

摘要 医疗质量是一个关乎个人及社会的重大问题，提高医疗质量是每个以患者为中心的医院努力的目标。随着国家新医保政策的出台、医院之间优劣评比方式的改变，以及当代社会公众对医疗质量问题的高度关注，如何在降低医疗成本的同时保证以及提高医疗质量，成为目前每家医院都亟待解决的问题。

基于 Android 的医疗质量分析系统，结合如今用户量最多的移动系统——Android，使医院人员录入病例信息时更加方便，不需要 PC，直接使用手机录入，也直接能在手机上观察到对医生的医疗质量的评分和数据。利用 Android Studio 使用 Java 以及 XML 对系统的 Android 端进行开发，利用 Visual Studio 使用 C#(.NET)对系统开发 Web Service。基于 Android 的医疗质量分析系统集成录入病例，医生列表功能。

关键字 Android；医疗质量；评价

ABSTRACT Quality of medical care is a major issue relating to personal and social development, improve the quality of health care is the goal of every hospital patient-centered efforts. With the introduction of the new national health insurance policy, changing the way the merits of competitions between hospitals, as well as contemporary public attention to the quality of medical problems, how to reduce health care costs while maintaining and improving the quality of medical care, become every hospital Problems to be solved.

Android-based health care quality analysis system, combined with the most amount of users now mobile systems --Android, hospital staff make entry more convenient case information, no PC, direct entry using the phone, can be observed directly on the phone to the doctor's medical data and quality ratings. Use Android Studio for Android using Java and XML-end system developed using Visual Studio using C # (. NET) for system development Web Service. Based on the analysis of medical quality system Android integrated entry of cases, doctors list feature.

KEYWORD Android; Medical quality;Comment

目录

前 言	1
1 绪论	2
1.1 项目研究的背景及意义	2
1.2 医疗质量系统介绍	3
1.3 Android 系统及其版本研究概况	3
2 系统开发环境介绍	4
2.1 Android 操作系统介绍	4
2.2 Web Service 介绍	7
2.3 Microsoft SQL Server 介绍	9
3 系统需求分析	10
3.1 系统功能分析	10
3.2 系统用例分析	12
3.3 可行性研究	14
3.4 系统性能需求分析	15
4 系统设计	16
4.1 数据库设计	16
4.2 程序模块设计	17
4.3 API 接口设计	17
4.4 业务时序	18
4.5 用户界面设计	20
5 系统实现	21
5.1 文件结构	21
5.2 核心方法(函数)	22
5.3 关键代码	23
6 系统测试	27
6.1 测试说明	27
6.2 测试平台	27
6.3 测试用例	27

6.4	单元测试.....	28
6.5	系统测试.....	28
项目总结		29
参考文献		30
致 谢		31

前言

20 世纪以来，在人口越来越多，而且环境也越来越差，弭患病痛的人也自然越来越多，但是，有经验的医生，和能用的医疗器械，药物却并不能跟上患者的增长率，而且一些没有经验的新进实习医生和一些有顽固要求的患者，而这些都会导致原本为数不多不多的医疗资源浪费，所以在 20 世纪，美国的华德和潘顿开始研究医疗评价系统，通过这个评价系统可以评价医疗质量如何，而医疗评价系统的出现可以使医疗质量一步一步的往上推进，医疗质量越好，医疗资源就越能得到大部分的利用，不浪费任何医疗资源。可见一个医疗质量分析系统在一所医院里是显得多么的重要。

Android 从 2003 年的成立开发，到 2005 年被 Google 收购，再到 2007 年开源，如今 Android 已经取代了当年 Symbian 的市场，成为世界上占有率最高的移动操作系统，恰逢互联网时代的兴起，越来越多的 App 登陆 Android 与 iOS 平台，Android 也开始占据人们的空余时间。

Android 系统与医疗质量分析系统的结合，就如同两种化学药剂混合发生的反应一般强烈，现在的医院系统，都是依附在电脑完成，也就是说，当某位患者需要出院，需要开药，需要看病，都需要数个护士坐在登记台在电脑面前做登记，这是何等的浪费医疗资源啊！如果医疗质量分析系统不在依托在电脑系统上，可以在手机上运行，那么原本做在登记台的护士就不用一整天斋坐在登记台前做登记浪费医疗资源了，他们可以与医生，主任共同巡房，一来可以充分利用医疗资源，二来可以更加快速录入患者的出院情况以及治愈情况，对于医疗质量分析的时效性和便捷性的提升不是一丝半点。

总而言之，基于 Android 平台上开发医疗质量分析系统，对于整个医院的医疗质量是有提升帮助的。

1 绪论

1.1 项目研究的背景及意义

医疗质量是通过由一些测量确定的医疗资源提供的值的水平。作为与其他领域的质量，医疗质量是否足够良好的评估，以及它是否是适合于它的目的。高质量医疗的目标是提供给医生告知病人他们所需要的医疗资源；也就是说，保证良好的医疗质量，在医疗的时候尽可能的延长病人的寿命，等等。研究人员使用许多不同的医疗措施，试图确定医疗质量的评定标准，包括治疗的减少或医疗诊断辨病数量的减持，在风险因素的数量减少的情况下，人们可以调查人群的医疗情况而做出医疗质量或健康指标的标准。

由专业医疗人员提供的医疗服务的质量可以通过其成果，技术性能来判断。

“成果”是患者的健康变化，如治愈，复发或死亡率。在结果差异大可为个别医疗服务提供者进行测量，而较小的差异可以通过研究大基团，例如低和高水平的医生来测量。

“技术性能”是到卫生专业符合医疗指南确定的最佳实践的程度。推定医疗指导提供给最好的医疗，并给予一个最希望的好结果。技术性能是从质量角度判断，而不考虑实际的结果——例如，如果一个医生根据指引给医疗，而且病人的健康没有改善，随后通过这一措施，的“技术性能”的质量是仍然很高。

已经提出了改善的准确性和医疗质量的措施的研究发展一些地区的情况如下：

- 确定医疗质量指标
- 允许患者提供信息到他们的医疗记录系统
- 标准算法测量质量
- 报告结果的标准体系

1.2 医疗质量系统介绍

该医疗质量分析系统主要使用对象医院内部人员使用的客户端,该客户端主要针对通过患者信息计算出该患者主治医师治疗的医疗质量。结合 Android 操作系统的普遍性和广泛运用性,形成一套基于 Android 的医疗质量分析系统的有效解决方案

医疗质量分析系统,首先【客户端】可以获取医生的基本信息(包括医生名字,平均住院天数、7 日出院率、治愈率),可以根据该医生的医疗质量做排序,以列表形式显示在首页;其次【客户端】也可以添加患者的病历信息(包括该患者名字,患者的住院日期与出院日期,治愈率以及该患者的主治医师的 ID);【服务端】使用 Web Service 与 SQL Server 作数据交换,提供后台 API 接口给【客户端】使用

以上描述就是整个基于 Android 平台的医疗质量分析系统的基本介绍

1.3 Android 系统及其版本研究概况

Android 最初由安迪·鲁宾(Andy Rubin)等人开发制作,最初开发这个系统的目的是创建一个数码相机的先进操作系统;但是后来发现市场需求不够大,加上智能手机市场快速成长,于是 Android 被改造为一款面向智能手机的操作系统。于 2005 年 7 月 11 日被美国科技企业 Google 收购。2007 年 11 月,Google 与 84 家硬件制造商、软件开发商及电信营运商成立开放手持设备联盟来共同研发改良 Android,随后,Google 以 Apache 免费开放源代码许可证的授权方式,发布了 Android 的源代码,让生产商推出搭载 Android 的智能手机,Android 后来更逐渐拓展到平板电脑及其他领域上。

2010 年末数据显示,仅正式推出两年的 Android 操作系统在市场占有率上已经超越称霸逾十年的诺基亚 Symbian 系统,成为全球第一大智能手机操作系统。

在 2014 年 Google I/O 开发者大会上 Google 宣布过去 30 天里有 10 亿台活跃的安卓设备,相较于 2013 年 6 月则是 5.38 亿。

2 系统开发环境介绍

2.1 Android 操作系统介绍

2.1.1 历史

2003 年 10 月，有“Android 之父”之称的安迪·鲁宾（Andy Rubin）在美国加利福尼亚州帕洛阿尔托创建了 Android 科技公司（Android Inc.）。尽管 Android 科技公司的创始人和员工过去都具有各自的科技成就，但是 Android 科技公司的经营只显露出它在智能手机软件的方面。

2005 年 7 月 11 日，Google 收购了 Android 科技公司，Android 科技公司成为 Google 旗下的一部分。

2007 年 11 月 5 日，在 Google 的领导下，成立开放手持设备联盟（Open Handset Alliance）。开放手持设备联盟的创建目的是为了创建一个更加开放自由的移动电话环境。而在开放手持设备联盟创建的同日，联盟对外展示了他们的第一个产品：一部搭载了以 Linux 2.6 为核心基础的 Android 操作系统的智能手机。

2.1.2 Linux 内核

Android 操作系统的核心属于 Linux 内核的一个分支，具有典型的 Linux 调度和功能，除此之外，Google 为了能让 Linux 在移动设备上良好的运行，对其进行了修改和扩充。Android 去除了 Linux 中的本地 X Window System，也不支持标准的 GNU 库，这使得 Linux 平台上的应用程序移植到 Android 平台上变得困难。2008 年，Patrick Brady 于 Google I/O 演讲“Anatomy & Physiology of an Android”，并提出的 Android HAL 架构图。HAL 以*.so 档的形式存在，可以把 Android framework 与 Linux kernel 隔开，这种中介层的方式使得 Android 能在移动设备上获得更高的运行效率。Google 还在 Android 的核心中加入了自己开发制作的一个名为“wakelocks”的移动设备电源管理功能，该功能用于管理移动设备的电池性能

2.1.3 系统架构

Android 的系统架构如图 2-1 所示

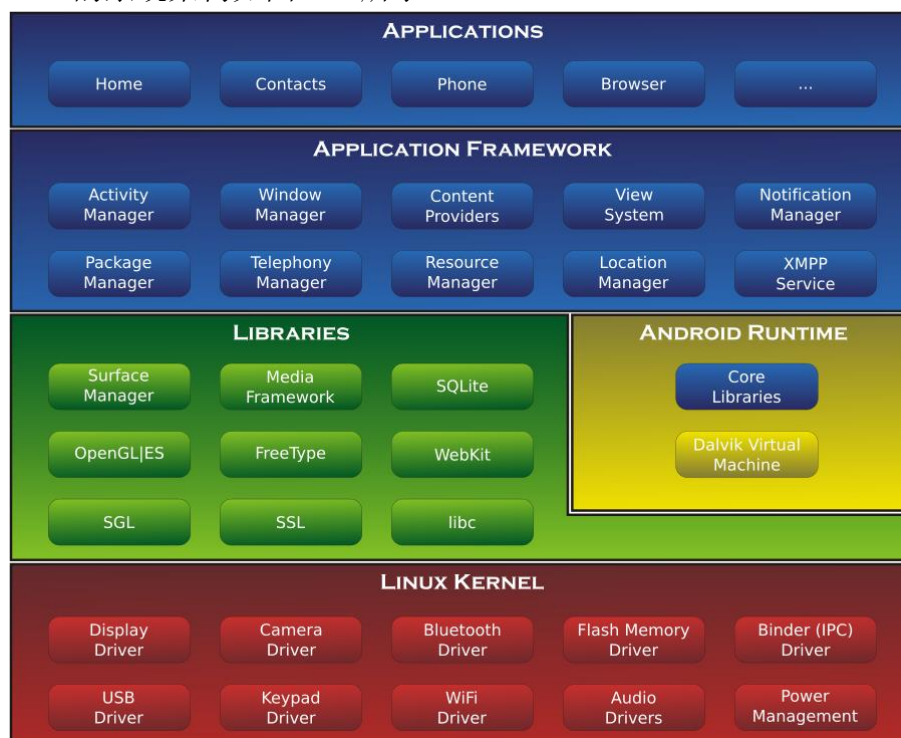


图 2-1 Android 系统内核架构

2.1.4 系统版本比例

截止 2016 年 3 月，不同的 Android 操作系统版本的用户比例如表 2-1 所示。

表 2-1 Android 各个版本用户比例

版本	代号	API	用户分布
6.0	Marshmallow 棉花糖	23	2.3%
5.1	Lollipop 棒棒糖	22	19.2%
5.0		21	16.9%
4.4	KitKat 奇巧巧克力	19	34.3%
4.3	Jelly Bean 果冻豆	18	3.2%
4.2.x		17	11.0%
4.1.x		16	8.1%
4.0.3-4.0.4	Ice Cream Sandwich 冰淇淋三明治	15	2.3%

2.1.5. Android 开发环境介绍

简介：

Android Studio 是一个为 Android 平台开发程序的集成开发环境。2013 年 5 月 16 日在 Google I/O 上发布，可供开发者免费使用。

2013 年 5 月发布早期预览版本，版本号为 0.1。2014 年 6 月发布 0.8 版本，至此进入 beta 阶段。第一个稳定版本 1.0 于 2014 年 12 月 8 日发布。

Android Studio 基于 JetBrains IntelliJ IDEA，为 Android 开发特殊定制，并在 Windows、OS X 和 Linux 平台上均可运行。

特点：

- 1) 可视化布局，WYSIWYG 编辑器 - 实时编码 - 实时程序界面预览；
- 2) 开发者控制台，优化提示，协助翻译，来源跟踪，使用率度量；
- 3) Beta 版本测试，并阶段性展示；
- 4) 基于 Gradle 的构建支持；
- 5) Android 特定代码重构和快速修复；
- 6) Lint 提示工具更好地对程序性能、可用性、版本兼容进行控制捕捉；
- 7) 基于模板的向导来生成常用的 Android 应用设计和组件；
- 8) 自带布局编辑器，可让开发者拖放 UI 组件，并预览 UI 显示效果。

系统要求：

Android Studio 系统要求如表 2-2 所示

表 2-2 AndroidStudio 系统要求

操作系统	Windows	OS X	Linux
	Microsoft Windows 10/8.1/7	OS X 10.8.5 或更高	Ubuntu
内存	最低 2 GB，推荐 4 GB 内存		
磁盘空间	500 MB 磁盘空间，至少 1 GB 用于 Android SDK，模拟器系统映像和缓存		
JDK 版本	Java Development Kit (JDK) 7 或更高版本		
分辨率	最低 1280×800 屏幕分辨率		

2.2 Web Service 介绍

Web 服务是一种服务导向架构的技术，通过标准的 Web 协议提供服务，目的是保证不同平台的应用服务可以互操作。

根据 W3C 的定义，Web 服务（Web service）应当是一个软件系统，用以支持网络间不同机器的互动操作。网络服务通常是许多应用程序接口（API）所组成的，它们透过网络，例如国际互联网（Internet）的远程服务器端，执行客户所提交服务的请求。

尽管 W3C 的定义涵盖诸多相异且无法介分的系统，不过通常我们指有关于主从式架构（Client-server）之间根据 SOAP 协议进行传递 XML 格式消息。无论定义还是实现，WEB 服务过程中会由服务器提供一个机器可读的描述（通常基于 WSDL）以辨识服务器所提供的 WEB 服务。另外，虽然 WSDL 不是 SOAP 服务端点的必要条件，但目前基于 Java 的主流 WEB 服务开发框架往往需要 WSDL 实现客户端的源代码生成。一些工业标准化组织，比如 WS-I，就在 WEB 服务定义中强制包含 SOAP 和 WSDL。

2.1.6. Web Service 开发环境介绍

简介:

Microsoft Visual Studio (简称 VS) 是微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集, 它包括了整个软件生命周期所需要的大部分工具, 如 UML 工具、代码管控工具、集成开发环境 (IDE) 等等。所写的目标代码适用于微软支持的所有平台, 包括 Microsoft Windows、Windows Phone、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight。而 Visual Studio .NET 是用于快速生成企业级 ASP.NET Web 应用程序和高性能桌面应用程序的工具。Visual Studio 包含基于组件的开发工具 (如 Visual C#、Visual J#、Visual Basic 和 Visual C++), 以及许多用于简化基于小组的解决方案的设计、开发和部署的其他技术。

特点:

Visual Studio 2010 将会带来 .NET Framework 4.0 并且支持开发面向 Windows 7 的应用程序。除了 Microsoft SQL Server, 它还将会支持 IBM DB2 和 Oracle 数据库 (详情参见 IBM.com 和 TeamFuze.net)。它将具有内置的 Microsoft Silverlight 开发支持, 包含一个交互设计器。Visual Studio 2010 将会提供一些工具来使并行计算更加简单: 除了 .NET Framework 的本地代码并行扩展以及并行模式库 (Parallel Patterns Library), Visual Studio 2010 还包含了用于调试并行程序的工具。这些新工具使并行任务以及它们的运行时堆栈可视化。这些工具可以用来可视化线程等待时间以及线程在多核心之间的移动。

2.3 Microsoft SQL Server 介绍

Microsoft SQL Server 是由美国微软公司所推出的关系数据库解决方案，最新的版本是 SQL Server 2014，已经在 2014 年 4 月 1 日发布。数据库的内置语言原本是采用美国标准局（ANSI）和国际标准组织（ISO）所定义的 SQL 语言，但是微软公司对它进行了部分扩充而成为作业用 SQL（Transact-SQL）。几个初始版本适用于中小企业的数据库管理，但是近年来它的应用范围有所扩展，已经触及到大型、跨国企业的数据库管理。

SQL Server 一开始并不是微软自己研发的产品，而是当时为了要和 IBM 竞争时，与 Sybase 合作所产生的，其最早的发展者是 Sybase，同时微软也和 Sybase 合作过 SQL Server 4.2 版本的研发，微软亦将 SQL Server 4.2 移植到 Windows NT（当时为 3.1 版），在与 Sybase 终止合作关系后，自力开发出 SQL Server 6.0 版，往后的 SQL Server 即均由微软自行研发。

3 系统需求分析

3.1 系统功能分析

3.1.1 管理人员登录

该系统必须只由 1-2 个人维护管理，并不能整个医院任何人都能观看以及录入任何信息，首先为了医生自己本身医疗质量分析信息不被篡改，而且避免多人维护管理会导致数据重复/混乱，其次若让医生知道自己或其他人的医疗质量，难免激起攀比心而忘记救人救命的初衷。

病人更不能登录该系统查看医生的医疗质量分析成绩，否则会导致更严重的医患关系，而且难免会导致病人会对自己目前的主治医生不满而想办法去找另一医生治疗，这对整个医疗系统会造成崩溃

3.1.2 医疗质量分析

后台可以通过医疗质量分析算法，根据病人病历表的数据，方便快速的完成对医生的医疗质量分析。医生质量分析系统管理人员登陆后，能立即看到每个医生的医疗信息及其医疗质量分析结果分数，以便对医生的绩效做评定。

系统会对每个医生的病人的病历情况分别由平均住院天数，7 天内出院率以及治愈率和相对应的病情严重度作出一个科学，详尽的分析并统计出一个综合医疗质量评分，该评分完全由算法统计，不存在任何人为干涉，综合医疗质量评分分为三部分给分：1、病人住院天数，住院天数越少分数越高；2、根据 7 天内出院率给分，百分比越高分数越高；3、根据治愈率及其病情严重度给分，治愈率百分比越高分数越高，而且当治愈的病人病情严重度越高，分数将会越高。三部分加起来既是综合评分。

3.1.3 病历录入

医生质量分析系统管理人员必须严格根据病人的住院信息进行录入，包括病人名字，病人住院日期，病人出院日期，该病人的主治医生 ID，以及治愈率及其病情严重程度。

治愈率根据病人出院后 7 天内是否由于同一病因复诊而确定：由于人体抗体产生后病毒必定被杀死且抗体在人体的存活时间有 7 天时间。即若某病人在医生治疗下出院后 7 天内再次发病，则该病人未治愈；若病人 7 天内并没有由于同一病因发病，则该病人治愈。所以医疗质量分析系统管理人员必须在病人出院后 7 天确定病人已经治愈后再对病人病历信息进行录入。

而病情严重程度则分为 10 个等级，等级越高医疗治愈难度越高，感冒咳嗽等系属 1-3 级，而癌症，急性濒临死亡病情则系属高等级 9-10 级严重度。所以医疗质量分析系统管理人员必须确定该病人的入院时的病情严重程度录入该严重度。

当病人病历信息被录入后，将记录在数据库 Patient 表中，并会立即对医生的医疗质量分析影响。

3.2 系统用例分析

3.2.1 管理人员登陆

管理人员登录作为该系统的入口模块功能，严谨性必须做好，不能被随意登录进去，而且对于空用户名/空密码，用户名/密码错误必须做出一个对用户的提示。

1、登录并进入医疗质量分析系统

管理人员登陆用例图如图 3-1 所示

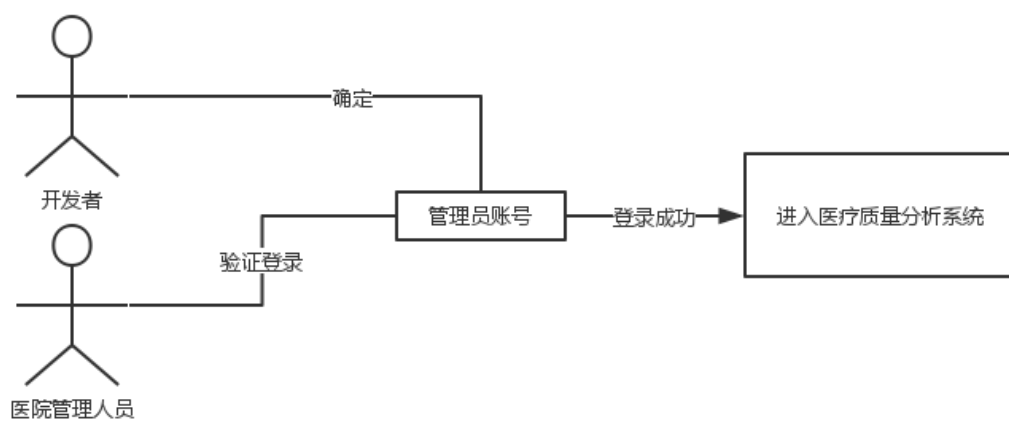


图 3-1 登录用例

3.2.2 医疗质量分析

医疗质量分析模块作为该系统的核心功能模块，是根据医生治疗病人的病历信息对整个医院的医生进行一个详尽，科学的分析并给出综合分数，管理人员对该分数不能作任何直接人为修改，在该医疗质量分析模块，管理员只能看到医生治疗的病人的病历信息以及该医生的医疗质量分析结果。

- 1、读取医生信息
- 2、读取该医生所治疗所有病人的病历信息
- 3、根据病历信息分析该医生的医疗质量

医疗质量分析用例如图 3-2 所示

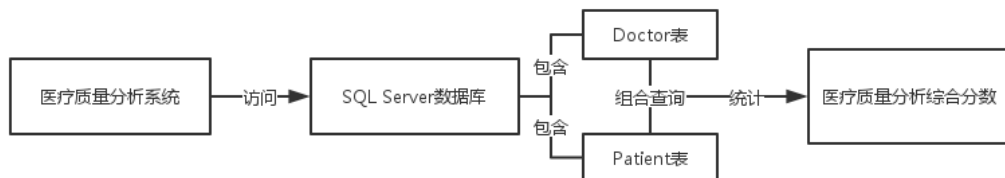


图 3-2 医疗质量分析用例

3.2.3 病历录入

病历录入模块属于该系统唯一一个输入数据模块，管理人员必须对该模块的录入数据十分严谨以及认真，该模块录入信息将直接影响决定医生的医疗质量分析综合分数。

录入病人病历信息

病历录入用例如图 3-3 所示

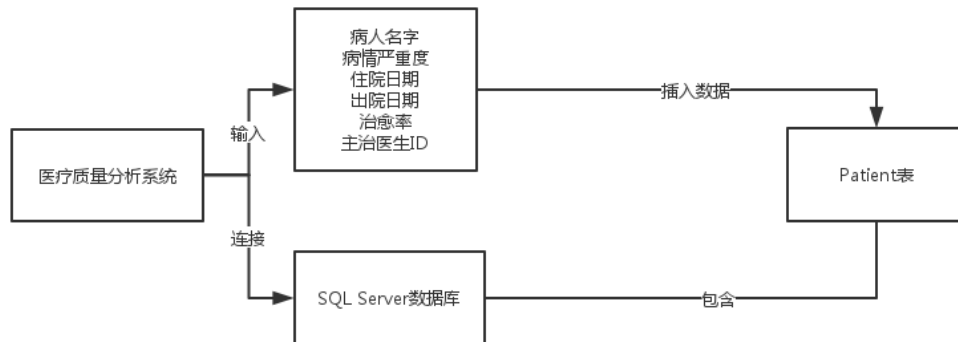


图 3-3 病历录入用例

3.3 可行性研究

可行性研究，简称“可研”，是工程学中常见的术语和前期准备过程，简单而言即是计划及考虑某项目建议的可行与否。大型项目开始之前，当然必须要有一个理性的及详实的可行性研究，以确定工程计划是否可行，一般包括技术评价、安全评价、经济评价价等多方面。

3.3.1 技术上的可行性

本医疗质量分析系统是基于 Android 操作系统开发，Android 的开发语言为 Java，Google 在 Android 开发上也向开发者提供了一整套的 API 接口使用，而且也有 Google 开发的 Android 开发编译器——Android Studio，而且 GitHub 上也有大量提供给 Android 用的第三方开源库以便开发使用，省去自己造轮子的时间与精力

3.3.2 数据上的可行性

本医疗质量分析系统的数据保存采用的是来自 Microsoft 开发的 SQL Server 数据库，SQL Server 数据库已经被 Microsoft 耕耘多年，从 1999 年到现在 2016 年，Microsoft 的 SQL Server 数据库的安全性已无需赘述，而且 SQL Server 面向非商用产品也是一律完全免费使用

3.3.3 运行可行性

本医疗质量分析系统是基于 Android 操作系统开发，而 2013 年 Android 的市场份额占据全球移动设备操作系统的 61.9%，可以保证有一半的移动设备都可以安装使用该系统

3.3.4 经济可行性

本医疗质量分析系统使用了 Android Studio、Microsoft Visual Studio 开发，数据库试用 Microsoft SQL Server，这三款开发工具面向非商用产品都是可以免费授权使用，并不会产生任何成本。

3.4 系统性能需求分析

医疗质量分析系统的首要要求是可用、稳定和安全，并能确实为医院带来便利和价值。基于此目标，系统的后台采用了WebService+IIS+SQL Server，客户端使用了当前移动系统使用最多的Android操作系统。

后台采用WebService+IIS+SQL Server的理由是IIS具有方便的特性，在Windows桌面版系统自带，在Windows服务器版本更是直接启用，不许用户开启，免去复杂繁琐的部署工作，通过简单的配置，就可让系统运行起来。数据存储采用SQL Server是因为微软对数据库管理的硬实力的确就在哪里，而且相比较Oracle更加便宜。

作为移动平台的医疗质量分析系统，界面设计要简介大方并能让使用者在有限的屏幕中能够快速的看到自己想要看到的信息。并且运行速度要足够快，因为面向的用户是一群在医院工作的工作者，不能让卡顿而影响了使用者的工作时效。作为移动端的软件，要节省使用者的资源，不能浪费有限的资源，否则会让使用者产生消极情绪极大的提升不再继续使用的情绪。

因为这是基于移动端的医疗质量分析系统，因此更应该注重安全，不应该轻易被入侵泄露医院病人，医生隐私信息造成损失。要保证医院的隐私安全不被轻易破解。

4 系统设计

4.1 数据库设计

本医疗质量分析系统有两部分数据需要存储，其中一个医生的信息，医生信息需要存储医生的 ID (doc_id) 与医生的名字 (doc_name)；另一个是病例信息，病例信息则需要存储病人名字 (patient_name)、病人的住院日期 (in_date)、病人的出院日期(out_date)、该病人的主治医师的 ID(doc_id)、该病人是否治愈的情况(cure)以及该病人的病情严重度(disease_lv)

本系统使用 Microsoft SQL Server 数据库存取数据，建立数据库 Medical，其中建立两个表 Doctor 如表 4-1 所示存储医生信息，Patient 如表 4-2 所示存储与病例信息。

表 4-1 Doctor 表设计

列名	数据类型	备注
doc_id	int	主键, 每个医生独有的 ID
doc_name	char(10)	医生的名字

表 4-2 Patient 表设计

列名	数据类型	备注
patient_name	char(10)	病例中病人名字
in_date	date	病人住院日期
out_date	date	病人出院日期
doc_id	int	表 Doctor 的 doc_id, 用于联结查询
cure	bit	治愈与否, 使用布尔类型, 0 (否) 或 1 (是)
disease_lv	int	病情严重度

4.2 程序模块设计

该医疗质量分析系统分为两个大模块：Web Service 端链接数据库，对数据库进行读取，写入操作；Android 端调用 Web Service 的获取数据函数，获取返回的数组对其进行解析后输出在页面上。两个模块关系如图 4-1 所示

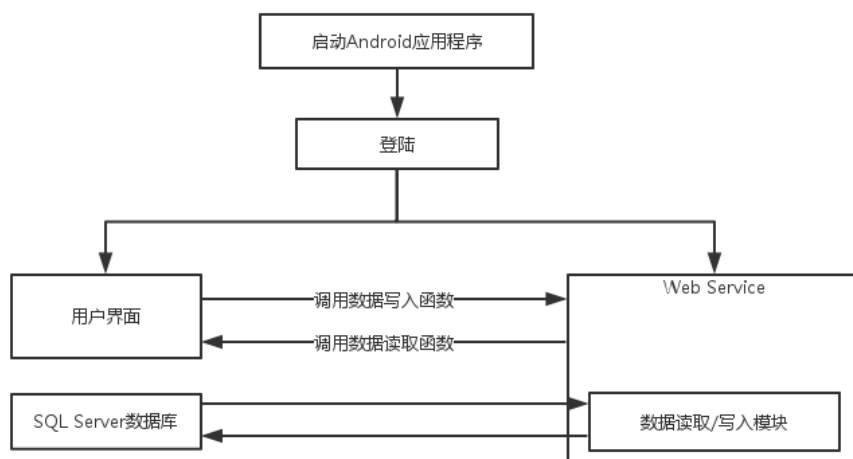


图 4-1 模块设计

4.3 API 接口设计

4.3.1. selectDoctor

selectDoctor 接口如表 4-3 所示

表 4-3 selectDoctor 接口说明

功能说明	医疗质量分析结果查询		
访问函数	selectDoctor		
输出参数			
参数名	类型	必须	说明
doctorName	String	是	医生名字
dateAvg	Int	是	医生治疗病人的平均住院日
day7Persent	String	是	医生治疗病人 7 天内出院概率
curePersent	String	是	医生治疗病人治愈概率
diseaseLvAvg	Int	是	医生平均治疗病人病情严重度
score	Int	是	医生的医疗质量综合评分

4.3.2. insertPatient

insertPatient 接口如表 4-4 所示

表 4-4 insertPatient 接口说明

功能说明	添加病历信息		
访问函数	insertPatient		
输入参数			
参数名	类型	必须	说明
sPatientName	String	是	病人名字
sInDate	String	是	病人住院日期
sOutDate	String	是	病人出院日期
sDoctorID	String	是	病人主治医生的 ID
sCure	String	是	病人治愈与否；是：1，否：0
sDiseaseLv	String	是	病人的病情严重程度
输出参数			
参数名	类型	必须	说明
flag	String	是	成功 true，失败 false

4.4 业务时序

4.4.1 用户登录时序

用户登录时序图如图 4-2 所示

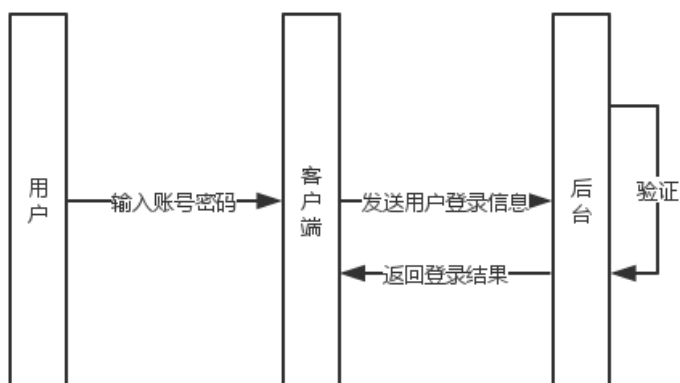


图 4-2 登录时序图

4.4.2 获取医疗分析结果时序

获取医疗分析结果时序如图所示

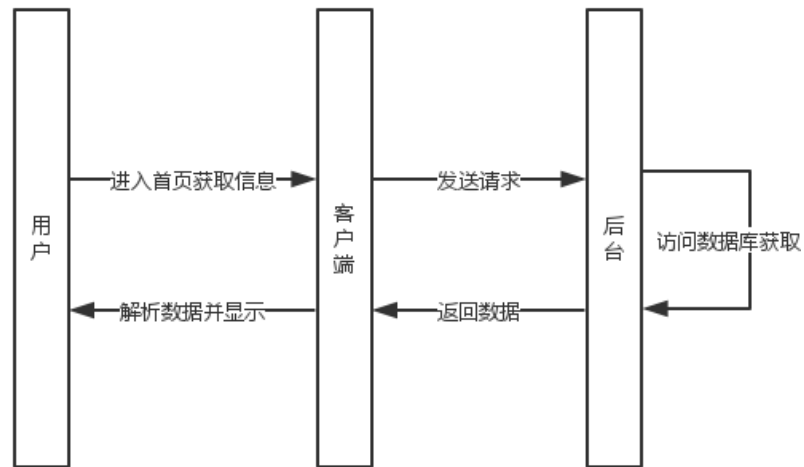


图 4-3 获取医疗质量分析结果时序

4.4.3 录入病历信息时序

录入病历信息时序如图 4-4 所示

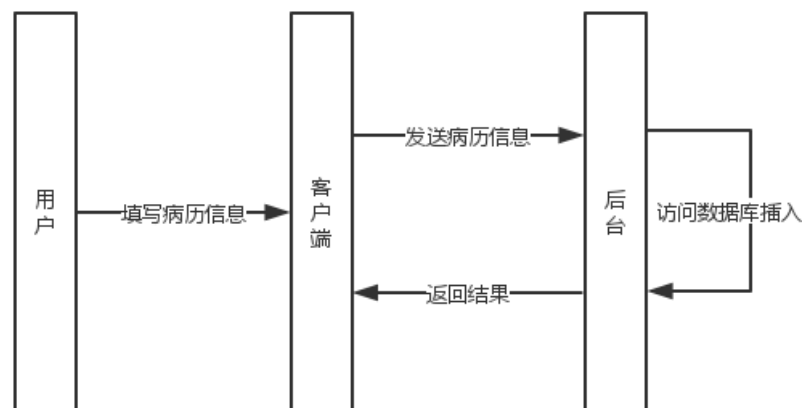


图 4-4 录入病历时序图

4.5 用户界面设计

根据需求文档，该医疗质量分析系统拥有两个页面：首页与添加病例页。这里需要进一步分析每个用户界面中应该包括哪些显示内容。

登陆页需要管理员输入自己的账号与密码登录；截图如图 4-2 所示

首页显示医生的列表，列表内容包括医生名字、平均住院数、7 日出院率、治愈率、平均治愈病情严重度以及分析后的医生医疗质量综合评分；截图如图 4-3 所示

添加病例页需要用户填写病人名字、住院日期、出院日期、治愈与否、病情严重度以及主治医师的 ID；截图如图 4-4 所示



图 4-7 登录页

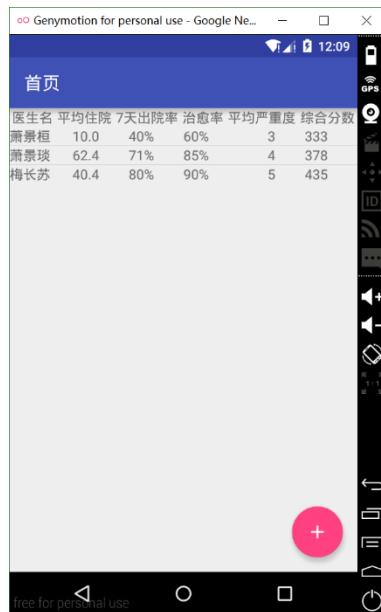


图 4-6 首页



图 4-5 添加病历页

5 系统实现

5.1 文件结构

5.1.1 Web Service

Web Service 端项目命名为 Android_SQL_WebService, 项目文件结构如图 5-1 所示:

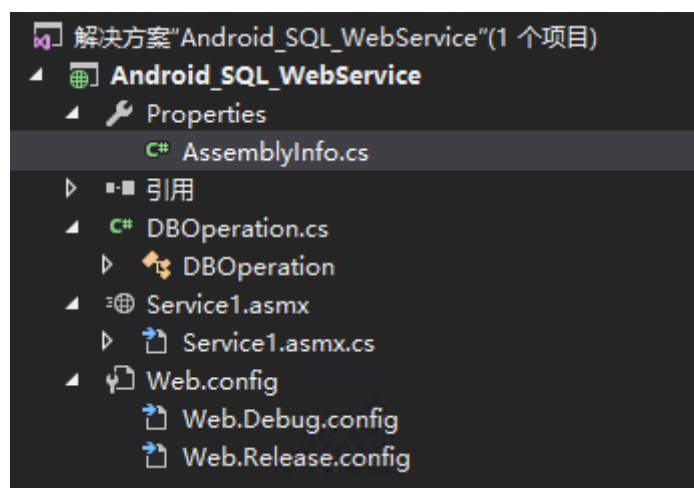


图 5-1 Web Service 文件目录

5.1.2 Android

Android 端项目命名为 MedicalProject, 项目的文件结构如图 5-2 所示:

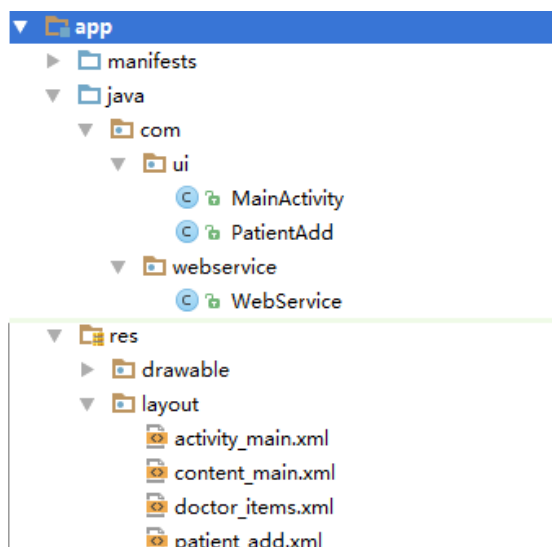


图 5-2 Android 端文件目录

5.2 核心方法(函数)

5.2.1 Web Service

Web Service 核心函数如表 5-1 所示

表 5-1 Webservice 主要函数

函数名(类型)	传入参数(类型)	返回参数
selectDoctor(list)	NULL	ArrayList
insertPatient(bool)	PatientName(string) InDate(string) OutDate(string) DoctorID(string) Cure(string)	true/false

5.2.2 Android

Android 核心方法如表 5-2 所示

表 5-2 Android 核心方法

方法名(类型)	传入参数(类型)	返回参数
callWebService(void)	url(String) methodName(String) properties(HashMap)	NULL
setPatient(void)	sPatientName(String) sInDate(String) sOutDate(String) sDoctorI(String) sCure(String)	NULL
DoctorSoapObject (List<Map<String, Object>>)	result(SoapObject)	DoctorItems

5.3 关键代码

5.3.1 Web Service

ConServerStr

```
private String ConServerStr = @"Data Source=MingkinHo\SQLEXPRESS ; Initial
Catalog=Medical ; Integrated Security=True";
```

该代码重点在于这是用于连接数据库的语句，通过它即刻 Web Service 即可链接到数据库

insertPatient

```
public bool insertPatient(string PatientName, string InDate, string
OutDate, string DoctorID, string Cure)
{
    try
    {
        string sql = "INSERT INTO Medical.dbo.Patient(pat_name ,
in_date , out_date , doc_id , cure) VALUES(' " + PatientName + "', ' " +
InDate + "', ' " + OutDate + "', ' " + DoctorID + "', ' " + Cure + "')";
        SqlCommand cmd = new SqlCommand(sql, sqlCon);
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}
```

该函数需要传入5个参数分别是 PatientName , InDate , OutDate , DoctorID , Cure,这5个参数通过 `string sql = "INSERT INTO Medical.dbo.Patient(pat_name , in_date , out_date , doc_id , cure) VALUES(' " + PatientName + "', ' " + InDate + "', ' " + OutDate + "', ' " + DoctorID + "', ' " + Cure + "')"`;将数据插入进数据库

SelectDoctor

```
public List<string> selectDoctor()
{
    List<string> list = new List<string>();

    try
    {
        string sql = "";
        SqlCommand cmd = new SqlCommand(sql, sqlCon);
        SqlDataReader reader = cmd.ExecuteReader();

        while (reader.Read())
        {
            //将结果集信息添加到返回向量中
            list.Add(reader[0].ToString());
            list.Add(reader[1].ToString());
            list.Add(reader[2].ToString());
            list.Add(reader[3].ToString());
            list.Add(reader[4].ToString());
        }

        reader.Close();
        cmd.Dispose();
    }
    catch (Exception)
    {
    }

    return list;
}
```

该函数重点在于通过 SQL 语句获取了数据后,通过一个循环将数据变成一个数组输出,由于该函数的 SQL 语句获取的数据表有 5 列所以循环体中有 5 个 list.Add。

5.3.2 Android

WebService.java

```
public class Webservice {

    [...]

    public static void callWebService(String url, final String methodName,
    HashMap<String, String> properties, final WebserviceCallBack
    webServiceCallBack) {

        [...]
        soapEnvelope.setOutputSoapObject(soapObject);
        soapEnvelope.dotNet = true;
        httpTransportSE.debug = true;

        final Handler mHandler = new Handler() {
            @Override
            public void handleMessage(Message msg) {
                super.handleMessage(msg);
                webServiceCallBack.callBack((SoapObject) msg.obj);
            }
        };

        executorService.submit(new Runnable() {
            @Override
            public void run() {
                [...]
                finally {
                    mHandler.sendMessage(mHandler.obtainMessage(0,
resultSoapObject));
                }
            }
        });
    }

    public interface WebserviceCallBack {
        void callBack(SoapObject result);
    }
}
```

该工具类,用于新建一个线程池(内含一个子线程),使用子线程去向 Web Service 做 HTTP 访问请求而不会报错,通过 Handler 将数据回调到使用此工作类的 callback 方法的地方

setPatient

```

public void setPatient(String sPatientName, String sInDate, String
sOutDate, String sDoctorID, String sCure) {

    String nameSpace = "http://tempuri.org/";

    String methodName = "insertPatient";

    String endPoint = "http://192.168.0.3:8080/Service1.asmx";

    String soapAction = "http://tempuri.org/insertPatient";

    SoapObject rpc = new SoapObject(nameSpace, methodName);

    rpc.addProperty("PatientName", sPatientName);
    rpc.addProperty("InDate", sInDate);
    rpc.addProperty("OutDate", sOutDate);
    rpc.addProperty("DoctorID", sDoctorID);
    rpc.addProperty("Cure", sCure);

    SoapSerializationEnvelope envelope = new
    SoapSerializationEnvelope(SoapEnvelope.VER11);

    envelope.bodyOut = rpc;

    envelope.dotNet = true;

    envelope.setOutputSoapObject(rpc);
    HttpTransportSE transport = new HttpTransportSE(endPoint);

    try {
        transport.call(soapAction, envelope);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
    
```

该方法重点在于指定了 Web Service 的域名以及需要调用的函数名，可以将数据放入 rpc 后调用 Web Service 的“insertPatient”函数

6 系统测试

6.1 测试说明

经过两个月的设计、开发，医疗质量分析系统大致功能已经完成，以下为对该系统进行一个粗略简单的功能测试。

该测试主要使用 Android 虚拟机进行测试

6.2 测试平台

➤ Android 虚拟机：

操作系统：Android 5.1.1

6.3 测试用例

ID	1
被测试功能描述	主界面显示
输入数据/运行要求	无
操作	在模拟器打开 MedicalProject App
预期结果	界面显示正常
评判标准	界面平滑，布局无问题
执行结果	界面显示正常

表 6-1 Medical Project App 测试用例 1

ID	2
被测试功能描述	显示医生数据
输入数据/运行要求	获取医生的数据以及显示
操作	打开 MedicalProject App
预期结果	医生数据显示正常并以 List 显示
评判标准	与数据库显示一致且没有出现布局问题
执行结果	显示成功

表 6-2 Medical Project App 测试用例 2

ID	3
被测试功能描述	录入病人数据
输入数据/运行要求	向数据库录入病人的兵力数据
操作	点击 FAB 后填写数据后点击添加按钮
预期结果	数据库增加数据，并返回首页且相对应的医生数据发生变化
评判标准	成功录入且医生数据也相对应变化
执行结果	录入成功

表 6-3 Medical Project App 测试用例 3

6.4 单元测试

单元测试的主要集中在系统开发过程中，配合数据库控制台“Log”对单元方法的数据处理进行测试。单元方法运行时，输入初始数据，然后方法运行后在页面输出预期的界面，则表示该单元能够正确运行。

6.5 系统测试

把每个功能模块的各个单元组合起来后就可以进行系统测试，系统测试除了测试各个模块的整合测试外，还要对应用的跨平台性进行检查，测试应用在各个 Android 版本上面的适应性。

项目总结

在这个系统从调研、设计、开发再到设计过程中，涉及到了许多方方面面的问题，也买了不少书填补基础知识的薄弱地区与缺失，也体验到理论调研与实际开发之间的差距。本论文主要是基于 Android 的医疗质量分析系统设计与开发，实现了对医生信息的获取、显示和新增病例信息的传输、插入。

本文第 1 章介绍了医疗质量的重要性以及医疗质量分析系统的存在意义；第 2 章描述了该系统需要的技术、开发语言及其开发平台的简介；第 3 章对该系统做了一个简要的调研需求分析；第 4 章对开发该系统所需环境进行了介绍及其安装步骤；第 5 章对该系统实际开发前做一个粗略的模块设计；第 6 章对系统的 Android 端和 Web Service 端进行了详尽的开发；最后第 7 章对该系统做了一个简要的测试，功能实现通过。

参考文献

- [1]Ben. SQL 必知必会(第4版)[M]. 北京:人民邮电出版社, 2013.
- [2]Sierra. Head First Java[M]. 北京:中国电力出版社, 2005.
- [3]郭霖. 第一行代码——Android[M]. 北京:人民邮电出版社, 2014.
- [4]徐宜生. Android 群英传[M]. 北京:电子工业出版社, 2015.
- [5]任玉刚. Android 开发艺术探索[M]. 北京:电子工业出版社, 2015.
- [6]何红辉. Android 源码设计模式解析与实战[M]. 北京:人民邮电出版社, 2015.
- [7]Griffiths. Head First Android Development[M]. 英国:O Reilly Media, 2015.
- [8]郭永洪. .NET Web 应用开发[M]. 北京:清华大学出版社, 2011.
- [9]Robert. 算法(第4版)[M]. 北京:人民邮电出版社, 2012.
- [10]李元峰. 医疗质量评价体系与考核标准[M]. 北京:人民卫生出版社, 2011.

致 谢

这篇论文与系统用时将近两个月，就系统开发用了大半的时间，最后半个月主要是完善 BUG 以及写这篇文章，这期间经历了很多，经历过卡在一个点上卡了一个星期；经历过因为一个字母的错误导致找了一个下午的 BUG；经历过失眠、惊喜，经历了很多。单人的毕业设计虽然功能简单但是却没人帮忙，不像公司工作，一个 App 分为 4 各模块，每个人各负责一个，不懂得可以问前辈。做毕业设计的两个月以来，这些都没有，很多时候，不懂的东西只能 Google，Google 到得方法不知道适不适用，大概适用也要看懂别人的代码再敲到自己的项目里面，到现在论文快要写完了，系统内有部分代码自己都还没看懂或者说只是大概了解，毕竟时间太短，工作太忙。

大学四年，转眼而逝，感谢的人不多，感谢各位教导过我老师，这个系统的开发尤其需要感谢教 Java 的王红勤老师，教算法的钟迅科老师，教 Android 的梁郁君老师，最后需要感谢一位师兄，他的论坛 ID 叫“洛米鸡”，在大三，是他介绍了他公司的一个 Android 项目给我做，让我赚到了第一桶金，让我发现敲代码是一个痛并快乐着的事情，也让我认识到未来找工作很重要的一点，就是办公室再小，也必须有一个小组共同做一个项目，因为自己一个人死脑筋想 BUG，想解决方案的过程实在太辛苦了