**班级**

**学号**

西安电子科技大学

**本科毕业设计论文**



题 目 多选择强隐私的个人博客系统的设计与实现

学 院 计算机学院

专 业 计算机科学与技术

学生姓名 张世欣

学生学号 03121311

导师姓名 黄伯虎老师

目 录

[第一章 绪 论](#_Toc200355145)

[1.1 研究背景及意义](#_Toc200355146)

[1.2 国内外研究现状](#_Toc200355147)

[1.3 本文内容及章节安排](#_Toc200355148)

[第二章 相关技术简介](#_Toc200355150)

[2.1 项目框架概述](#_Toc200355151)

[2.2 安卓MVP架构简介](#_Toc200355155)

[2.3 数据库greenDAO简介](#_Toc200355156)

[2.4 服务端LeanCloud简介](#_Toc200355156)

[2.5 安卓材料设计规范简介](#_Toc200355156)

[2.6 Markdown简介](#_Toc200355156)

[第三章 个人博客系统的设计](#_Toc200355157)

[3.1 系统需求分析](#_Toc200355158)

[3.1.1 需求数据分析](#_Toc200355159)

[3.1.2 需求分析结果](#_Toc200355160)

[3.2 系统概要设计](#_Toc200355158)

[3.2.1 系统流程设计](#_Toc200355159)

[3.2.2 系统功能设计](#_Toc200355160)

[3.3 系统详细设计](#_Toc200355158)

[3.3.1 服务端设计](#_Toc200355159)

[3.3.2 数据库设计](#_Toc200355160)

[3.3.3 客户端设计](#_Toc200355160)

[3.4 界面设计](#_Toc200355158)

[3.4.1 服务端界面设计](#_Toc200355159)

[3.4.2 客户端界面设计](#_Toc200355159)

[第四章 个人博客系统的实现](#_Toc200355157)

[4.1 文章模块的实现](#_Toc200355163)

[4.1.1 编写普通文章的实现](#_Toc200355164)

[4.1.2 MarkDown语法编写文章的实现](#_Toc200355165)

[4.2 待做事项模块的实现](#_Toc200355163)

[4.2.1 编写待做事项的实现](#_Toc200355164)

[4.2.2 待做事项锁屏提醒的实现](#_Toc200355165)

[4.2.3 待做事项定时提醒的实现](#_Toc200355165)

[4.3 推荐模块的实现](#_Toc200355165)

[4.3.1 推荐模块定时更新的实现](#_Toc200355164)

[4.3.2 推荐视频播放的实现](#_Toc200355165)

[4.4 云同步模块的实现](#_Toc200355165)

[4.4.1 本地文章、待做上传服务器的实现](#_Toc200355164)

[4.4.2 服务器数据同步到本地的实现](#_Toc200355165)

[4.5 导入导出模块的实现](#_Toc200355165)

[4.5.1 文章、待做导出为文件的实现](#_Toc200355164)

[4.5.2 本地指定文件导入系统的实现](#_Toc200355165)

[4.5 设置模块的实现](#_Toc200355165)

[4.5.1 文件保存相关设置的实现](#_Toc200355164)

[4.5.2 同步相关设置的实现](#_Toc200355165)

[4.5.1 安全相关设置的实现](#_Toc200355164)

[4.5.2 提醒相关设置的实现](#_Toc200355165)

[4.6 数据库的实现](#_Toc200355165)

[4.6.1 数据库、相关表格的创建](#_Toc200355164)

[4.6.2 数据库升级及迁移的实现](#_Toc200355165)

[4.7 服务端的实现](#_Toc200355165)

[4.7.1 服务端接口的实现](#_Toc200355164)

[4.7.2 服务端数据库的实现](#_Toc200355165)

[第五章 个人博客系统的测试](#_Toc200355157)

[5.1 测试用例的设计](#_Toc200355168)

[5.2 测试情况及结果分析](#_Toc200355168)

[5.3 问题及解决方案](#_Toc200355169)

[第六章 总结](#_Toc200355170)[与展望](#_Toc200355170)

[6.1 总 结](#_Toc200355171)

[6.2 展 望](#_Toc200355172)

[致 谢](#_Toc200355173)

[参 考 文 献](#_Toc200355174)

1. 绪论

本章主要介绍本课题《强隐私多选择的个人博客系统的设计与实现》的研究出发点、研究背景及意义、国内外相关研究的现状以及对本论文内容及章节安排的梗概。

1.1 研究背景及意义

随着科技的发展，知识变得空前容易获取，无论图书馆、书籍、报纸、网站或者博客，都可以让人轻松找到自己想要的知识。然而事有利弊，随着移动互联网的爆炸式发展，每天有太多的应用在产生，导致人们的注意力被大大分散。在信息爆炸的同时，年青一代(30岁以下)的历史意识、公民意识、阅读成绩、国际竞争力的提高却不甚对称[1]。

究其原因，是因为年轻人对社交网络服务的过度依赖。过去人们一天外界相连接的时间有限，所以会有剩余相当一部分时间用来阅读、自我思考或者与父母交谈。而如今世界互联，只要有手机、电脑，人们就一直与世界、与朋友相连接。从早上起床刷下微信、微博，地铁、饭桌、上课、上班、开会到中午休息跟朋友聊天，晚上睡前道个晚安，几乎二十四小时不停止。因此很少有时间集中精力的阅读，他们必须时不时地在社交软件上更新下状态，或者查看一下朋友们在做什么，否则就会坐立不安，无法专注。

以我国社交软件用户量最大的微信为例，近一半用户年龄低于26岁，九成用户低于36岁，平均每天打开微信10次以上的用户占55.2%，每天超过30次的重度用户占36%，而61.4%的用户每天必刷朋友圈，从来不看朋友圈的用户占比仅为1.3%[2]。而由于朋友圈的强关系性质，发表的内容会被其他人看到、评价。人们在其上更新状态后，朋友会给赞或者评论，因为都是认识的人，难免想要顾全面子，所以在发表、评论状态时会为了维持更好的形象有意识的修改自己的言论。

微信朋友圈只是一个缩影，微信、微博、以及其他使用率较高的强关系、社交网络服务平台，用户在彼此之间的互动上，于是，正确、适当与否，多了很多种评判标准，为了了解别人的状态，为了及时回应别人的评论，人们不得不停地刷着微博、朋友圈，以维持一种“与朋友沟通”的错觉。此外为了被他人认可、点赞，很多人发表的内容与实际生活相差甚大，努力的伪装只是为了给他人呈现一个“虚拟的自己”，这对使用者生活影响非常大，几乎可以说是活在他人的评价中。

据美国《福布斯》双周刊网站3月7日报道，凯度集团发布的《中国社交媒体影响报告》显示，与上一年相比，认为社交媒体让自己的生活变糟的人增加了一倍。对社交媒体越来越提不起兴趣的一个重要原因是，它剥夺了读书的时间。其他原因还包括，社交媒体侵犯了隐私，导致睡眠不足，并对健康产生不利影响。社交媒体已经渗透进中国人的生活，他们现在意识到自己在这上面花费了过多时间。此外，他们接收到的多是质量不高、抄来抄去的内容。

最近，心理学家贝丝安德森（Beth Anderson）和同事在评估Facebook的影响后认为，使用Facebook会让人很快上瘾，也同样会产生令人烦躁的负能量，让用户讨厌社交网络，有些讨厌的因由和当初让人注册Facebook的原因一样。我们想知道别人的情况，也希望别人了解我们——但就是在这种彼此了解的过程中，我们可能开始怨恨彼此的生活，也开始怨恨自己需要不断保持并不真实的网上形象。

针对这种情况，提出了本课题--《一种强隐私、多选择的个人博客系统的设计与实现》。本课题旨在为用户提供一个弱关系的、隐私安全更高的、相对封闭的个人博客系统，为用户提供便捷的记录想法、观点功能，使用户可以有一个独立的思考、记录空间，让用户能够更专注与个人提升，而非意义不大的浏览社交软件。此外软件并非完全封闭，会在筛选后为用户推荐一些高质量的资讯、视频。

1.2 国内外研究现状

上一节提到的社交网络对人的不良影响用一个专有名字来说就是“社交网络依赖症”。社交网络依赖症是指必须具备以下三个条件，缺一不可：第一，患者表现出由于沉迷于网络社交和网络游戏而导致出现焦虑、抑郁、恐惧、强迫、疑病、躯体化或神经衰弱等症状；第二，表现出来的症状已经导致社会功能受损或具有无法摆脱的精神痛苦；第三，前面两点的表现和行为至少要有一个月[3]。

实际上，近年来过度依赖社交网络，已经逐渐成为了一种世界性现象。

国外有市场研究公司调查发现，以“脸谱”（Facebook）和“推特”（Twitter）为代表的社交媒体，会像毒品一样令人上瘾。在这些上网成瘾的人群中，53%的人在无法上网时感到不安，40%的人会感到寂寞。

随着个人博客、微博等新的社交方式的出现，人们泡在社交网络上的时间也日益增多。根据2010年末的数据，人们在网络上花费的平均时间比10年前增加了82%。而根据杜克大学教授丹·艾瑞利研究得出的数据，他认为人们花在社交网络上的时间也是突飞猛进，是占据上网时间中最长的，目前为上网时间的22.7%，而且有不断向上的趋势。“也就是说，人们愿意花费在社交网络上的时间越来越多。”

美国马萨诸塞州阿桑普逊学院最近进行了一项关于社交网络的有趣实验。该学院的老师马丽娅·艾尔对学院里的70个本科新生进行了一次调查。她发现，Facebook上面好友的数量可以预测学生对大学的适应性。有超过200个好友的新生在自尊、个人适应和学业适应问卷的分数，低于那些没有那么多好友的新生。对于那些初来乍到的新生而言，在Facebook上的“好友”越多，现实中的朋友就越少。可见，社交网络对现实生活中的正常社交，并没有多大帮助，反而起反作用。

心理学家蒂莫西威尔逊（Timothy Wilson）研究发现大学生如果没了手机或电脑，只要呆在房间里几分钟就开始“发疯”。威尔逊说：“可能你会觉得我们可以用这时间精神自娱，可是我们做不到。我们已经忘了该怎么做。”只要我们有闲，互联网就是一个极诱人又可迅速填补空隙的解决办法。我们无聊，刷一下Facebook或者Twitter，然后变得更无聊。摒弃Facebook并不能改变的事实是，我们在越来越多的时候忘记了该如何正确而有意义地集中注意力。从这点来看，Facebook并不是问题。症状才是。

刘杰、王冰莹做了一次关于大学生网络社交成瘾现状、原因及对策的研究。他们选取哈尔滨工业大学、东北林业大学及黑龙江大学的大一至研二的学生，共发放问卷450份，回收有效问410份，有效率为91.1%。根据问卷分析发现，在“由于上网与人交流影响了我的工作效率”等依赖行为问题上，选择“比较符合”和“非常符合”的比例均在15%以上[4]，由此可见，社交网络使用对大学生的学习以及日常生活的负面影响较为严重。

1.3 本文内容及章节安排

为了给用户提供一个弱关系、强隐私、多选择的记录想法、总结提升的场所，本人拟搭建一款封闭性个人博客系统，本论文主要介绍该系统从设计到实现到测试过程中的内容。

本文共6章，具体各章介绍内容如下安排：

1. 绪论。概述本课题的出发点、解决方案及意义，列举相关内容国内外研究现状，同时简要说明本文的章节安排；
2. 相关技术简介。介绍本课题系统从设计到实现的过程中涉及到的一些技术点，比如项目框架、服务端、安卓MVP框架、安卓材料设计规范等；
3. 个人博客系统的设计。介绍本课题系统从需求分析、概要设计、详细设计到界面设计等；
4. 个人博客系统的实现。详细介绍了本课题系统的一些模块的实现过程及方法。
5. 个人博客系统的测试。介绍了针对本课题系统设计的测试用例、测试结果及解决方案。
6. 总结与展望。介绍了做完本课题系统的感受以及对本课题研究内容未来的展望。
7. 相关技术简介

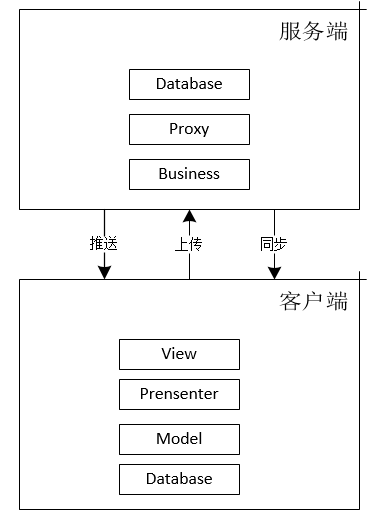
本章主要介绍本课题项目的框架结构、项目中Android端使用的架构 - MVP架构的简介、项目中Android端使用的数据库ORM（对象关系映射）框架 - GreenDao的简介、 项目中服务端采用的BAAS（后端即服务）平台 - LeanCloud的简介、项目中Android端采用的Google官方UI设计规范 - MaterialDesign简介、项目中Android端提供的编辑方式之一Markdown简介。

2.1 项目框架概述

本项目主要采用Client/Server架构，即客户端/服务端架构，通过合理分配客户端和服务端的功能，可以使系统耦合度更低、通讯成本降低。客户端负责收集用户行为、保存用户创建内容、展示推荐内容等，服务端负责持久化保存用户信息和用户内容，为项目实现信息持久化、跨平台同步提供可能。

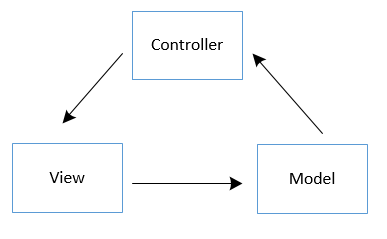
服务端主要由Business层（业务层）、Proxy层（代理层）和Database层（数据库）组成。业务层主要负责处理业务，比如用户注册、登录、文章上传备份、推荐内容更新等；代理层的作用是层隔离，隔断业务层与底层数据库层，这样任何一个层的修改都不会影响到其他层，同时还可以对业务层的操作做一些过滤，避免直接操作数据库导致问题。

客户端主要由View层、Presenter层、Model层、Database层组成，通过使用MVP架构来解耦界面与数据，具体细节下节介绍。



2.2 安卓MVP架构简介

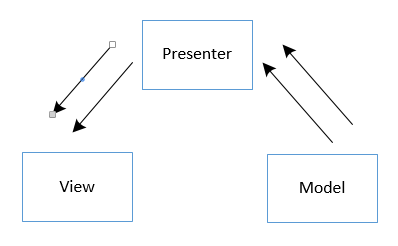
本项目中安卓客户端采用当前比较流行的MVP(Model-View-Presenter)框架。介绍MVP之前要先了解下之前被广泛使用的MVC(Model-View-Controller)架构，Model指的是业务模型，View是指用户操作的界面，Controller是指控制器。使用MVC架构的目的是为了将Model和View实现代码分离，从而使程序的耦合性更低。



然而在过去的使用过程中，太多的程序中Model被理解成提供View要展示的数据的实体类，而忽略了Model更重要的功能：处理业务逻辑。这样处理业务逻辑的代码就都放到了Controller(也就是Activity、Fragment)中，导致Activity、Fragment最后代码十分臃肿，维护困难，不利于重用。Model其实是MVC中非常重要的一部分，它提供展示的数据，同时负责实现业务逻辑。而Controller仅仅起到一个“桥梁”的作用就好了，它负责把View层接收到的用户行为转发给Model，然后再把Model的处理结果通知给View显示。所以Controller是用来解耦用户界面与业务逻辑的，准确的说就是让UI与逻辑隔离开。

为了解决MVC中Controller臃肿的问题，出现了MVP架构。在MVP中，Presenter完全把Model和View隔离开来，两者间没有直接交互。主要业务逻辑在Presenter中实现。MVP中的View代表的是将界面抽象出的接口。Presenter不操作具体的View，而是View的实现类，这样可以实现View有修改时Presenter几乎不收到影响，同时任何实现了该接口的界面都可以复用对应的Presenter。除此外，我们还可以针对View写对应的测试用例，模拟用户的行为，从而测试Presenter，即单元测试。

MVP真正意义上做到了View与业务逻辑的解耦。这样的好处是可以把整个的、复杂的项目拆分成多个模块，每个模块间相互独立，及时修改、替换其中一个模块，其他模块也不会受到影响，这样就实现了项目的可拓展、可维护性。每个模块实现特定的功能，内部聚合的同时也方便了单独测试。这样拆分后可以便于分工合作，原来需要一个人做六个月的工作量，采用MVP架构后可以交给六个人来做，一个月的时间就能做完。模块划分地够清晰后，每个人只需要专注于实现自己的那部分，完成后进行模块对接就好，这对于大项目，或者工期紧的项目是非常重要的。不过还要注意掌握好度，对于十分简单的功能，如果也分的稀碎，就有点过度设计之嫌了。解耦另一方面提高了程序的维护性，即容易区分边界，一旦出了问题，能够立刻定位是哪个模块出了问题。然后让相关责任人去修改。



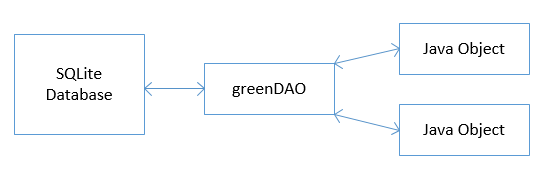
2.3 数据库greenDAO简介

本项目中数据库操作使用的是greenDAO。greenDAO是一个为Android开发者提供便捷的、有效率的SQLite操作数据接口的开源框架，使用它后开发者就不需要与底层数据事务打交道，可以节省很多开发时间。虽然SQLite是一个优秀的轻量级安卓内嵌关系型数据库，然而写SQL语句、解析查询结果还是很费时间。greenDAO通过提供Java对象与数据库中表的映射关系（通常称为关系对象模型）让你免于这些操作。这样你就可以使用面向对象接口来进行Java对象的增删改查。

greenDAO有以下几点特性：

1. 性能最优化(可能是Android最快的ORM框架)；
2. 数据的覆盖关系、加入接口使用方便，同时功能也很强大；
3. 耗费内存极少；

(4)本身体积小，让你编译地更快，同时也能避免“65k方法溢出”。



2.4 服务端LeanCloud简介

如今市场上有很多BaaS服务平台(后端即服务:Backend as a Service)，为移动应用开发者提供了很多功能的SDK，其中包括数据统计分析、推送、支付、定位、数据存储以及托管环境等服务。BaaS作为应用开发的省时省力助手，近一步促进了移动应用开发的成本，推动了应用市场的繁荣。

本项目使用的LeanCloud的存储功能就是一种为开发者提供便捷高效的数据存储服务的。它配合 Node.js 云端服务器环境，为开发者提供一站式的解决方案。

我只要把我的代码部署到LeanCloud的云端来执行，不用自己筹备服务器，同时也不必担心流量跟压力，可以轻松应对多种需求，LeanCloud提供了很成熟的命令行工具，部署、发布、回滚都十分方便。客户端调用时也很方便，只需要满足JSONObject这一基本格式就可轻松访问。

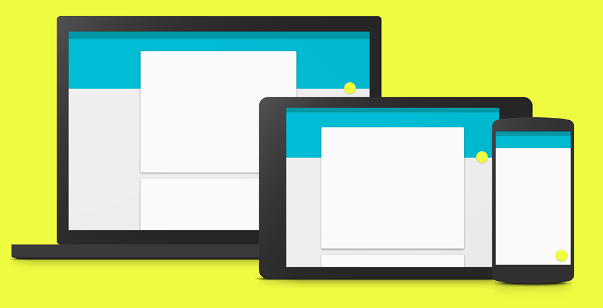
除此外LeanCloud还有以下特点：

1. 支持定时任务，beforeSave 等 Hook 调用；
2. 支持在线编辑，本地部署，方便调试；
3. 提供离线数据分析：使用 SQL 语句分析应用数据；
4. 文件存储采用国内全网加速 CDN，支持多种类型文件；
5. 全平台 SDK 支持：iOS、Android、Windows Phone、Unity 3D、JavaScript等。

2.5 安卓材料设计规范简介

本项目中Android端的设计规范是采用的Google官方UI设计规范 - Material design(又名材料设计)。Material design的核心思想就是把我们在物理世界得到的体验带进屏幕上去，保留物理世界中最原始纯净的形态、空间关系与过渡动画效果，同时需要去掉现实中的杂质和不确定性，然后配合虚拟世界的灵活特性，为用户创造最贴近真实感受的体验，达到简洁与美观的体验。

Material design的设计目标是让应用在不同平台、不同屏幕上都有一致的体验。因此它的规范十分严格细致，保证它在各个平台上使用体验高度一致。



本项目中用到的Material design的特性主要有：

1. 风格：限制主色调数量、图片规格、文字尺寸等；
2. 布局：凸出与省略、三维层级合理分布、适当运用阴影等；
3. 动画：模拟真实动画，考虑材质、加速度等，过渡动画的优美使用等；
4. 组件：合理使用材料风格的卡片、按钮、列表、网格、开关、提示和对话框等。

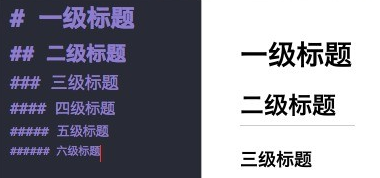
2.6 Markdown简介

Markdown之前一直用来在Web上将按语法规则输入的文本转换成HTML的标签。用简洁的语法取代复杂的排版，不像常用的word一样有大量的排版、字体等设置功能。使用Markdown后我们可以专心于内容本身，而不必为排版费太多的力气。目前Markdown被越来越多的写作者所使用，几乎已经成为文字编辑平台的通用功能。国内比较著名的如“简书”、“CSDN”等都提供了Markdown编辑方式。

为了便于用户编辑内容，本项目中客户端也提供了Markdown编辑，在Android平台上要提供Markdown语法，就需要构建一套文字排版规则，根据规则生成对应的HTML标签，然后再用WebView加载，目前项目中已实现支持Markdown的主流标签规则。

使用Markdown有以下优点：

1. 使用户专注于内容本身，而不是纠结于排版；
2. 可读、直观，学习成本低；
3. 便于后续导出为.md文件等。



1. 个人博客系统的设计

成熟的软件项目中，设计不必可少而且一定要先行，设计阶段要把项目要解决的问题抽象成不同层次、角度的逻辑问题。只有设计时考虑到足够完全的情况，将项目分解并模块化，使得问题清晰、解决方案明确，后续开发才不至于遇到白做、重做的尴尬。

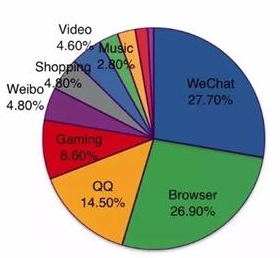
本章主要介绍本课题项目的设计过程，主要包括系统需求分析(明确问题，寻找方案)、系统概要设计(整体上设计系统的主要流程、功能以及软件架构)、系统详细设计(具体到服务端接口的设计、数据库表的设计、客户端模块的设计 )、界面设计(服务端、客户端用户界面的设计)。

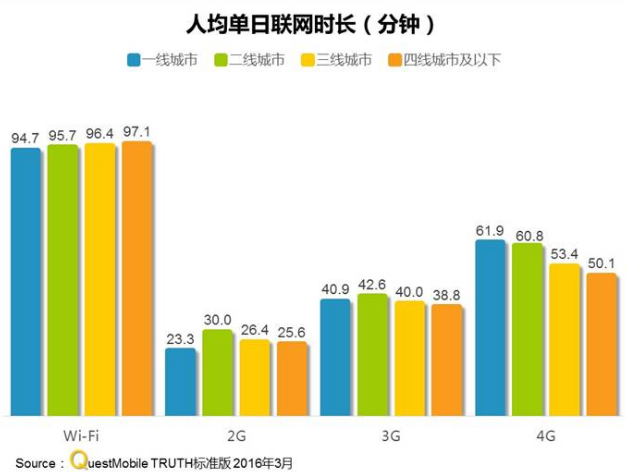
3.1 系统需求分析

本节主要对当前要解决问题的一些数据进行分析，在数据分析之后总结出具体的问题原因，继而寻找出合适的解决方案。主要包括需求数据分析、需求分析报告两部分。

3.1.1 需求数据分析

根据QuestMobile 2016春季APP数据报告结果，中国人均日上网时长约78.3分钟，其中使用社交网络服务(微信、QQ、微博等)的时间占据了47%。





此外据央视财经报道 中国人每天的休闲时间平均是2.55小时，只有发达经济体的一半。中国人1/3的休闲时间用在了互联网上，尤其是手机上。此外，中国人1/6的休闲时间在看电视。中国人1/10的休闲时间在进行纸质本阅读，仅有15.3分钟。

根据GlobalWebIndex最近的一份报告，网民使用社交网络的十大原因中，“与朋友保持联络”居首，占比过半为55%。其次是浏览最新时事、新闻和打发零散时间，占比均为41%。排名垫底的原因是“记录生活细节”，占比为27%。

在关于18岁以上成年人使用社交网络目的的调查中：

保持与目前好友联系这一目的方面，67%的人赞成这是主要动机，9%认为不存在这个动机；而重新与老朋友取得联系这一目的方面，64%的人赞成这是主要动机，13%的人认为不存在这个动机；寻找兴趣小组这一方面，14%的人赞成这是主要动机，而50%的人认为不存在这个动机；结识新朋友这一方面，9%的人赞成这是主要动机，而57%的人认为不存在这个动机；

另一个研究团队提出，使用Facebook也会增强嫉妒心：用户花越多时间浏览Facebook（而不是发原创信息或者和Facebook上的内容互动），他们的嫉妒感就越强。汉娜克拉斯诺夫（Hanna Krasnova）和同事认为，这是由社会心理学中“社会对比效应”造成的结果。由于人们多和自己大致相似的人交往，这种对比效应就愈发凸显：正因为比较的对象是思维方式类似的同龄人，所以得知他们的成就会造成更大的打击。专门研究社交媒体的使用和社交网络分享动机的心理学家塞缪尔戈斯林（Samuel Gosling）说：“很可能，正是那些一开始很吸引人的东西，到最后反而让人避之不及。”

3.1.2 需求分析结果

结合上一节的研究数据，我们不难发现，当今人们对社交网络的依赖程度达到一个比较高的程度，人们需要有一个工具来帮助他们独立思考、记录，同时思考的内容远离其他人的视线，这样就可以避免他人的评价对自己的影响。既然提供记录功能，就需要为这些数据提供备份功能，同时也要支持用户导出使用。除此外既然涉及个人信息，就需要有隐私保护，包括软件锁、内容加密、网络传输时加密等。

结合以上数据及分析，总结出本项目系统的主要功能如下：

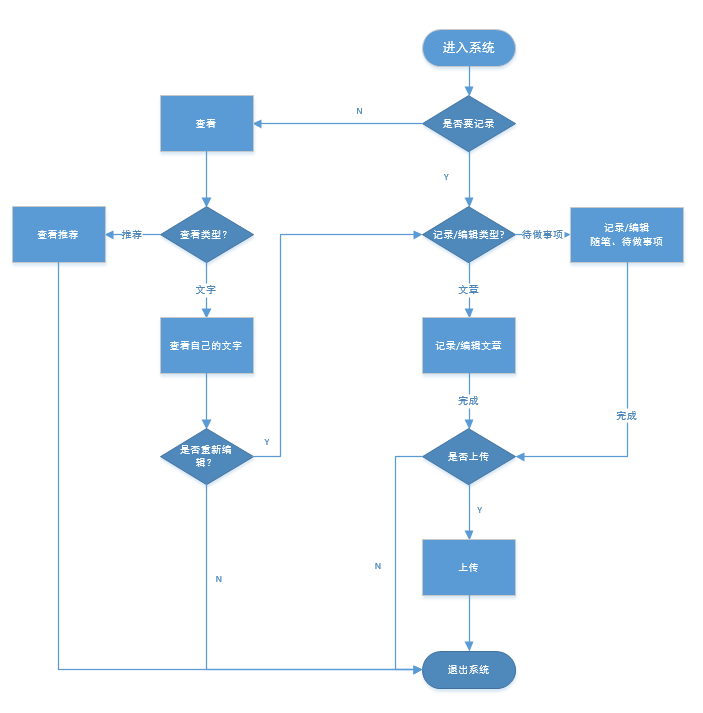
1. 支持记录随笔，一些转瞬即逝的灵感需要快速记录；
2. 支持记录文章，记录方式多样化，包括文字、Markdown、图片等；
3. 支持云端备份，保证在意外情况下用户信息可以找回；
4. 支持本地导出为文件，方便用户二次编辑或使用；
5. 支持隐私保护，包括软件手势密码锁、内容加密等；
6. 此外还可以考虑适当推荐优质内容，为用户推荐高质量的文字、视频等。

3.2 系统概要设计

上一节在进行需求分析后得出解决方案及主要软件功能。本节主要进行系统使用流程的设计、系统主要功能具体使用的设计、以及具体到系统软件架构的设计。

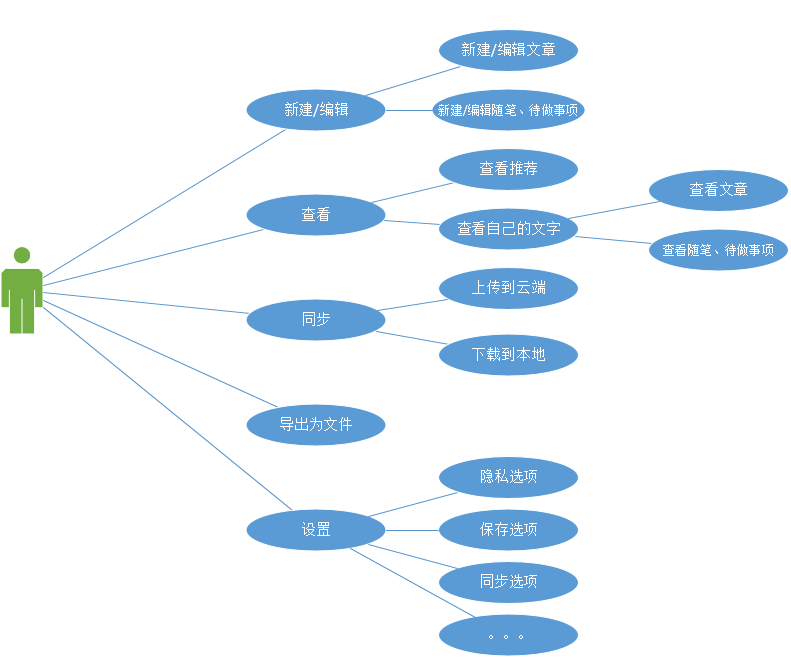
3.2.1 系统流程设计

用户进入系统后使用流程设计如下：



3.2.2 系统功能设计

结合3.1需求分析以及上一节的流程图，可以总结出系统的主要功能为新建/编辑、查看、同步、导入导出以及一些个性化设置等，见下面的用例图：



3.3 系统详细设计

上一节主要讲了系统的概要设计，包括系统主要功能及使用流程。这节将介绍系统的具体设计细节，包括服务端接口设计、数据库的设计以及客户端软件架构设计。

3.3.1 服务端设计

本项目中服务端主要负责以下功能：

1. 用户账户体系的管理，包括用户用户注册、登录、找回密码；
2. 用户文章、随笔(待做事项)的同步处理，包括上传、下载；
3. 用户推荐内容的更新等。

由于项目中使用了BAAS服务LeanCloud，免去了服务端的具体搭建、发布流程，我们只需在LeanCloud上创建应用并进行一些配置就可以进行相关表、接口的开发。

下面列出针对上述功能设计的主要几个接口，如无特殊标出均为get请求：

1. 用户注册/重置密码时发送验证码:

**URL**: Account/PhoneVerify

**参数**:

phoneNumber 手机号码

**返回值**：

status 发送状态，int，有如下几个取值范围:

0 成功发送

1 因为某些原因【短信 API 接口调用失败等等】无法发送信息

2 请求过于频繁，请等待一分钟后重试

1. 用户提交注册：

**URL:** Account/Register

**参数**:

username 用户名

password 密码

phoneNumber 手机号

verifyCode验证码

**返回值:**

status 注册状态，int，有如下几个取值范围：

0 注册成功

1 因为某些原因【比如数据库爆炸了之类】无法完成注册

2 verifyCode 无效

3 用户名已经存在

4 手机号已经存在

1. 用户登录:

**URL**: Account/Login

**参数**:

username 用户名

password 密码

**返回值:**

status 登录状态，int，有如下几个取值范围：

0 登录成功

1 因为某些原因【比如网络问题、服务器内部问题等】无法完成登录

2 用户名或密码不匹配

token 用户的令牌, 用于验证用户身份，string

1. 上传文章:

**URL**: Article/Upload

**参数**:

username 用户名

Article 文章实体类

**返回值:**

status 登录状态，int，有如下几个取值范围：

0 上传成功

1 因为某些原因【比如网络问题、服务器内部问题等】无法完成上传

1. 下载文章:

**URL**: Article/Download

**参数**:

username 用户名

**返回值:**

status 登录状态，int，有如下几个取值范围：

0 上传成功

1 因为某些原因【比如网络问题、服务器内部问题等】无法完成上传

List<Article> 文章实体类列表

1. 上传待做事项:

**URL**: Todo/Upload

**参数**:

username 用户名

Todo 文章实体类

**返回值:**

status 登录状态，int，有如下几个取值范围：

0 上传成功

1 因为某些原因【比如网络问题、服务器内部问题等】无法完成上传

1. 下载待做事项:

**URL**: Todo/Download

**参数**:

username 用户名

**返回值:**

status 登录状态，int，有如下几个取值范围：

0 上传成功

1 因为某些原因【比如网络问题、服务器内部问题等】无法完成上传

List<Todo> 待做事项实体类列表

3.3.2 数据库设计

针对前几章节总结的功能，本项目中需要创建的数据主要有用户实体表、文章实体表、待做事项实体表、推荐内容实体表等。

上述表的具体字段设计如下：

1. 用户表：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | ID | Username | Password | Token | ACL | CreateAt | ArticleCount | TodoCount |
| 数据  类型 | String | String | String | String | String | Date | Int | Int |
| 备注 | 用户ID | 用户名 | 密码 | 唯一标示 | 控制权限 | 注册时间 | 文章数量 | 待做事项数量 |

1. 文章实体表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | ID | Username | Type | Content | Title | Date | ImgPath | CreateAt | UpdateAt |
| 数据类型 | String | String | Int | String | String | Date | String | Date | Date |
| 备注 |  | 创建者用户名 | 类型：  0草稿  1文章 | 文章内容 | 文章标题 | 文章时间 | 文章配图路径 | 文章创建时间 | 文章更新时间 |

1. 待做事项/随笔实体表

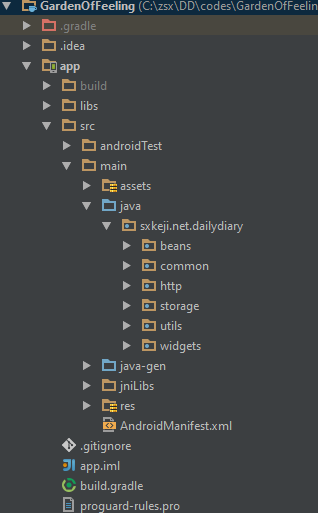
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | ID | Username | Color | Content | Title | Date | IsFinished | CreateAt | UpdateAt |
| 数据类型 | String | String | Int | String | String | Date | Boolean | Date | Date |
| 备注 |  | 创建者用户名 | 背景色 | 内容 | 标题 | 时间 | 是否完成 | 创建时间 | 更新时间 |

1. 推荐内容实体表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | ID | Type | Url | Title | MP4URL | Date | CreateAt | UpdateAt |
| 数据类型 | String | String | String | String | String | Date | Date | Date |
| 备注 |  | 推荐内容类型 | 内容链接 | 标题 | 视频地址 | 时间 | 创建时间 | 更新时间 |

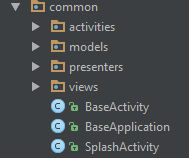
3.3.3 客户端设计

本项目客户端采用Android平台，主要软件结构如下图：



主要分为6个部分：实体类、业务相关、网络请求相关、内存管理相关、工具类以及自定义组件。

业务相关包下具体结构如下图：



前文已经提到，本项目中客户端架构主要采用MVP结构，在MVP中，Presenter完全把Model和View隔离开来，两者间没有直接交互。主要业务逻辑在Presenter中实现。MVP中的View代表的是将界面抽象出的接口。Presenter不操作具体的View，而是View的实现类。Model、View、Presenters分别对应上图的文件夹名称。

3.4 界面设计

本节主要介绍项目中的UI设计，包括服务端的界面和客户端的界面设计。

3.4.1 服务端界面设计

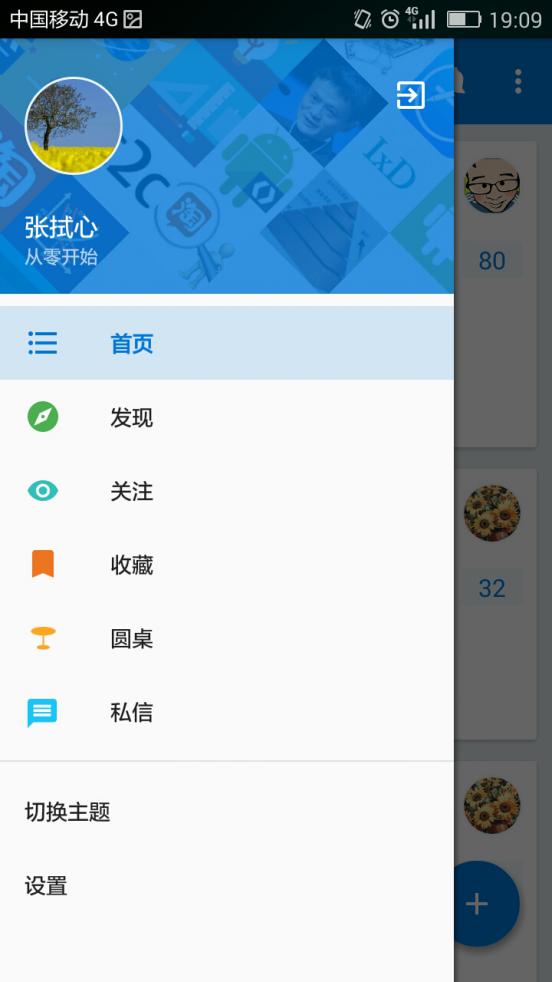
服务端主要是查看、编辑用户表、文章表、待做事项表等，所以初步原型图设计如下：



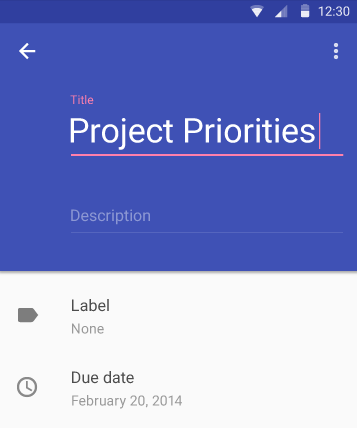
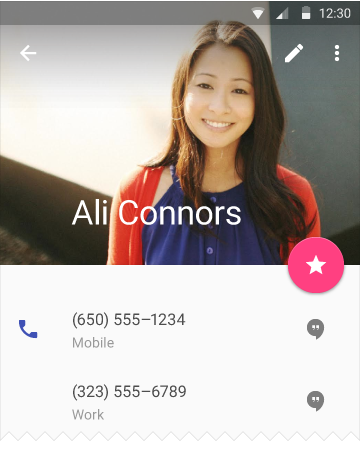
左端点击表名切换浏览数据表，右侧显示该表下有多少条数据及对应内容，下面可以对选中的表进行增删改查操作。

3.4.2 客户端界面设计

由于本项目客户端的设计要采用谷歌官方提供的材料设计规范，所以在设计上可以参考国内外的材料设计典型，比如知乎的Android客户端主页及侧滑栏：

颜色方面遵循材料设计的规范，在 UI 中的大块区域内使用醒目的颜色。 UI 中不同的元素适合主题中不同的色彩，如下面的样图为例：

1. 个人博客系统的实现

本章主要介绍个人博客系统的具体实现，包括：文章模块的实现、待做事项模块的实现、推荐内容模块的实现、同步模块的实现、导入导出模块的实现、设置模块的实现、数据库模块的实现以及服务端模块的实现。

4.1 文章模块的实现

文章模块作为本项目比较重要的模块，分为两种编辑方法：普通文字输入以及Markdown输入。本节分别介绍这两种实现方式。

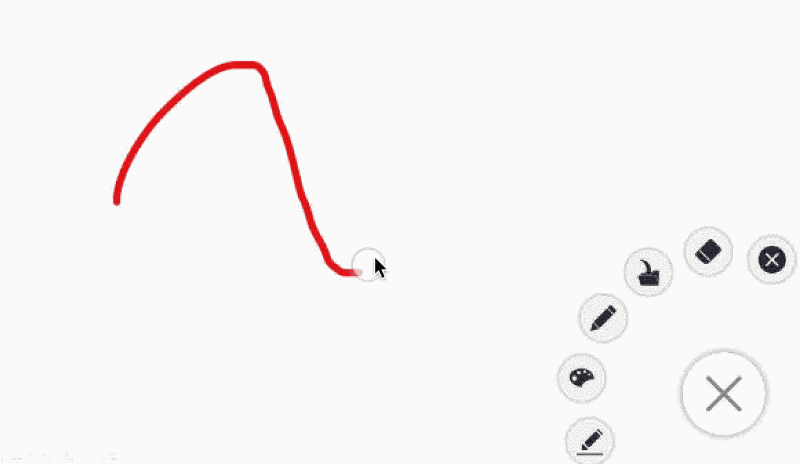
4.1.1 编写普通文章的实现

编写普通文章时主要提供的编写方式有：文字、图片、画图等。其中文字比较简单，比较复杂的是图片和画图的实现。

首先是图片的处理，Android中图片的加载、处理主要与Bitmap有关，而由于Bitmap的特殊性以及Android系统对每个应用占用内存的限制，使得图片的加载前处理变得十分重要，否则会导致内存溢出、应用奔溃。

在用户选择好图片后，我需要对Bitmap进行缩放处理：即根据要显示的视图大小对图片进行缩放处理，以达到不多占用内存。Android中简单有效的Bitmap高效加载方式是使用BitmapFactory.decodeFile方法，这个方法需要传入一个Options参数，我们要做得就是解析出图片尺寸信息然后进行一定比例的缩小，返回合适的Options，这样调用BitmapFactory.decodeFile方法就可以对图片进行缩小处理了。

其次是画图的实现，为了方便用户想要用手势绘制一些奇思妙想的需求，本项目提供了画图的功能。要实现的效果如下图：



要实现根据用户手指轨迹绘制指定的图形，就要用到Android中的Canvas，它的功能类似HTML5的canvas，都可以让我们在上面进行绘制操作。通过Canvas，绘制到Surface上的内容会被存储到与之相关的Bitmap中，该Bitmap最终会被呈现到窗口上。可以通过创建一个自定义View，监听手势动作，在手指滑动时绘制调用onDraw()方法绘制轨迹，最后保存为一个Bitmap，保存时要进行上述图片处理。

4.1.2 Markdown语法编写文章的实现

Markdown作为一种简洁、易用的编辑方式在PC端已经使用十分广泛，为了给喜爱使用Markdown编辑文字的用户提供支持，本项目中提供了Markdown语法支持。

Android原生是不支持Markdown语法识别的，但是由于此前web端已经有了Markdown语法解析，我们只需保存用户输入的符合Markdown语法的内容，然后使用js、css进行语法分析，最后使用WebView将分析后的内容以Markdown正确的格式显示出来，最后效果例如下图：

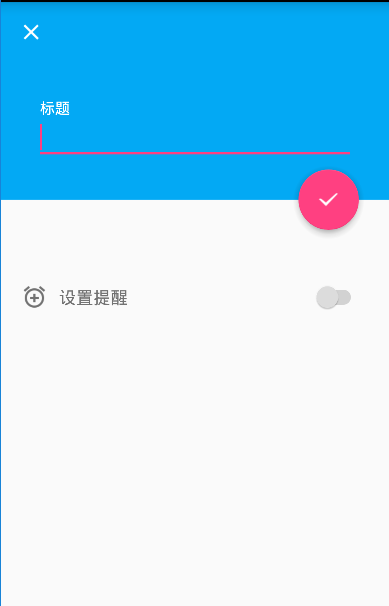


4.2 随笔/待做事项模块的实现

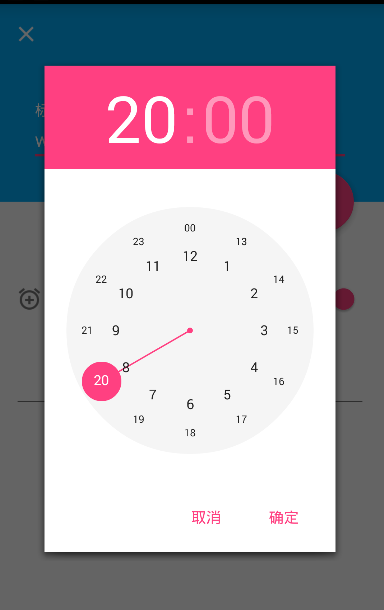
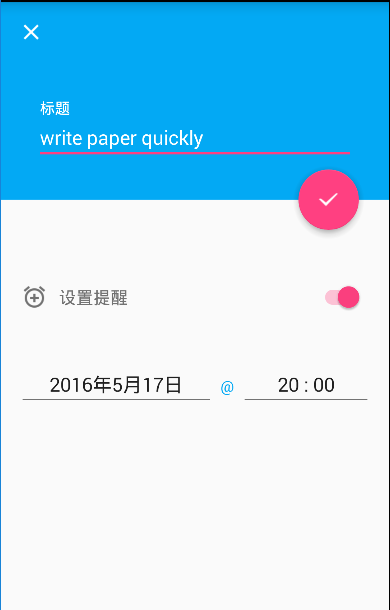
随笔/待做事项的目的是给用户提供一个快捷的（只记一段话）的记录方式，同时还支持设置提醒。

4.2.1 编写待做事项的实现

待做事项用一句话足矣，用户输入后可以选择是否设置提醒，界面如下：



如果用户需要设置提醒，选择提醒时间后保存即可：

4.2.2 待做事项锁屏提醒的实现

为了方便的提醒用户，本项目支持将提醒添加到锁屏页面，这样用户解锁时边可以看到提醒内容，更加印象深刻。

锁屏提醒功能的实现是基于监听用户开启屏幕时系统发出的广播，然后关闭系统的锁屏页面，开启自定义锁屏页面。目前已基本实现功能，遗留了两个问题：

1. 软件长时间不打开的情况下，监听服务所在线程可能被系统关闭，会出现锁屏提醒不出现的情况，虽然可以通过双守护线程实现，但是成本略高，待寻找更好解决方案；
2. 如果用户安装了第三方锁屏应用，会导致用户打开手机时第一时间看到的是该第三方应用界面，而不是本项目界面，不过影响不大，只要能让用户看到便达到了提醒的目的，后续寻找解决方案吧。

4.2.3 待做事项定时提醒的实现

用户设置提醒时间后，到指定时间软件需要发出通知提醒用户。这样还要考虑一个服务被关闭或未启动的问题，如果应用长时间未启动或者用户重启手机，可能我们的定时提醒服务线程会被关闭。

解决方案有两种：推送或者轮询。考虑到本项目的时间和人力成本，使用轮询更为简单，这样不需要额外的搭建推送服务器或者引入更多SDK。另外现在推送都是以长连接的方式实现，比较耗费用户手机的电量。所以我采用Android端的一种轮询方式的提醒：AlarmManager。

Android中的AlarmManager主要用来定时处理一个事件，本项目中要使用它的定期执行功能来实现轮询的功能。定期执行的任务可以采用TimerTask来调用。这里涉及一个Android系统锁的机制，即系统在检测到一段时间没有活跃以后，会关闭一些不必要的服务来减少资源和电量消耗，采用AlarmManager可以避免被关闭。

4.3 推荐模块的实现

本项目中有推荐模块，主要用来向用户定时推荐一些优质内容，其中的推荐内容是从一些应用中抓取的API。

4.3.1 推荐模块定时更新的实现

为了降低服务端压力、用户流量，本项目中拟定每隔一天更新推荐内容。这里就涉及到一个缓存和定时更新的问题。服务端保存的数据接口，隔一天刷新一次。而客户端在每次更新时将请求到的内容即JSON数据缓存到本地，同时记录缓存时间，下次打开时判断是否到更新时间，等到要更新的时候再重新请求，这样就避免了频繁请求带来的一系列问题。

4.3.2 推荐视频播放的实现

推荐内容中有视频播放功能，接口返回的是不同分辨率的mp4文件的URL，客户端要做的就是根据用户当前使用的网络模式（即2G、3G、4G、WIFI）来访问播放相应分辨率的视频。

Android系统提供了一个视频播放接口MediaPlayer。

参考文献

[1](美)马克·鲍尔莱恩,最愚蠢的一代,天津社会科学院出版社,2011年07月01日

[2] 2016年微信用户数据报告

[3] 百度百科 虚拟社交依赖症

[4] 刘杰、王冰莹,大学生社交网络成瘾现状、原因及其对策,2014