**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：

年 月 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的一个前后端分离的框架。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，使用户可以随时发布帖子来分享自己的Artery学习心得。

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

Design and Implementation of Artery Developer Community App

**Abstract:** Artery is a front-end and back-end separation framework based on Spring Boot independently developed by Beijing Huayu Information Technology Co., Ltd. According to the company's specific business types, the Artery framework further encapsulates Spring Boot, 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 supporting use of 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 sharing and communication platform for Artery developers. The App is based on the Android operating system, using Retrofit + RxJava + OkHttp and MVP code structure. The server uses Artery for development, so that users can post messages at any time to share their own Artery learning experience.

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

目录

[1 绪论 1](#_Toc39000322)

[1.1 开发背景 1](#_Toc39000323)

[1.2 开发目标 1](#_Toc39000324)

[1.3 项目意义 1](#_Toc39000325)

[1.4 可行性分析 1](#_Toc39000326)

[1.4.1 操作的可行性 1](#_Toc39000327)

[1.4.2 技术的可行性 1](#_Toc39000328)

[1.4.3 经济的可行性 1](#_Toc39000329)

[2 参考技术 2](#_Toc39000330)

[2.1 Android 语言 2](#_Toc39000331)

[2.2 网络框架 2](#_Toc39000332)

[2.2.1 Retrofit 2](#_Toc39000333)

[2.2.2 RxJava 2](#_Toc39000334)

[2.2.3 OkHttp 2](#_Toc39000335)

[2.3 MVP 代码架构 3](#_Toc39000336)

[2.4 QMUI Android 3](#_Toc39000337)

[2.5 Artery 4](#_Toc39000338)

[2.6 Elasticsearch 4](#_Toc39000339)

[3 需求分析 4](#_Toc39000340)

[3.1 功能需求 4](#_Toc39000341)

[3.2 非功能需求 4](#_Toc39000342)

[3.3 用例图分析 4](#_Toc39000343)

[4 系统设计 4](#_Toc39000344)

[4.1 系统功能结构图 4](#_Toc39000345)

[4.2 数据库设计 5](#_Toc39000346)

[4.3 模块划分 5](#_Toc39000347)

[5 系统实现 5](#_Toc39000348)

[5.1 用户模块 5](#_Toc39000349)

[5.2 文章模块 5](#_Toc39000350)

[5.3 消息模块 5](#_Toc39000351)

[6 系统测试 5](#_Toc39000352)

[6.1 系统功能测试 5](#_Toc39000353)

[7 总结与展望 5](#_Toc39000354)

绪论

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，同时会有许多与员工在学习使用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。

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。

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。

本项目的提出是为了解决Artery

可行性分析

可行性研究的目的，就是确定问题是否能够解决，而且要求用最小的代价和尽可能短的时间。要达到这个目的，必须分析利弊，从而判断原定的系统规模和目标是否可以实现，系统完成后所能带来的效益是否值得投资开发这个系统。因此可行性研究性质是要进行一次大大压缩简化了的系统分析和设计的过程，也就是在较高层次上以较抽象的方式进行的系统分析和设计的过程。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。

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，只要用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。

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，这是现在最为主流的移动端开发语言。

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，两者只需要约定好数据的传输格式和内容格式便可进行分离开发。客户端采用MVP代码结构，方便开发人员对代码进行编写和维护。

服务端采用Artery框架进行，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。

综上所述， Artery开发者社区APP使用流行的开发方式与技术让整个系统满足客户开发要求。因此，它在技术上可行。

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。

参考技术

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，由Google公司和开放手机联盟领导及开发。Android操作系统最初由Andy Rubin开发，主要支持手机,后来逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机、智能手表等。

Android 系统结构大致可以分为五层，自顶向下分别是系统应用程序、Java API框架、系统运行库、硬件抽象层以及Linux内核。Linux内核是Android平台的基础；Android的硬件抽象层，简单来说，就是对Linux内核驱动程序的封装，向上提供接口，屏蔽低层的实现细节；系统运行库主要包含两个部分核心库以及Android运行环境，这一层提供了一个很关键的模块，Art（早期为Dalvik）虚拟机；系统应用程序包含一系列系统App，例如通讯录、浏览器等等；Java API框架为应用程序提供了许多更高层的服务，例如Activity Manager、Resource Manager、Notifications Manager等。

网络框架

系统采用的是RxJava + Retrofit + OKhttp的网络框架，这套网络框架功能非常强大并且简单易用。

Retrofit

Retrofit 是一个 RESTful 的 HTTP 网络请求框架的封装，网络请求的工作本质上是 OkHttp 完成，而 Retrofit 仅负责 网络请求接口的封装。在服务端返回数据之后，OkHttp 将原始的结果交给 Retrofit，Retrofit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。

Retrofit将 Http请求 抽象成 Java接口：采用 注解 描述网络请求参数 和配置网络请求参数，用 动态代理 动态 将该接口的注解“翻译”成一个 Http的url请求，最后再执行 Http 请求。

RxJava

RxJava是ReactiveX中使用Java语言实现的版本，那什么是ReactiveX呢？ReactiveX就是一种新兴的编程模式--响应式编程，定义为一种基于异步数据流概念的编程模式，可分解为"观察者模式+迭代器模式+函数式编程"。

RxJava的核心便是被观察者Observables与观察者Observer，由Observables发出一系列的事件，Observer通过subscribe()方法进行订阅接收事件并进行处理，类似观察者模式，不同之处在于，若没有观察者，被观察者是不会发出任何事件的。

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。

OkHttp

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，其基础功能相当于Android自带的HttpURLConnection和Apache HTTP Client,但他却比自带的2个Http客户端优越很多，一者是写法简单，二者okhttp处理很多网络复杂问题，如会从很多常用的连接问题中自动恢复。如果系统中服务器配置了多个IP地址，当第一个IP连接失败的时候，OkHttp会自动尝试下一个IP。OkHttp还处理了代理服务器问题和SSL握手失败等等很多问题。

MVP 代码架构

本系统采用MVN架构实现。MVP模式（Model-View-Presenter）可以说是MVC模式（Model-View-Controller）在Android开发上的一种变种、进化模式。MVP模式和MVC模式一样，主要分为三层：Model 层、 View 层、 Presenter 层。其中Model层负责获取数据，数据的来源可以是网络或本地数据库等；View层负责用户交互和视图显示，在android中对应activity和fragment；Presenter 层是 Model 层与 View 层之间通信的桥梁，将Model 层与 View 层分离开来。 MVC 结构框架与 MVP 结构框架的主要区别如图 2.1 所示

图 2.1 MVC 与 MVP 区别

在 Android 传统开发中，使用的依然是 MVC 结构，但在这其中存在许多问题，代表 View 的 layout 资源不能完全解决如何渲染的问题，还需要 Acitvity 的帮助，所以在 Activity 中也有 View 的内容，但是同时， Activity 主要代表了生命周期，拥有生命周期回调，并且常常设置所关联的 View 的监听，进而接收来自用户的输入，所以在一个完整的 Android 项目中 Activity 必然包含 Control 的功能。因此，使用 MVC 并不能完全将 View 与 Control 分离开。这种模式虽然在 web 端开发中有着高效使用，但由于 Android 开发与 web 开发的区别，较大的项目为了降低耦合性，更应该选择 MVP 结构。

QMUI Android

QMUI Android 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。

QMUI Android 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。

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SpringMVC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。

Elasticsearch

Elasticsearch 是一个分布式、高扩展、高实时的搜索与数据分析引擎。它能很方便的使大量数据具有搜索、分析和探索的能力。充分利用Elasticsearch的水平伸缩性，能使数据在生产环境变得更有价值。Elasticsearch 的实现原理主要分为以下几个步骤，首先用户将数据提交到Elasticsearch 数据库中，再通过分词控制器去将对应的语句分词，将其权重和分词结果一并存入数据，当用户搜索数据时候，再根据权重将结果排名，打分，再将返回结果呈现给用户。

Elasticsearch可以用于搜索各种文档。它提供可扩展的搜索，具有接近实时的搜索，并支持多租户。”Elasticsearch是分布式的，这意味着索引可以被分成分片，每个分片可以有0个或多个副本。每个节点托管一个或多个分片，并充当协调器将操作委托给正确的分片。再平衡和路由是自动完成的。“相关数据通常存储在同一个索引中，该索引由一个或多个主分片和零个或多个复制分片组成。一旦创建了索引，就不能更改主分片的数量。

Elasticsearch使用Lucene，并试图通过JSON和Java API提供其所有特性。它支持facetting和percolating，如果新文档与注册查询匹配，这对于通知非常有用。另一个特性称为“网关”，处理索引的长期持久性；例如，在服务器崩溃的情况下，可以从网关恢复索引。Elasticsearch支持实时GET请求，适合作为NoSQL数据存储，但缺少分布式事务。

需求分析

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，后面所有的需要用户信息的操作都使用系统所使用的数据库中获取，但是用户登录验证还是通过Cocall后台服务器进行。系统还需要实现记住密码功能，当用户勾选记住密码后，能够保存当前的用户名和密码，在下次登录时自动填入用户的账号和密码。

1. 文章的编辑、查看

文章使用Markdown的方式进行存储，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。

1. 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。

1. 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。

1. 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。

1. 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。

1. 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。

1. 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）。

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。

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

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CoCall账户进行登录 |
| 触发条件 | 用户点击登录 |
| 前置条件 | 用户拥有CoCall账户 |
| 用例描述 | 1. 用户进入登陆页面 2. 用户输入CoCall账号和密码，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   1. 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 | 1. 用户在首页点击发布文章选项    1. 用户如果未登录，则提示用户 2. 用户输入文章标题和文章内容 3. 用户点击发布，跳转到下一页 4. 用户选择主题和标签，点击发布    1. 用户未选择主题或标签，提示用户需要选择主题或标签    2. 用户发布文章失败，提示用户发布文章失败 5. 发布文章成功，提示用户并跳转到文章详情页面 |
| 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 | 1. 用户在首页点击发起问题选项    1. 用户如果未登录，则提示用户 2. 用户输入问题标题和问题内容 3. 用户点击发布，跳转到下一页 4. 用户选择主题和标签，点击发布    1. 用户未选择主题或标签，提示用户需要选择主题或标签    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条文章或问题 |
| 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 3. 文章或问题内容不存在，提示文章或问题内容不存在 4. 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 3. 用户未登录，提示用户未登录 4. 用户输入要回复的内容，点击回复 5. 文章不存在，提示回复失败 6. 输入框中没有输入内容，提示请输入要回复的内容 7. 用户没有回复权限，提示没有权限 8. 用户未登录或登录失效，提示用户未登录 9. 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 3. 用户未登录，提示用户登录 4. 点赞失败，提示用户点赞失败 5. 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 3. 用户未登录，提示用户登录 4. 收藏失败，提示用户收藏失败 5. 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 3. 用户未登录，提示用户登录 4. 文章不存在，提示文章不存在 5. 用户没有置顶文章的权限，提示用户没有权限 6. 置顶失败，提示用户置顶失败 7. 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 3. 用户未登录，提示用户登录 4. 文章不存在，提示文章不存在 5. 用户没有设置精华文章的权限，提示用户没有权限 6. 设置精华文章失败，提示用户设置精华文章失败 7. 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 3. 用户未登录，提示用户登录 4. 关注失败，提示用户关注失败 5. 关注用户成功 |
| 后置条件 | 无 |

系统设计

系统功能结构图

系统功能结构图

数据库设计

本系统主要使用MySQL进行存储数据，使用elasticsearch实现全文检索。用户、角色、权限、标签、主题、代码、消息及他们的关系表存储在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elasticsearch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。

数据库关系图模型

合理数据库设计对一个项目的成功起着至关重要的作用，为了使系统能够正确高效的运行，经过分析，在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。分析这些实体之间的联系并画出E-R图。由于系统实体较多且属性较多，简化版的E-R图如图4.1所示：

图4.1 Artery开发者社区APP E-R图

数据库主要关系表

最终经过分析，形成了10个数据表；并且按照这些表的格式将数据存入MySQL数据库中。这些表中的表名、属性名、属性类型及其它约束性条件在下面的表中显示。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 |

代码表图表4.8所示：

表4.8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 | 代码类型编号 |
| C\_Code | varchar | 50 | 否 | 代码 |
| C\_Name | varchar | 150 | 否 | 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4.6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\_RECEIVER | varchar | 32 | 是 | 接收人 |
| C\_ID\_CONTENT | 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4.6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\_CONTENT | 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4.6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\_CONTENT | 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4.6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\_USER | 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4.6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\_CONTENT | varchar | 32 | 是 | 点赞的文章 |
| C\_ID\_UP\_USER | varchar | 32 | 是 | 文章发布人 |
| D\_DZSJ | datetimt | 0 | 是 | 点赞时间 |

模块划分

功能模块划分对一个系统设计至关重要。一个合理的模块划分，要做到模块之间独立程度相对较高，做到低耦合和高内聚，从而大幅度的减少代码冗余和增加代码的重用性，进而减少编码的工作量。根据功能需求分析的结果，将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：包括文章的查看、搜索、发布、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。
4. 系统维护：包括对主题、标签的维护。

界面设计

客户端的界面主要以下几部分：登录页、首页、标签页、消息页面、我的页面、文章编辑页、文章详情页、文章搜索页、文章列表页。

1. 首页

系统实现

用户模块

用户登录

系统的用户信息使用CoCall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CoCall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

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5.2所示：

图5.2 关注流程图

文章模块

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。发布文章的流程图如图5.3所示：

图5.3发布文章流程图

发起问题

用户可以在问题编辑界面输入问题的基本信息，然后发起问题。发起问题的流程图如图5.4所示：

图5.4 发起问题流程图

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5.5所示

图5.5搜索文章/问题流程图

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5.6所示：

图5.6查看文章/问题详情流程图

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5.7所示：

图5.7回复功能流程图

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5.8所示：

图5.8点赞功能流程图

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5.9所示：

图5.9收藏功能流程图

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5.10所示：

图5.10置顶功能流程图

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5.11所示：

图5.11设置精华功能流程图

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5.12所示：

图5.12修改问题状态功能流程图

消息模块

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5.13所示：

图5.12查看回复消息流程图

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5.14所示：

图5.12查看关注消息流程图

系统维护

主题维护

内容管理员可以对主题进行维护，包括添加或删除主题。主题维护功能流程图如图5.13所示：

图5.12主题维护功能流程图

标签维护

内容管理员可以对标签进行维护，包括添加或删除标签。标签维护功能流程图如图5.13所示：

图5.12标签维护功能流程图

系统测试

主要功能测试

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 | 登录 | | |
| 用例目的 | 测试App登录功能 | | |
| 前提条件 | 用户已拥有Cocall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 |

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：  主题：技术案例  标签：API | | 提示请输入文章内容 | 符合预期 |

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 |

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 |

总结与展望

本系统在设计和实现的过程中，主要用到了java语言、Android语言；数据存储使用了Mysql和elasticsearch；客户端使用了腾讯开源QMUI来实现界面风格的同一和美化，使用了MVP模式简化了与服务端的数据交互流程。在实现项目的过程中，我不断的接触和学习了新的技术，对以前所学过的相关技术的掌握程度也越来越高。

因此，通过这次开发经历，我明白在今后的工作生涯中不要固步自封，要不断的提高自己、持续的学习新技术，最好养成终生学习的习惯。当达到这样的目标后，会发现许多更简单的解决问题的方法，从而大幅度的提高自己的工作效率和工作能力，做到持续进步，不被社会的快速发展所淘汰。

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.

# 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。在论文撰写的的过程中，李晓宇老师认真的指点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，我要感谢四年的大学生活，感谢我所有老师同学以及我的家人和那些永远也不能忘记的朋友，他们的支持与情感，是我永远的财富。