

# 郑州大学毕业设计（论文）

题    目：Artery 开发者社区 APP 的设计与实现

指导教师：  李晓宇   职称：  副教授  

指导教师(校外)：  张创伟   职称：  研发经理  

学生姓名：  李伟   学号：  20162430211  

专    业：                    软件工程                    

院（系）：                    信息工程学院                    

完成时间：                    2020 年 6 月 5 日                    

2020 年 6 月 5 日

## Artery 开发者社区 APP 的设计与实现

**摘要：**Artery 是北京华宇信息技术有限公司自主研发的基于 SpringBoot 的前后端分离的框架。根据公司的具体业务类型，Artery 框架对 SpringBoot 进行了进一步的封装，将基本的权限管理、组织机构管理、用户管理封装到了框架中，使用户能够更加专注于业务的开发。同时 Artery 还提供了配套使用前端框架 artery-ui。artery-ui 是基于 Vue 进行开发，提供了丰富的组件库，使用户能够更快的开发出一套漂亮的网页。Artery 开发者社区 APP 为 Artery 开发者提供了一个分享和交流的平台。App 基于 Android 操作系统，采用 Retrofit + RxJava + OkHttp 和 MVP 代码结构，服务端使用 Artery 进行开发，同时使用 MySQL 进行数据存储，使用 ElasticSearch 进行全文检索。Artery 开发者社区主要实现了使用 Cocall 账户进行登录、使用 markdown 格式编辑文章或问题、分类查看文章或问题、全文检索文章以及高亮关键字、评论文章、置顶文章、收藏文章、关注用户、设置精华文章等功能。上述功能满足了 Artery 开发者在使用 Artery 的过程中交流使用心得的基本需求，同时可以加快完善 Artery 的生态系统，吸引更多的人使用 Artery。

**关键字：**Artery；社区；SpringBoot；Android；App；Retrofit；RxJava；OkHttp

# Design and Implementation of Artery Developer Community App

**Abstract:** Artery is a front-end separation framework based on SpringBoot independently developed by Beijing Huayu Information Technology Co., Ltd. According to the company's specific business type, the Artery framework further encapsulates SpringBoot, encapsulating basic rights management, organization management, and user management into the framework, so that users can focus more on business development. At the same time Artery also provides the supporting front-end framework artery-ui. artery-ui is developed based on Vue, providing a rich component library, enabling users to develop a beautiful set of web pages faster. The Artery Developer Community APP provides a platform for Artery developers to share and communicate. The app is based on the Android operating system and uses Retrofit + RxJava + OkHttp and MVP code structure. The server uses Artery for development, MySQL for data storage, and Elasticsearch for full-text search. Artery developer community mainly implements login with Cocall account, edit articles or questions using markdown format, view articles or questions by category, full-text search articles and highlight keywords, comment articles, sticky articles, favorite articles, follow users, set essence Articles and other functions. The above functions meet the basic needs of Artery developers to exchange their experience in the process of using Artery. At the same time, they can speed up the improvement of the Artery ecosystem and attract more people to use Artery.

**Keywords:** Artery; Community; SpringBoot; Android; App; Retrofit; RxJava; OkHttp

# 目录

1	绪论	1
1.1	开发背景	1
1.2	开发目标	1
1.3	项目意义	2
1.4	可行性分析	2
1.4.1	操作的可行性	2
1.4.2	技术的可行性	2
1.4.3	经济的可行性	3
2	参考技术	4
2.1	Android 语言	4
2.2	网络框架	4
2.2.1	Retrofit	4
2.2.2	RxJava	4
2.2.3	OkHttp	5
2.3	MVP 代码架构	5
2.4	QMUI Android	6
2.5	Artery	6
2.6	ElasticSearch	7
3	需求分析	8
3.1	功能需求	8
3.2	非功能需求	9
3.3	用例图分析	9
3.4	需求用例	11
4	系统设计	17
4.1	系统模块图	17
4.2	数据库设计	17
4.2.1	数据库关系图模型	18
4.2.2	数据库主要关系表	18
4.3	模块划分	23
5	系统实现	24
5.1	用户模块	24
5.1.1	用户登录	24
5.1.2	关注用户	26
5.2	文章模块	27
5.2.1	发布文章	27
5.2.2	搜索文章/问题	29
5.2.3	查看文章/问题详情	30
5.2.4	回复文章/问题	31
5.2.5	点赞	33
5.2.6	收藏	34
5.2.7	置顶	35
5.2.8	设置精华	36

5.2.9	修改问题状态.....	37
5.3	消息模块.....	38
5.3.1	查看回复消息.....	38
5.3.2	查看关注消息.....	38
6	系统测试.....	39
6.1	主要功能测试.....	39
6.1.1	用户登录测试.....	39
6.1.2	发布文章测试.....	39
6.1.3	评论文章测试.....	40
6.1.4	搜索文章测试.....	40
7	总结与展望.....	40
	参考文献.....	42
	致谢.....	43

# 1 绪论

## 1.1 开发背景

Artery 是北京华宇信息技术有限公司自主研发的一个前后端分离的框架。Artery 框架包含前后端两部分，后端是对 SpringBoot 进行了进一步的封装，同时还集成的 Shiro 和 Mybatis 等流行技术；前端采用 Vue 框架进行实现，为用户提供了丰富的组件库。公司中的大部分项目都使用了 Artery 进行开发，因此在公司，Artery 拥有许多的用户。不过随着公司的不断扩大，使用 Artery 框架的用户也越来越多，许多员工在使用 Artery 的过程中可能会遇到一些问题，但是不知道找谁解答，同时会有许多员工在学习使用 Artery 框架的过程中有一些心得体会想要分享给其他人。因此大部分 Artery 的用户希望拥有一个平台可以交流学习 Artery，整合 Artery 的相关资源，分享 Artery 的学习体会，获取 Artery 相关问题的答案。

## 1.2 开发目标

本项目是一个基于 C/S 的一个手机 App，服务端主要使用 Artery 框架实现，提供方便简洁的操作界面使用户可以快速的发布帖子和问题，主要的实现目标有如下：

- 1) 本系统使用公司 Cocall 账户进行登录
- 2) 本系统设置三个用户角色，分别是内容管理员、一般用户、问题解决人。
- 3) 内容管理员：可以编辑问题和内容、分配问题；可以在 Cocall 中收到问题待分配的通知；可以管理主题和话题信息；可以设置文章精华、置顶。
- 4) 一般用户：可以创建问题或文章；可以回复、收藏和点赞问题或文章；可以关注其他用户；可以收到自己文章的回复提示；可以查看关注用户的动态信息。
- 5) 问题解决人：解决的问题的人，新问题可以分配此拥有此角色的用户去处理
- 6) 本系统能实现用户使用 Markdown 编辑器编辑文章和问题。
- 7) 界面要简约大方美观且实用。

## 1.3 项目意义

Artery 开发者社区 APP 是 Artery 开发者的一个交流平台，不仅限于 Artery 开发平台，通常说的 Artery 其实只是大 Artery 体系下的一个开发平台（Artery Develop Studio），简称 ADS，其它相关的技术帖子都可以在此发布。Artery 开发者社区 APP 可以帮助 Artery 开发者快速的了解 Artery 相关知识，解决 Artery 开发者在开发过程中的一些问题。

## 1.4 可行性分析

可行性研究就是通过分析问题的利与弊，从而判断问题是否能够用最少的代价和尽可能短的时间去解决。在对一个项目进行可行性研究时，我们要判断项目的规模和目标是否可以实现，以及完成项目后带来的收益是否值得我们去投资开发这个项目。因此，可行性研究的性质就是在较高抽象层次上对项目进行一次简化的系统分析和设计。系统的可行性分析主要包括经济上的可行性，技术上的可行性和操作上的可行性。

### 1.4.1 操作的可行性

Artery 开发者 APP 是一款 Android 应用，主要分为两个部分，一个是运行在 Android 手机上的客户端，一个是运行在服务器上的后台服务。客户端应用主要运行在 Android 6.0 以上的手机中，并且应用界面简单大方，操作流程比较简单，用户可以很快熟悉应用的操作；后台服务可以部署在 linux 操作系统上，只要有 java 的基本环境，就能很快进行部署。因此，它在操作上是可行的。

### 1.4.2 技术的可行性

Artery 开发者社区 APP 使用原生的 Android 语言进行编写，这是目前比较先进、可靠、稳定并且使用最为广泛的移动端开发语言。

客户端通过 Retrofit + RxJava + OkHttp 网络框架组合调用后台服务。客户端和服务端通过 http 协议进行通信，客户端从服务端中获取 Json 格式的数据，两者只需提前预定好数据的传输格式便可分开进行开发，不会过多的影响对方的开发进度。客户端使用 QMUI Android 和 MVP 代码结果，可以降低代码之间的耦合度，使开发人员对代码的编写和维护的难度大大降低。

服务端采用 Artery 框架进行开发，Artery 框架封装了一系列的基础服务，包括对数据库的操作等，简化的开发者的开发流程，使开发者更加专注于业务流

程的开发。在数据方面，系统使用了 MySQL 进行数据存储，同时使用 ElasticSearch 实现全文检索，提高了用户的搜索体验和搜索速度。

综上所述， Artery 开发者社区 APP 使用目前流行的开发方法和先进的技术使整个系统满足了用户的基本需求。因此，它在技术上可行。

### **1.4.3经济的可行性**

Artery 开发者社区 App 是本人自己开发，使用自己的电脑，后台服务部署在自己租用的阿里云服务器上。因此，它在经济上是可行的。



## 2 参考技术

### 2.1 Android 语言

Android 由 JAVA 语言发展而来，是一种基于 Linux 内核（不包含 GNU 组件）的自由及开放源代码的操作系统。主要使用于移动设备，如智能手机和平板电脑。

Android 的系统架构采用了分层的架构设计，Android 操作系统从高到低主要分为四层，分别是应用程序层、应用程序框架层、系统运行库（C/C++库以及 Android 运行库）层和 Linux 内核层。

### 2.2 网络框架

RxJava + Retrofit + OKhttp 网络框架是目前 Android 开发使用最多的网络框架，这套网络框架功能非常强大并且简单易用。

#### 2.2.1 Retrofit

Retrofit 是一个 RESTful 的 HTTP 网络请求框架的封装，网络请求实际上是使用 OkHttp 完成的，Retrofit 仅对网络请求接口进行的封装。在服务端返回数据之后，OkHttp 将接受这些数据并进行简单的解析并将解析的结果返回给 Retrofit，Retrofit 会根据用户的需求对结果进行解析。

Retrofit 主要定义了 4 个接口：

Callback<T>：请求数据的返回；

Converter<F, T>：对返回数据进行解析，一般用 GSON ；

Call<T>：发送请求，Retrofit 默认的实现是 OkHttpCall<T>，也可以依需自定义 Call<T>；

CallAdapter<T>：将 Call 对象转换成其他对象，如转换成支持 RxJava 的 Observable 对象。

#### 2.2.2 RxJava

RxJava 是一个在 Java 虚拟机上，使用可观测的序列来组成异步的、基于事件的程序的库，简单的说就是让异步变得更加简洁方便。

RxJava 主要包含两个主要的类 Observable (观察者)与 Subscriber (订阅者)。Observable 可以发出数据流和事件，而 Subscriber 对这些数据流或事件进行吹，一个 Observable 可以包含多个 Subscribers。这个和观察者模式很相似，但是它

们之间的不同之处在于，如果没有订阅者，观察者是不会发出任何数据流或事件的。

所以说，RxJava 本质上是一个异步操作库，是一个能让开发者用极其简洁的逻辑去处理繁琐复杂任务的异步事件库。。

## 2.2. 3OkHttp

OkHttp 是 Square 公司开发的一套网络请求库，已经得到 Google 官方的认可。OkHttp 的基础功能类似于 Android 自带的 Apache HTTP Client 和 HttpURLConnection，但是它却要比 Android 自带的两个 Http 客户端好的多，一者是 OkHttp 的写法更加简单，开发人员可以很快的熟悉它的用法；二者是 OkHttp 可以处理大部分网络复杂问题，比如我们经常在系统中为服务器配置多个 IP 地址，当第一个 IP 连接失败时，OkHttp 就会自动的去尝试连接下一个 IP，这样可以减少开发人员许多的工作量，且使系统更加稳定。OkHttp 还处理了 SSL 握手失败和代理服务器问题等等许多问题。

## 2.3 MVP 代码架构

MVP 是 MVC 的一种衍生模式。MVP(Model-View-Presenter)模式是 2000 年左右由 IBM 开发出来的一个针对 C++和 Java 的编程模型，是 MVC (Model-View-Controller) 模式演变而来的，在应用开发中它们具有基本相通的设计思想：Controller/Presenter 负责逻辑的处理，Model 提供数据，View 负责显示<sup>[8]</sup>。MVC 的设计模型图如图 2.1 所示，MVP 的设计模型图如图 2.2 所示。

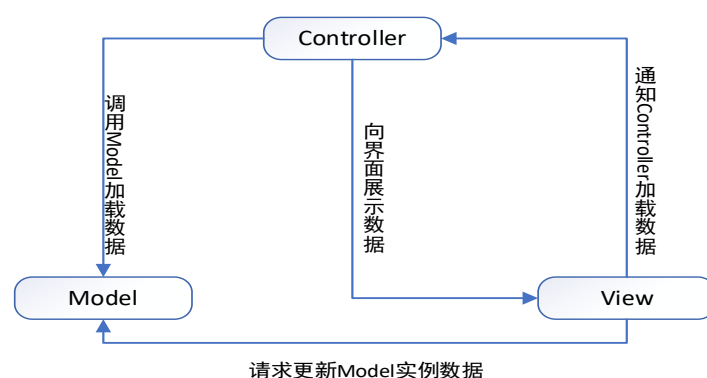


图 2.1 MVC 设计模式

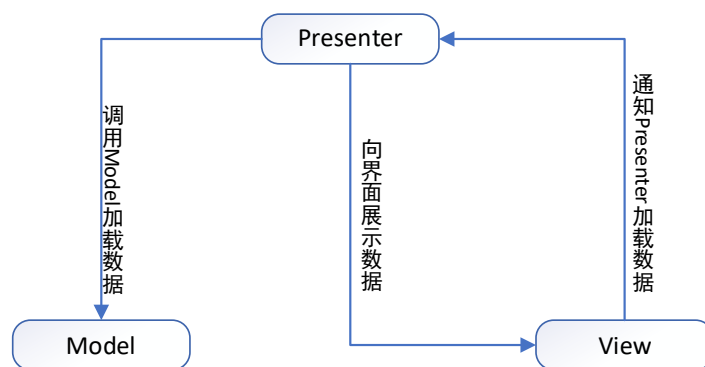


图 2.2 MVP 设计模式

在传统的 Android 开发中，大部分使用的是 MVC 模式，由于代表 View 的 Layout 资源在没有 Activity 帮助时并不能完全解决界面如何渲染的问题，导致 Activity 中无法避免的包含了大量的 View 的内容，然而 Activity 又需要监听 View，以此来接收用户的输入，所以 Activity 又包含了 Control 的功能。因此 MVC 模式无法将 View 和 Control 完全分离开来。为了降低 Android 项目的耦合性，在进行 Android 开发时 MVP 模式是一个更好选择。

## 2.4 QMUI Android

QMUIAndroid 是一个 Android 平台的 UI 开发库，它提供了丰富控件及兼容处理，让开发者能专注于业务需求而无需耗费精力在基础代码建设上。

QMUIAndroid 项目源自于项目团队对多个项目的迭代思考，既能方便地统一项目整体的 UI 样式，也能方便地应对频繁的变动，既提供了如圆角按钮、对话框、下拉刷新等丰富的 UI 控件给开发者单独地应用到各种常见场景中，也提供了对沉浸式状态栏、手势返回等全局配置的实现方案。不管是新项目的创建，或是已有项目的维护，使用 QMUI Android 均可使开发效率和项目质量得到大幅度提升。

## 2.5 Artery

Artery 是北京华宇信息技术有限公司开发的基于 SpringBoot 的，前后端分离的框架，它的中文名称是通用电子政务快速构建平台。

Artery 主要分为前端 ui 组件和后端两个部分，框架基于 SpringBoot2.2.x，Thymeleaf，控件采用 Vuejs 开发。Artery 主要实现了以下几大革新：

- 1) 表单 html 标准化。
- 2) 前后台交互方式使用 Spring MVC 官方注解，减少黑盒。
- 3) 插件人性化，插件尽可能的辅助开发人员生成代码。

- 4) 组件模块化, Artery 功能保持单一。
- 5) 采用约定大于配置的设计理念, 简化配置。

Artery 对 SpringBoot 进行了进一步的封装, 实现了基本的用户管理、权限管理等, 使用户可以更加专注于业务方面的开发, 极大的加快了软件开发的速度。

## 2.6 Elasticsearch

分布式搜索引擎 Elasticsearch(以下简称 ES) 是一个开源的、基于 Lucene 的分布式搜索引擎, ES 可以提供稳定、实时、可靠的检索服务, 具有高可用、易扩展以及近实时的特点<sup>[9]</sup>。ES 使用 Java 语言和 Lucene 作为其核心来实现所有索引和检索功能, 但它通过简单的 RESTful API 隐藏了 Lucene 框架的复杂性。除此之外, ES 还使用分布式实时文件存储, 可执行实时分析搜索, 还可以扩展到上百台服务器, 处理 PB 级结构化或非结构化数据。

在本系统中主要通过 ES 存储用户发布的文章信息、问题信息和评论, 方便在用户搜索时可以进行全文检索, 更加精确的找到用户所需要的内容。

## 3 需求分析

### 3.1 功能需求

#### 1、用户登录验证

本系统使用公司内部使用的 Cocall 账户进行登录，系统将用户提供的用户名和密码发送到服务端，在服务端与 Cocall 的后台服务器交互进行验证，如果用户名和密码正确并且是第一次登录本系统，则将用户的信息存在系统自己的数据库中，后面步骤中所有的用户信息都在系统所使用的数据库中获取，但是用户登录还是通过与 Cocall 后台服务器交互进行验证。系统还需要实现记住密码功能，当用户勾选记住密码后，系统能够自动保存当前的用户名和密码，并在用户下次登录时自动填入账号和密码。

#### 2、文章的编辑、查看

本系统中的文章使用的是 Markdown 的格式进行编辑，因此本系统需要实现在手机上编辑和解析 Markdown 文件。在查看文章时，能够通过主题、标签分类查看文章列表信息。

#### 3、文章的回复

本系统需要支持回复文章。如果文章的类型是问题，回复人可以修改问题的状态，标记问题是否被解决。

#### 4、文章的置顶、设置精华、点赞、收藏

用户如果具有相关的权限，用户可以将文章置顶，设置文章为精华文章。同时，用户在查看文章时可以对文章点赞和收藏文章。

#### 5、文章的搜索

本系统需要支持对文章的全文检索，通过输入的关键字根据文章的标题、内容、发布人等信息中查询是否含有该关键字，并根据关键字在文章中出现的次数对检索出来的文章进行排序。

#### 6、关注发帖人

本系统需要支持用户关注其他用户，并且如果关注的用户发布文章时，用户可以获取到提示。

### 7、动态信息通知

本系统需要支持消息通知。当用户发布的文章被其他人回复、点赞时，当用户被其他用户关注时系统需要提示用户。

### 8、问题自动分配负责人

如果用户发布问题时，系统要对问题自动分配负责人（负责人是具有一定权限的人，主要负责回答用户提出的问题）。

## 3.2 非功能需求

为了使用户在使用本系统的过程中具有良好的用户体验，本系统需要实现以下需求：

- 1、界面友好：APP 的界面应该简约大方并且具有合理的布局，对于用户的一些错误操作应该有相应的提示信息，使用户在操作的过程中具有良好的视觉效果。同时，应当尽量简化用户的操作流程，避免用户进行大量的输入，尽可能的使用户选择相关的内容。
- 2、响应迅速：系统的响应速度是影响用户体验的重要原因之一，因此对于用户的操作应该尽快的给出相应，对于一些比较耗时的操作应该给出相应的提示信息。
- 3、系统安全：系统安全是每个系统都应该保障的，对于用户的登录信息应当进行加密传输，对于一些操作，比如删除文章、修改文章、评论文章等应当设置一些权限。

## 3.3 用例图分析

本系统中有三种用户类型：内容管理员、一般用户、问题解决者，以下是它们的用例图。

内容管理员拥有系统所有权限，可以管理文章，管理问题，管理主题和标签。内容管理员用例如图 3.1 所示：

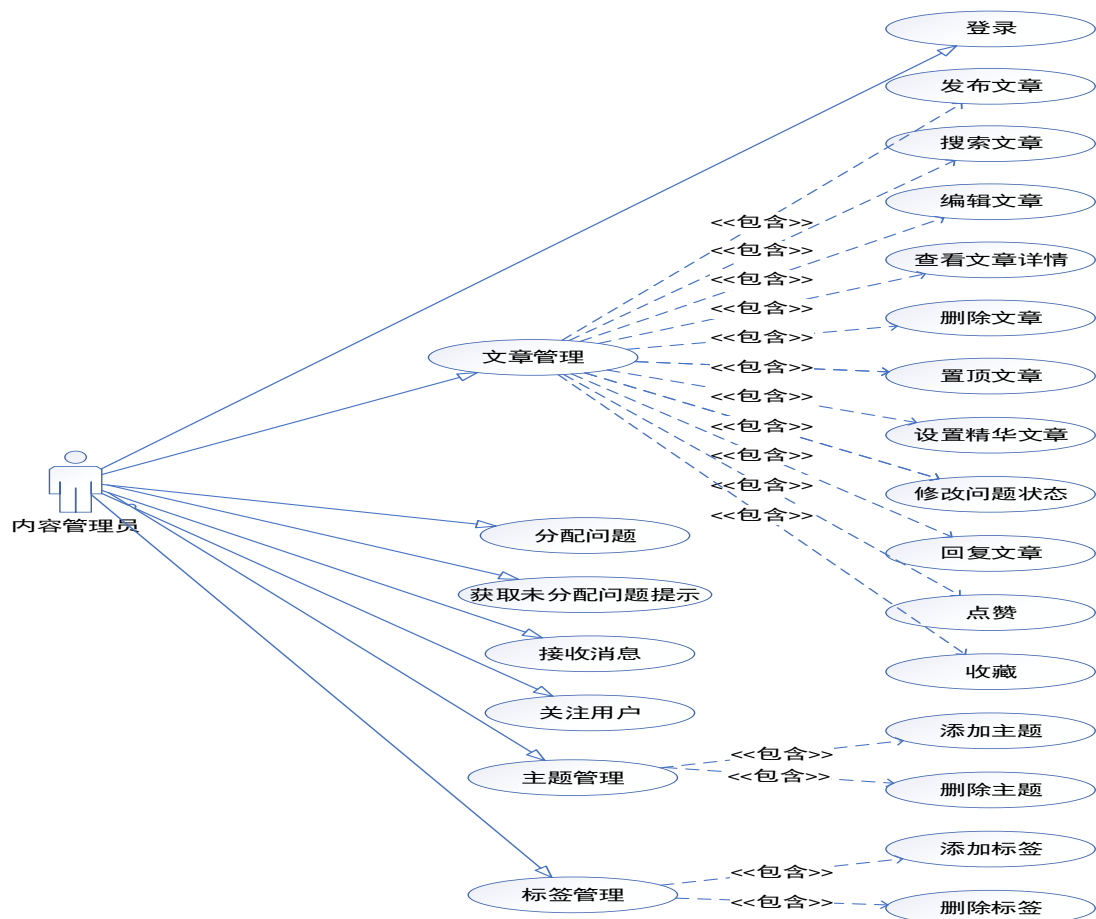


图 3.1 内容管理员用例图

一般用户用例如图 3.2 所示：

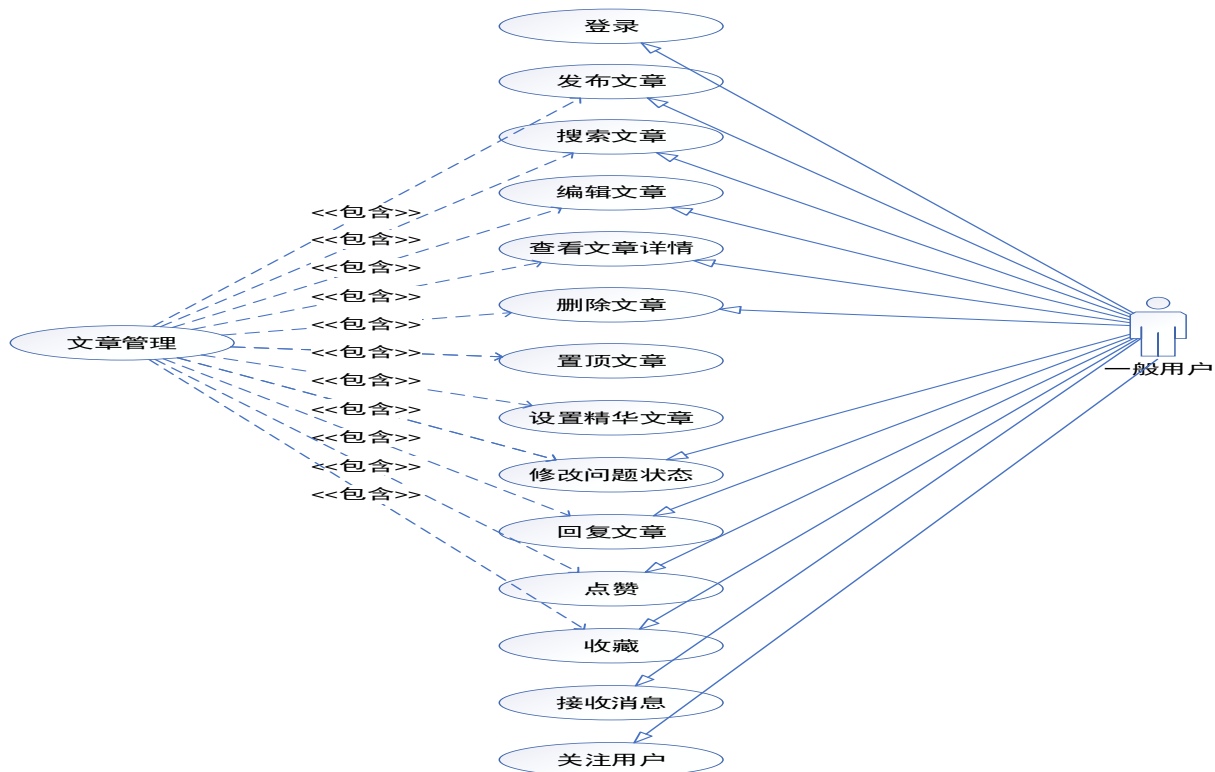


图 3.2 一般用户用例图

问题解决人用例如图 3.3 所示：

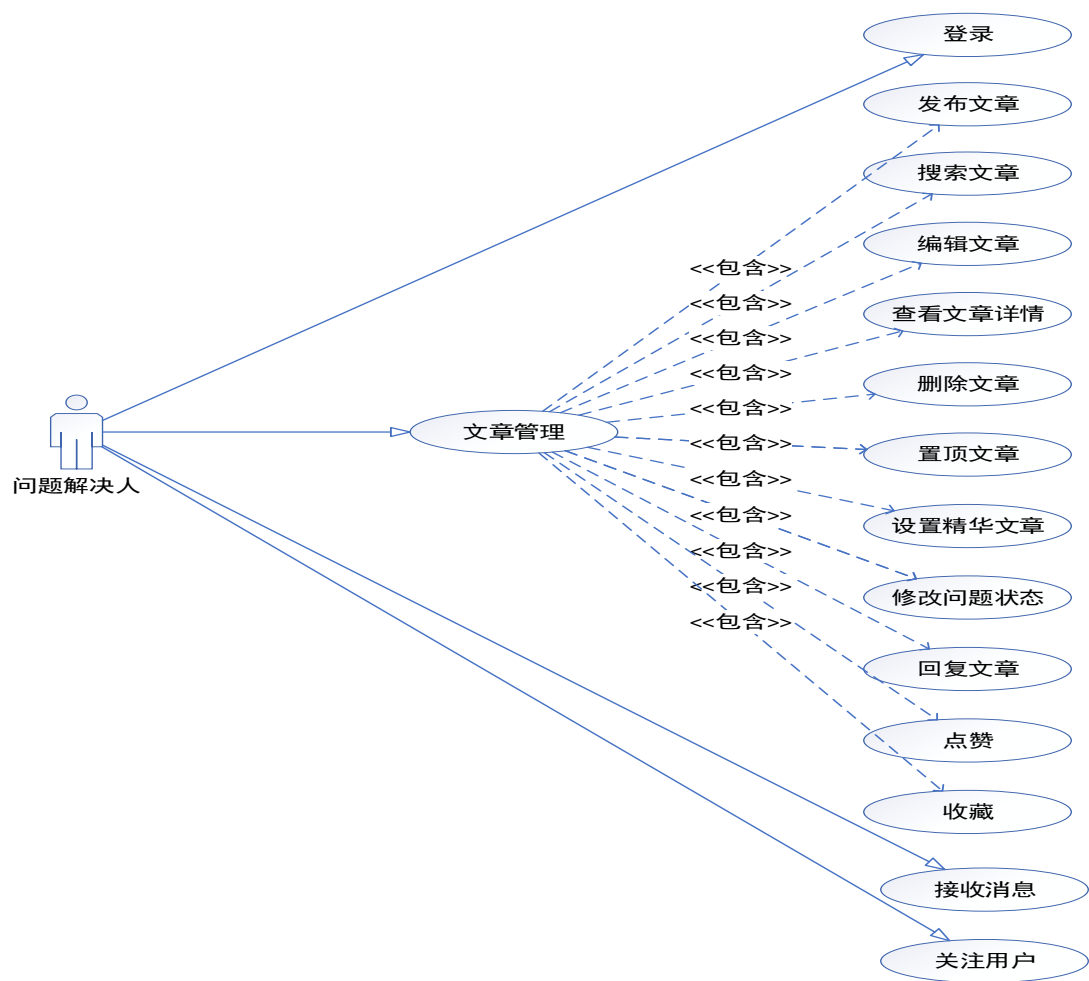


图 3.2 问题解决人用例图

### 3.4 需求用例

用户登录用例如表 3.1 所示：

表 3.1 登录用例表

名称	登录
参与者	用户
描述	用户使用 CoCa11 账户进行登录
触发条件	用户点击登录
前置条件	用户拥有 CoCa11 账户
用例描述	1、用户进入登陆页面 2、用户输入 CoCa11 账号和密码，选择登录服务器，点击登陆 2.1、用户账号或密码错误，提示用户账号或密码错误



	3、用户登录成功，自动跳转到首页
后置条件	提示用户登录成功，跳转到首页

用户发布文章用例如表 3.2 所示：

表 3.2 发布文章用例

名称	发布文章
参与者	用户
描述	用户发布文章
触发条件	用户点击发布文章选项
前置条件	用户已使用 CoCall 账号登录
用例描述	<ol style="list-style-type: none"> <li>1、用户在首页点击发布文章选项 <ol style="list-style-type: none"> <li>1.1 用户如果未登录，则提示用户</li> </ol> </li> <li>2、用户输入文章标题和文章内容</li> <li>3、用户点击发布，跳转到下一页</li> <li>4、用户选择主题和标签，点击发布 <ol style="list-style-type: none"> <li>4.1 用户未选择主题或标签，提示用户需要选择主题或标签</li> <li>4.2 用户发布文章失败，提示用户发布文章失败</li> </ol> </li> <li>5、发布文章成功，提示用户并跳转到文章详情页面</li> </ol>
后置条件	提示用户发布文章成功，跳转到文章详情页面

用户发起问题用例如表 3.3 所示：

表 3.3 发起问题用例

名称	发起问题
参与者	用户
描述	用户发起问题
触发条件	用户点击发起问题选项
前置条件	用户已使用 CoCall 账号登录
用例描述	<ol style="list-style-type: none"> <li>1、用户在首页点击发起问题选项 <ol style="list-style-type: none"> <li>1.2 用户如果未登录，则提示用户</li> </ol> </li> <li>2、用户输入问题标题和问题内容</li> <li>3、用户点击发布，跳转到下一页</li> <li>4、用户选择主题和标签，点击发布</li> </ol>

	4.1 用户未选择主题或标签，提示用户需要选择主题或标签 4.2 用户发起问题失败，提示用户发起问题失败 5、发起问题成功，提示用户并跳转到问题详情页面
后置条件	提示用户发起问题成功，跳转到问题详情页面

查看文章/问题详情用例如表 3.4 所示：

表 3.4 查看文章/问题详情用例

名称	查看文章/问题详情
参与者	用户
描述	用户查看文章或者问题详情
触发条件	用户点击文章或问题列表中的某篇文章或问题
前置条件	无
用例描述	1、用户点击文章或问题列表中的某项内容 2、跳转到文章或问题详情页面，加载文章或问题详细信息 2.1 文章或问题内容不存在，提示文章或问题内容不存在 3、显示文章或问题详细信息
后置条件	无

搜索文章/问题用例如表 3.5 所示：

表 3.5 搜索文章/问题用例

名称	搜索文章/问题
参与者	用户
描述	输入关键字搜索文章/问题
触发条件	点击搜索按钮
前置条件	无
用例描述	1、用户点击首页的搜索框，跳转到搜索页面 2、用户输入要搜索的关键字，点击搜索按钮 3、显示根据用户输入的关键字的查询结果。
后置条件	显示查询结果

回复文章用例如表 3.6 所示：

表 3.6 回复文章用例

名称	回复文章
参与者	用户
描述	用户回复某篇文章
触发条件	用户点击输入框
前置条件	用户已登录，且用户进入文章详情页面
用例描述	1、 用户进入文章详情页面 2、 用户点击输入框 2.1 用户未登录，提示用户未登录 3、 用户输入要回复的内容，点击回复 3.1 文章不存在，提示回复失败 3.2 输入框中没有输入内容，提示请输入要回复的内容 3.3 用户没有回复权限，提示没有权限 3.4 用户未登录或登录失效，提示用户未登录 4、 回复成功
后置条件	无

点赞文章用例如表 3.7 所示：

表 3.7 点赞文章用例

名称	点赞文章
参与者	用户
描述	用户点击点赞按钮，对文章进行点赞
触发条件	点击点赞按钮
前置条件	用户已登录
用例描述	1、 用户进入文章详情页面 2、 用户点击点赞按钮 2.1 用户未登录，提示用户登录 2.2 点赞失败，提示用户点赞失败 3、 点赞成功，提示用户点赞成功
后置条件	无

收藏文章用例如表 3.8 所示：

表 3.8 收藏文章用例

名称	点赞文章
参与者	用户
描述	用户点击收藏按钮，收藏文章
触发条件	点击收藏按钮
前置条件	用户已登录
用例描述	1、用户进入文章详情页面 2、用户点击收藏按钮 2.1 用户未登录，提示用户登录 2.2 收藏失败，提示用户收藏失败 3、收藏成功，提示用户收藏成功
后置条件	无

置顶文章用例如表 3.9 所示：

表 3.9 置顶文章用例

名称	置顶文章
参与者	内容管理员、问题解决人
描述	内容管理员和问题解决人查看文章详情时，可以点击置顶选项置顶文章
触发条件	点击置顶选项
前置条件	用户已登录且具有置顶文章权限
用例描述	1、进入文章详情页面 2、点击置顶选项 2.1 用户未登录，提示用户登录 2.2 文章不存在，提示文章不存在 2.3 用户没有置顶文章的权限，提示用户没有权限 2.4 置顶失败，提示用户置顶失败 3、置顶成功，提示用户置顶成功
后置条件	无

设置精华用例如表 3.10 所示：

表 3.10 设置精华用例

名称	设置精华
参与者	内容管理员、问题解决人

描述	内容管理员和问题解决人查看文章详情时，可以点击精华选项设置文章为精华文章
触发条件	点击精华选项
前置条件	用户已登录且用户具有设置精华权限
用例描述	1、进入文章详情页面 2、点击精华选项 2.1 用户未登录，提示用户登录 2.2 文章不存在，提示文章不存在 2.3 用户没有设置精华文章的权限，提示用户没有权限 2.4 设置精华文章失败，提示用户设置精华文章失败 3、设置精华文章成功，提示用户设置精华文章成功
后置条件	无

关注发帖人用例如表 3.11 所示：

表 3.11 关注发帖人用例

名称	关注发帖人
参与者	用户
描述	用户可以在文章详情页面关注文章的发起人
触发条件	点击关注按钮
前置条件	用户已登录
用例描述	1、进入文章详情页面 2、点击关注按钮 2.1 用户未登录，提示用户登录 2.2 关注失败，提示用户关注失败 3、关注用户成功
后置条件	无

## 4 系统设计

### 4.1 系统模块图

系统模块图如图 4.1 所示

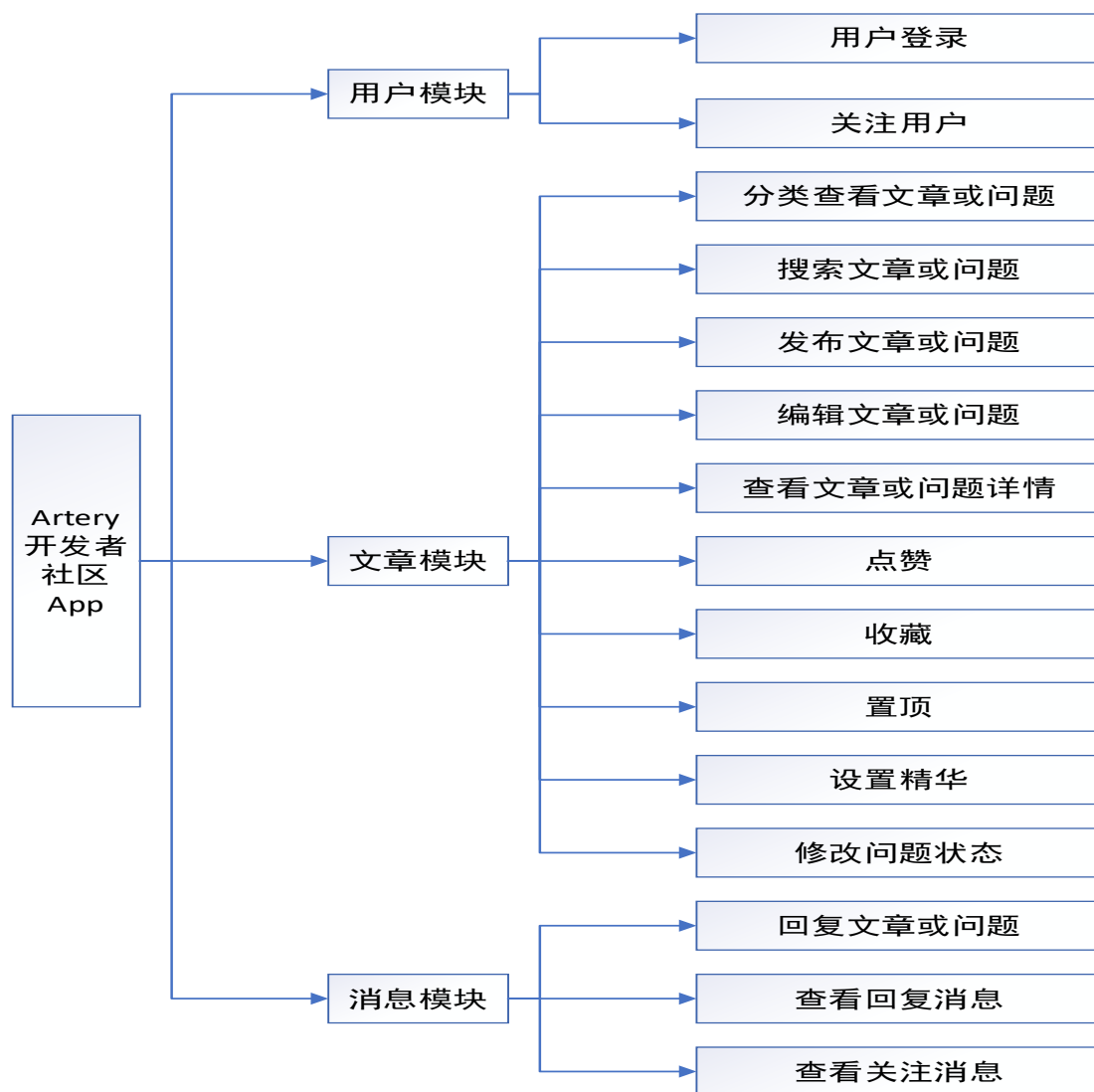


图 4.1 系统模块图

### 4.2 数据库设计

本系统主要使用 MySQL 进行存储数据，使用 ES 实现全文检索。用户、角色、权限、标签、主题、单值代码、消息及它们的关系表存储在 MySQL 中，方便通过关系查询他们的详细信息；而文章和回复信息存储在 ES 中，方便通过文章或回复中的内容的关键字查询文章或回复的信息。

### 4.2.1 数据库关系图模型

数据库设计是否合理对一个项目有着至关重要的作用,为了系统能够高效正确的运行,通过对系统进行需求分析和功能设计,得到了本系统中主要的实体,分别是:用户、角色、权限、文章、标签、主题、代码、消息、回复。下面是分析过后得到的 E-R 图,但由于系统实体比较多只给出了简化版的 E-R 图如图 4.2 所示:

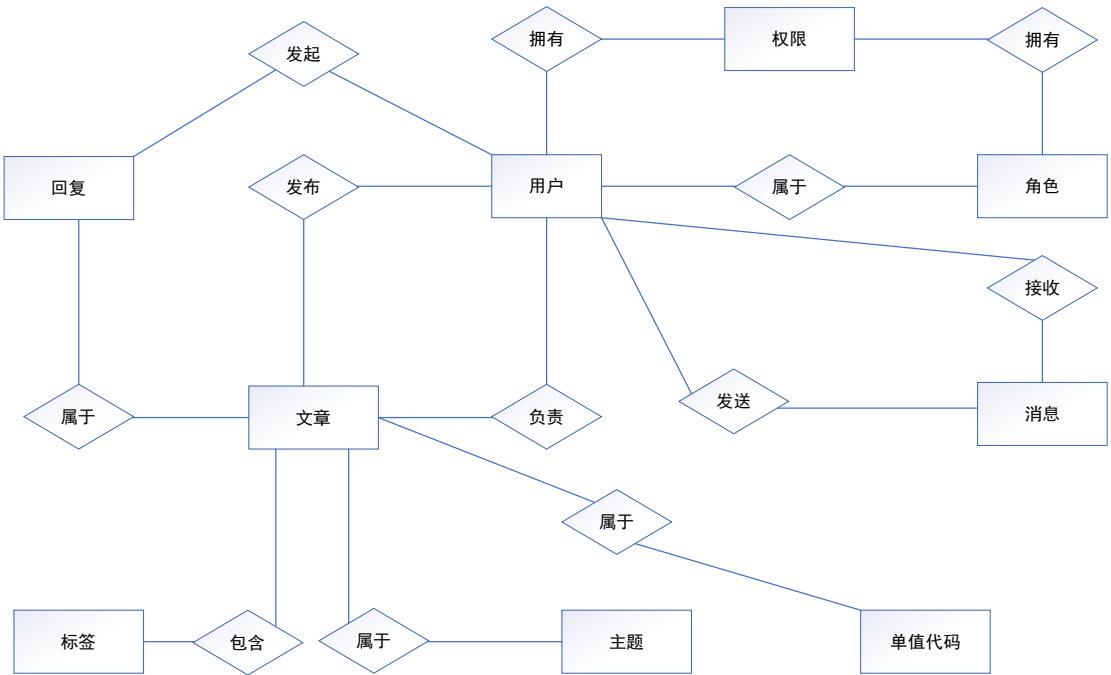


图 4.2 Artery 开发者社区 APP E-R 图

### 4.2.2 数据库主要关系表

根据图 4.2 Artery 开发者社区 APP E-R 图,可以得到 15 张数据表,并将数据按照这些表的格式存入数据库中。以下是数据库表的详细信息。

用户表如表 4.1 所示:

表 4.1 用户表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	50	否	帐号 ID
C_LoginID	varchar	300	否	登录标识
C_Name	varchar	300	否	姓名
C_Password	varchar	32	否	密码
C_Mail	varchar	300	是	电子邮件
C_IP	varchar	300	是	IP
C_XMJP	varchar	300	是	姓名简拼
C_CORP	varchar	300	是	所在单位

C_DEPT	varchar	300	是	所在部门
N_Valid	tinyint	0	否	是否有效
N_Order	smallint	0	否	显示顺序
C_CCUSER_ID	varchar	50	是	CoCall 用户 ID
C_CCUSER_DOMAIN	varchar	50	是	CoCall 用户 DOMAIN

角色表如表 4.2 所示：

表 4.2 角色表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	50	否	角色 ID
C_Name	varchar	150	否	角色名称
C_Descript	varchar	255	是	角色描述
N_XTGY	int	0	是	系统固有
N_Valid	int	0	是	是否有效
N_Order	int	0	是	显示顺序

权限表如表 4.3 所示：

表 4.3 权限表

字段名称	数据类型	长度	是否为空	字段描述
C_RightKey	varchar	32	否	权限字
C_Name	varchar	150	否	权限名称
C_Descript	varchar	255	是	权限描述
N_Order	int	0	是	显示顺序

角色权限表如表 4.4 所示：

表 4.4 角色权限表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	Id
C_RoleID	varchar	50	否	角色 Id
C_RightKey	varchar	32	否	权限字

用户权限表如表 4.5 所示：

表 4.5 用户权限表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	char	32	否	ID
C_UserID	varchar	150	否	帐号 ID
N_Type	int	0	否	类型
C_RoleID	varchar	255	是	角色 ID
C_RightKey	varchar	150	是	权限字

标签表如表 4.6 所示：

表 4.6 标签表

字段名称	数据类型	长度	是否为空	字段描述
------	------	----	------	------



C_ID	varchar	32	否	主键
C_MC	varchar	300	是	名称
C_MS	varchar	900	是	描述
C_PID	varchar	32	是	父 ID
N_ORDER	int	0	是	顺序
C_RIGHT	varchar	300	是	权限表达式

主题表如表 4.7 所示：

表 4.7 主题表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_NAME	varchar	100	是	名称
C_ICON	varchar	300	是	图标
C_DESC	varchar	900	是	描述
N_PUBLIC	int	0	是	公开状态
N_ORDER	Int	0	是	顺序

单值代码表图表 4.8 所示：

表 4.8 单值代码表

字段名称	数据类型	长度	是否为空	字段描述
C_PID	varchar	50	否	单值代码类型编号
C_Code	varchar	50	否	单值代码
C_Name	varchar	150	否	单值代码名称
N_KWH	int	0	否	可否维护
C_LevelInfo	varchar	64	是	分级信息
N_Valid	int	0	否	是否有效
N_Order	int	0	否	显示顺序
C_Dmjp	varchar	150	是	代码简拼

文章表如表 4.9 所示：

表 4.9 文章表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_BT	varchar	300	是	标题
C_NR	text	0	是	内容，编辑框内容即 markdown
C_NR_HTML	text	0	是	内容 Html

C_NR_TEXT	text	0	是	内容 Text
N_NRLX	int	0	是	内容类型, 1 问题 2 文章
C_FJM_TOPIC	varchar	600	是	关联标签的分机码, 逗号分隔多个, 首尾加逗号
C_ID_TOPIC	varchar	600	是	关联的标签 ID, 逗号分隔多个, 首尾加逗号
C_ID_SUBJECT	varchar	32	是	关联的主题 ID
C_FQR	varchar	50	是	发起人
D_FQSJ	datetime	0	是	发起时间
C_XGR	varchar	50	是	修改人
D_XGSJ	datetime	0	是	修改时间
N_CLZT	int	0	是	处理状态
C_FZR	varchar	50	是	负责人
C_ZXHFR	varchar	50	是	最新回复人
D_ZXHFSJ	datetime	0	是	最新回复时间
N_SFQX	int	0	是	是否缺陷
N_LLCS	int	0	是	浏览次数

回复表如表 4.10 所示:

表 4.10 回复表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_CONTENT	varchar	32	是	关联文章
C_NR	text	0	是	回复内容, Markdown
C_NR_HTML	text	0	是	内容 html
C_NR_TEXT	text	0	是	内容 text
N_HFR	varchar	32	是	回复人
D_HFSJ	datetime	0	是	回复时间
C_CLZT	int	0	是	处理状态

消息表如表 4.11 所示:

表 4.11 消息表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_SENDER	varchar	32	是	发消息的人
C_ID_RECEIV	varchar	32	是	接收人

ER				
C_ID_CONTE NT	varchar	32	是	关联的文章
C_MESSAGE	varchar	600	是	消息内容
D_FSSJ	datetime	0	是	发送时间
N_STATUS	int	0	是	消息状态 (1 未接受 2 已接收)
N_TYPE	int	0	是	消息类别 (1 关注 2 赞 3 回复)

文章标签表如表 4.12 所示：

表 4.12 文章标签表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_CONTE NT	varchar	32	是	关联的文章
C_ID_TOPIC	varchar	32	是	关联的标签

收藏表如表 4.13 所示：

表 4.13 收藏表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_USER	varchar	32	是	关联的用户
C_ID_CONTE NT	varchar	32	是	关联的文章
D_SCSJ	datetime	0	是	收藏时间

关注表如表 4.14 所示：

表 4.14 关注表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_USER	varchar	32	是	粉丝用户
C_ID_LIKE_U SER	varchar	32	是	被关注的用户
D_GZSJ	datetime	0	是	关注时间

点赞表如表 4.15 所示：

表 4.15 点赞表

字段名称	数据类型	长度	是否为空	字段描述
C_ID	varchar	32	否	主键
C_ID_USER	varchar	32	是	点赞用户

C_ID_CONTE NT	varchar	32	是	点赞的文章
C_ID_UP_USE R	varchar	32	是	文章发布人
D_DZSJ	datetimt	0	是	点赞时间

### 4.3 模块划分

根据前面需求分析所得到的结果，可以叫本系统的模块划分为以下几个部分：

- 1、用户模块：包括用户的登录、登出以及关注用户
- 2、文章模块：包括文章的发布、查看、搜索、点赞、置顶、设置精华，收藏、回复和删除。
- 3、消息模块：包括获取回复文章时的消息提示、关注用户的消息提示和用户发布新文章时提示具有相关权限的用户。

## 5 系统实现

### 5.1 用户模块

#### 5.1.1 用户登录

系统的用户信息使用 CoCa11 中的用户信息，不需要在本系统中添加。用户每次登录本系统时，服务端都要从 CoCa11 的服务器中验证用户信息是否合法，如果用户是第一次登录本系统，则将用户的基本信息存储到本系统使用的数据库中；如果在登录的过程中发现用户的系统被修改，则更新本系统中用户的信息。本功能流程图如图 5.1 所示：

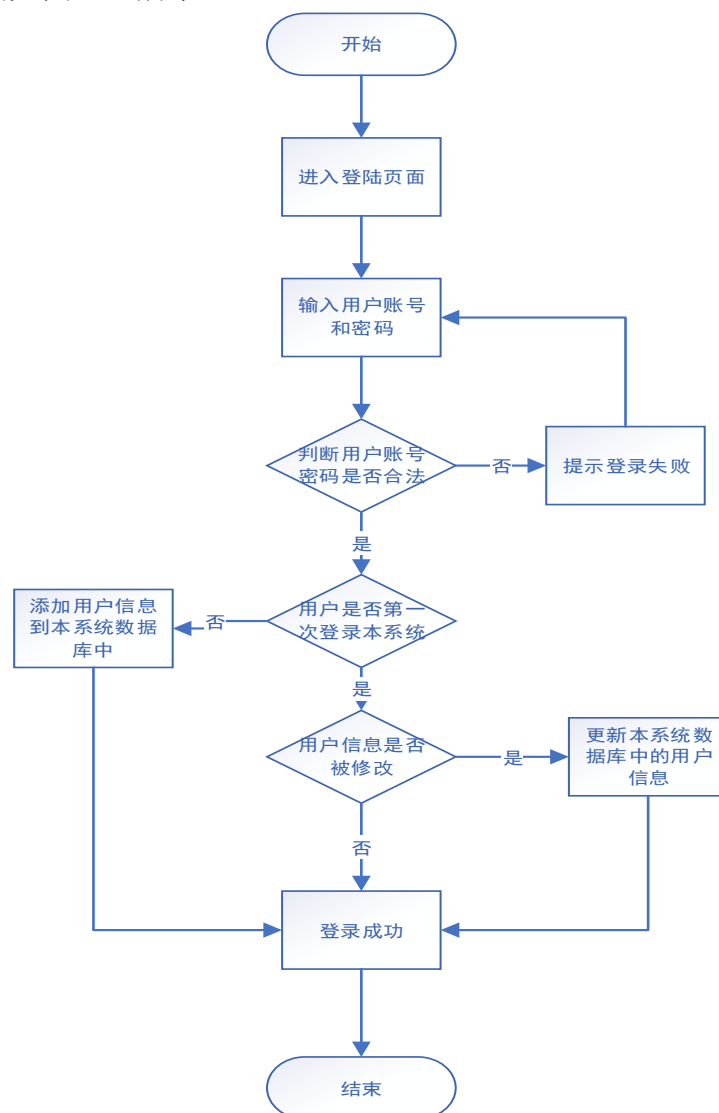


图 5.1 登录流程图

系统的登录界面如图 5.2 所示：



图 5.2 登录界面

## 5.1.2 关注用户

用户可以关注其他用户，当关注的用户发布文章或问题时，用户收到相应的通知消息。关注功能的流程图如图 5.3 所示：

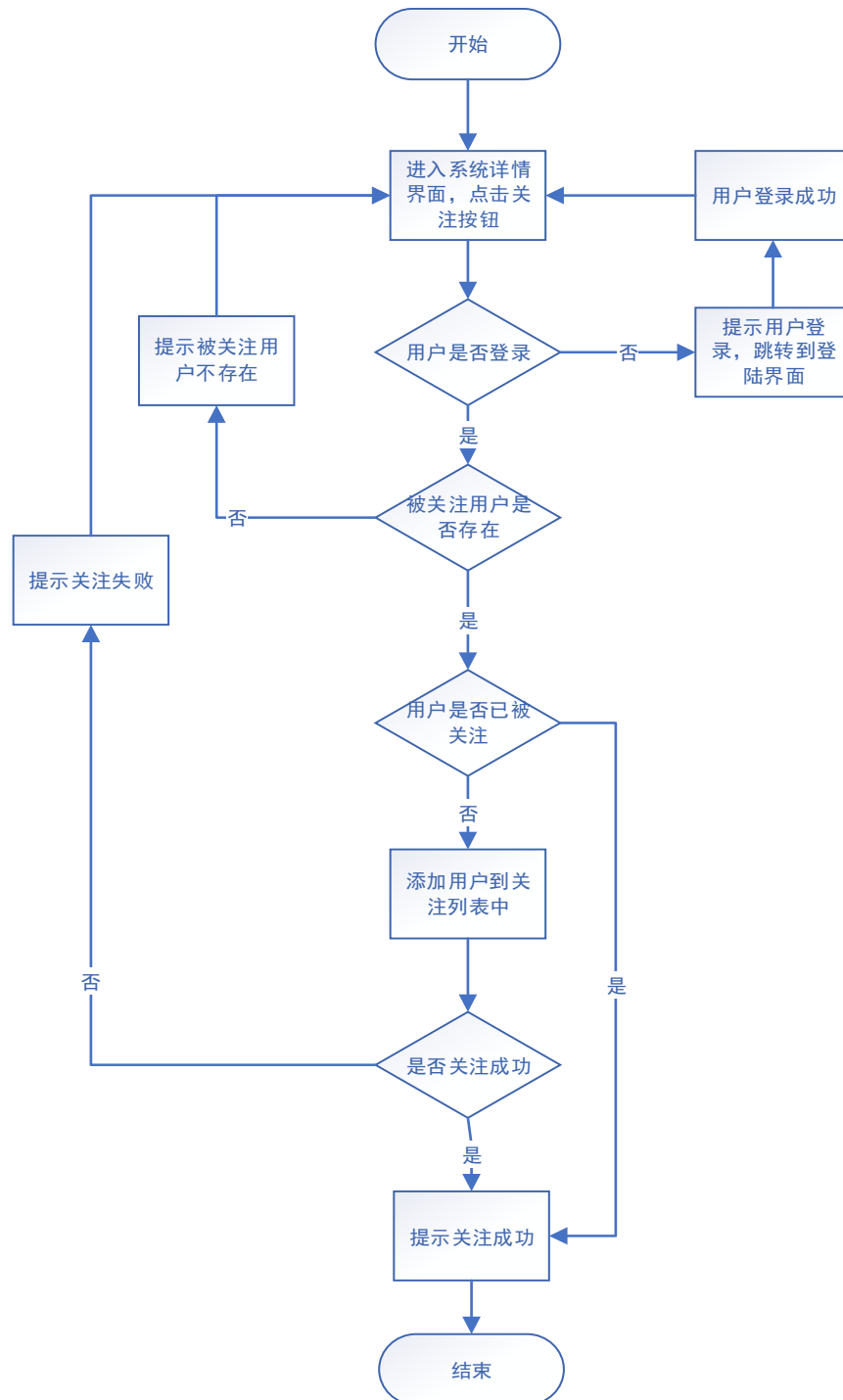


图 5.3 关注流程图

## 5.2 文章模块

### 5.2.1 发布文章

用户可以在文章编辑界面输入文章的基本信息，然后发布文章。同时，用户发布文章后会在 Cocall 中通知内容管理员和问题负责人。发布文章的流程图如图 5.4 所示：

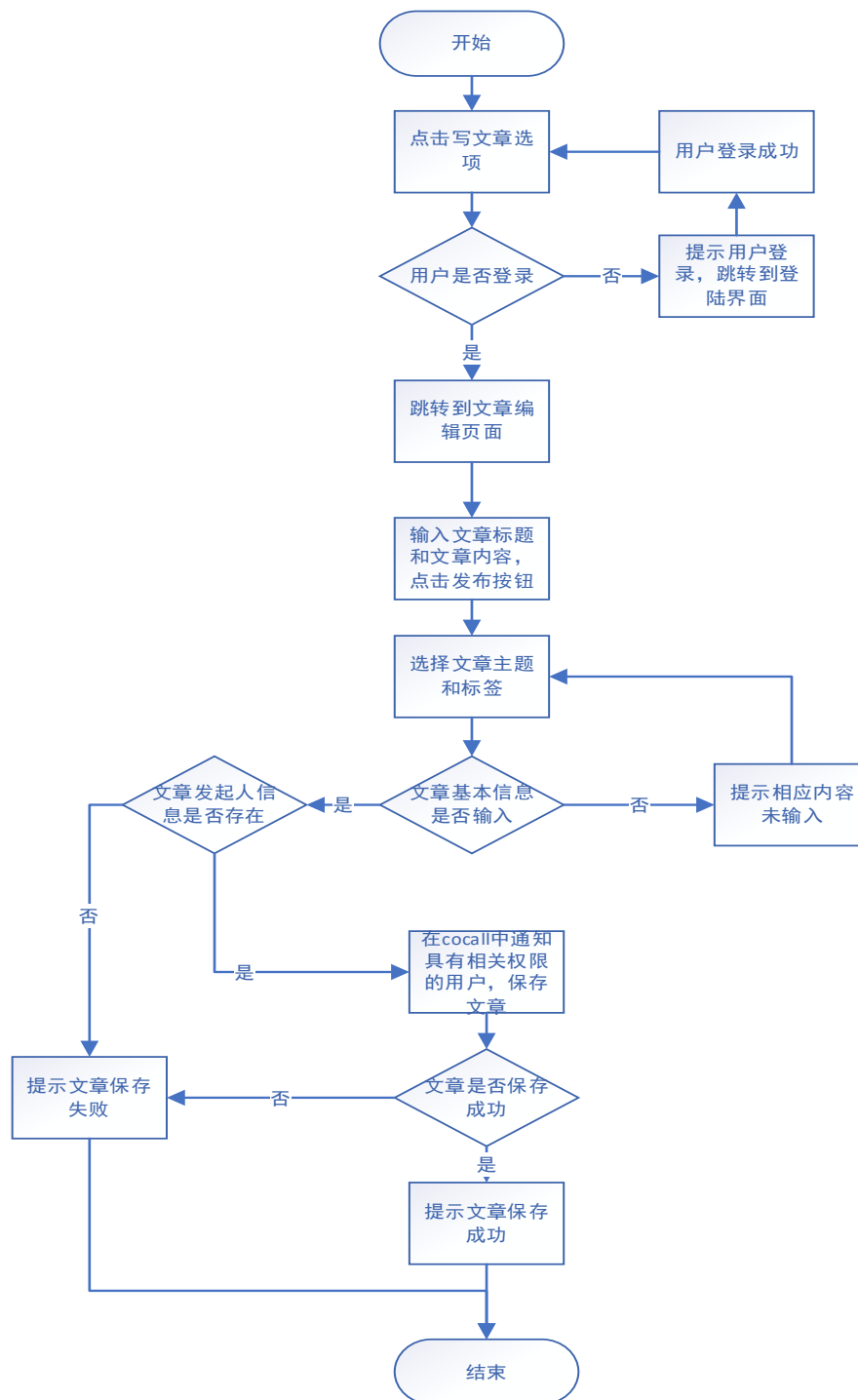


图 5.4 发布文章流程图



文章发布页面分别如图 5.5、图 5.6、图 5.7、图 5.8 所示：

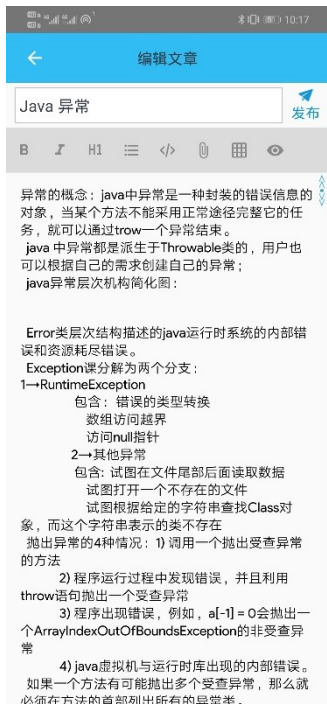


图 5.5 文章编辑页面

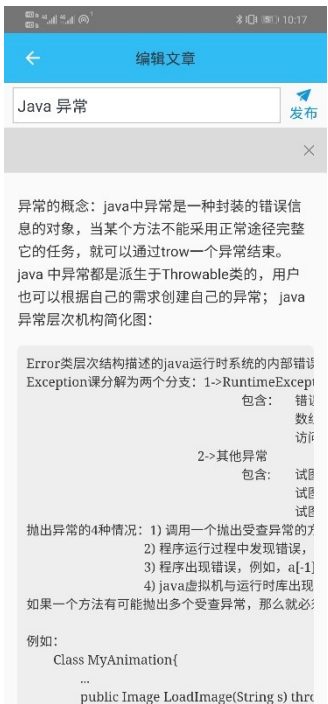


图 5.6 文章预览页面

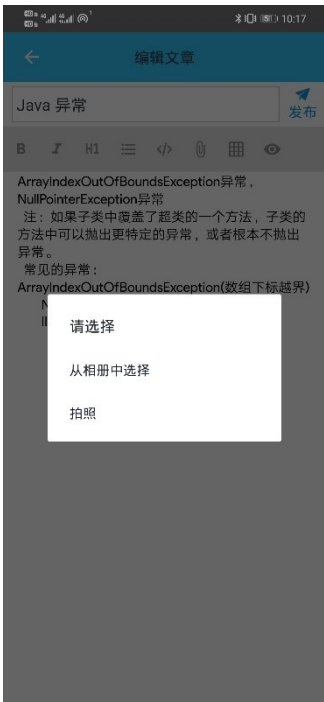


图 5.7 插入图片

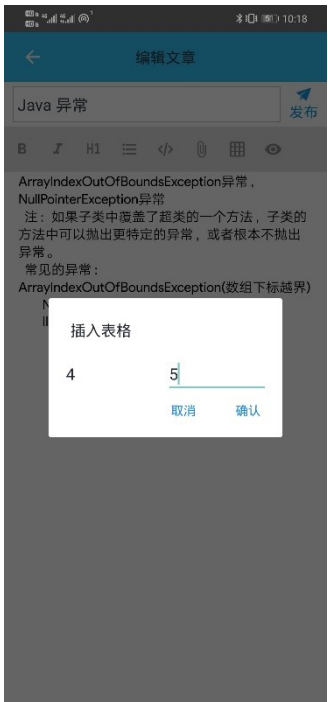


图 5.8 插入表格

## 5.2.2 搜索文章/问题

在首页顶部的输入框中输入关键字，可以全文检索文章或问题的内容，以列表的形式显示搜索的结果。搜索文章/问题的流程图如图 5.9 所示

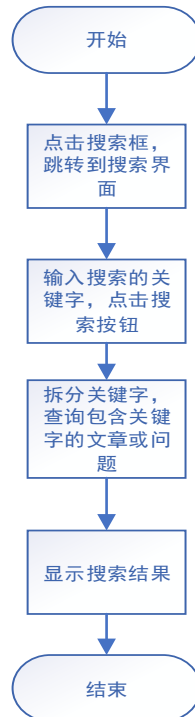


图 5.9 搜索文章/问题流程图

文章/问题搜索界面如图 5.10 所示：

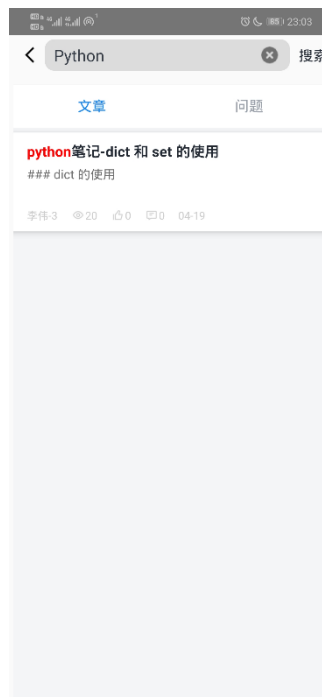


图 5.10 搜索文章/问题

### 5.2.3 查看文章/问题详情

当点击文章/问题列表中的某一项内容时，会跳转到文章详情页面。查看文章/问题详情流程图如图 5.11 所示：

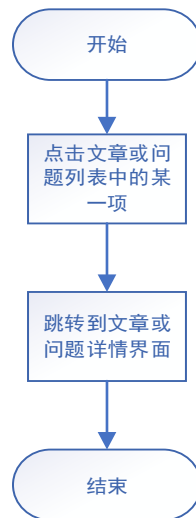


图 5.11 查看文章/问题详情流程图

文章详情界面如图 5.12 所示：

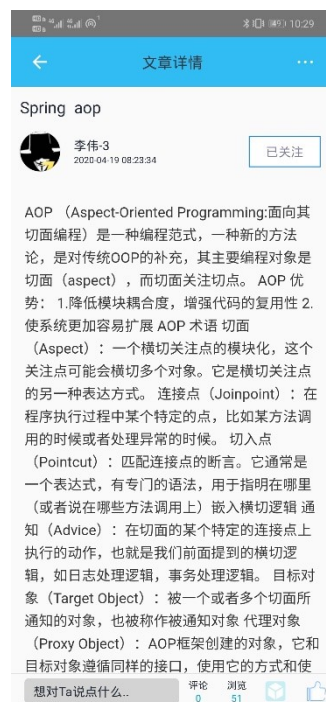


图 5.12 文章详情

## 5.2.4 回复文章/问题

用户可以在文章或问题详情界面点击输入框，然后输入回复内容进行回复。  
回复功能流程图如图 5.13 所示：

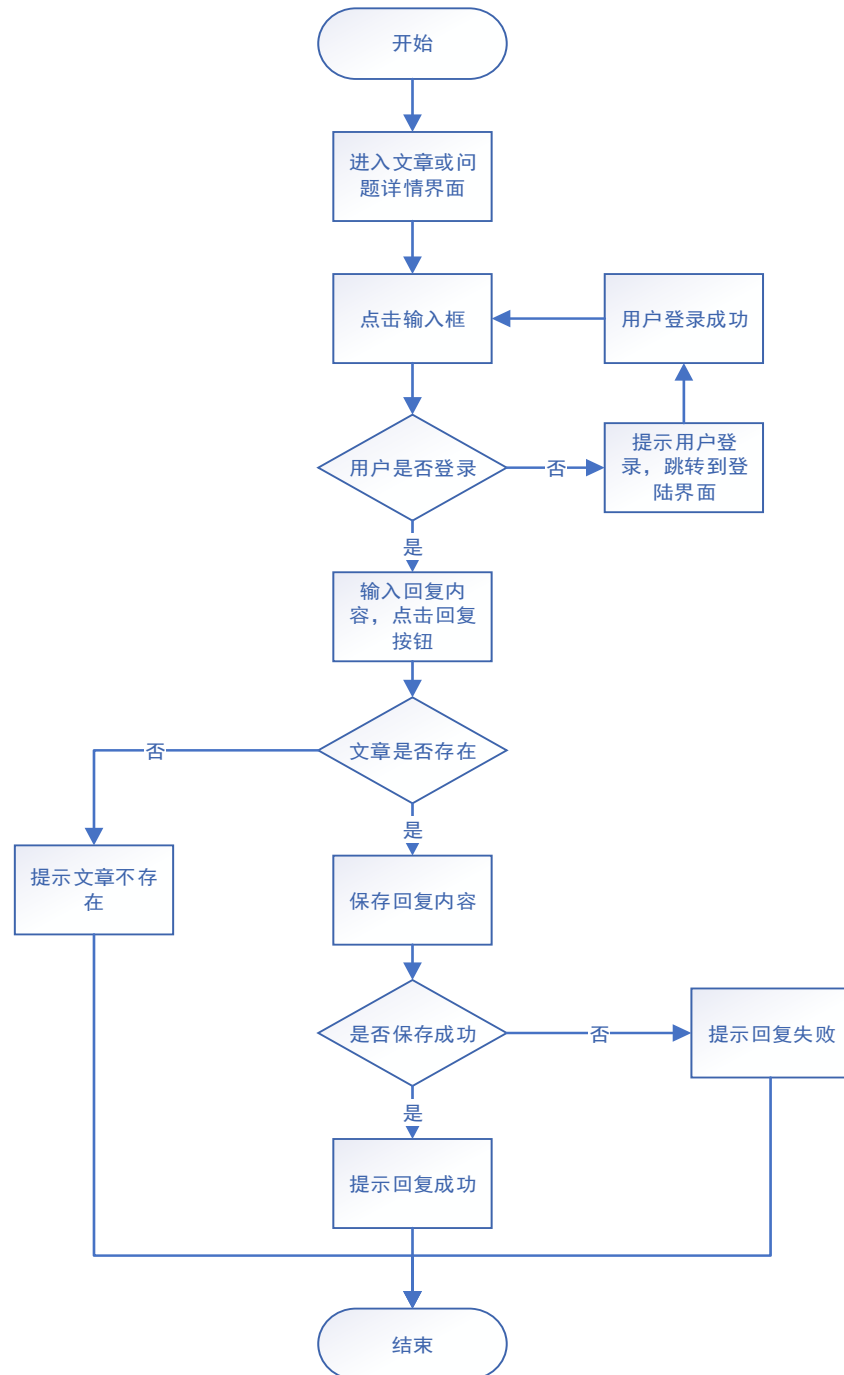


图 5.13 回复功能流程图

文章回复界面如图 5.14、图 5.15 所示：



图 5.14 文章回复

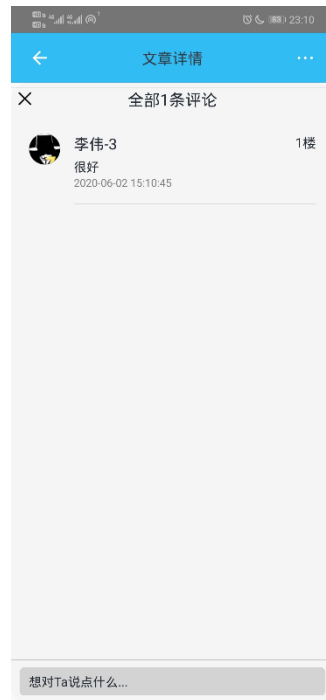


图 5.15 回复列表

### 5.2.5 点赞

用户可以在文章或问题详情界面点击点赞按钮对文章或问题进行点赞。点赞功能流程图如图 5.16 所示：

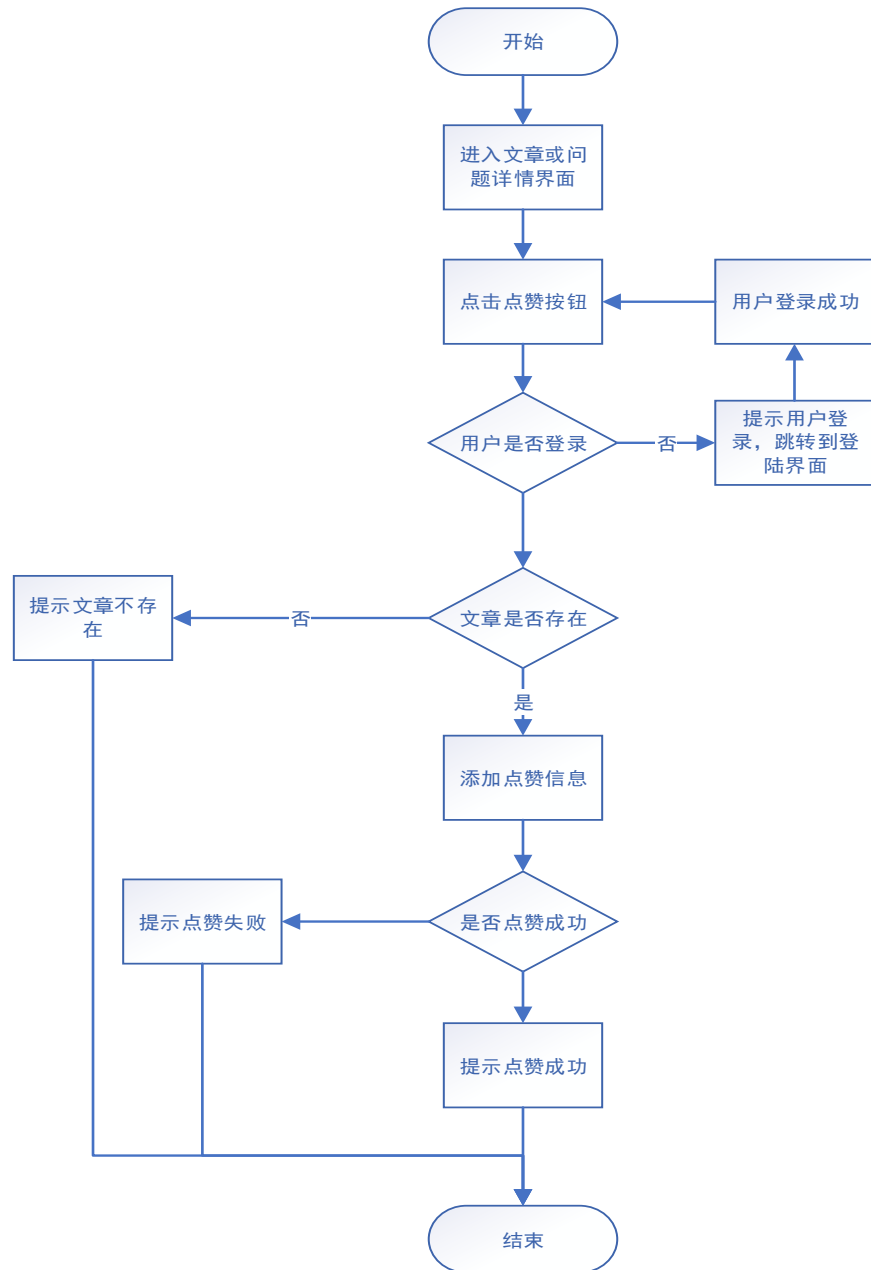


图 5.16 点赞功能流程图

## 5.2.6 收藏

用户可以在文章/问题详情界面内点击收藏按钮收藏文章/问题。收藏功能流程图如图 5.17 所示：

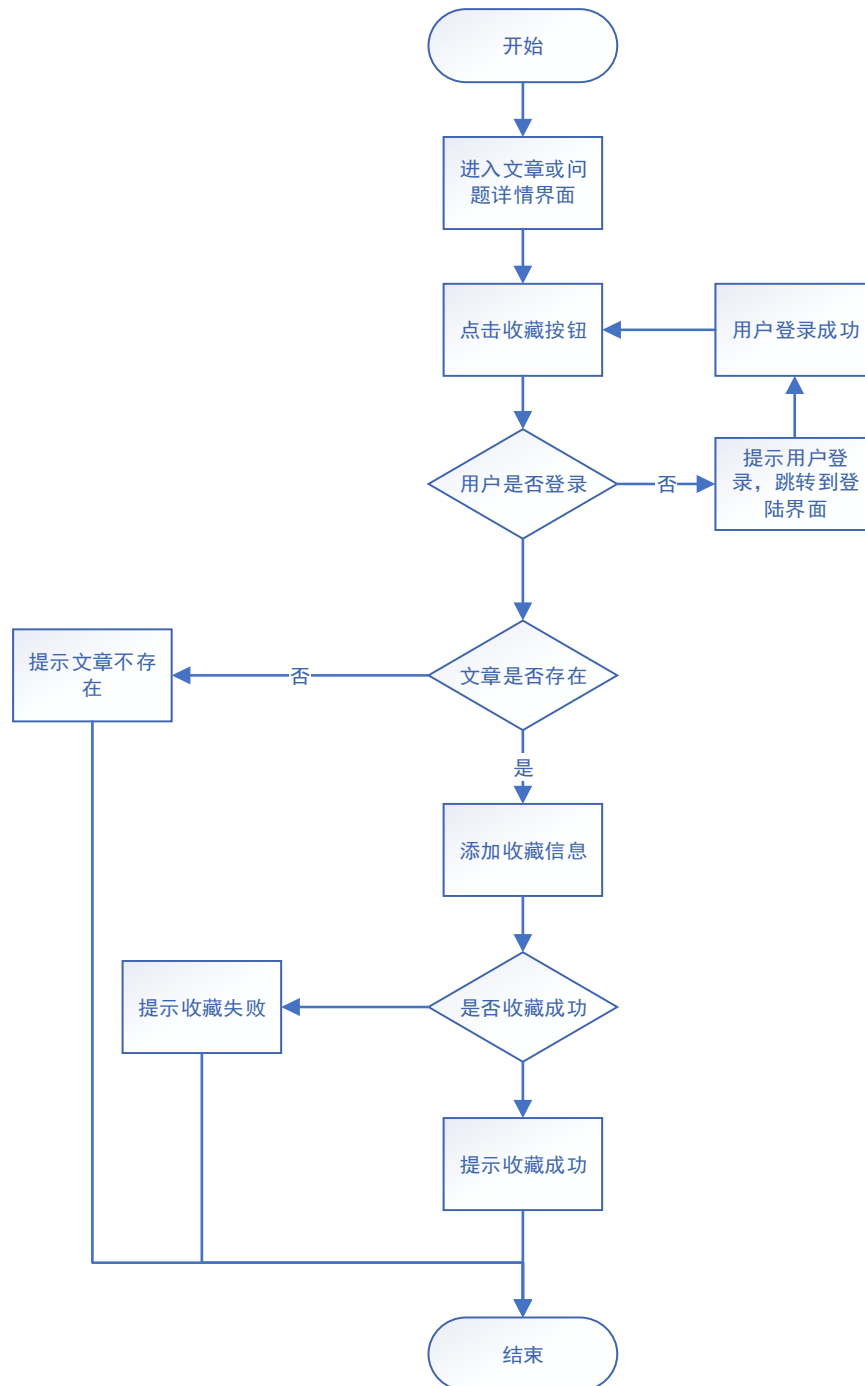


图 5.17 收藏功能流程图

### 5.2.7 置顶

内容管理员、问题解决人可以在文章/问题详情界面设置文章置顶。置顶功能流程图如图 5.18 所示：

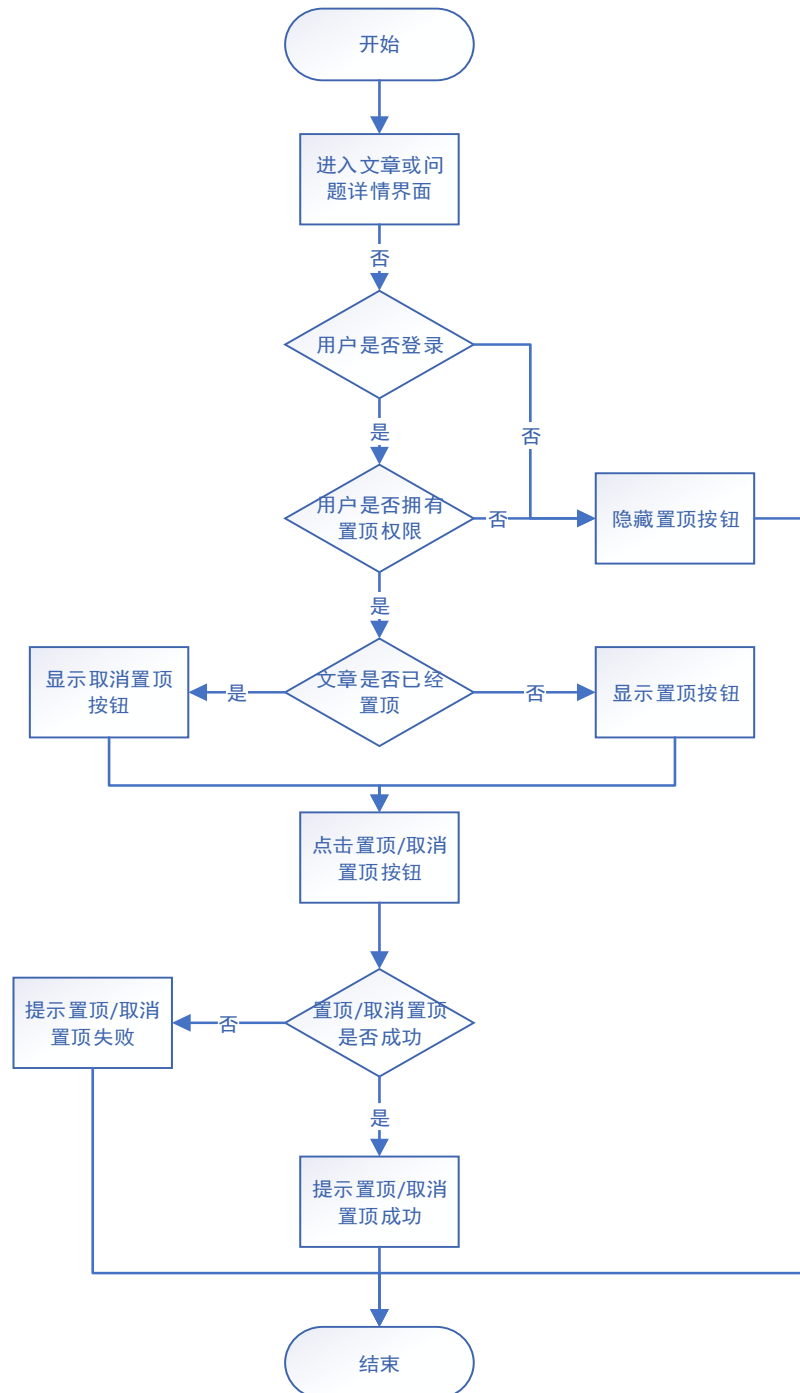


图 5.18 置顶功能流程图



## 5.2.8 设置精华

内容管理员、问题解决人可以在文章/问题详情界面设置文章为精华文章。  
设置精华功能流程图如图 5.19 所示：

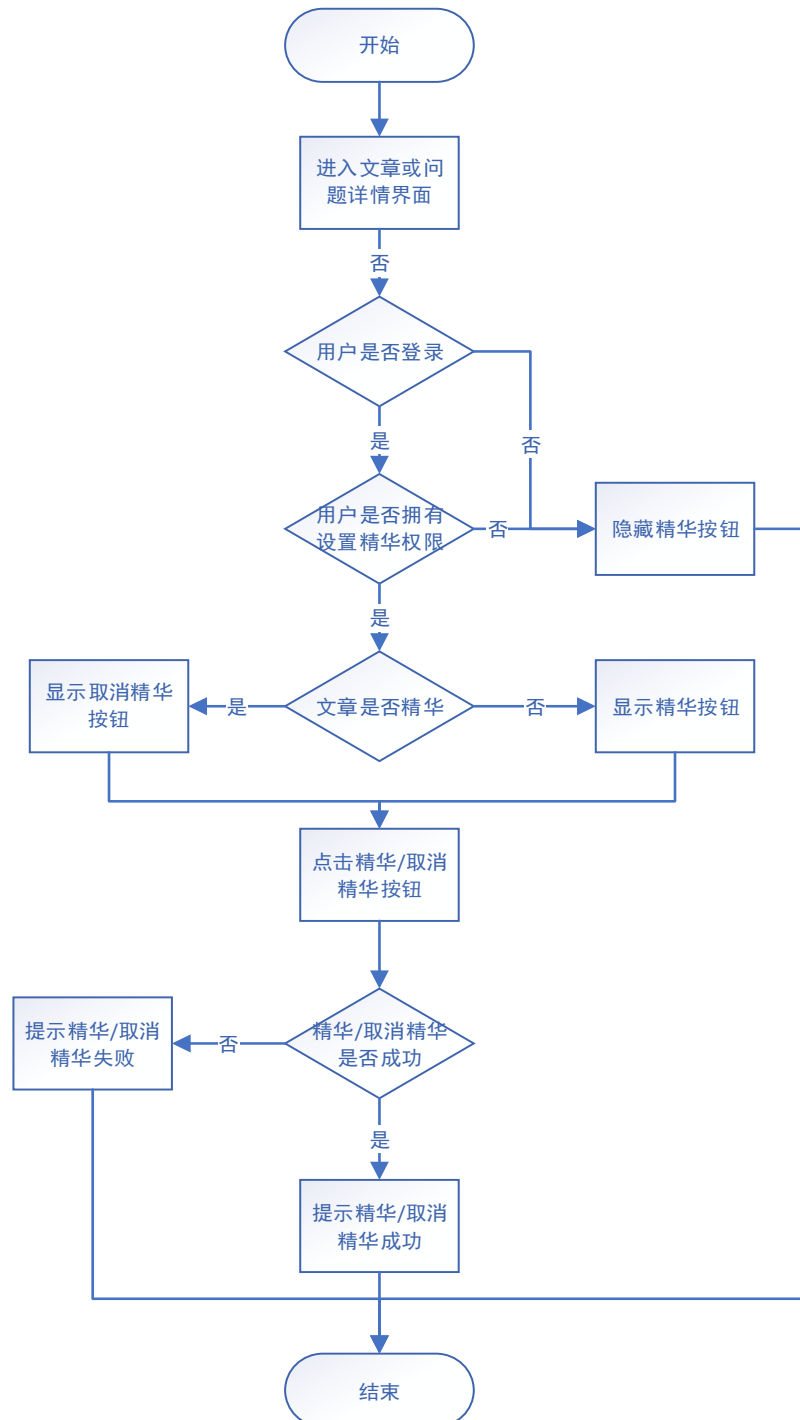


图 5.19 设置精华功能流程图

### 5.2.9 修改问题状态

用户可以在文章/问题详情界面修改问题状态，标记问题是否已被解决。修改问题状态功能流程图如图 5.20 所示：

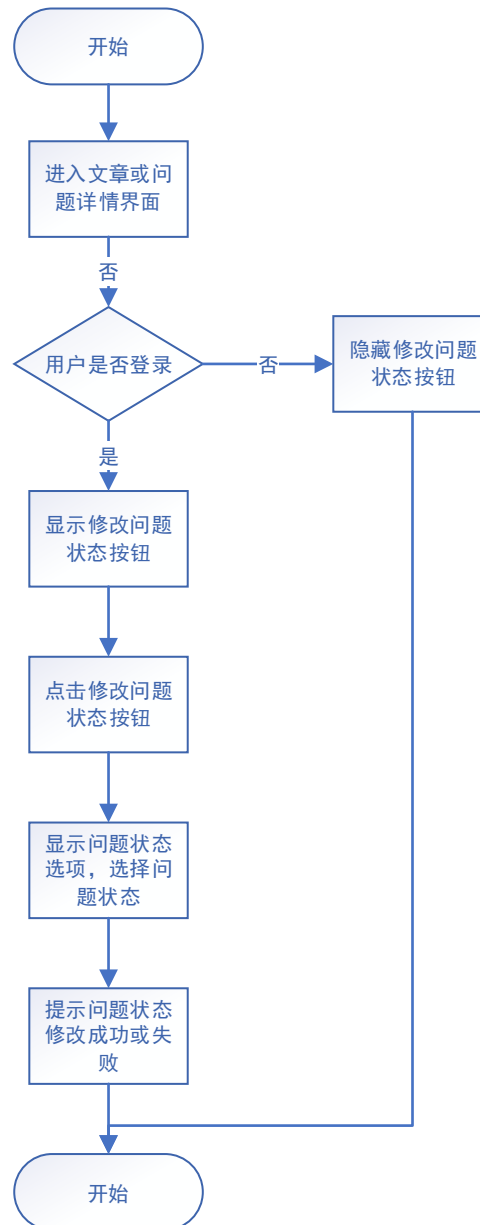


图 5.20 修改问题状态功能流程图

## 5.3 消息模块

### 5.3.1 查看回复消息

用户消息界面点击评论选项,可以查看其他用户回复该用户发布的文章的消息。查看回复消息流程图如图 5.21 所示:

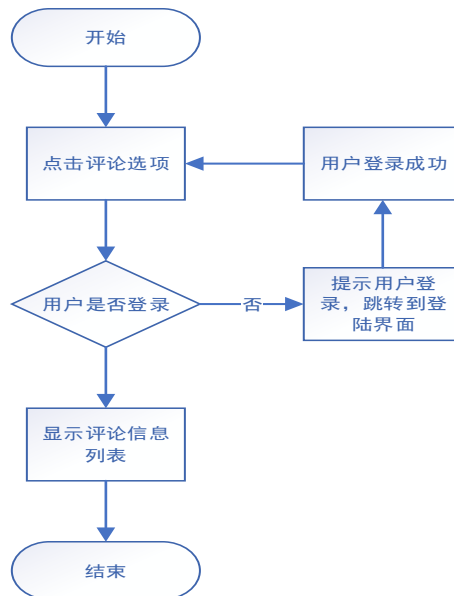


图 5.21 查看回复消息流程图

### 5.3.2 查看关注消息

用户消息界面点击粉丝选项,可以查看其他用户关注该用户的消息。查看关注消息流程图如图 5.22 所示:

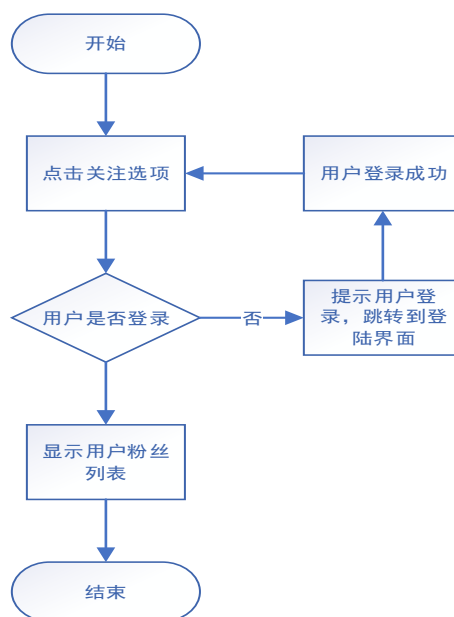


图 5.22 查看关注消息流程图

## 6 系统测试

### 6.1 主要功能测试

#### 6.1.1 用户登录测试

尝试使用不同的账号登录本系统，测试登录功能是否正常。登录测试见表 6.1 所示：

表 6.1 登录测试

编号	ADC_DL		
描述	用户登录		
用例目的	测试 App 用户登录功能		
前提条件	用户已拥有 Coca11 账号		
输入/动作	期望的输出/相应	实际情况	
账号：liwei-4 密码：1759840027	登录成功	符合预期	
账号：liwei-4 密码：1234	登录失败	符合预期	
账号：lijunhua 密码：6789@jkl	登陆成功	符合预期	

#### 6.1.2 发布文章测试

输入文章的标题和正文，然后点击发布，随后跳转到下一界面，选择主题和标签。发布文章测试见表 6.2 所示：

表 6.2 发布文章测试

编号	ADC_FBWZ		
描述	发布文章		
用例目的	测试 App 发布文章功能		
前提条件	用户已登录		
输入/动作	期望的输出/相应	实际情况	
标题：文章测试 1 内容：文章测试 1 主题：技术案例 标签：API	发布成功	符合预期	
标题： 内容：文章测试 2 主题：技术案例 标签：API	提示请输入标题	符合预期	
标题：文章测试 3 内容：	提示请输入文章内容	符合预期	

主题：技术案例		
标签：API		

### 6.1.3 评论文章测试

在文章详情界面输入评论内容评论文章。评论文章测试见表 6.3 所示：

表 6.2 评论文章测试

编号	ADC_PLWZ	
描述	评论文章	
用例目的	测试 App 文章评论功能	
前提条件	用户已登录	
输入/动作	期望的输出/相应	实际情况
评论内容：评论测试 1	评论成功	符合预期
评论内容：评论测试 2	评论成功	符合预期
评论内容：	请输入评论内容	符合预期

### 6.1.4 搜索文章测试

输入搜索关键字可以通过文章标题或内容搜索文章。搜索文章测试见表 6.4 所示：

表 6.2 搜索文章测试

编号	ADC_SSWZ	
描述	搜索文章	
用例目的	测试 App 搜索文章功能	
前提条件	无	
输入/动作	期望的输出/相应	实际情况
搜索关键字：测试 1	显示搜索到的文章列表	符合预期
搜索关键字：测试 2	显示搜索到的文章列表	符合预期
搜索关键字：测试 3	显示搜索到的文章列表	符合预期

## 7 总结与展望

经过长达两个月的开发，整个项目已经接近尾声。Artery 开发者 APP 主要实现了 Cocall 用户登录、关注用户、使用 Markdown 编辑文章/问题、评论、点

赞、置顶、设置精华、自动分配问题、分类查看文章/问题、修改问题状态等功能，满足了用户在使用过程中的发帖评论的基本需求。

本系统采用 C/S 架构，主要分为两个部分——服务端和客户端。

服务端采用 Java 语言进行开发，其中还使用了 Artery 框架、redis 进行数据缓存。系统所使用的数据库是 MySQL 和 ElasticSearch，其中 ElasticSearch 主要用来存储文章/问题信息和评论信息，方便用户通过关键字查询文章/问题信息；而其他的数据则存储到 MySQL 数据库中，方便用户数据之间的关系进行数据的查询。

客户端是一个 Android APP，主要采用原生的 Android 语言进行开发。客户端和服务端通过 JSON 格式的数据进行交互，同时使用 RxJava + Retrofit + OKhttp 网络框架访问服务端的数据。客户端采用了 MVP 的模式降低了代码的耦合性。APP 的界面采用腾讯开源的 QMUI Android 进行开发，其提供的丰富的 Android 控件是 APP 的界面更加的美观和统一，同时降低了界面开发的难度。

由于长时间没有进行过 Android 方面的开发，因此对 Android 项目中所需要使用的 API、设计方法等有一定的生疏，所以对 Android 项目的开发进行了一段时间的学习，同时也加深了自身对 Android 程序的理解。

通过这次开发。我明白大学毕业并不意味着以后可以放弃学习，在今后的工作中，还用要不停的去学习新的技术，通过这些技术去提高自己，增加自己的社会竞争力。

## 参考文献

- [1] 王盛夏. 企业安全投资决策的经济学研究及其系统的开发[D]. 中南大学, 2011.
- [2] 靳岩, 姚尚朗. **Android** 入门开发与实践[M]. 北京: 人民邮电出版社, 2012.
- [3] J.F.DiMarzio. **Android a programmer's Guide**[M].McGraw.Hill, 2012.
- [4] 杨炜. 基于 **Android** 的腾讯微博客户端应用软件设计[D].西安电子科技大学,2013.
- [5] 李佐彬. **Android** 开发入门与实战体验[M].北京: 机械工业出版社, 2010.
- [6] 霍斯特曼科内尔. **Java** 核心技术[M]. 北京: 机械工业出版社, 2014.
- [7] 曾露.MVP 模式在 **Android** 中的应用研究[J].软件,2016,37(06):75-78.
- [8] 倪红军. 基于 MVP 模式的 **Android** 应用开发研究[J]. 电子设计工程, 2018, 26(11):6-9+13.
- [9] 陈亚杰, 王锋, 邓辉, 刘应波. ElasticSearch 分布式搜索引擎在天文大数据检索中的应用研究[J]. 天文学报, 2016, 57(02):241-251.
- [10] 张哲, 刘云鹤, 王乃生. ElasticSearch 分布式搜索引擎在地名地址检索中的应用[J]. 测绘与空间地理信息, 2020, 43(01):184-187.
- [11] 张阔. 基于行为分析的 **Android** 智能移动终端应用安全研判方案研究[D]. 北京交通大学, 2015.
- [12] 陈德林, 赵良福, 王颖. **Android** 操作系统与数字电视中间件的对比分析[J]. 广播与电视技术, 2012, 39(01):87-91.
- [13] 高奕璠. 警方调证客服系统的设计与实现[D]. 北京交通大学, 2019.
- [14] 刀怡婕, 云利军, 刘乔波, 刘智峰, 唐鹏程. 基于物联网技术的旅游团队自动点名系统设计[J]. 电脑知识与技术, 2017, 13(29):55-57.
- [15] 周巍, 何涛, 林嘉宇. **Android** 系统智能手机语音应用开发环境构架[J]. 微处理机, 2011, 32(06):28-32.
- [16] 徐昊. 基于北斗定位的电网柱上断路器监控系统的设计[D]. 华南理工大学, 2019.

## 致谢

天下没有不散的宴席，随着毕业设计的完成，我的大学生活也接近了尾声。在此，我想感谢李晓宇老师和张创伟导师。在我实习期间他们给予了我许多在学习上和在生活上的帮助。

这篇论文所涉及的课题是我和我的指导老师交流后定下的，在前期的实习积累经验，到中期的修改和讨论，及最后的反复斟酌，我希望能尽自己最大的努力，写出一篇具有现实意义的论文。在论文撰写的过程中，李晓宇老师认真的指出我论文中存在的问题，耐心的解答我的疑惑，帮助我完善论文的结构和内容；张创伟导师在我实习过程中指导我完成了许多困难的工作，帮助我确定论文的大方向，为我解决开发中所遇到的一些技术难题。

最后，我要感谢大学生活中的点点滴滴，正是这些点点滴滴使我的大学生活变得更加丰富多彩。同时也要感谢所有的老师同学和我的家人，他们的陪伴与支持让我能够更好面对学习过程中所遇到的困难。