

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。项目使用IntelliJ IDEA作为开发工具，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](#_Toc25691551)

[Abstract II](#_Toc25691552)

[目 录 III](#_Toc25691553)

[1 绪论 1](#_Toc25691554)

[1.1 研究背景 1](#_Toc25691555)

[1.2 研究的目的和内容 3](#_Toc25691556)

[1.3 国内外应用发展现状 4](#_Toc25691557)

[1.4 论文结构 5](#_Toc25691558)

[1.5 本章小结 5](#_Toc25691559)

[2 系统分析 6](#_Toc25691560)

[2.1 需求分析 6](#_Toc25691561)

[2.2 可行性分析 6](#_Toc25691562)

[2.3 系统功能分析 7](#_Toc25691563)

[2.4 数据库需求分析 8](#_Toc25691564)

[2.5 开发环境及硬件需求 9](#_Toc25691565)

[2.6 本章小结 10](#_Toc25691566)

[3 系统设计 11](#_Toc25691567)

[3.1 系统总体设计 11](#_Toc25691568)

[3.2 系统详细设计 13](#_Toc25691569)

[3.3 系统数据库设计 14](#_Toc25691570)

[3.4 用户界面设计 17](#_Toc25691571)

[3.5 本章小结 20](#_Toc25691572)

[4 系统实现 21](#_Toc25691573)

[4.1 登录模块 21](#_Toc25691574)

[4.2 注册模块 22](#_Toc25691575)

[4.3 查询收藏 23](#_Toc25691576)

[4.4 评论收藏 23](#_Toc25691577)

[4.5 复制收藏链接 24](#_Toc25691578)

[4.6 创建收藏 24](#_Toc25691579)

[4.7 编辑收藏 26](#_Toc25691580)

[4.8 删除收藏 27](#_Toc25691581)

[4.9 编辑个人资料 28](#_Toc25691582)

[4.10 用户档案系统 29](#_Toc25691583)

[4.11 搜索系统 31](#_Toc25691584)

[4.12 本章小结 31](#_Toc25691585)

[5 系统测试 32](#_Toc25691586)

[5.1 系统功能测试 32](#_Toc25691587)

[5.2本章小结 37](#_Toc25691588)

[总结与展望 38](#_Toc25691589)

[致 谢 39](#_Toc25691590)

[参考文献 40](#_Toc25691591)

# 1 绪论

**[TODO]**

近几年来，移动无线互联网的不断发展，早在2010年12月，手机上网用户达3.03亿，占所有的互联网用户的66.2%[1]。随着经济的发展，越来越多人使用移动设备，包括手机、ipad和微型计算机。而人们进入Inter网的主要工具也逐渐从电脑变成了手机。随着平面设计、UI设计等艺术设计的不断发展，许多人开始从网上获取设计资源、素材和学习资料。设计师们在网上讨论设计的优缺点和创作的灵感，但这些信息交流网站大部分都是基于PC端的，所以基于Android端的设计交流平台市场目前来说还是一片空白。

在于移动通讯迅速发展之后，网络与用户对于移动设备的要求也越来越高，而Symbian，Windows Mobie，PalmOS这些平台过于封闭，无法很好地符合用户的需求，所以市场急切需要一个很强的开放性平台。经过前几年的发展，第三代通信（3G）获得了大多数人的接收，紧接着（4G）也跟着普遍开来，3G和4G的迅速发展带来了更快的传输速率。所以，移动终端的需求和应用软件将会有很大的发展空间。2007年11月Google公司推出了一款专为移动端设计的软件平台--Android。Android平台有着真正意义上的开放性，由Linux和Java构成的开源软件，它允许所有的开发商和个人在其发表的Android的基础上进行再次开发。Android在拥有开放性一系列特性之后促进了移动端技术的创新，同时还有助于开发商和个人降低开发成本，以及方便运营商根据自己的特色去开发拥有自己特色的产品。综上所述，Android平台拥有强大的市场。

## 1.1 研究背景

### 1.1.1 Vue

**[TODO]**UI就是User Interface（用户界面）的简称。主要就是指对软件的人机及交互、操作逻辑、界面美观的整体设计[2]。随着电子产品的飞速发展，UI设计在软件开发当中处于越来越重要的位置，好的UI设计可以使整个软件变得更加具有个性和品位，还让用户对于软件的操作变得更加舒适、简单和自由，充分体现软件的定位和特点。一个好的软件界面设计就好像工业产品中的工业造型一样，是产品的一个重要的卖点。一个电子产品拥有美观的界面会给用户带来舒适的视觉享受，可以拉近人与商品的距离，这是建立在科学之上的艺术视觉设计。检验一个界面的好坏的标准不是某个项目的开发组领导的意见也不是项目成员的投票结果，而是终端用户的感受。

传统的UI设计师职能包括三大方面，一是图形设计，主要是按美术角度来思考界面的美观性。二是交互设计，主要在于设计软件的操作流程、树状结构、操作规范等。一个软件产品在编码之前就需要报交互设计做好，并且确立交互的模型和交互规范。早期的交互设计主要关注的还是人与机器，尤其是与计算机的交互，其主要目的是通过提供简单易懂的操作界面使计算机和其他新兴的数字产品能够为更多的普通用户所接受[3]。三是用户测试和研究，这里指的“测试”是测试交互设计的合理性及图形设计的没关系。

### 1.1.2 Spring Boot

**[TODO]**Android 是 Google 开发的基于 Linux 平台的开源手机操作系统，它是一个移动终端平台，包括了移动电话工作所需的全部软件，由操作系统、中间件、 UI设计（ UserInterface ）及应用程序组成[4]。号称是首个为移动终端打造的真正开放和完整的移动软件[5]。

根据IDC2016年的最新全球智能手机市场分析报告显示，安卓的市场份额自从2012年来一直遥遥领先位于第一。在最新的2016年第二季度的安卓的占有率达到了87.6%之多。可见安卓市场的强大，并具有一定的稳定性。相对去安卓操作系统的其它操作系统iOS和Windows目前来说占有一定的份额但是对于安卓操作系统来说占有的比例比较小，已经很难再超越安卓的地位并取而代之。虽然安卓操作系统占有的份额波动比较大，相比较来说，其他的操作系统表现的则要平稳更多。

### 1.1.3 TypeScript

**[TODO]**

Bmob是用户开发一个云存储的应用软件，给应用软件快速添加一个安全且灵活的后台管理系统，方便终端保存各种信息。传统的网络数据存储、用户管理、消息推送等功能都需要数据库以及服务器的支持，个人开发者要是想在 App 中加入这些功能，就必须学习 Java 、 xml语言之外的其他语言，例如 .net 、 php 、 spq ， 增加了开发App 的难度，并且搭建服务器也额外增加了开发成本。Bmob 云平台简化了这一过程，使开发者能够将更多的精力投向软件功能的改善和界面的美化设计，使应用的开发变得更加简单[6]。对于数据的安全，除了在服务器上部署上的安全处理之外，采用了多租户的虚拟隔离模式，即任何一个开发者的流量变化或者数据变化，都不会给其他的用户造成影响。

### 1.1.4 Kotlin

**[TODO]**

## 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：用户可以根据多种方式，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，用户也可以拷贝收藏、评论收藏、点赞收藏、以多种方式复制收藏的链接等。

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

软件需求的工作，即是深入地描述软件的性能和功能，确定软件设计的边界和软件同其他元素的接口，定义软件的其他有效性要求[8]。

## 2.1 需求分析

需求分析的阶段，我们需要了解的是用户的需求，一个受用户喜欢的应用，必然需要满足用户的真正需求。软件需求分析的任务，即是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，本系统还系统了链接跳转和链接复制功能。

## 2.2 可行性分析

可行性分析也叫可行性研究，它是所有工程项目在开始阶段必须要进行的一项工作。可行性分析是指在项目正式开发之前，先投入一定的精力，通过一套准则，从经济、技术、社会等方面对项目的必要时、可能性、合理性，以及项目所面临到的重大风险进行分析和评价，得出项目是否可行的结论[9]。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，从而避免一些不必要的问题。可行性分析一般包含以下的三个方面。

### 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：用户忘记密码时，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：用户可以通过多种方式查询收藏，也可仅查询自己的收藏。

（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。

（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面查看到相关收藏，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：用户可以根据多种方式搜索收藏。

（2）搜索用户：用户可以根据多种方式搜索用户。

## 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，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，首先明确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

概述内容……

## 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，前端通过发送Http请求调用后台API，从而进数据库操作。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。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等基本信息。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 等基本信息。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 等基本信息。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 等基本信息。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 等基本信息。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 等基本信息。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、激活状态等基本信息。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，但却能够为用户带来良好的操作体验，提供莫大的吸引力。

### 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。

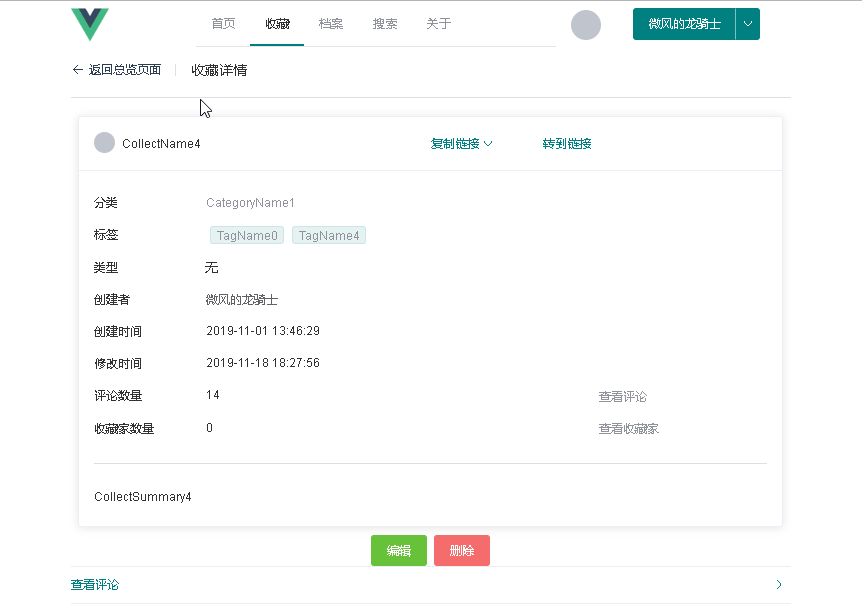


图3-12总览界面设计

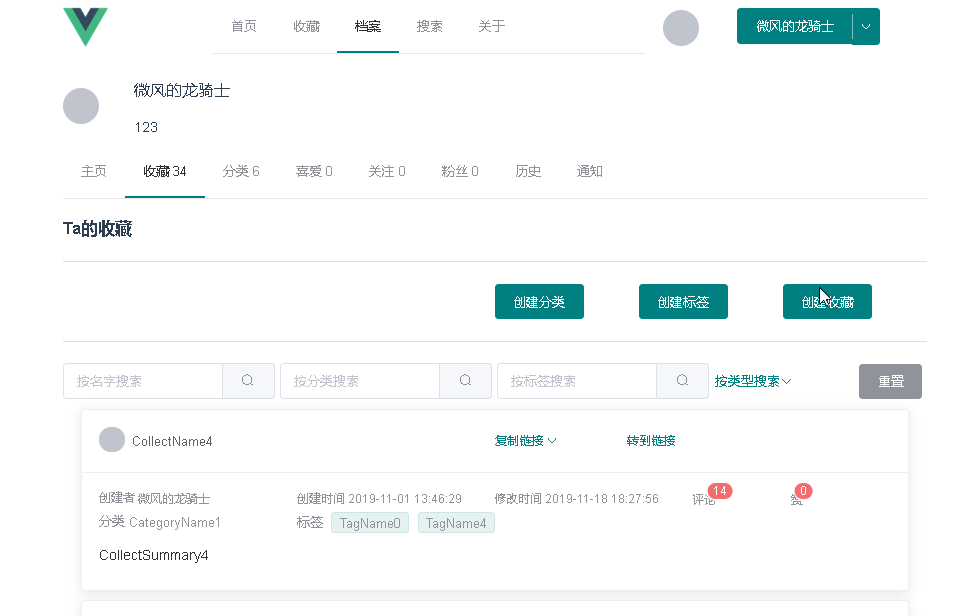


图3-13详情界面设计

## 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

对于软件开发而言，系统实现是一个非常重要的环节。系统实现是对系统中算法的设计，并且按照已经设计好的算法去实现系统的功能。这些功能模块虽然不算复杂，但是若是想让这个系统有更好的运行效果的性能，就必须设计最佳的算法。

## 4.1 登录模块

在登录模块，用户需要使用自己事先注册好的账号的密码进行登录，当用户点击登录按钮之后，相关的后台会将用户输入的数据与数据库的内容做一个对比，如果账号和密码同时匹配成功就可以登录系统进入到主界面。凡是有不存在该账号或者密码和账号不匹配的问题出现，系统都会给出相应的提示。

图4-1 客户端登录流程

如图4-1所示，登录的时候，Android端会根据Bmob SDK提供的查找接口查找账号是否存在和查找账号对应的密码是否跟输入的账号密码可以同时匹配，如果可以，则可登录。

## 4.2 注册模块

用户在注册的时候，可以按照一般软件的注册方法，首先输入自己现在使用的手机号，然后后台会将验证码发送到输入的手机号中，用户在接收了验证码之后就可以将验证码输入验证码的输入框中，最后输入自己需要设置的密码点击注册就可以为这个号码注册一个账号。



图4-2 客户端注册流程图

如图4-2所示，注册的时候，Android端输入一个正确的手机号码，通过后台将数据提交给服务器，服务器将验证码发送给该号码。用户输入验证码后后台会将输入的内容和发送的验证码做对比，如果正确则在数据库中创建一个新的用户。当然用户如果不按提示去输入无效的手机号码，系统也会给出相应的提示。一个用户已经存在也无法进行再次注册。

## 4.3 查询收藏

本系统中作品的搜索可以按照很多中方式来搜索，虽然目前只可以按照现有的字段进行准确的查询，但是这个查询绝对不是简单的作品名字查询，当然，用户也可根据自己喜欢的设计师的名字来查询。模糊查询虽然尚未完成，但是模糊查询可以涵盖作品当中所有的字段，比如作品的分类和作品的解释部分。关键在于模糊查询有更好的查询效率，不需要用户完全记住自己需要查询的内容。



图4-3 客户端作品搜索流程图

如图4-3所示，进行作品查询的整体过程是通过客户在搜索框中输入想要进行搜索的内容，客户端将要查找的字符以Bmob SDK提供的搜索接口去服务器中进行模糊查找，查找在分类、详细分类、标题、作者名和提示这些列中是否含有用户输入的内容，如果匹配成功就将匹配的内容显示到下面的列表之中。如果没有一个匹配，则会显示暂时无此内容来进行一个友好的提示。

## 4.4 评论收藏

TODO

## 4.5 复制收藏链接

真正地去下载一个作品对于用户移动端用户来说，手机的内存是远远不够的。所以这里的下载并不是真正意义上的下载，而是作者已经将作品素材打包在下载的文件当中并以云盘的方式生成了一个链接。用户点下载的时候实际上是跳转到了云盘的链接当中，在新的网页链接中，用户可以选择存入自己的云盘，也可以选择下载到本地。



图4-4 客户端作品下载流程图

如图4-4所示，在作品下载的这个功能中，主要的是要从作品的信息中获取到作者提供的作品的素材的下载地址也就是云盘的地址，用户在点击下载之后，后台会将云数据库中的数据调取出来。并将这个链接以网页的形式打开，获取到云盘的地址并在客户端里将这个云盘的网页链接打开。客户可以在打开的云盘链接中选中下载到本地还是保存到云盘，多项选择，让设计师们可以通过云盘在电脑上获取自己想要的信息和素材。

## 4.6 创建收藏

作品的购买这个功能主要针对需要付费的作品下载，一些作品或者素材在发表的时候设计师就设定了一定的下载支付费用。所以在针对某些下载需要费用的作品，本系统就增加了作品购买这个功能。用户点击购买之后会跳转到支付宝中支付相应的费用之后才会出现一般作品出现的云盘链接。

图4-5客户端作品购买流程图

如图4-5所示，作品的购买功能主要是在作品下载前判断该作品是否收费，如果是免费则直接跳转到下载功能，如果显示收费则要进行收费。在收费的过程中我们选择通过Bmob接口来实现支付宝的收费，Bmob在收费之后会返回是否收费成功的信息，若支付成功则用户可以继续进行下载的逻辑功能。若支付不成功则返回到支付的界面继续支付，当然用户也可以返回再从头再次进行支付或者取消当前的支付。取消当前支付可以直接关闭该界面。

分类模块和标签模块中的相关功能与该功能十分相似，在此不再赘述。

## 4.7 编辑收藏

用户在上传作品的时候可以根据自己将要上传作品的性质选择不同的分类，并且设置好该作品是免费还是收费，如果是收费的还需要设置收费的具体费用是多少。在输入完自己的标题以及作品介绍并为之选择好了相应的图片介绍之后，用户就可以上传自己的作品。如果上传的过程中没有问题，在点击上传之后就会有相应的提示。



图4-6客户端作品上传流程图

如图4-6所示，作品的上传实在Android客户端中进行的，用户可以按照自己的需求上传不一样的作品，也可以对作品进行免费或者收费的定义，也就是在客户端输入一个作品必要的信息后，我们可以按照不同的需求添加不一样的信息，并将这些信息打包好并发送给Bmob云后台，在Bmob云数据库的相应的作品表中建立相应的信息，并将建立成功与否的信息返回给客户端以告知用户上传的数据是否上传成功。

分类模块和标签模块中的相关功能与该功能十分相似，在此不再赘述。

## 4.8 删除收藏

作品的删除功能主要是由用户也就是作品的上传者决定的，作者可以在个人中心找到自己发表过的作品，在这个界面作者可以点击删除按钮删除自己想要删除的作品。



图4-7 客户端作品删除流程图

如图4-7所示，系统在删除的时候会在人机交互的基础上提示用户是否需要删除自己的作品，确认之后系统会将该条数据的信息通过Bmob云后台的SDK接口传递给云后台并找到相应的数据条目，并在数据库中的作品表中删除这一条数据。最后云后台会及时返回是否删除成功的信息，若返回成功则代表已完成数据的删除。若返回失败则返回浏览界面并提示用户删除失败 。用户在删除失败之后可以再次点击删除按钮。

分类模块和标签模块中的相关功能与该功能十分相似，在此不再赘述。

## 4.9 编辑个人资料



图4-8 客户端作品修改流程图

如图4-8所示，作品的修改功能是作者在浏览自己发表过的作品的时候，若觉得信息有误或者需要更改而不想删除作品信息的时候可以根据自己的需要更改作品的信息，更改作品的信息时需要输入数据，然后系统将数据提交给服务器，服务器查询到相应的数据条目时，将更新的数据存储到作品表中。成功将数据库的数据更新之后会返回相应的信息，成功则会返回成功，失败则返回失败。并同样会返回到个人作品的浏览界面。

## 4.10 用户档案系统

在个人信息的功能中集成了当前用户的个人信息的修改、发表记录的浏览和购买记录的浏览。



图4-9 客户端个人信息流程图

如图4-9所示，在个人信息修改中用户输入自己的信息后，系统会根据该用户的登录名在服务器的数据库中找到该作者的ID，然后系统再调用服务器的接口将用户输入的数据更新在作者表中并返回相应的更新信息。客户端根据服务器返回的信息来提示用户是否更新成功个人信息。浏览发表记录就是在云后台中调用接口将数据库中所有作者为当前用户的作品全部查找出来并返回给客户端，客户端在界面中显示返回的数据。

## 4.11 搜索系统

系统的设置和普通的软件设置都差不多，一般就是版本更新与意见反馈等主要的软件设置。所以系统测试是比较简单，更注重对于系统来说是比较实用的功能。既然是系统的设置，就在一定程度上表现的是系统的功能。

图4-10 客户端个人信息流程图

如图4-10所示，在系统设置功能当中，用户可以根据自己的需求来设置自己的系统，在系统设置中最主要的就是可以退出当前用户的登录和更新最设计的版本。辅以用户用来发表对系统的意见的意见反馈和查看用户协议的用户协议以及关于最设计的介绍等等。

## 4.12 本章小结

本章主要介绍了系统功能及实现几个主要的功能是如何实现的，并用了图文结合的方式细致阐述了每一个功能是如何实现的，以及后台的运行机制是如何运作的。一个功能完善的系统需要的是周密又完整的算法设计，而这些算法设计就需要完整的流程图来表现出来。

# 5 系统测试

软件测试是指使用人工或自动手段来运行或测试某个系统的过程，其目的在于检验被测试系统是否满足规定的要求或弄清预期的结果与实际结果之间的差别[15]。

## 5.1 系统功能测试

### 5.1.1 登录注册功能

登录注册功能是该应用的基本功能之一。在测试登录注册时，需要对表单进行验证。当表单未通过前端参数验证时，需要弹出相应提示，并禁止用户点击提交按钮。当表单未通过后台参数验证时，需要弹出相应提示，并让用户的此次操作无效。当用户注册成功时，需要打开到登录对话框。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

本系统的重点不在于用户的个人资料管理功能，尽管如此，本系统仍然实现了基本的用户个人资料编辑功能。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 | | | |
| 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.点击导航栏中的档案导航，跳转到用户的个人资料主页  3.点击次级导航栏中的收藏导航，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，找到先前创建的收藏，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，找到先前创建的收藏，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 | | | |
| 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，收藏列表中的数据将会随之发生变化 | | | |
| 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。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。

### 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（如：badu.com）  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（如：<a href=”baidu.com” title=”百度一下，你就知道”>百度</a>） | | | |
| 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。这是考虑到用户也可以将一个文件的相对或绝对路径作为链接，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（其值允许为null，且此时为null），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] [罗军舟](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFQ&sfield=au&skey=%e7%bd%97%e5%86%9b%e8%88%9f&code=06616573;26679602;06620387;)，[吴文甲](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFQ&sfield=au&skey=%e5%90%b4%e6%96%87%e7%94%b2&code=06616573;26679602;06620387;)，[杨明](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFQ&sfield=au&skey=%e6%9d%a8%e6%98%8e&code=06616573;26679602;06620387;).移动互联网:终端、网络与服务[J].计算机学报，2011，12（11）：2019~2051.

[2] 屠秀栋.浅谈UI设计[J].电脑知识与技术，2010，36（07）：1706~1707.

[3] 辛向阳.交互设计:从物理逻辑到行为逻辑[J].装饰，2015，12（01）：58~62.

[4] 陈璟，陈平华，李文亮.Android内核分析[J].现代计算机，2009，12（11）：112~115.

[5] 宋小倩，周东升.基于Android平台的应用开发研究[J].软件导刊，2011，12（02）：104~106.

[6] 周冉，高玉竹.Bmob云平台在Android App开发中的应用[J].微型机与应用，2015，24（01）：26~28.

[7] 候长海.2016年上半年智能手机市场分析[J].互联网天地，2016，10（10）：74~76.

[8] 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.

[9] 卫红春.软件工程概论[M].北京：清华大学出版社，2007.72~74.

[10] 杜巍.Android App开发中Bmob云平台的运用[J].通讯世界，2016，23（03）：220~221.

[11] [惠燕](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFD&sfield=au&skey=%e6%83%a0%e7%87%95&code=10171900;09043127;09064462;)，[潘煜](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFD&sfield=au&skey=%e6%bd%98%e7%85%9c&code=10171900;09043127;09064462;)，[徐光辉](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFD&sfield=au&skey=%e5%be%90%e5%85%89%e8%be%89&code=10171900;09043127;09064462;).数据库设计范式[J].现代电子技术，2003，24（24）：107~108 .

[12] 毛昕，赵倩[雲](http://www.cnki.net/kcms/detail/search.aspx?dbcode=CJFQ&sfield=au&skey=%e8%b5%b5%e5%80%a9%e9%9b%b2&code=31285873;31285874;).浅析移动终端设备的扁平化设计风格[J].设计，2014，12（06）：

85~86.

[13] 罗仕鏊，朱上上.用户体验与产品创新设计[M].北京：机械工业出版社，2010.145.

[14] 李勇. 模块化设计在用户界面设计教学中的应用[J].设计，2016，22（12）：128~129.

[15] 周元哲.软件测试基础[M].西安：西安电子科技大学出版社，2011.19~20.