

注：教师姓名后留有一个空格，后面填写教师职称。下面加下划线。

阅后删除此文本框。

**本科毕业设计**

**好软分享App设计与实现**

|  |  |  |
| --- | --- | --- |
| **学生姓名** | **：** | 陈浩扬 |
| **学号** | **：** | 151543207 |
| **学院** | **：** | 互联网金融与信息工程 |
| **专业** | **：** | 计算机科学与技术 |
| **指导教师** | **：** | 刘宇熹 **职称：**副教授 |
| **提交日期** | **：** | 年 月 日 |

**本科毕业论文（设计）诚信声明**

本人郑重声明：所呈交的本科毕业论文（设计），是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议，除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

 学生签名：

时间： 年 月 日

**关于论文（设计）使用授权的说明**

本人完全了解广东金融学院关于收集、保存、使用学位论文的规定，即：

1.按照学校要求提交学位论文的印刷本和电子版本；

2.学校有权保存学位论文的印刷本和电子版本，并提供目录检索与阅览服务，在校园网上提供服务；

3.学校可以采用影印、缩印、数字化或其它复制手段保存论文。

本人同意上述规定。

学生签名：

时间： 年 月 日

摘 要

如今，信息技术高速发展，手机、电脑等智能设备已经成为了人们生活中不可或缺的一部分。人们通过不同的软件来使用这些智能设备，每时每刻都在享受软件带来的好处。但是，由于软件发展迅速，开发门槛也随之降低，各式各样的软件层出不穷，应用市场达到了饱和的状态。其中软件的质量参差不齐，极大地影响了用户的体验。

本文设计实现的好软分享App提供一个用户分享优质软件的平台，实现了查看、分享、收藏、评论软件的功能，用户可以在这个平台上分享自己使用过的体验良好的软件，也可以从其他用户的分享中根据自身需求找到喜欢的软件，从而提高用户对软件的满意度。

本文先介绍了研究背景，分析了国内外软件市场的发展状况，发现了当今软件市场存在的问题，并提出了本文的目的和意义。在说明开发该系统所需要相关技术后，进行了需求分析，然后对系统进行设计。在设计过程中采用MVC模式，保证了系统功能的同时具有可维护性。最后进行编码并进行了测试。

**[关键词]**：软件；用户；分享；MVC模式；评论

Abstract

Nowadays, mobile phones, computers and other smart devices have become an indispensable part of people’s lives with the rapid development of information technology. People use these smart devices with different software and enjoy the benefits of them at every turn. However, a variety of software emerging and cause the application market reached a state of saturation, for the reason that the rapid development of software and the development threshold is also reduced. The quality of the software is uneven, which greatly affects the user’s experience.

The good software sharing app designed and implemented in this paper is a platform for users to share high quality software, realize the function of viewing, sharing, collecting and commenting on software. Users can share the experience of good software they have used on this platform, or they can find the software they like according to their needs from the sharing of other users, so as to improve the user’s satisfaction with the software.

At first, this article introduces the research background and analyzes the development situation of software market at home and abroad. Then I discover the problems that exist in today’s software market and the purpose and significance of this paper are put forward. After describe the relevant technologies required for the development of the system, I carried out a demand analysis and designing the system with MVC mode, which ensures the maintainability of the system at the same time. Finally, the encoding is carried out and tested.

**[Key Words]:** Software; User; Share; MVC mode; Review

目 录

[摘 要 I](#_Toc3907171)

[Abstract II](#_Toc3907172)

[目 录 III](#_Toc3907173)

[1 绪论 1](#_Toc3907174)

[1.1 研究背景 1](#_Toc3907175)

[1.2 研究的目的与意义 1](#_Toc3907176)

[1.3 主要研究工作 2](#_Toc3907177)

[2 相关技术 2](#_Toc3907178)

[2.1 Android技术与理论 2](#_Toc3907179)

[2.1.1 概述 2](#_Toc3907180)

[2.1.2 体系结构 2](#_Toc3907181)

[2.1.3 Android四大基本组件 3](#_Toc3907182)

[2.1.4 MVC模式 3](#_Toc3907183)

[3 文献综述 4](#_Toc3907184)

[4 需求分析 6](#_Toc3907185)

[4.1 功能需求 6](#_Toc3907186)

[4.1.1 管理角色 6](#_Toc3907187)

[4.1.2 系统功能 7](#_Toc3907188)

[4.2 数据需求 14](#_Toc3907189)

[4.3 非功能需求 15](#_Toc3907190)

[4.3.1 环境需求 15](#_Toc3907191)

[4.3.2 性能需求 16](#_Toc3907192)

[5 总体设计 16](#_Toc3907193)

[5.1 总体设计 16](#_Toc3907194)

[5.2 功能模块设计 17](#_Toc3907195)

[5.2.1 用户管理 17](#_Toc3907196)

[5.2.2 软件管理 22](#_Toc3907197)

[5.2.3 评论管理 26](#_Toc3907198)

[6 详细设计 29](#_Toc3907199)

[6.1 数据库的设计 29](#_Toc3907200)

[6.1.1 主要实体图 29](#_Toc3907201)

[6.1.2 主要实体关系图 31](#_Toc3907202)

[6.1.3 主要关系数据表设计 31](#_Toc3907203)

[7 系统实现 34](#_Toc3907204)

[8 系统测试 39](#_Toc3907205)

[8.1 测试原因 39](#_Toc3907206)

[8.2 测试方法 39](#_Toc3907207)

[8.2.1 正确性测试 39](#_Toc3907208)

[8.2.2 容错性测试 39](#_Toc3907209)

[8.2.3 边界性测试 39](#_Toc3907210)

[8.2.4 性能与效率测试 39](#_Toc3907211)

[8.2.5 易用性测试 39](#_Toc3907212)

[8.3 测试情况 40](#_Toc3907213)

[9 结论与不足 40](#_Toc3907214)

[参考文献 41](#_Toc3907215)

[致 谢 42](#_Toc3907216)

**好软分享App设计与实现**

1 绪论

1.1 研究背景

随着互联网的快速发展和科技的进步，手机、电脑等电子产品与人们的日常生活的联系越来越紧密，给人们带来了许多便利。在生活中或者工作中，人们无时无刻不使用手机中的软件来进行相关的操作[1]。用户需求的增加，推动了软件的发展。随着Android手机用户数量的快速增长，第三方应用商店不断涌现，为用户提供了大量功能丰富的应用程序。应用商店模式的出现，拉近了手机软件开发者与用户之间的距离，使得手机软件的开发、发布、售卖都更加简单，人们也能更容易地进入手机应用软件领域[2]。无论是在电脑还是手机，都能找到成千上万的软件。软件数量庞大，质量自然也就鱼龙混杂，良莠不齐。应用商店作为获取应用信息的主要渠道[3]，如果不能很好地给用户推荐高品质的应用，用户找不到优质的应用，就会影响使用智能手机的体验[4]。目前在国内各大APP市场中提供下载的APP有数十万款，然而多数下载量并不乐观，甚至有部分无人问津。手机商家尚且需要专注于提高产品的质量以满足人们对手机的要求，手机软件更应该如此，只有尽力在各个方面都做到最好，才能够在激烈的市场竞争当中始终占有一席之地[5]。商家必须以消费者的需求为基础，对手机进行优化升级改造，从而不断地提供更高端的手机软件服务[6]。

1.2 研究的目的与意义

本文的主要研究工作是针对软件利用率不高和用户体验不佳的现状，设计一款用户互相分享好软件的APP，并加以实现。用户如果使用到了界面美观，响应速度快，功能可以满足需求的软件，就可以分享在好软分享APP这个平台上，让其他用户了解到这款软件。目的是让更多的用户能够使用优质的软件，同时让这些优质的软件也能够被更多的用户使用。意义在于既能改善用户体验，又能提高优质软件的利用率。

1.3 主要研究工作

通过查看手机用户在应用商店对软件的评价，来了解用户对软件的需求；通过互联网查阅与软件有关的资料，了解应用软件的发展以及用户需求的变化，进行分析和归纳；对好软分享APP进行设计并实现。

2 相关技术

2.1 Android技术与理论

2.1.1 概述

Android是基于Linux平台的开源移动操作系统，包括操作系统、中间件、用户界面和应用软件。Android以层压的方式构建，将各层彼此分离来明确跨层的分工。

2.1.2 体系结构

Android体系结构分三层。底层以Linux内核工作为基础，提供基本功能；中间层包括函数库Library和Dalvik虚拟机；最上层是各种应用框架和应用软件。

（1）应用程序

Android内有一系列的核心应用和应用程序框架，所有的应用都是用Java语言编写的。

（2）应用程序框架

应用程序框架用于简化和重用应用程序的组件。如四大组件：Activity、Service、 ContentProvider、BroadcastReceiver。

（3）系统运行库层

Android定义了一组C/C++开发库提供Android平台的其他组件使用。应用程序框架为开发人员提供了这些功能，但开发人员不能直接使用。

（4）Linux内核层

Android的核心系统服务依赖于Linux2.6内核，如安全性、内存管理、进程管理、网络协议栈和驱动模型。

2.1.3 Android四大基本组件

（1）Activity

Activity是Android中与用户交互最主要的方式，用于将数据展示给用户，并获取用户输入的数据。每个Activity都被实现成一个单独的类，和相应的布局绑定设置显示的界面，并对该界面所产生的时间做出响应。一个Activity控制一个界面，每个Android应用需要多个用户界面，这就代表这个应用将会包含多个Activity，这些Activity组成了Activity栈，当前活动的Activity位于栈顶。

（2）Service

Service和Activity类似，也代表一个单独的Android组件，运行之后也会拥有自己独立的生命周期。不同的事，Service主要用来给另外的组件提供后台服务或者监控其他组件的运行状态，与Windows中的服务类似，所以通常位于后台运行，一般不与用户进行交互，没有图形用户界面。

（3）BroadcastReceiver

广播消息接收器和事件监听器类似，不同的是它所监听的事件源是Android应用中的其他组件，用来进行不同组件之间的通信。

（4）ContentProvider

内容提供者用来进行不同应用之间的数据交换。应用程序继承ContentProvider类，并重写此类用于提供数据和存储数据的方法，就可以将自己的数据提供给其他应用程序共享。

2.1.4 MVC模式

Android程序开发使用MVC(Model-View-Controller)模式。其中，M、V、C分别指的是模型层、视图层、控制层。

MVC模式能够将应用程序的模型层与视图层代码分离，因此界面设计人员和程序员可以分工合作，分别专注于他们所负责开发的部分，从而可以提高开发的效率，最后通过控制层来保持模型层与视图层的数据的一致性。在Android中MVC各部分对应的关系如下。

（1）视图层

在Android中，所有界面控件都从View类继承，一个界面可以包含多个View对象。在Android中，为每个View定义了相应的XML标签，并为XML标签定义了各种属性，这些属性通常在XML布局中定义。

（2）控制层

控制层的重任通常由Activity和Intent来实现，一个Activity可以使用多种界面，通过setContentView()方法指定用来显示数据的视图模型。

（3）模型层

这一层主要处理数据库、网络和业务计算等操作，模型层主要使用Java来实现。

# 3 文献综述

从目前的数据来看，中国是世界手机用户最多的国家，也是智能手机用户群体最大的国家，所以移动互联网在中国具有很大的发展空间和潜力。根据GNNIC调查数据显示，2010年中国智能手机销售额达到了400亿元，专家预测2011年中国智能手机将比2010年增长30%。总的来说，智能手机和移动互联网已经在中国形成了一定的规模[7]。

Android应用商店作为移动设备流量的入口，在市场中占据着重要的地位。应用商店之所以是移动互联网时代厂商互相争夺的目标，是因为应用商店会给厂商带来不少的好处。应用商店不仅能够增加消费者的用户粘性，保证厂商的用户群体的稳定性，还能给厂商带来全新的利润增长。以上这些都足以见得，推出应用商店对厂商而言是一举多得的。为此，各大公司纷纷推出应用商店来抢占市场份额，但这些应用商店普遍存在某些问题，比如过于复杂臃肿，没有抓住用户的核心需求；或者广告和推送过多，使用户不胜其烦；还有可能提供的应用的可用度不高，甚至有一些盗版和不安全的应用存在[8]。

2016年，只能手机出货量增长放缓，存量市场中APP覆盖率趋近饱和。仅从应用市场的数据情况来看，2015年诞生的APP中有90.19%倒下。因为各个行业的移动互联网产品已发展成型，无论是该领域出现的一枝独秀的独角兽公司，还是两虎相争、三足鼎立，几乎容不下任何新起之秀的摸索和试错。截止2016年4月，我国市场中的移动应用累计数量达到671万款，我国本土第三方应用商店移动应用累计数量超过500万款，苹果商店移动应用累计数量超过163万款，而尚未官方进入我国市场的谷歌商店移动应用分析公司Flurry最新报告，从2016年1月到10月这期间，国内Android应用的下载量就较2015年全年增长了870%[9]。

目前在国内各大APP市场中提供下载的APP有数十万款，然而多数下载量并不乐观，甚至有部分无人问津。大多数APP存在的问题是，它们给用户带来的体验并不是非常好。它们满足不了用户的需求，操作方式过于繁琐，甚至会使用户产生不好的情绪[10]。同时，移动APP版本也有许多变种，互联的画面极其复杂。例如，在Android热门应用Top10中，只计算不同名称，十个热点移动应用就又有186款变种，其中QQ就有近50款变种[11]。由于选择带来的麻烦，用户下载APP的满意度会随所下载APP数量的增加而递减。

Hammad Khalid根据APP销售平台的信息，对手机APP的用户评价进行整理，总结出一些用户体验的情况，并进行分类[12]。比较表明，功能性错误、配置要求和APP不兼容是最让用户头疼，为IOS APP的软件修复提供了好方向。

对移动APP的特殊研究有对新闻类手机APP的研究，主要为了应对新闻营销和使用方法，Pew Research Center（2012）[13]在研究中指出，美国正在进入媒体新时代，通常使用的平台有智能手机、电脑和平板，今天最常见的方式就是进入专门的新闻APP来获取信息。Josh Clark(2011)[14]在苹果移动APP的专门期刊上详细地讨论了苹果手机APP的交互和视觉设计。此外，Gilbert等[15]在对价格折扣对手机APP使用行为影响的研究中证实了价格降低可以提高消费者的购买力度[16]。

Google Play前名为Android Market，是Google公司为Android系统开发的应用商店，以最多最全的应用存储占领了应用商店的主流地位。Google账号相关功能的统一性，更导致了Google Play一家独大的现状[17]。此外，还有一些和品牌联系较为密切的应用商店也占据了一些份额，例如亚马逊商店、三星商店等。相反，其他应用商店只在特定地区或者特定版本上有一定的用户群体。

综上所述，现如今软件越来越多，但是质量却良莠不齐。有些软件体型臃肿，操作繁琐，附带一些不需要的功能，造成了卡顿的情况，影响使用；有些软件隐藏恶意木马，收集个人信息，威胁信息安全甚至手机安全；有些好的软件由于经费不足，用户较少，难以推广等等。这一现状不利于软件的发展，为了改善这一现状，使用户更容易地找到满足需求的软件，对好软件进行分享是有必要的。好软件需要满足一些条件：能够满足用户的需求；使用简单易上手，界面美观，响应速度快，用户体验良好；没有多余的功能，不携带木马病毒。用户如果用到了好的软件，会有一种满足感，也就是有良好的用户体验，并会产生一种想分享给其他人的心情，让更多的人能够用到这款好软件。在一些资源网站上就有好软分享的栏目，但是总体来看比较凌乱，而且有一些是重复的。因此需要好软分享APP这个平台，用户可以分享自己使用过的好软件，也可以搜索其他用户分享的好软件。看到喜欢的软件可以收藏起来，看到使用过的软件也可以发表自己的评论，给其他用户参考。

4 需求分析

4.1 功能需求

4.1.1 管理角色

系统用户包括系统管理员、游客、登陆用户3类人群用户。不同用户设置使用的系统资源不同，用户权限明晰。

系统管理员：最高管理权限。对用户、软件以及评论进行管理，维护系统环境。

游客：只拥有搜索和查看权限。能够对分享的软件进行搜索和查看，能够对分享的软件的留言进行查看。

登陆用户：拥有搜索、分享、查看和收藏权限。能够搜索、查看、收藏用户分享的软件；能够自己分享软件；能够发表和回复评论。

系统管理角色见下表4.1:

表4.1 系统管理角色情况

|  |  |  |
| --- | --- | --- |
| **角 色** | **职责或功能** | **备 注** |
| 管理员 | 完整的系统权限 |  |
| 游客 | 有查询权限 |  |
| 登陆用户 | 使用系统权限 |  |

4.1.2 系统功能

经过系统需求分析，将好软分享APP用户分成以下三类：

（1）管理员管理

管理员通过系统管理模块管理登陆用户的个人信息、分享的软件信息与评论信息等，见下图4.1。



图4.1 管理员管理用例图

管理员通过软件管理对用户分享的软件进行管理，例如管理员将违规软件删除等操作，见下表4.2。

表4.2 软件管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 软件管理 |
| 用例描述 | 管理员对用户分享的软件进行管理 |
| 参与者 | 管理员 |
| 前置条件 | 管理员成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 管理员进入系统后，查看软件分享的列表，对用户分享的软件进行相应的管理，如删除。 2. 管理员进入系统后，点击已删除软件的列表，对误删除的软件进行恢复。 |
| 可选操作流 | 1. 无 |

管理员通过用户管理对用户管理，例如管理员对违规用户进行禁言、封禁等操作，见下表4.3。

表4.3 用户管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 用户管理 |
| 用例描述 | 管理员进行用户管理 |
| 参与者 | 管理员 |
| 前置条件 | 管理员成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 管理员进入系统后，点击进入用户列表界面，对违规的用户进行相应的管理，如禁言、封禁等。 2. 管理员进入系统后，点击进入用户列表界面，对被封禁的用户进行解封。 |
| 可选操作流 | 1. 无 |

管理员通过评论管理对用户发表的评论进行管理，例如管理员对违规的评论进行删除等操作，见下表4.4。

表4.4 评论管理用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 评论管理 |
| 用例描述 | 管理员进行评论管理 |
| 参与者 | 管理员 |
| 前置条件 | 管理员成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 管理员进入系统后，点击软件分享，进入软件分享界面，对用户的评论进行相应的管理，如删除。 |
| 可选操作流 | 1. 无 |

（2）游客管理

游客主要搜索和查看分享的软件，以及查看评论，见下图4.2。



图4.2 游客管理用例图

游客能够对登陆用户分享的软件进行查看，见下表4.5。

表4.5 查看软件用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 查看软件 |
| 用例描述 | 游客浏览用户分享的软件 |
| 参与者 | 游客 |
| 前置条件 | 游客成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 游客进入系统后，点击软件分享，进入软件分享界面，浏览用户的分享的软件。 |
| 可选操作流 | 1. 无 |

游客能够查看登陆用户对分享的软件的评论，见下表4.6 。

表4.6 查看评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 查看评论 |
| 用例描述 | 游客查看用户评论 |
| 参与者 | 游客 |
| 前置条件 | 游客成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 游客进入系统后，点击软件分享，进入软件分享界面，查看用户对软件的评论。 |
| 可选操作流 | 1. 无 |

（3）登陆用户管理

登陆用户主要查看、搜索、收藏、分享软件，发表评论和回复评论等，见下图4.3。



图4.3 登陆用户管理用例图

登陆用户通过本系统分享软件，见下表4.7。

表4.7 分享软件用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 分享软件 |
| 用例描述 | 登陆用户分享软件 |
| 参与者 | 登陆用户 |
| 前置条件 | 用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 登陆用户进入系统后，点击软件分享，进入软件分享界面，填写软件的名字等相关信息并上传相关截图。 2. 点击发表按钮，完成对软件的分享。 |
| 可选操作流 | 1. 登陆用户放弃分享时，软件分享不成功，数据库没有保存纪录。 |

登陆用户通过本系统查看其他用户分享的软件，见下表4.8。

表4.8 查看软件用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 查看软件 |
| 用例描述 | 登陆用户浏览用户分享的软件 |
| 参与者 | 登陆用户 |
| 前置条件 | 登陆用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 登陆用户进入系统后，点击软件分享，进入软件分享界面，浏览用户的分享的软件。 |
| 可选操作流 | 1. 无 |

登陆用户看到喜欢的软件，可以进行收藏等操作，见下表4.9。

表4.9 收藏软件用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 收藏软件 |
| 用例描述 | 登陆用户收藏其他用户分享的软件 |
| 参与者 | 登陆用户 |
| 前置条件 | 用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 登陆用户进入系统后，点击软件分享，进入软件分享界面，浏览用户的分享的软件，遇到喜欢的软件点击收藏按钮即可收藏。 |
| 可选操作流 | 1. 登陆用户放弃收藏时，再点击一次可以取消收藏 |

登陆用户在查看软件时可以对软件进行评论，见下表4.10。

表4.10 进行评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 进行评论 |
| 用例描述 | 登陆用户对用户分享的软件进行评论 |
| 参与者 | 登陆用户 |
| 前置条件 | 用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 游客进入系统后，点击软件分享，进入软件分享界面，浏览用户的分享的软件，并对已使用过的软件进行评论。 |
| 可选操作流 | 1. 无 |

登陆用户在查看软件时，可以查看其他用户对软件的评论，见下表4.11。

表4.11 查看评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 查看评论 |
| 用例描述 | 登陆用户查看用户评论 |
| 参与者 | 登陆用户 |
| 前置条件 | 用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 登陆用户进入系统后，点击软件分享，进入软件分享界面，查看用户对软件的评论。 |
| 可选操作流 | 1. 无 |

登陆用户对发表的评论不满意时，可以删除自己发表的评论，见下表4.12。

表4.12 删除评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 删除评论 |
| 用例描述 | 登陆用户删除自己评论 |
| 参与者 | 登陆用户 |
| 前置条件 | 用户成功进入系统 |
| 后置条件 | 系统给出操作后的提示信息 |
| 基本操作流 | 1. 登陆用户进入系统后，点击软件分享，进入软件分享界面，查看用户对软件的评论，若不满意自己的评论，可以删除。 |
| 可选操作流 | 1. 无 |

4.2 数据需求

（1）系统概念类图见下图4.4。



图4.4 好软分享APP概念类图

（2）用户管理类图缩略表示见下图4.5。



图4.5 用户管理类图缩略表示

（3）软件管理类图缩略表示见下图4.6。



图4.6 软件管理类图缩略表示

（4）评论管理类图缩略表示见下图4.7。



图4.7 评论管理类图缩略表示

4.3 非功能需求

4.3.1 环境需求

客户端配置要求见下表4.14。

表4.14 客户端配置要求

|  |  |
| --- | --- |
| 操作系统 | Android5.0或以上版本操作系统 |
| 内存 | 2GB以上 |
| 分辨率 | 1280\*720以上 |

4.3.2 性能需求

（1）易操作性

界面设计：提供简约、美观、实用、方便和直观的用户图形界面，简单易用，操作方便。

（2）可维护性

维护方法：提前进行备份系统，在系统出现无法回复的错误时可以进行恢复。提供对系统自身的集中操作维护的功能，在发生故障时以最快的速度排除故障并恢复正常运行。

（3）可扩展性

代码易添加新功能或者删除已有的功能，软件能适应环境的变化以达到用户的需求，这就要求在设计时各个组件之间的耦合性不能过高。

（4）可靠性

系统能提供完整的运行日志；系统提供的功能完善，能准确反映用户的需求；没有较大的错误，数据格式应复合国家相关标准及行业标准。系统崩溃或出错时可以回滚到上一次操作中，忽略当前此次操作[18]。

（5）容错性

在用户进行软件操作时，若出现误操作或者输入错误，系统应具有一定的容错机制，对用户进行提示引导至正确的操作。

5 总体设计

5.1 总体设计

系统共分为3个子模块，在好软分享APP中包含用户管理、软件管理、评论管理共3个模块，见下图5.1。



图5.1 系统体系结构图

5.2 功能模块设计

5.2.1 用户管理

（1）功能结构设计

用户管理包图见下图5.2。



图5.2 用户管理包图

（2）类图设计

用户管理类图见下图5.3。



图5.3 用户管理类图

（3）动态图设计

用户注册动态图见下图5.4。



图5.4 用户注册动态图

用户登陆动态图见下图5.5。



图5.5 用户登陆动态图

用户封禁动态图见下图5.6。



图5.6 用户封禁动态图

（4）核心处理流程

用户注册活动图见下图5.7。



图5.7 用户注册活动图

用户登陆活动图见下图5.8。



图5.8 用户登陆活动图

用户封禁活动图见下图5.9。



图5.9 用户封禁活动图

5.2.2 软件管理

（1）功能结构设计

软件管理包图见下图5.10。



图5.10 软件管理包图

（2）类图设计

软件管理类图见下图5.11。



图5.11 软件管理类图

（3）动态图设计

软件分享动态图见下图5.12。



图5.12 软件分享动态图

软件收藏动态图见下图5.13。



图5.13 软件收藏动态图

软件删除动态图见下图5.14。



图5.14 软件删除动态图

（4）核心处理流程

软件分享活动图见下图5.15。



图5.15 软件分享活动图

软件收藏活动图见下图5.16。



图5.16 软件收藏活动图

软件删除活动图见下图5.17。



图5.17 软件删除活动图

5.2.3 评论管理

（1）功能结构设计

评论管理包图见下图5.18。



图5.18 评论管理包图

（2）类图设计

评论管理类图见下图5.19。



图5.19 评论管理类图

（3）动态图设计

发表评论动态图见下图5.20。



图5.20 发表评论动态图

删除评论动态图见下图5.21。



图5.21 删除评论动态图

（4）核心处理流程

发表评论活动图见下图5.22。



图5.22 发表评论活动图

删除评论活动图见下图5.23。



图5.23 删除评论活动图

6 详细设计

6.1 数据库的设计

6.1.1 主要实体图

管理员信息表实体属性图见下图6.1。



图6.1 管理员信息表实体属性图

用户信息表实体属性图见下图6.2。



图6.2 用户信息表实体属性图

分享信息实体属性图见下图6.3。



图6.3 分享信息实体属性图

评论信息实体属性图见下图6.4。



图6.4 评论信息实体属性图

软件类别实体属性图见下图6.5。



图6.5 软件类别实体属性图

收藏信息实体属性图见下图6.6。



图6.6 收藏信息实体属性图

6.1.2 主要实体关系图

系统ER图见下图6.7。

图6.7 系统ER图

6.1.3 主要关系数据表设计

（1）管理员信息：保存管理员的基本信息，见下表6.1。

表6.1 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| admin\_id | int | 5 | 是 | 管理员ID |
| admin\_account | varchar | 10 | 否 | 管理员账号 |
| admin\_password | varchar | 20 | 否 | 管理员密码 |

（2）用户信息：保存用户的基本信息，见下表6.2。

表6.2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| user\_id | int | 10 | 主键 | 用户ID |
| user\_account | varchar | 10 | 否 | 用户账号 |
| user\_password | varchar | 20 | 否 | 用户密码 |
| user\_name | varchar | 10 | 否 | 用户昵称 |
| user\_head | blob |  | 否 | 用户头像 |
| user\_register | datetime |  | 否 | 注册时间 |
| user\_status | varchar | 10 | 否 | 用户状态 |

（3）软件分享信息：保存用户分享的软件的信息，见下表6.3。

表6.3 软件分享信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| share\_id | int | 10 | 是 | 分享ID |
| share\_software | varchar | 10 | 否 | 软件名称 |
| share\_icon | blob |  | 否 | 软件图标 |
| share\_type | varchar | 10 | 否 | 软件类型 |
| share\_publish | varchar | 10 | 否 | 发布者 |
| share\_intro | varvhar | 50 | 否 | 简介 |
| share\_screenshot | blob |  | 否 | 截图 |
| share\_details | varchar | 100 | 否 | 详细介绍 |
| share\_url | varchar | 50 | 否 | 下载地址 |
| share\_user | varvhar | 10 | 否 | 分享人 |
| share\_time | datetime |  | 否 | 分享时间 |

（4）评论信息：保存用户对软件评论的信息，见下表6.5。

表6.5 评论信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| comment\_id | int | 10 | 是 | 评论ID |
| share\_id | int | 10 | 否 | 分享ID |
| comment\_content | varchar | 150 | 否 | 评论 |
| comment\_user\_id | int | 10 | 否 | 评论人ID |
| comment\_time | datetime |  | 否 | 评论时间 |
| reply\_id | int | 10 | 否 | 回复的评论的ID |

（5）软件类别信息：保存分享的软件的类别信息，见下表6.6。

表6.6 软件类别信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| type\_id | int | 10 | 是 | 类别ID |
| type\_name | varchar | 10 | 否 | 类别名称 |

（6）软件收藏信息：保存用户收藏的软件的信息，见下表6.7。

表6.7 软件收藏信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **是否主键** | **描述** |
| collect\_id | int | 10 | 是 | 收藏ID |
| collect\_user\_id | int | 10 | 否 | 收藏人ID |
| collect\_share\_id | int | 10 | 否 | 分享ID |
| collect\_time | datetime |  | 否 | 收藏时间 |

7 系统实现

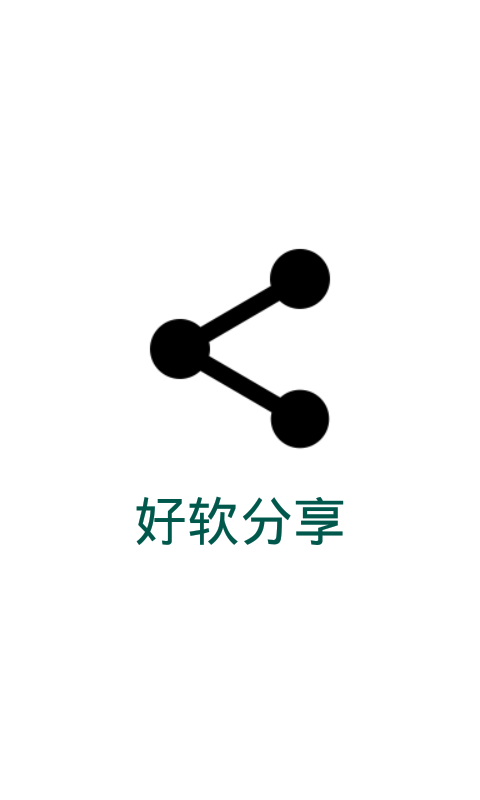


图7.1 启动界面

启动页如图7.1所示。使用ImageView显示图标，使用TextView显示软件名称，并居中显示。在LauncherActivity中设置全屏显示，并使用Intent跳转到MainActivity，设置延迟3秒。在AndroidManifest.xml中配置过滤器，优先启动LauncherActivity，并隐藏标题栏ActionBar。

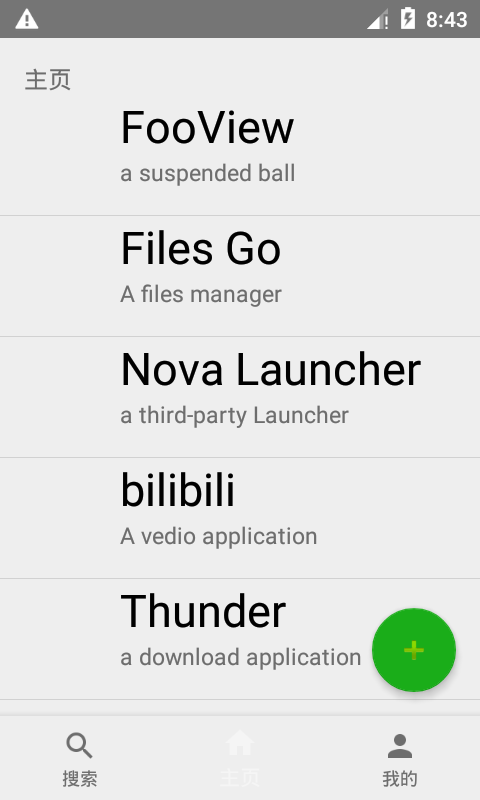


图7.2 主页界面

主页如图7.2所示，底部使用系统默认的navigation导航栏，在res目录中单独创建一个Menu文件夹，并创建一个navigation.xml定义导航栏中各个项的属性。上面使用fragment来为各个项显示不同的页面，标题栏显示当前项的名称。主页中使用的是垂直的线性布局，采用ListView来显示软件列表，列表中的每一项用item来布局，左边显示图标，右边显示软件的名称和一句简短的介绍。右下角绿色图标用于添加分享，采用了FloatingActionButton，点击后进入填写信息的分享页面。



图7.3 搜索界面

搜索界面如图7.3所示，也是使用垂直线性布局，采用ListView来显示，在values文件夹创建一个arrays.xml用来存放数组，这里用一个命名为category数组来保存软件的类别信息。

我的界面如图7.4所示，使用了垂直线性布局，采用ListView来显示，同样在arrays.xml定义一个命名为my的数组来保存我的信息。下面设置一个按钮，用于退出软件。如图7.5所示。

图7.4 我的界面 图7.5 退出程序界面

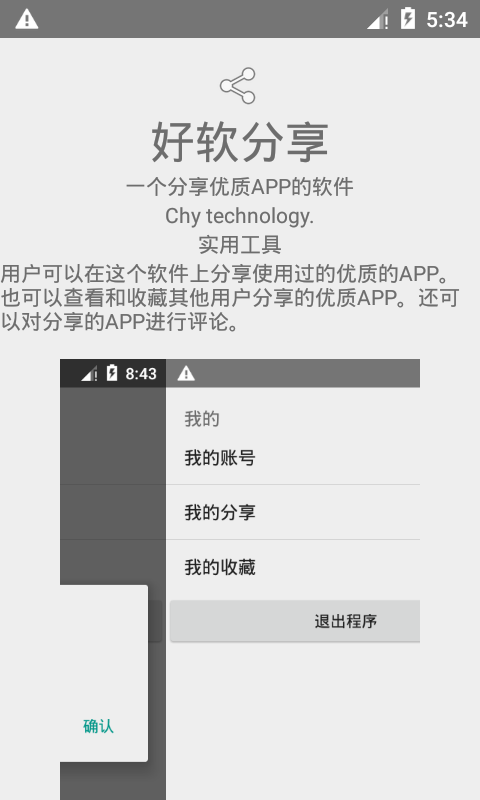
 

图7.6 分享详情页面

分享详情页面如图7.6所示，使用垂直线型布局，采用了ScrollView，在页面中添加了滚动条，以查看页面不能完全显示的部分。其中采用ViewFlipper，通过图片轮播的方式来展示软件截图。在res文件夹中创建animate文件夹，并添加两个文件，slide\_from\_right.xml和slide\_to\_left.xml，通过这两个文件来设置图片从右侧平移进入，从左侧平移退出。显示下载地址的网站使用TextView的autoLink属性识别链接用于点击。发表评论所在行使用水平线性布局，并通过使用FrameLayout以显示在最上层。

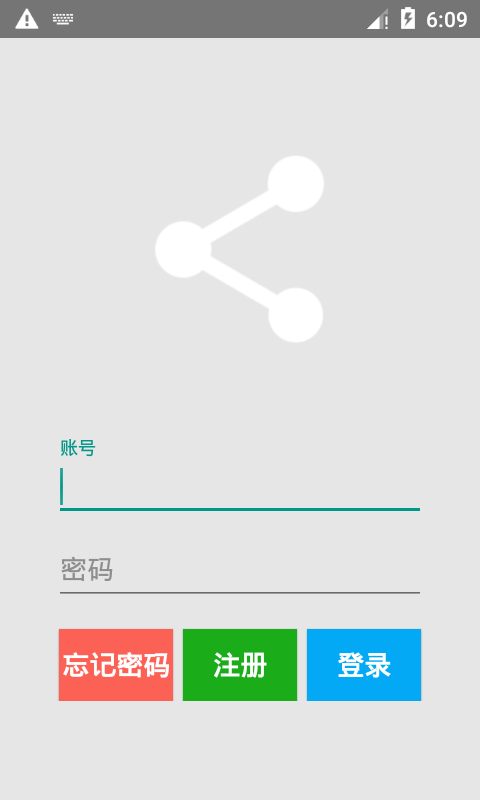


图7.7 登陆界面

登陆界面如图7.7所示，整体采用垂直线性布局，按钮一栏再嵌套一个水平线性布局，其中每个按钮通过layout\_weight属性来平分空白区域，使得每个按钮大小相同。

注册界面如图7.8所示，使用垂直线型布局，其中验证码一栏也是通过layout\_weight属性来平分空白区域，达到平分秋色的效果。



图7.8 注册界面

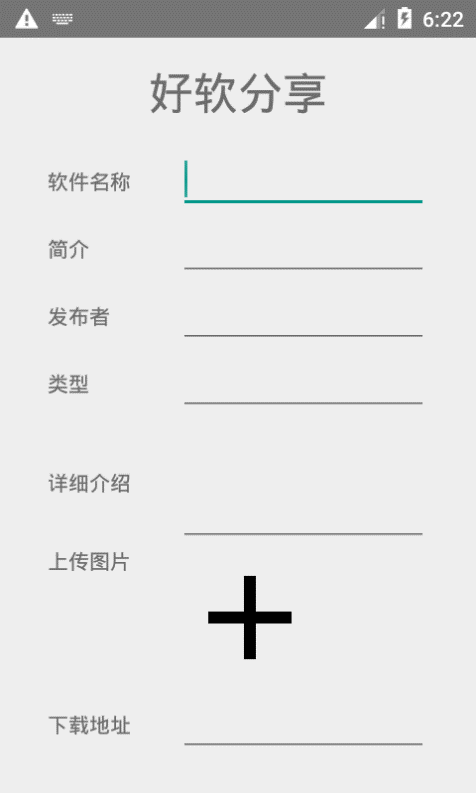
 

图7.9 软件分享界面上 图7.10 软件分享界面下

软件分享界面如图7.9和7.10所示，采用表格布局，左边一列显示提示信息，右边一列用于用户输入。

8 系统测试

8.1 测试原因

一个原因是软件的工程化程度越来越高。复杂度越来越高，因此出现问题的概率越来越大；另一个原因是只有通过测试，才能发现更多的bug，只有修复了这些bug才能保证软件的高质量。

8.2 测试方法

8.2.1 正确性测试

正确性测试又称为功能测试，主要测试软件的功能是否满足软件规格说明书的要求，能否正确的运行。使用黑盒测试的方法，通过构造一些合理输入来查看能否获得所需的输出。

8.2.2 容错性测试

容错性测试是检查软件对异常情况的处理。

8.2.3 边界性测试

因为边界一直是比较敏感的地方，而且是程序员最容易忽略的地方，所以，这种测试也往往最容易奏效。

8.2.4 性能与效率测试

性能与效率测试主要是测试软件在不同条件下的运行速度是否足够快以及软件在运行过程中对系统资源的利用率是否适中。性能与效率测试中很重要的一项是测试软件能正常运行的最大限度，因为很多软件系统会在这项测试中崩溃。

8.2.5 易用性测试

易用性测试主要是从用户的角度去考虑软件使用起来是否方便，没有具体的指标[19]。

8.3 测试情况

本文使用Monkey自动化测试工具，通过模拟用户的按键输入，触摸屏输入，手势输入等对系统进行性能测试，观察系统多长时间会出现异常。

本文使用黑盒测试技术，根据功能需求设计测试用例，采用了等价类划分法和边界值分析法，对系统功能进行测试。

9 结论与不足

本文首先分析了软件发展的背景以及国内外发展现状，指出了目前软件市场存在的一些问题。然后对本项目相关的一些技术做了简单的介绍。之后对系统进行了需求分析、总体设计以及详细设计，在编码后对系统进行了测试。系统能够稳定运行，且基本实现了主要的功能。

在开发过程中，我查阅了大量的相关技术资料，对Android系统有了进一步的了解，熟悉了对开发工具的使用。第一次开发完整的系统，我意识到了代码规范的重要性，代码规范不仅美观，还能提高开发效率。也遇到了很多问题，积攒了一些解决问题的方法。遇到问题时，我都是先检查可能产生问题的原因，并尝试修改，若未能解决问题，就上网查找解决方案，直到解决问题为止。若能解决问题，则做纪录，下次遇到同样的问题就可以查找之前的记录，这样不仅能更快地解决问题，还能加深印象。

目前好软分享APP能实现基本的分享功能，搜索功能，收藏功能，评论功能以及删除功能。但是存在一些不足，有待进行改进。首先界面不够美观，对界面的美化还有很大的空间；可以在用户之间添加相互关注的功能，用来实时接受关注人分享的动态；可以添加更人性化的下载功能，下载按钮直接获取官方网站的下载链接，可以免去用户打开官方网站的步骤；评论时可以添加对指定评论进行评论的功能，加强软件的互动性等等。这就需要提高系统的可扩展性，在必要的时候就可以增加系统的功能。

参考文献

1. 孙永波, 高雪. 移动APP营销研究评述与展望[J]. 管理现代化, 2016(1): 82-85.
2. 马德棚. 安卓应用市场可信性分析[D]. 2016.
3. 艾媒咨询. 2016Q3中国移动应用商店季度监测报告[R]. 香港: 艾媒咨询, 2016-12-13.
4. 盛晓峰. 从小屏幕到大屏幕交互转换的用户体验研究[D]. 上海交通大学, 2009.
5. 蒋健健. 移动APP的现状与发展[J]. 现代工业经济和信息化, 2017, 7(3):74-75.
6. 顾村来. APP应用程序开发模式探究[J]. 硅谷, 2014(5): 35-36.
7. 申之民. 基于用户体验的移动互联网应用商店研究[D]. 华中科技大学, 2012.
8. 马垚. 应用商店Android客户端的设计与实现[D]. 2017.
9. 邹继, 薛莲, 罗布超, John, 白俊丰. 更接地气-第三方Android应用商店不完全体验[J]. 移动信息, 2013, 35(8): 60-65.
10. 黄晟. 基于用户体验的APP设计研究[D]. 陕西科技大学, 2012.
11. 王跃, 路博. 我国移动互联网发展态势分析[J]. 互联网天地. 2016(09): 35-40.
12. Hammad Khalid. On Identifying User Complaints of iOS Apps[C]. ICSE 2013, San Francisco, CA, USA ACM Student Research Competition.
13. Pew Research Center. Future of Mobile News[J]. 2012.
14. Josh Clark. Designing Great iphone Apps[J]. Business week, 2011.
15. Gilbert, D C, and N Jackaria. The efficacy of sales promotionsin UK supermarkets: A consumer view[J]. International Journal of RetailD & istribution Management, 2002(6/7).
16. 张子悦. 大学生手机APP使用行为影响因素研究[D]. 黑龙江大学, 2017.
17. Butler, M. Android: changing the mobile landscape[J]. Pervasive Computing, IEEE, 2010, 10(1): 4-7.
18. 刘焱. 广东宏达建设工程项目管理信息系统的分析与设计[D]. 云南大学, 2014.
19. 闫毓. 手机测试系统的研究[D]. 北京邮电大学, 2009.

致 谢

这次毕业设计，是对我这四年学习的知识的一次实践。通过这次实践，我体验了一次开发的过程，感受到了技术人员的艰辛，也认识到了自身的不足。在这个过程中，我意识到了课本的知识的局限性，只有通过实践，才能发现更多的问题，学到更多的知识。在本文的结尾，由衷地感谢我的导师刘宇熹副教授对我的悉心指导。刘老师循循善诱，给我提出了建议，又指出了论文的不足。从论文的选题，到论文的撰写和修改，都倾注了刘宇熹导师的心血和汗水。在学习期间，刘老师的言传身教让我受益匪浅，他认真严谨的治学态度、豁达宽广的胸怀、平易近人的处事风格是我一生的楷模。值此提交论文之际，向刘宇熹导师表示衷心的感谢！