本系统同样具有足够的交互性。

## 2.3 系统功能分析

本系统的主要功能可以按照登录注册模块、收藏模块、分类模块、标签模块、用户档案模块、搜索模块进行划分。

登录注册模块的功能如下。

（1）登录功能：用户可以点击首页或导航栏的登录按钮登录账户。

（2）注册功能：用户可以点击首页或导航栏的注册按钮注册账户。注册资料基于用户名和邮箱地址，且要求是合法的。

（3）找回密码：用户遗忘密码时，可以在登录或注册表单中点击该按钮，输入自己的邮箱地址重置密码。

收藏模块功能如下。

（1）查询功能：用户可以通过诸多方式查询收藏，也可仅查询自己的收藏。允许进行模糊查询和分页查询。可以对查询结果进行排序。

（2）管理功能：用户可以创建、编辑和删除自己的收藏。

（3）链接跳转功能：用户可从收藏总览或收藏详情页面跳转到对应的链接。

（4）链接复制功能：用户可以通过多种方式复制收藏的链接。

（5）跳转功能：用户可从收藏页面跳转到对应的分类、标签或用户页面。

分类模块和标签模块的功能非常相似，它们的功能如下。

（1）查询功能：用户不仅可以查询自己的分类和标签，也可以查询他人的分类和标签。允许进行模糊查询和分页查询。可以对查询结果进行排序。

（2）管理功能：用户可以创建、编辑和删除自己的分类和标签。

（3）跳转功能：用户可从分类和标签的详情页查看到相关收藏，从而跳转到相关页面。

用户档案模块的功能如下。

（1）主页：用户可以查看和管理自己的个人资料。

（2）收藏：用户可以查看和管理自己的收藏，并对其进行过滤。

（3）喜爱的收藏：用户可以查看自己喜爱的收藏，并对其进行过滤。

（4）分类：用户可以查看和管理自己的分类。

（5）关注用户：用户可以查看关注自己的用户，并关注或者取消关注他们。

（6）粉丝用户：用户可以查看自己关注的用户，并关注或者取消关注他们。

（7）通知：用户可以查看自己的通知。

（8）浏览历史：用户可以查看自己的收藏浏览历史，并删除或清空历史记录。

搜索模块的功能入校：

（1）搜索收藏：用户可以根据数种方式搜索收藏。

（2）搜索用户：用户可以根据数种方式搜索用户。

## 2.4 数据库需求分析

根据系统需求，需要建立以下几张表。

收藏表：包括收藏的名字、概述、链接、分类id、评论id、用户id、类型和时间信息等。

分类表：包括分类的名字、概述、用户id和时间信息等。

标签表：包括标签的名字、概述、用户id和时间信息等。

评论表：包括评论的内容、收藏id、回复评论id、发起用户id和时间信息等。

浏览历史表：包括收藏id、用户id和时间信息等。

通知表：包括通知的标题、内容、用户id和时间信息等。

用户表：包括用户的用户名、密码、邮箱、昵称、头像地址、简介和时间信息等。

此外，考虑到实体类之间存在一些多对多关系，还需建立收藏-标签表、收藏-点赞用户表、用户-关注用户表等，以建立必要的级联。

## 2.5 开发环境及硬件需求

该系统的开发环境需求如下。

（1）开发工具：IntelliJ IDEA。

（2）SDK：JDK11。

（3）前端依赖：TypeScript、Vue、Axios等。

（4）后台依赖：Kotlin、Spring Boot、Spring Security、Redis等。

（3）其他工具：Maven、MySQL、Redis等。

（3）操作系统：Windows 10。

该系统的硬件需求如下。

（1）电脑内存：8G以上。

（2）电脑CPU：双核及以上。

**IDEA是一款功能强大且极具扩展性的Java IDE，可以大大提高项目的开发效率[12]。相比Eclipse、MyEclipse等开发工具，可谓没有缺点，只有优点。Kotlin是一款基于JVM的优雅而实用的语言，相比Java拥有诸多的特性，搭配IDEA能够进一步提高项目的开发效率。TypeScript可以视为带有类型系统的JavaScript，可以有效避免编写代码时产生的诸多低级错误。**

**考虑到项目使用到了许多新技术，项目的软件版本需求和硬件需求比较高。**

## 2.6 本章小结

**本章主要讨论了系统的需求分析，首先确定了该系统的需求和可行性，接着进一步分析了整个系统的功能，然后分析了数据库的设计，最后分析了开发环境和硬件的需求。**

# 3 系统设计

系统设计概括了系统的设计思路，包括总体流程、路由逻辑、实体关系、数据库、页面设计等方面。

## 3.1 系统总体设计

### 3.1.1 总体功能设计

本系统由前端和后台两部分组成，前端按照组件化思想进行开发，后台则采用经典的MVC模式进行开发。前端通过发送http请求调用后台的API，从而进行相关的数据库操作，然后将状态码和数据返回给前端。

本系统采用Vue + Spring Boot的方式进行开发，拥有相当大的优势。这种开发方式一方面能够提供较为美观的用户界面，另一方面能够带来良好的用户体验，同时还能提高项目的开发效率[11]。

本系统主要包括登录注册、收藏、分类、标签、用户档案、搜索等模块。如图3-1所示。



图3-1 总体功能图

### 3.1.2 实体-联系图

本系统的实体主要包括收藏、分类、标签、评论和用户。用户拥有属于自己的收藏、分类、标签和评论，而收藏也拥有自己的分类、标签和评论。另外用户还有它的关注用户和点赞收藏，评论还有它的回复评论。大部分实体都拥有自己的名字和概述，但是命名略有不同。所有实体都拥有自己的创建时间和最后修改时间（除非该实体是只读的）。如图3-2所示。

图3-2 E-R图

### 3.1.3 总体流程设计

本系统的操作流程是由路由主导的。当用户来到系统首页后，用户可以通过顶级导航栏跳转到收藏、档案、搜索等页面，也可以直接进行登录或注册。当用户登录以后，方可进入档案页面，通过次级导航栏跳转到不同的页面，对自己的收藏、分类等进行进一步的查看和管理。大多数导航都会跳转到某个总览页面，通过总览页面又可跳转到详情页面。如果系统出错，还会跳转到对应的错误页面。

总体来说，当用户未登录时，用户可以查看和搜索绝大部分数据。当用户登录后，用户还可以查看、搜索和管理自己的数据。如图3-3所示。

图3-3 系统业务流程图

## 3.2 系统详细设计

登录注册模块：主要包括登录和注册两个功能。

收藏模块：包括了收藏的展示、过滤、查询、评论、点赞、链接复制和链接跳转等功能。此外，用户可以创建、编辑和删除自己的收藏。

分类模块：包括了分类的展示和查询功能。此外，用户可以创建、编辑和删除自己的分类。

标签模块：包括了标签的展示和查询功能。此外，用户可以创建、编辑和删除自己的标签。

用户档案模块：包含了用户的首页、收藏、喜爱的收藏、关注用户、粉丝用户、浏览记录、通知等部分。用户可对这些数据进行查看和管理。

搜索模块：包含了收藏搜索和用户搜索两大部分。用户可以根据多种条件查询这些数据，允许进行模糊查询，也允许进行分页查询。

## 3.3 系统数据库设计

本系统选用Spring Data Jpa作为后台的持久层框架，这个框架的一大特点便是不需要开发者手动设计数据库，而是根据项目中的实体类自动创建数据表。相比传统方式，Spring Data Jpa将开发人员从繁琐的数据访问层中解放出来，仅需少量代码即可实现对数据库的访问，从而能够更专注于业务逻辑的开发，提高了项目的开发效率[13]。因此，本系统设计数据库的过程，实际上也就是编写实体类代码的过程。这些代码可以写的非常简单，不过，考虑到性能因素，本系统仍然通过注解的方式，对一些字段的可空性、长度等进行了必要的限制。主键为Long类型的数据，以自增的方式自动生成。另外，本系统还使用到了Spring Data Jpa的审计功能，每个数据表都包含创建时间和最后修改时间（除非对应的实体是只读的）这两个字段。当用户执行创建和修改操作时，将会自动更新这两个字段。

（1）收藏表

收藏具有id、名称、概述、分类id、类型、链接、所属用户id等信息。收藏的属性图如图3-4所示。



图3-4 收藏的属性图

根据图3-4可定义出收藏的数据表collect。如表3-1所示。

表3-1 收藏表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| created\_time | datatime(6) | N/A | 创建时间 |
| last\_modified\_time | datatime(6) | N/A | 最后修改时间 |
| logo\_url | varchar(64) | N/A | 图标地址 |
| name | varchar(64) | N/A | 名字 |
| summary | varchar(255) | N/A | 概述 |
| type | varchar(255) | N/A | 类型 |
| url | varchar(255) | N/A | 链接地址 |
| category\_id | bigint(20) | 外键 | 分类id |
| user\_id | bigint(20) | 外键 | 用户id |

（2）分类表

分类具有id、名字、概述、用户id等信息。分类的属性图如图3-5所示。



图3-5 分类的属性图

根据图3-5可定义出分类的数据表category。如表3-2所示。

表3-2分类表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| created\_time | datatime(6) | N/A | 创建时间 |
| last\_modified\_time | datatime(60 | N/A | 最后修改时间 |
| name | varchar(64) | N/A | 名字 |
| summary | varchar(255) | N/A | 概述 |
| user\_id | bigint(20) | 外键 | 用户id |

（3）标签表

标签具有id、名字、概述、用户id 等信息。标签的属性图如图3-6所示。



图 3-6 标签的属性图

根据图3-6可定义出标签的数据表tag。如表3-3所示。

表3-3 标签表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| created\_time | datatime(6) | N/A | 创建时间 |
| last\_modified\_time | datatime(60 | N/A | 最后修改时间 |
| name | varchar(64) | N/A | 名字 |
| summary | varchar(255) | N/A | 概述 |
| user\_id | bigint(20) | 外键 | 用户id |

（4）评论表

评论具有id、内容、创建时间、收藏id、回复评论id、发起用户id等信息。评论的属性图如图3-7所示。



图 3-7 评论的属性图

根据图3-7可定义出评论的数据表comment。如表3-4所示。

表3-4 评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| content | varchar(512) | N/A | 内容 |
| created\_time | datatime(6) | N/A | 创建时间 |
| collect\_id | bigint(20) | 外键 | 收藏id |
| reply\_to\_comment\_id | bigint(20) | 外键 | 回复评论id |
| sponsor\_by\_user\_id | bigint(20) | 外键 | 发起用户id |

（5）浏览历史表

浏览历史具有id、创建时间、收藏id、用户id等信息。浏览历史的属性图如图3-8所示。



图 3-8 浏览历史的属性图

根据图3-8可定义出浏览历史的数据表history。如表3-5所示。

表3-5 浏览历史表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| created\_time | datatime(6) | N/A | 创建时间 |
| collect\_id | bigint(20) | 外键 | 收藏id |
| user\_id | bigint(20) | 外键 | 用户id |

（6）通知表

通知具有id、内容、创建时间、标题、类型、用户id等信息。通知的属性图如图3-9所示。



图 3-9 通知的属性图

根据图3-9可定义出通知的数据表notice。如表3-6所示。

表3-6 通知表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| content | varchar(512) | N/A | 内容 |
| created\_time | datatime(6) | N/A | 创建时间 |
| title | varchar(255) | N/A | 标题 |
| type | varchar(255) | N/A | 类型 |
| user\_id | bigint(20) | 外键 | 用户id |

（7）用户表

用户具有id、头像地址、邮箱、简介、昵称、密码、注册时间、角色、更新时间、用户名、激活状态等信息。用户的属性图如图3-10所示。



图 3-10 用户的属性图

根据图3-10可定义出用户的数据表user。如表3-7所示。

表3-7 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 主键/外键 | 说明 |
| id | bigint(20) | 主键 | 编号 |
| avatar\_url | varchar(255) | N/A | 头像地址 |
| email | varchar(64) | N/A | 邮箱 |
| introduce | varchar(255) | N/A | 简介 |
| nickname | varchar(64) | N/A | 昵称 |
| password | varchar(255) | N/A | 密码 |
| register\_time | datatime(6) | N/A | 注册时间 |
| role | varchar(255) | N/A | 角色 |
| update\_time | datatime(6) | N/A | 更新时间 |
| username | varchar(16) | N/A | 用户名 |
| activate\_status | bit(1) | N/A | 激活状态 |

（8）其他

考虑到实体之间的多对多关系，另外还有收藏的标签表collect-tags、用户的关注用户表user\_follow\_to\_users，以及用户的点赞用户表user\_praise\_to\_collects。这里不再多加说明。

## 3.4 用户界面设计

一款软件的成功与否，很大程度上在于它的界面设计。拥有美观而流畅的界面的软件，虽然并不能直接体现其技术含量，也不能由此直接判断是否符合需求，但却能为用户带来良好的操作体验，从而提供吸引力和用户粘性。合理化的UI设计不仅可以展示出软件的自身特色，也可以让软件操作过程呈现出舒适化、简单化与自由化的特点[14]。

### 3.4.1 整体风格与路由逻辑

由于本系统采用了Element-UI框架，在界面设计上并未花费太多功夫。不过，Element-UI框架虽然提供了许多开箱即用的页面组件，但在页面排版、字体格式等方面仍需花费一些功夫。总的来说，本系统的页面风格简洁朴素、清晰明了，没有过多的动画效果。本系统的数据大多采用卡片的形式进行展示，其中包含不少链接和按钮，满足用户交互性的需要。而表单大多以对话框的形式提供，由于不需要进行页面跳转，能为用户提供更加流畅的操作体验。

本系统的页面之间通过路由相互联系，路由是本系统前端的重要组成部分。当用户进行该系统时，用户首先会来到本系统的首页。通过点击顶级导航栏中的不同导航，将会分别来到首页、档案页、收藏页、搜索页、关于页等页面。当用户点击收藏页后，用户将会进入收藏总览页，点击收藏列表中的某一收藏，则会进入收藏详情页。当用户点击档案页后，将会要求用户登录。如果已登录，则会来到用户的档案首页。通过点击次级导航栏中的不同导航，将会分别来到用户档案的首页、收藏页、喜爱的收藏页、关注用户页、粉丝用户页、浏览浏览历史页等页面，其中又可以进行进一步的交互。当用户点击搜索页后，用户将会进入搜索页面，接着可以进入收藏搜索页和用户搜索页。当用户未登录时，点击页面右上角的登录/注册按钮，将会分别打开登录/注册对话框。当用户已登录时，右上角将会出现用户档案的下拉菜单以及注销按钮，通过它们可以进入指定的用户档案页面，或是注销用户。当系统发生错误时，则会跳转到对应的错误页面。

### 3.4.2 首页页面设计

本系统的首页中并未包含太多的页面元素，除了页面头部和页面尾部之外，作为页面主体的部分中，仅包含显眼的应用图标和标题，以及登录和注册按钮。除此之外还提供了一个随便看看按钮，当用户点击以后，将会跳转到随机的收藏详情页面。由于本系统的大部分页面，都包含共同的页面头部和页面尾部，如果用户想浏览其他页面，用户自然可以通过页面头部中的导航栏进行跳转。另外，页面顶部的导航栏是固定的，当用户下拉页面到一定高度后，用户可通过点击页面右下角的回到顶部组件，回到页面的顶部。如图3-11所示。

### 3.4.3 总览页面设计

本系统中的绝大部分页面都可视作总览页面，这类页面以卡片列表的方式展示了多条数据，附带功能完善的分页导航，方便用户进行分页跳转、指定每页页数、上一页、下一页等分页操作。根据不同的功能需求，可能还允许用户进行搜索、过滤、新建、删除等操作。



图3-11首页界面设计

以用户档案的收藏页面为例。该页面的主题部分由操作按钮栏、过滤按钮栏、卡片组三部分组成。用户可通过点击操作按钮栏中的不同按钮，创建收藏、分类或标签；可通过点击过滤按钮栏中的按钮，按照分类、标签等条件过滤自己的收藏，或是重置过滤条件。在收藏卡片中，用户可以通过点击收藏名字，跳转到对应的收藏详情页面，也可以通过点击相应的按钮，直接复制收藏链接，或是打开收藏链接。这里并未显示收藏的所有数据，用户可能需要来到收藏的详情页面，以便对自己的收藏做进一步的查看和管理。如图3-12所示。

### 3.4.5 详情页面设计

收藏、分类、标签……这些数据都拥有属于自己的详情页面，以对应的详情卡片为主。由于功能需求的不同，不同数据的详情页面之间也存在一些微妙的区别。

这里仍以收藏的详情页面为例。收藏详情卡片以整洁统一的排版，详尽展示了该收藏的相关数据。某些数据是可以交互的，例如，用户可点击收藏所属用户的用户名，跳转到该用户的用户档案首页。除此之外，这里同样提供了链接复制和链接跳转的按钮，方便用户进行相应的操作。如果当前用户是该收藏的所属用户，详情卡片下方将会出现编辑收藏和删除收藏按钮，允许用户进行相应的操作。最后，在页面的最下方，则是收藏的评论列表，用户可以在此查看、创建、回复和删除评论。这一部分是可以折叠的。如图3-13所示。

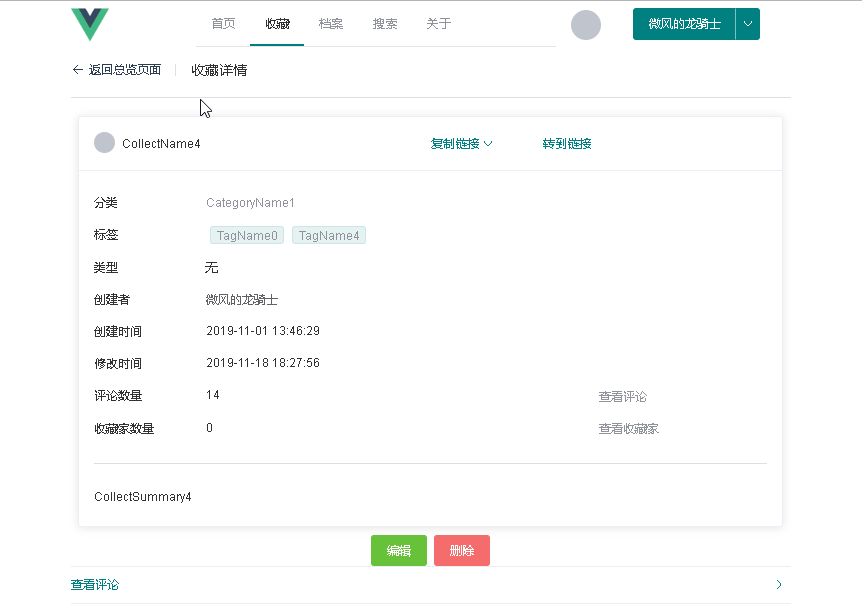


图3-12总览界面设计

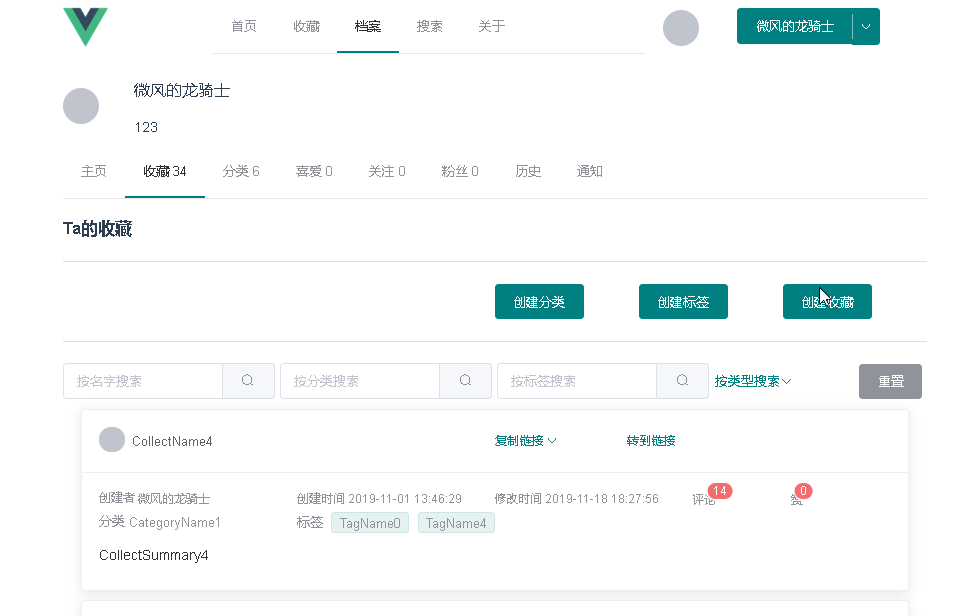


图3-13详情界面设计

## 3.5 本章小结

本章主要叙述了系统的整体设计，首先讨论了总体的功能设计，然后则是整个系统的详细设计，最后分别讨论了数据库设计和界面设计。一个好的项目的完成，必然离不开一个充分而完善的整体设计的支持。总体设计和详细设计可谓基础中的基础，而对应后台的数据库设计和对应前端的页面设计，则显得同样重要。

# 4 系统实现

系统实现是软件开发的重要环节之一。系统实现的重点在于系统的代码逻辑和算法设计，以求达到更好的系统性能和更好的代码结构。

## 4.1 登录模块

在登录模块中，用户可以使用已注册的用户名和密码进行登录。当用户打开登录对话框，输入必要的数据，并点击登录按钮之后，前端将会发送http请求到后台，调用后台的登录方法。

图4-1 登录功能的流程图

如图4-1所示，当用户登录时，系统后台将会根据用户名和密码生成一个UsernamePasswordAuthenticationToken，然后使用authenticationManager对其进行验证。如果通过验证，则将其存储到SecurityContextHolder中，并返回用户信息到前端，然后前端提示登录成功，并且关闭登录对话框。如果未通过，则会抛出对应的异常到前端，然后前端提示对应的错误消息。当成功登录后，前端会再次向后台请求根据登录信息生成JWT令牌，然后存储到前端的Vuex Store中。此后，只要未注销用户，当发送http请求时，前端都会在请求头中加上这段JWT令牌。后台根据JwtFilter解析令牌，判断用户是否已登录，从而保存用户的登录状态。

## 4.2 注册模块

本系统的注册模块是基于邮箱的。用户在注册，需要输入合法的用户名、密码、确认密码、邮箱和昵称，然后才能点击注册按钮。如果表单输入不合法，将会显示对应的错误提示。点击注册按钮后，前端将会发送http请求到后台，调用后台的登录方法。

图4-2 注册功能的流程图

如图4-2所示，当用户注册时，后台首先会进行一次参数验证，如果验证失败，例如发生用户名重复、邮箱重复等情况，则会抛出异常到前端，前端接着会显示相应的错误消息。如果验证失败，则会将用户数据存储到数据库中，接着注册成功，前端将会转到登录对话框。如果发生意外错误，则会跳转到对应的错误页面。

## 4.3 查询收藏

本系统中对于收藏的查询功能算不上强大，查询算法也并不复杂，但是已经能够满足多种情况下的需要。用户可以通过名字、分类、标签、类型等多种方式查询收藏，也可以仅限查询自己的收藏。另外，本系统支持模糊查询和分页查询，也可以对查询结果进行排序。得益于此，本系统的查询功能兼具灵活性和性能。根据用户查询收藏（即展示用户的收藏）、根据点赞用户查询收藏（即展示收藏的点赞者）等特殊的查询方式，由于不能即时改变查询内容，不认为包含在本系统的查询功能当中。

图4-3 查询收藏功能的流程图

如图4-3所示，当用户选择了查询方式（选择查询输入框前端的下拉菜单中的下拉项，或者使用不同的查询输入框）并输入了查询内容之后，点击查询按钮，系统后台就会调用对应的查询方法，并传入对应的查询内容和分页参数，从数据库查询内容并进行分页，然后返回给前端。接着，数据库就会将这些分页后的数据显示出来。如果查询结果为空，视情况的不同，可能会不显示任何数据，也可能会显示“查询结果为空”之类的提示文字。

本系统基于Spring框架的Page类，实现了查询结果的分页功能。本系统中所有的多结果查询都是分页查询，当用户成功查询到数据后，页面就会显示对应的分页栏。通过分页栏，用户就能进行上一页、下一页、页面跳转、切换每页显示页数等操作。

## 4.4 评论收藏

当用户进入收藏的详情页面之后，用户就可以在页面下方查看该收藏的评论列表。如果用户已登录，则可以发表新的评论，也可以回复某一评论。点击创建评论或回复评论按钮之后，系统就会打开创建评论对话框。用户接着需要做的事情，就是输入一些评论内容，然后点击发送。

图4-4 评论收藏功能的流程图

如图4-4所示，用户可以通过点击收藏列表右上方的创建评论按钮，或者任意收藏卡片中的回复评论按钮，打开一个创建评论对话框。输入不为空且在字数范围内的内容，点击发送按钮，就会发送http请求到后台。系统后台会调用创建评论方法，创建相应的评论。等待后台操作完成后，系统前端还会即时刷新页面中的数据。

## 4.5 复制收藏链接

点击收藏总览卡片或者收藏详情卡片中的复制链接按钮，即可打开复制链接下来菜单。这时，用户可以按自己的需要选择一种复制方式，点击相应的下拉项。



图4-5 复制收藏链接的流程图

如图4-5所示，当选择了复制方式并点击下拉项之后，前端就会发送http请求到后台，后台调用对应的复制收藏链接方法，位于服务层的此方法又会调用工具类UrlGenerator中的相关方法，从而得到要生成的链接字符串，并返回给前端。前端又会通过Vue插件clipboard2，将该链接字符串加入系统的剪贴板。接着，用户就能如普通的粘贴操作一样，粘贴出对应的链接字符串。

## 4.6 创建收藏

在用户的收藏页面，用户可以通过点击页面上方的创建收藏按钮，弹出创建收藏对话框，从而创建自己的收藏。这时将会进行前端参数验证，只有输入数据合法时，用户才能点击创建收藏按钮，真正进行创建收藏操作。否则将会显示错误消息。当用户创建收藏时，用户可以通过输入关键字，查询并指定收藏的分类和标签。不同之处在于，收藏仅能拥有最多一个分类，但能同时拥有多个标签。

图4-6 创建收藏的流程图

如图4-5所示，当用户点击完成创建按钮之后，系统就会发送http请求到后台。系统后台在通过参数验证后，就会将收藏数据存储到数据库中。创建成功后，与创建评论功能类似，系统会关闭创建收藏对话框，然后即时刷新页面数据。

## 4.7 编辑收藏

在收藏的详情页面中，当用户为收藏的创建者时，页面下方就会显示编辑收藏按钮。点击打开编辑收藏对话框，在输入了合法的数据之后，即可进一步点击完成编辑按钮。编辑收藏时，后台还会就收藏是否重复进行参数验证。

图4-7 编辑收藏的流程图

如图4-7所示，用户可以编辑自己的收藏。具体的操作与创建收藏类似，这里不再赘述。

## 4.8 删除收藏

在收藏的详情页面中，当用户为收藏的创建者时，页面下方就会显示删除收藏按钮。点击删除收藏按钮，再次点击确认后，就会执行删除收藏操作。

如图4-8所示，系统后台在受到http请求后，就会通过调用删除收藏方法，删除数据库中的对应数据。接着，如果用户已登录，系统会跳转到用户的档案首页，否则跳转到收藏总览页。如果删除失败，则会显示提示信息。



图4-8 删除收藏的流程图

## 4.9 浏览历史功能

在用户已登录的情况下，当用户进入某个收藏的详情页面时，系统会自动为该用户添加一条浏览记录。需要注意的一点是浏览历史是不可重复的，对于同一个收藏，不管用户浏览了多少次，数据库中仅会保存最近的一条浏览记录。浏览历史记录了浏览的收藏的id和上次浏览时间。

用户可以通过进入自己的档案页面，选择浏览历史导航，查看自己的浏览历史列表。在这个页面中，用户可以删除浏览历史、清空浏览历史，或者从浏览历史跳转到对应的的收藏详情页面。

添加浏览历史的流程图如图4-9所示。



图4-8 删除收藏的流程图

## 4.10 用户档案系统

用户档案系统基于本系统的次级导航栏。次级导航栏上方展示了用户的头像、昵称和简介，部分导航的文字后会附带统计文字，即这类数据的总数。

点击不同的导航，可以分别进入用户的首页、收藏页、喜爱的收藏页、分类页、关注用户页、粉丝用户页、浏览历史页、通知页等页面。

用户可在首页中打开编辑个人资料对话框，编辑自己的个人资料；可在收藏页、喜爱的收藏页、分类页等页面中，按照不同的条件过滤数据；可在收藏页、分类页、浏览历史页、通知页等页面中，管理自己的数据。当前用户与显示用户档案的用户不一致时，不能编辑个人资料，也不会显示浏览历史、通知这两项数据。系统结构如图4-9所示。

图4-9 用户档案系统的流程图

## 4.12 搜索系统

搜索系统可用于搜索收藏和用户。当用户进入搜索页面时，用户可通过点击搜索用户按钮和搜索收藏按钮，分别进入对应的页面。当用户进入搜索收藏页时，用户可通过名字、分类名、标签名等方式查询收藏。当用户进入搜索用户页面时，用户可通过昵称、用户名、邮箱等方式查询用户。允许进行模糊查询和分页查询。可以对查询结果进行排序。

搜索系统是对其他模块的查询功能的一个整合，包含了本系统中常见的几种查询功能，本身并未添加多少新的功能。系统结构如图4-10所示。



图4-10 搜索系统的流程图

## 4.12 本章小结

本章主要阐述了部分系统主要功能的实现方式，结合流程图对其进行了详细而富有逻辑性的说明。为了提高代码的利用效率，尽可能地统一了功能的实现思路，因而某些功能在实现方式上颇具相似性。这也大大提高了项目的开发效率。

本系统的其余功能与上述功能的实现方式大多相似，在此不再赘述。

# 5 系统测试

软件质量是贯穿软件生存期的一个极为重要的问题, 是软件开发过程中所使用的各种开发技术和验证方法的最终体现[15]。

## 5.1 系统功能测试

### 5.1.1 登录注册功能

测试登录注册时，需要对表单进行验证。无论是未通过前端的，还是未通过后台的，都需要弹出相应的验证错误提示。当用户注册成功时，需要转到登录对话框。当用户登录成功时，需要关闭登录对话框。具体的测试用例如表5-1所示。

表5-1登陆注册功能的测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 1 | 测试项目 | | 云收藏应用 |
| 测试目的 | 验证云收藏应用的功能是否满足需求 | | | |
| 预置条件 | 打开云收藏应用的网站主页 | | | |
| 测试步骤 | 1.点击首页右上角的注册按钮，打开注册对话框  2.输入用户名TestUser、密码123456、邮箱294\*\*\*41a@163.com和昵称TestUser（以上数据合法且不重复）  3.点击注册按钮  4.在登录对话框输入之前的用户名TestUser和密码123456  5.点击登录按钮 | | | |
| 预期结果 | 1.提示注册成功，来到登录对话框  2.提示登录成功，关闭登录对话框，回到首页 | | | |
| 测试结果 | 注册成功，并且用于注册的用户名和密码能够顺利登录 | | | |
| 测试是否通过 | ☑是□否 | | | |
| 测试人员 | 施诚 | 测试日期 | 2019年11月20日 | |

进行注册时，如果用户输入了非法的数据，页面会进行提示，要求用户重新输入。只有用户输入的数据合法时，用户才能点击注册按钮。如果用户名或邮箱重复，在点击注册按钮后，用户仍会收到相应的错误提示。如果注册成功，页面会弹出提示并打开登录对话框。进行登录时，如果用户输入了合法的数据，并且能在后台通过用户名和密码查询到对应的用户，则登录成功，并关闭登录对话框。否则登录失败，用户也会收到相应的错误提示。

### 5.1.2 用户的个人资料编辑功能

用户可以修改自己的昵称和简介，但不能修改自己的用户名、邮箱和密码。每个用户的用户名和邮箱都是唯一且固定的。另外，由于密码经过加密，如果用户忘记了密码，或者需要修改密码，必须通过点击登录注册表单中的忘记密码按钮，根据提示重置自己的密码。具体的测试用例如表5-2所示。

表5-2用户个人资料编辑功能的测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 2 | 测试项目 | 云收藏应用 |
| 测试目的 | 验证云收藏应用的功能是否满足需求 | | |
| 预置条件 | 打开云收藏应用的网站主页 | | |
| 测试步骤 | 1.通过路由跳转、搜索或者收藏列表  2.点击导航栏中的档案导航，跳转到用户的个人资料主页  3.点击页面下方的编辑资料按钮  4.在编辑个人资料对话框中，修改自己的昵称和简介  5.输入合法的数据后，点击下方的完成编辑按钮 | | |
| 预期结果 | 1.成功打开编辑个人资料对话框  2.在编辑个人资料对话框中修改输入数据时，作为背景的用户个人资料主页中的数据不会随之发生变化  3.点击完成编辑按钮后，将会提示编辑成功，并关闭编辑个人资料对话框，用户个人资料主页中的数据也会随之发生变化 | | |

续表5-2用户个人资料编辑功能的测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试结果 | 成功编辑用户个人资料 | | |
| 测试是否通过 | ☑是□否 | | |
| 测试人员 | 施诚 | 测试日期 | 2019年11月20日 |

用户的个人资料编辑功能虽然比较简单，由于涉及到页面的显示与切换，实际实现中仍然需要考虑不少问题。例如，用户仅能编辑自己的一部分资料，需要验证用户的表单输入，当编辑时显示的数据不能随之变化，当完成编辑后显示的数据则需要即时发生变化。总的来说，这是一个需要考虑诸多细节的功能。

### 5.1.3 收藏的创建、编辑和删除功能

由于收藏的查询和管理是本系统的核心功能，收藏的创建、编辑和删除自然是本系统必不可少的功能之一。用户可以查看所有人的收藏，但仅能管理自己的收藏。当用户并非收藏的所属用户时，用户的收藏页面中不会显示相应的创建按钮，收藏详情页面中也不会显示相应的编辑和删除按钮。具体的测试用例如表5-3所示。

表5-3收藏管理功能的测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 3 | 测试项目 | 云收藏应用 |
| 测试目的 | 验证云收藏应用的功能是否满足需求 | | |
| 预置条件 | 打开云收藏应用的网站主页 | | |
| 测试步骤 | 1.登录云收藏应用  2.点击跳转到用户的收藏页面  4.点击页面右侧的创建收藏按钮  5.在创建收藏对话框中，输入合法的数据，点击创建按钮  6.在收藏列表中，找到之前创建的收藏，点击进入详情页面  7.点击页面下方的编辑按钮，输入合法的数据，点击完成编辑按钮  8.在收藏列表中，找到之前创建的收藏，点击进入详情页面  9.点击页面下方的删除按钮，接着点击确认按钮 | | |

表5-3收藏管理功能的测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 预期结果 | 1.成功创建收藏  2.成功编辑收藏  3.成功删除收藏，但在删除之前，需要再次进行确认  4.操作成功后，收藏列表中的数据也会发生相应的变化 | | |
| 测试结果 | 可以比较顺利地创建、编辑和删除收藏。操作成功后，收藏列表也会发生相应的变化。 | | |
| 测试是否通过 | ☑是□否 | | |
| 测试人员 | 施诚 | 测试日期 | 2019年11月20日 |

虽然这项功能顺利通过了测试，但是仍然存在一些不尽人意之处。例如，虽然当用户删除了某个收藏之后，收藏列表需要即时发生变化，但本系统在这里是采用重新加载后台数据的方式实现的，而这可能会导致一些性能问题。也许仅仅删除相应的前端数据会是更好的选择。

另外，也可以考虑添加一些动画效果，让页面的改变和切换显得更加流畅，为用户提供更好的浏览体验。例如，当删除收藏时，收藏列表中的该项会被向右移除并消失，收藏列表会在之后重新闭合到一起。

也可以考虑添加一项扩展功能，当用户确认删除之后，在接下来的一段时间内，仍然允许用户恢复被删除的数据。

### 5.1.4 收藏的链接复制功能

收藏的链接复制功能，本质上是一种字符串处理功能。后台系统接收指定收藏的数据，通过某种格式转化为指定类型的链接。如果仅仅只是原样复制收藏的链接的话，那么链接复制功能的实现可谓相当简单了。不过，在本系统中，用户能够将收藏复制为多种类型的链接，如Html链接、Markdown链接等。可以说，本项功能能够完全满足用户“复制收藏为链接”的需求。具体的测试用例如表5-4所示。

表5-4收藏的链接复制功能的测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 4 | 测试项目 | | 云收藏应用 |
| 测试目的 | 验证云收藏应用的功能是否满足需求 | | | |
| 预置条件 | 打开云收藏应用的网站主页 | | | |
| 测试步骤 | 1.通过直接路由跳转或者查询的方式，，进入某个收藏的详情页面  2.点击复制链接按钮，在下拉菜单中，点击复制链接按钮  3.粘贴链接到地址栏  4.点击复制链接按钮，在下拉菜单中，点击复制为Html链接按钮  5.粘贴链接到地址栏 | | | |
| 预期结果 | 1.成功粘贴链接到地址栏，并且粘贴的是原样链接  2.成功粘贴链接到地址栏，并且粘贴的是合法且包含完整收藏信息的Html链接 | | | |
| 测试结果 | 成功依次粘贴了原样链接和Html链接。 | | | |
| 测试是否通过 | ☑是□否 | | | |
| 测试人员 | 施诚 | 测试日期 | 2019年12月20日 | |

收藏的链接复制功能，是本系统较有特色的功能之一，能够根据需要将收藏的信息复制为各种类型的链接，方便用户日后引用。这里需要注意的是，无论是在复制链接时，还是在创建时，本系统都不会对链接的有效性和合法性进行验证。因为用户也可以将一个文件路径作为链接，然后在另一个Html或Markdown文件中引用它。另一个需要注意的地方，则是当收藏的某些属性不存在时，最后生成的链接不能包含相应的字符串片段，以免引发空指针异常，或者导致链接格式错误。

## 5.2本章小结

本章主要说明了“云收藏应用”的系统测试，系统测试能够帮助我们发现一些低级的错误和浅显的问题，以便解决它们。因此，本章挑选了一些重要的功能，进行了基本的测试，从而排查了一些潜在的问题。

# 总结与展望

本次毕业设计是采用前后端分离的思路制作的，前端使用TypeScript + Vue，后台则使用Kotlin + Spring Boot。相对来说，这个项目的代码量并不大，系统也并不复杂，但是仍然很有挑战性，且颇有难度，于个人而言，是一次极好的项目实践。

一方面，该项目使用到了许多时下流行的技术（例如Vue和Spring Boot），由于学校并没有开设相关的专业课，因此不得不自主学习这些技术；而当掌握了基础知识后，在实战开发的过程中，往往又会遇到一些常常被忽略的问题。

另一方面，该项目使用到了一些新技术（例如TypeScript和Kotlin），且某些技术（例如Vue Class Component），网络上并没有完善的教程，当遇到一些没有被总结过的问题时，往往又需要花费成倍时间和精力才能妥善解决。

由于以上几点原因，虽然在项目开发之前，曾设想过不少有特色的功能，但是随着时间、精力以及耐心的消耗，最终都没有完全实现。不过，项目的基础功能和一些比较简单的特色功能，例如对收藏、分类、标签、用户的增删改查的功能，多用户互动的评论、关注、点赞等功能，收藏的浏览记录功能、收藏的复制链接功能，最终还是符合预期地实现了。这些功能的后代代码足够简洁、规范，算是个人比较满意的地方，足以作为日后项目的参考。前端页面虽然进行过良好的重构（即将页面拆分为多个Vue组件），满足代码规范，但是样式和排版仍然存在很多不足之处。以后可能还需再作一些调整，让页面更加美观。

这个项目仍然存在许多可以完善的地方，也有一些可以考虑添加的特色功能。由于时间以及个人能力的限制，并没有将其逐一解决，希望在以后的工作中，能够更进一步，将这个做得更好，成为GitHub上拥有较多收藏人数的项目之一。这次毕业设计虽然消耗了不少时间和精力，但是自己也从中学习到了不少知识，总结了不少问题的解决方案，另外还确定了个人的代码规范，可谓受益匪浅。

# 致 谢

此次毕业设计，是对自己的四年大学生涯的一次总结，也是对自己所学知识的一次全面的考核。相对来说，这个项目的代码量并不大，系统也并不复杂，但仍然很有难度，可以说是正式工作之前的一次入门测验，重点考察了自己所掌握的知识和和技术、项目实战能力以及解决问题的能力，为以后的工作打下了一个良好的基础。然而，这样的准备恐怕还远远不够，以后还需多加努力。

此次毕业设计的顺利完成，需要特别感谢各类网络资源以及指导老师吕慧的答疑解惑。感谢Vue、Spring Boot、TypeScript、Kotlin的官方参考文档，让我可以快速上手对应的技术，掌握基础知识。感谢网络上的某些博客、demo项目的作者，让我得以理清思路，解决许多让人十分困扰的疑难问题。另外，也感谢吕慧老师对于论文编写的指导。虽然此次项目是自己独立完成的，但是对于其中的部分难题，没有他人帮助的话，恐怕还需花费很多时间才能妥善解决。

最后，再次感谢网络和周围的朋友、老师的力量，为此次毕业设计带来的莫大帮助。

# 参考文献

[1] 跨平台云端书签[J]．电脑迷，2015（12）：37．

[2] 一鸣．独乐乐不如众乐乐的网络书签[J]．互联网天地，2008（11）：76．

[3] 李广宏．vue.js前端应用技术分析[J]．中国新通信，2019，21（20）：115．

[4] 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．

[5] 江雁．浅谈Spring Boot框架下如何快速进行后台开发[J]．海峡科技与产业，2019（02）：131~132．

[6] 彭灿华，韦晓敏，魏士伟．J2EE项目开发与设计[M]（第二版）．北京：中国铁道出版社，2016：8．

[7] 德米特里（俄），斯维特拉娜（俄）．Kotlin实战[M]．北京：电子工业出版社，2017：3．

[8] JetBrains开源其Kotlin语言，基于JVM的新编程语言[J]．硅谷，2012（04）：77．

[9] 徐赛华．软件需求分析研究[J]．吉林师范大学学报（自然科学版）．2006，04（01）：104~105．

[10] 卫红春．软件工程概论[M]．北京：清华大学出版社，2007：72~74．

[11] 杨妍．基于Spring Boot与Vue的系统管理模块开发探究[J]．电声技术，2019，43（02）：32~34．

[12] 陈立兵．IntelliJ IDEA——开发人员利器[J]．程序员，2009（01）：118~120．

[13] 傅莞龙，张传武，彭安金．使用Spring Data和JPA在JavaEE系统中简化持久层[J]．电子世界，2017（06）：88~89+91．

[14] 樊鸿烨，关玉欣．基于网页设计中UI界面的应用[J]．电子技术与软件工程，2019（02）：10．

[15] 刘敏．软件的质量保证[J]．计算机光盘软件与应用，2013，16（18）：313