|  |  |
| --- | --- |
| 学号： | 2011301500215 |
| 密级： |  |

**武汉大学本科毕业论文**

**ERP中移动审批系统的设计与实现**

|  |  |
| --- | --- |
| 院(系)名称： | 计算机学院 |
| 专业名称： | 计算机科学与技术 |
| 学生姓名： | 陈钱远 |
| 指导教师： | 朱福喜教授 |

二〇一五年五月

**BACHELOR'S DEGREE THESIS**

**OF WUHAN UNIVERSITY**

**The Design and Implementation of Mobile Approval System in ERP**

|  |  |
| --- | --- |
| College ： | Computer School |
| Subject ： | Computer Science and Technology |
| Name ： | Chen Qianyuan |
| Directed by ： | Prof. Zhu Fuxi |

May 2015

**郑重声明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 日期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**摘　　要**

在移动办公与电子商务告诉发展的今天，计算机的商业化应用开辟了企业管理信息处理的新纪元。这对企业管理所采用的方法产生了深远的影响。企业追求着基本相似的营运目标，即实现企业资源的合理有效的利用，以期企业利润最大化。于是，ERP系统作为其中一种方案对企业信息和业务的管理起到了至关重要的改变。它将资源信息的收集、处理、归档等一系列问题规范化，电子化，大幅度提高了业务处理的效率与准确性。

审批是许多日常工作中必不可少一部分。在人们认识到其传统方式的不足以及现阶段电子商务与电子办公的趋势后，越来越多的人开始尝试建立统一的电子移动审批信息平台。这意味着审批模式将发生重大转变，因为电子信息，而非传统信息媒介，将决定着审批决策。研究与开发者也将重点转移到如何合理有效地利用电子信息这一重要的途径优化审批这一传统而重要的步骤。

本论文所尝试设计的便是ERP系统下的移动审批子系统。移动审批在当今并未普及，但其概念和设计却被逐渐多的关注，这说明在不久的将来该类系统会越来越成为主流ERP开发关注的焦点。该系统采用C/S结构开发，功能上分为服务器端、用户端与通信端，实现了用户管理、合同管理、通信与传输等多方面功能。论文采用从总体到细节的方式介绍了整个系统功能设计，和实现上的要点问题与核心技术，综合文字、图例、代码段等方式介绍到每个重要问题的细节，并将展示系统实例测试下的演示情况。

**关键词：**ERP系统，移动审批，C/S

**ABSTRACT**

Under the significant development of mobile office and E-Commerce today, the commercial use of the computer, to which actually has had a profound impact, has opened up a new era of management and processing of enterprise information. Companies pursuing similar operational objectives, namely to achieve rational and efficient use of enterprise resources, in order to maximize their profits. Thus, ERP system, as one of the excellent solution, has played a crucial change to the management of program and business information. It standardizes the process of gathering, archiving, processing of resources information, and makes them all electronic, which greatly improve the business processes efficiency and accuracy.

The approval is an essential part of the most of daily work. After realizing the weakness of the traditional methods and the trend of E-commerce and electronic office, people began to try to establish a unified information platform for electronic mobile approval. This means that the approval mode will undergo significant transformation, as electronic information, rather than traditional information media, will determine the approval decision. Research and development are also shifting their focus to how to rational and effective use of electronic information in this important way, to optimize the traditional and important step of approval.

Mobile approval subsystem designing based in the ERP system is under this thesis attempts. Mobile approval has not yet been popular, but its concept and design is gradually attracting more attention, indicating that in the near future, such systems will increasingly become the focus of attention of the mainly ERP development. The system is developed under C/S structure, and it can be divided into three parts as: the server end, the client end and the communication end. The system manages to functions including achieving user information, contract management, communications and data transport, etc. The thesis presents the entire system functions as well as the key issues during designing and implementation by integrating text, legends, and snippets methods together. It also describes every important detail, and will show examples of demo testing of the system.

**Key words:** ERP system, Mobile approval, C/S

目　　录

[第1章 绪论 1](#_Toc419378152)

[1.1 课题背景 1](#_Toc419378153)

[1.2 目的与意义 2](#_Toc419378154)

[1.3 相关研究和工作 2](#_Toc419378155)

[1.4 论文章节内容安排 3](#_Toc419378156)

[第2章 相关理论 4](#_Toc419378157)

[2.1 ERP系统 4](#_Toc419378158)

[2.2 移动办公与电子审批 4](#_Toc419378159)

[2.3 C/S结构软件开发 5](#_Toc419378160)

[2.4 移动审批系统 6](#_Toc419378161)

[第3章 系统设计 7](#_Toc419378162)

[3.1 总体设计 7](#_Toc419378163)

[3.1.1 开发环境 7](#_Toc419378164)

[3.1.2 系统结构 7](#_Toc419378165)

[3.1.3 功能分析与模块划分 8](#_Toc419378166)

[3.1.4 模块间调用关系 10](#_Toc419378167)

[3.2 核心技术与工具 11](#_Toc419378168)

[3.2.1 通信技术与套接字 11](#_Toc419378169)

[3.2.2 线程 12](#_Toc419378170)

[3.2.3 SQL 12](#_Toc419378171)

[3.3 主要类 12](#_Toc419378172)

[3.3.1 Contract类 13](#_Toc419378173)

[3.3.2 SocketRequest类 13](#_Toc419378174)

[3.3.3 SocketResponse类 14](#_Toc419378175)

[3.4 服务器端 15](#_Toc419378176)

[3.4.1 主模块 15](#_Toc419378177)

[3.4.2 登陆模块 16](#_Toc419378178)

[3.4.3 用户管理模块 18](#_Toc419378179)

[3.4.4 合同管理模块 19](#_Toc419378180)

[3.5 用户端 21](#_Toc419378181)

[3.5.1 登陆模块 21](#_Toc419378182)

[3.5.2 主模块 22](#_Toc419378183)

[3.5.3 用户管理模块 24](#_Toc419378184)

[3.5.4 合同管理模块 24](#_Toc419378185)

[3.5.5 其他 27](#_Toc419378186)

[3.6 通信端 28](#_Toc419378187)

[3.6.1 通信连接 29](#_Toc419378188)

[3.6.2 数据传输 30](#_Toc419378189)

[3.7 数据库 30](#_Toc419378190)

[第4章 系统展示与测评 32](#_Toc419378191)

[4.1 运行环境 32](#_Toc419378192)

[4.2 安装与配置 36](#_Toc419378193)

[4.3 测试运行 32](#_Toc419378194)

[第5章 总结与展望 37](#_Toc419378195)

[参考文献 38](#_Toc419378196)

[致　　谢 40](#_Toc419378197)

# 绪论

## 课题背景

自18世纪产业革命以来，手工业作坊逐步向工厂生产的方向发展，便出现了制造业。随之而来的，所有企业几乎一致地追求着相似的经营目标，即实现企业资源(包括资金、设备、人力等)的高效而合理的利用，以获得企业利润最大化[1]。这一基本目标的追求使工业，尤其是制造业的管理者面临许多的挑战。日趋激烈的市场竞争环境使上述挑战对企业具有至关重要的意义。

直到约50年代的中期，计算机的商业化应用给了企业信息管理一种新的视角。这对企业管理所采用的方法产生了深远的影响。于是，应对上述挑战的各种理论和实践也就相应地产生。ERP系统[3,4]作为其中之一对企业信息和业务的管理起到了至关重要的改变，它将资源信息的收集、处理、归档等一系列问题规范化，电子化，大幅度提高了业务处理的效率与准确性。

图1 移动办公设计概念图

同时，电子商务[2]与移动办公[5]（如图1）的逐步普及也慢慢在改变人们对于工作的思考方式，工作不仅仅局限在繁杂的书卷与纸张中，整个工作流程都可以通过鼠标与键盘的轻松操作完成。进一步的，当人们已不满足于整天盯着电脑屏幕办公时，软件开发商们也开始将视线放在可移动与可穿戴设备上：在等公交时浏览报表，在洗澡时重听今天重要会议录音要点，在度假时及时处理紧急情况，……。

移动办公在今天即使算不上主流，也绝对是不可小视的一股新风潮。审批环节作为非常常规的工作之一，也逐步在朝着电子化与移动化的发展[1,2]。本篇论文所研究与开发技术与软件，便是在这种大背景下的一次尝试。

## 目的与意义

本设计选题来源于真实项目汉默特SRM，其目的是建设集中统一的信息平台，实现多种采购方式，灵活的流程策略配置，多种的采购组织模式，模板化的合同协议管理，快捷的网上电子交易，高效的供应商协同处理，完善的供应商业绩管理，全面的采购过程监控与完整的采购数据分析。

而该系统中涉及到很多合同、报表等文件的审批工作，为了简化该流程，并实现随时随地高效审批，于是考虑设计基于移动设备的审批程序，来极大地简化这一流程。移动审批系统的设计为SRM系统全面高效的管理提供保障，从而进一步提升集中采购的质量、效率和效益。

本论文的研究与设计不仅会在实际项目中用到，而且考虑到目前市场中发行与推广的移动审批系统几乎没有，或者是系统很特殊化不具有普遍价值这样的情况，这篇文章所开发的系统所做的尝试对于日后移动办公与移动审批系统的进一步开发也是一种参考。

## 相关研究和工作

在电子商务告诉发展的今天，供应商管理系统[15]作为电子办公与采购管理系统的一个子系统以及重要模块，在信息化管理的今天得到越来越多的重视。有包括SAP及Oracle公司在内的许多国内外大型软件公司研发过相关的系统。但由于规模，性能，费用等问题行业仍然需要开发新的管理系统。而作为子模块的移动审批系统，近年来也有部分学者或集体尝试过研究开发，有部分成型的产品，如软件“口袋助理”，也有相关部门开发的基于网站的审批程序，可以实现文件审批的基本功能[16]。

然而这个方面的产品并不多，研究也并不火热，并且和大型管理系统结合不紧密。考虑到其应用需求会日益增加，在该论题上的研究将会有显著增加的趋势。采购管理作为电子商务中重要一环今后将逐渐成为极为重要甚至必不可少的工具与平台[5]。

在对于移动审批问题的技术研究中，目前主流的工作采用基于工作流[10]的系统审批模式，如“军字一号”医院信息系统[7]，建设银行个人贷款移动审批系统[8]等，均采用了基于工作流的模式。还有的研究以数据访问中间件[12]作为设计突破口进行开发。然而现有的研究与开发工作并不多，也存在一部分的问题[6,9,11]。本文所开发的系统尝试用一种更加通用的办法解决移动审批这个问题。

## 论文章节内容安排

论文分为五章，从各方面介绍了移动审批系统的研究与设计细节，并给出已开发程序的运行展示和性能分析。论文第一章用于交代选题的背景与目的，简要地介绍了当前电子商务与移动办公的大背景，分析了相关研究工作的情况；第二章对于论文涉及的四个方面理论进行简要地介绍，包括本文开发基于的ERP系统，移动办公与电子审批的背景与特点，移动审批系统，以及本次开发所采用的C/S结构；第三章作为全文的重点，以模块作为划分详细地介绍了系统核心功能的研究与开发，并以重要代码为基础进行讲解；第四章则包括已开发系统的运行演示与分析；第五章对全文内容进行总结，以及对该问题未来发展进行展望。

# 相关理论

## ERP系统

在经过一个漫长的，大致分为四个阶段的发展过程后，ERP系统得以成型为一个现代而先进的理念。ERP系统(Enterprise Resource Planning，企业资源计划系统) 是指建立在信息技术基础上，以系统化的管理思想，为企业决策层及员工提供决策运行手段的管理平台[3]。

企业资源计划ERP整合了企业先进的管理思想，将敏捷制造、供应链管理、人办、物力、全面质量管理等同业务流程和先进的信息技术、第四代语言、图形用户界面集于一体。ERP 系统的应用是对企业的资源进行综合管理和优化配置的过程，减少了生产经营过程中劳动与资本的浪费，实现资金流、信息流和物流的有效整合[4]。根据市场需求，及时、准确地对企业的内外部资源进行优化配置，从而达到提高生产效率，降低成本，满足客户需求，增强企业可持续竞争力的目的，以供应链作为其管理的核心思想，为企业提供一套具有合理化、动态化及持续化的特色管理模式[14]。

本课题所研究开发的移动审批系统，来源于真实项目汉默特SRM，其目的是基于ERP系统建设集中统一的信息平台。作为一个与ERP系统相互配合与补充的电子系统，移动审批系统应当在充分了解并借鉴ERP系统特点的情况下进行研究与开发。

## 移动办公与电子审批

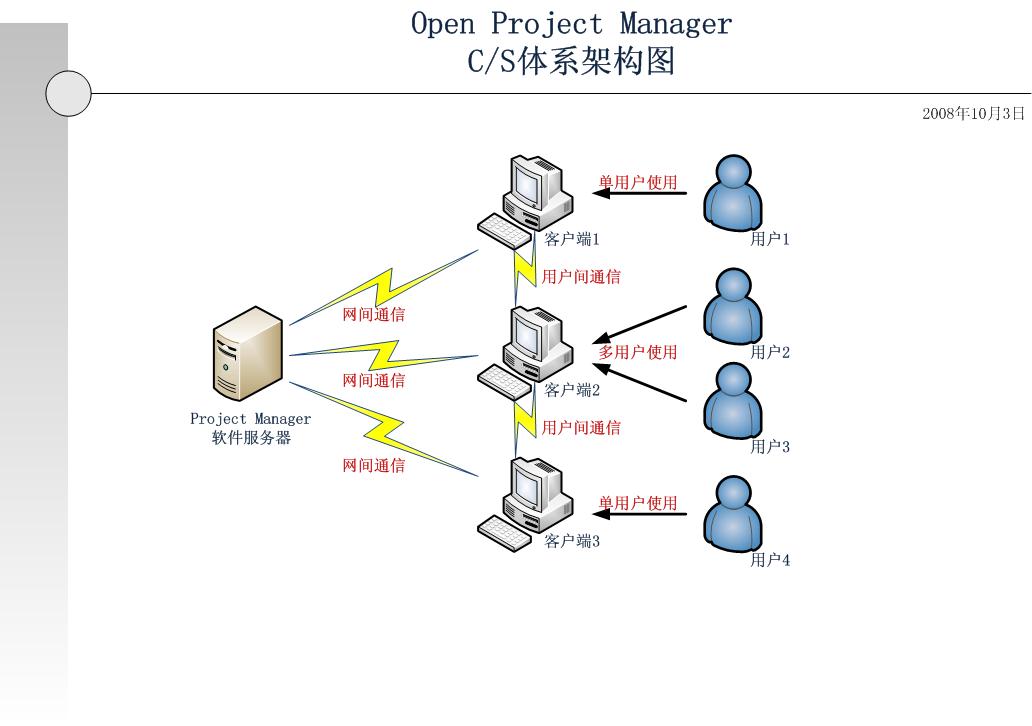
移动办公是高速发展的通信业与IT业交融的产物，随着3G时代的到来，WiFi日趋覆盖，4G通信也已开始试点，网络无处不在，这使实现广泛的移动办公应用成为可能[5]。智能手机和移动互联网的流行将通信业在沟通上的便捷，与IT业在软件应用上的高效和在业务内容上的多样，完美地融合到了一起，这使得办公变得自由、轻松，使人们能高效地开展工作。

审批是许多日常工作中必不可少一部分，传统的人工审批效率低，且审批后各类单据保管难、查询难、统计难；再者，手工审批管控难，极易产生错误或造成损失[2]。为了弥补传统人工审批的不足，并在充分考虑现阶段电子商务与电子办公的大环境下，越来越多的人开始尝试建立统一的电子审批信息平台。

实施电子审批[1]，意味着审批模式将发生重大转变，审批人审阅的纸质材料全部改变为电子信息，通过对电子信息的判断来做出最终的审批决策，也就是说电子信息决定着审批决策。研究与开发者也将重点转移到如何合理有效地利用电子信息这一重要的途径优化审批这一传统而重要的步骤。

## C/S结构软件开发

C/S 结构，即客户机和服务器结构，与B/S结构（浏览器/服务器结构）相对，它是软件系统组成的体系架构（如下图2所示）。C/S 结构的基本原则是将计算机应用任务分解成多个子任务，由多台设备分工完成[21]。客户端完成数据处理，数据显示以及用户接口功能，服务器端完成数据分析，数据库管理等核心功能。通过它可以充分利用两端硬件环境的优势，将任务合理分配到用户端和服务器端来实现，增强用户端应用程序自身的处理能力，同时降低了系统的通讯开销[19]。

图2 C/S系统体系架构图

Client和Server常常被分别部署在相距很远的两台设备上，用户端程序的任务是将用户的要求提交给服务器端程序，再将服务器端程序返回的结果以特定的形式显示给用户；服务器端程序的任务是接收客户程序提出的服务请求，进行相应的处理，再将结果返回给用户程序。同时由于C/S结构中客户端实现与服务器的直接相连，而没有中间环节的复杂过程，因此响应速度较快，适合处理办公类应用[17,18]。

## 移动审批系统

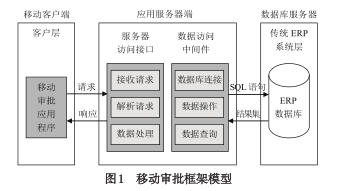
2.2中谈到的电子审批依然存在方便之处，比如审批人必须每天坐在电脑前才能完成业务操作与审批，若审批人出差、开会或因其它原因不在电脑前，就无法完成审批工作，因而影响待审批业务与工作的执行。联系到移动办公，人们开始想到由于手机是每个人必备的通讯工具，因此开发移动设备上的审批系统，将会大大提高审批工作的效率，从而提高整个办公系统的运行效率[13,20]。于是移动审批系统开始越来越多地被受到重视，目前研究下的主流系统结构如下图3所示：

图3移动审批系统结构模型

本文所研究并设计的移动审批系统也满足如上所述的用户端-服务器-数据库的结构，采用C/S结构开发，并考虑到与其隶属项目汉默特SRM的兼容性。下一章将详细介绍移动审批系统中重要的设计要点和细节。

# 系统设计

本章将系统地介绍移动审批系统在设计与开发中的内容，第一节从总体上陈述系统的架构与功能，从宏观上了解各部分功能的划分，第二节介绍了该系统中涉及到的核心技术，第三节则介绍系统中最重要也是使用最频繁的三个类，之后的三节会分别详细地陈述三大部分功能的设计细节与研究开发中碰到的问题，最后一节则介绍系统数据库的设计。

## 总体设计

### 开发环境

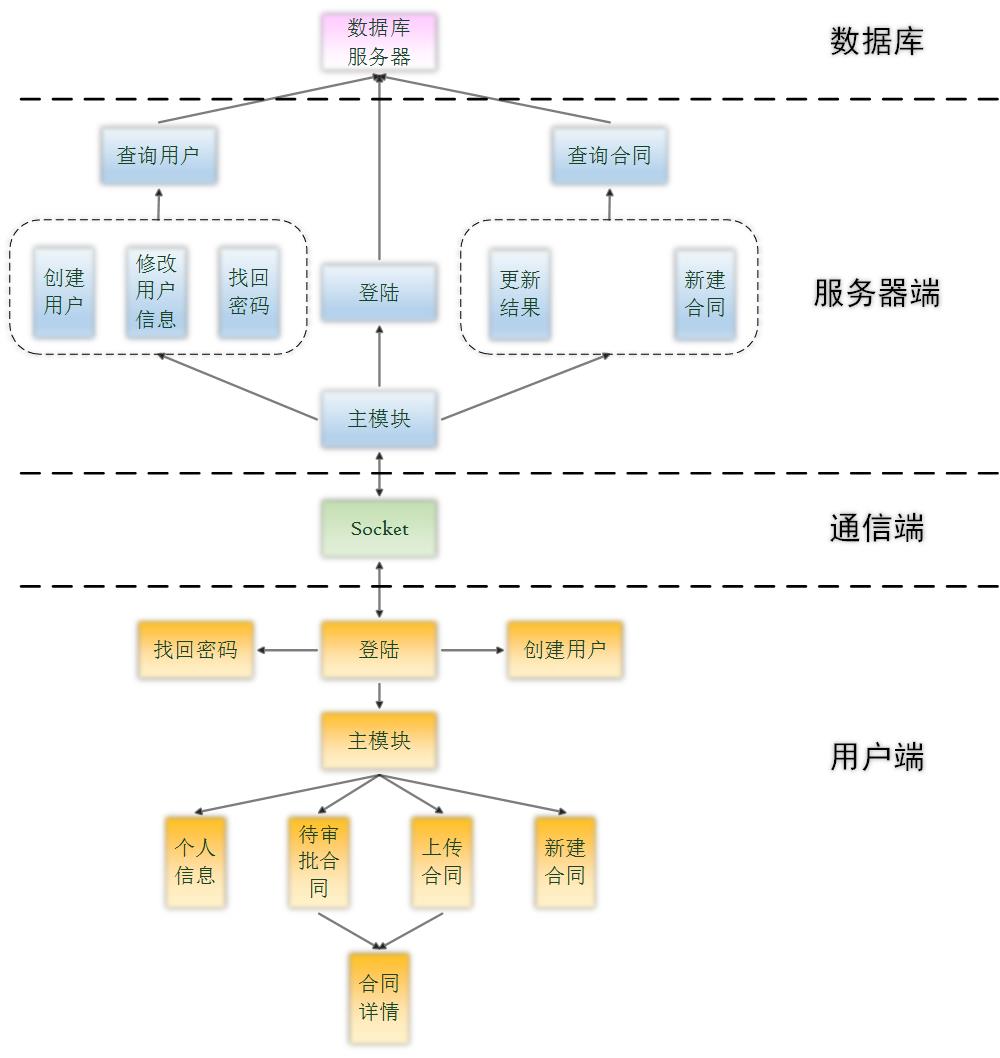
由于本移动审批系统属于真实项目的一部分，故采用与汉默特SRM项目相匹配的开发环境：服务器端用C#语言编程，在Visual Studio 2013环境下开发，数据库服务器使用SQL Server 2012平台；用户端则编写Android系统下的手机App，使用Java语言，在安装Android SDK的Eclipse平台下开发。

如上所述，本篇论文所对应的移动审批系统选用安卓手机作为移动端进行研究开发，而更加完整的系统开发可尝试在不同系统如iOS等平台下开发，或尝试在平板电脑、智能电视等设备下开发该系统。但由于时间所限本系统只能将移动设备简化为安卓手机进行的开发，同样的简化还包括对系统所审批对象的简化，本系统以电子合同作为例子，研究开发系统对于合同这种形式的文件进行审批以及其他的一系列工作，故从下文开始系统待审批的对象均指电子合同。

### 系统结构

本系统总体上使用C/S架构，以高性能计算机作为Server端、参与用户的移动设备作为Client端，来架构整个移动审批系统。C、S两端通过通信完成审批功能，并在服务器端连接数据库，以保存系统的所有数据。其总体架构如下图4所示：

由图4可以看出，系统主要功能由服务器部分、用户部分和通信部分三部分结构组成，其中通信部分作为系统纽带连接另外两部分功能。图中箭头的指向关系也表示了各模块间的调用关系，被指向端的模块为被调用模块。

图4移动审批系统程序结构

### 功能分析与模块划分

由图4可以看出，系统主要功能由服务器端、用户端和通信三部分组成，而每部分又可细化为若干功能模块，下面依次分析各模块的作用：

1. 服务器端：该部分为运行在高性能服务器上的处理程序，用于在远端管理用户，认证用户，接收合法用户的数据、并处理其合理的请求，建立与维护数据库等功能，故将其划分为如下功能模块：

主模块：伴随着服务器端程序的整个生命周期，主模块通过在服务器特定端口新建并监听套接字来开启服务，通过与通信模块的交互获得用户的请求数据包，解析其目的后调用相应的功能模块完成特定功能。

登陆模块：通过查询数据库来验证从用户端传来的登陆信息是否正确，若正确则调用查询合同模块，获得该用户当前相关联的合同信息，连同该用户的详细信息一起，返回给客户端；若错误则告知用户。

创建用户模块：验证上传的新用户信息，若有效，更新数据库并通知用户端；若无效返回错误。

查询用户模块：为其他模块服务的模块，用于向数据库查询包含特定信息的用户数据，返回给调用其的功能模块。

修改用户信息模块：验证所需修改的用户信息，若有效，更新数据库并通知用户端；若无效返回错误。

找回密码模块：通过注册时预留的邮箱帮助用户重置密码。

新建合同模块：验证用户端上传的合同信息，有效则更新数据库。

查询合同模块：为其他模块服务的模块，用于向数据库查询包含特定信息的合同数据，返回给调用其的功能模块。

更新结果模块：验证用户上传的合同审批信息，有效则更新数据库。

其中，创建用户模块、查询用户模块、修改用户信息模块、找回密码模块组成用户管理模块，新建合同模块、查询合同模块、更新合同模块组成合同管理模块。

1. 用户端：该部分为运行在使用者手机端的App，完成包括用户在手机上创建用户、管理用户信息、登陆服务器、上传新合同、查看并审批待回复合同等功能，故将其划分为如下功能模块：

登陆模块：打开应用后显示的主界面，可以通过通信端上传输入的登陆信息至服务器，并根据服务器返回数据决定调用的模块；也可用来调用创建用户和找回密码模块。

主模块：用于存放及调用其它功能模块（页面）。

创建用户模块：通过通信端与服务器交互，验证并上传新建用户信息。

找回密码模块：通过通信端上传邮箱至服务器，以验证是否具有重置密码的资格。

个人信息模块：将服务器传来的用户信息显示给用户，并能交互地修改部分用户信息。

待审批模块：将服务器传来的待审批合同以列表形式显示给用户，包含其简略信息，并通过与用户交互调用详情模块。

上传合同模块：将服务器传来的已上传合同以列表形式显示给用户，包含其简略信息，并通过与用户交互调用详情模块。

新建合同模块：通过通信端与服务器交互，验证并上传新建合同信息。

详情模块：显示合同详情，并根据调用模块的不同显示不同的操作：审批、查看结果或等待中。

其中，创建用户模块、找回密码模块和个人信息模块组成用户管理模块，待审批模块、上传合同模块、新建合同模块和详情模块组成合同管理模块。

1. 通信端：该部分分别运行于服务器端与用户端，用于在服务器与客户两端建立可靠的通信连接，并使用连接相互传输数据。值得强调的是用户和服务器两端采用不同的编程语言开发，其数据结构并不完全相容，需要对所传输数据进行封装与解析，使用统一的、独立于编程的结构进行传输，而JSON格式很好的满足了这个条件，采用其作为中间格式进行传输。据此将通信端功能划分为如下功能模块：

数据封装模块：将C#或Java对象转换为JSON对象以待发送。

数据解析模块：将JSON对象转换为相应编程语言能识别的对象，待其他部分使用。

连接模块：利用套接字在服务器与用户之间建立TCP连接，使其能够正常通信。

传输模块：利用TCP连接传输单个或多个数据对象。

### 模块间调用关系

由图4及3.1.3中各模块功能介绍可以大致了解程序各部分的调用关系，下面更详细地从服务器和用户两个系统主体分别出发，叙述其功能上的调用关系。

首先，服务器端会直接调用到数据库端和通信端，更具体地，服务器的主模块是其全部调用的来源。主模块会直接调用到其三大功能模块：登陆、用户管理、合同管理，使其完成相应的功能；用户和合同管理模块会调用各自的数据库查询模块来完成对数据库的访问或操作；两个数据库查询模块和登陆模块会连接数据库并调用数据库相应的功能，以完成来自主模块完整的调用需求，并返回。除此之外主模块还会在需要与用户端通信时，调用通信端。

用户端与服务器端类似，区别在于用户端不能直接访问数据库。用户端依然依靠主模块调用其它各功能模块（页面），但进入主模块之前，用户需要先通过登陆模块验证身份，或创建用户。相似的，用户端通过调用通信端在必要时与服务器通信。

从严格意义上，服务器端与用户端程序间并不存在调用关系，其功能分别独立运行在自己的设备上，只有当需要进行数据交换，即通信过程的时候，才会依靠通信端相互关联。但由于数据通信对两端功能上有重大影响，如在指令和操作上的数据交换，故可以认为服务器和用户端程序存在相互调用关系。

## 核心技术与工具

### 通信技术与套接字

本文谈及的通信不仅仅指设备上的宏观意义上的通信，主要指应用程序之间微观意义上的通信。应用程序之间的相互通信（数据交换）一直是困扰广大程序员的难题，尽管已经出现了各式各样的解决方案，但迄今为止没有哪一种方案是完美无缺的。因此，只有了解其分别的优缺点以及设计时在特定的情况下的要求，选择最佳方案，以满足最终的要求。主流的通信手段包括共享，Pipe通信，COM和DCOM，Socket等。

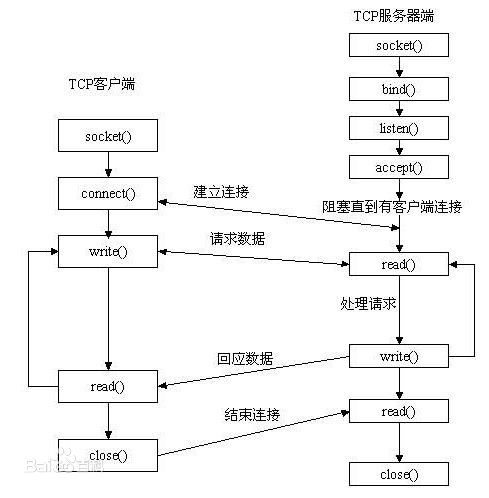
我们根据移动审批系统的特性，首先共享式并不合适，其效率太过于低下，不适合实时交互类的程序通信；审批系统在寻找待审批人等相关功能中需要用到广播，故Pipe通信也不适合；COM编程太过于繁杂，审批系统没有如此大的规模需要采用这种方式，所以Socket套接字方式成为最优选择，不仅因为其通讯效率高，更重要的是系统服务器与用户端采用了不同的平台，只有Socket的可移植性满足跨平台通信的要求。

图5Socket工作原理图

上图5所示即为实现TCP协议的套接字工作原理。关于套接字在移动审批系统中的具体应用，将在3.6通信端设计中详细介绍。

### 线程

线程，与进程相对，有时被称为轻量级进程，是程序执行流的最小单元。线程是进程中的一个实体，轻型实体，是被系统独立调度和分派的基本单位，线程自己不拥有系统资源，只拥有一点儿在运行中必不可少的资源，但它可与同属一个进程的其它线程共享进程所拥有的全部资源。一个线程可以创建和撤消另一个线程，同一进程中的多个线程之间可以并发执行。由于线程之间的相互制约，致使线程在运行中呈现出间断性[22]。每一个程序都至少有一个线程，若程序只有一个线程，那就是程序本身，我们习惯性称之为主线程。

在移动审批系统中，由于安卓系统对于通信功能的要求，需要使用线程来执行操作。而对于服务器端程序，若监听套接字可以实现多线程操作的话，更可以同时响应多个客户端的连接与请求，提高系统效率。

### SQL

SQL（结构化查询语言）是一种特殊目的的编程语言。用于作为一种数据库查询和程序设计语言，和存取数据以及查询、更新和管理关系数据库系统的特殊编程语言。结构化查询语言是高级的非过程化编程语言，允许用户在高层数据结构上工作。用户不需要指定对数据的存放方法，也不需要了解具体存放数据的方式，所以具有完全不同底层结构的不同数据库系统, 可以使用相同的结构化查询语言作为数据输入与管理的接口。结构化查询语言语句可以嵌套，这使它具有极大的灵活性和强大的功能。

SQL语言包括六部分内容：数据查询语言、数据操作语言、事务处理语言、数据控制语言、数据定义语言、指针控制语言。在移动审批系统中，只需要用到数据定义语言创建数据库及数据表，以及查询语言和操作语言来执行数据的操作即可。

## 主要类

本节将介绍审批系统中最重要的三个类，其被用不同的语言分别定义在服务器端和用户端，但结构是一样的，分别是合同类Contract、套接字接收类SocketRequest和套接字响应类SocketResponse。论文分别给出其在服务器端C#的定义，用户端类似。

### Contract类

定义合同类及其包含的23个属性，属性与项目汉默特SRM中保持一致，分别包括：

数值属性：合同状态Status，合同总额Sum，序号Id

其中合同状态有三种取值，分别是0（待审批中），1（已同意），2（已反对）；

字符串属性：合同标题Title，合同上传者Author，待审批人Signname, 合同承办人名Dealname，合同简介Content，合同编号Number，合同类型Category，供应商编号Pronum，供应商名Proname，物料编号Goodnum，物料名称Goodname，付款类型Sum\_cat，合同税率Rate，付款类型Payment，履约期限Period，交货地点Site，违约条件Punish合同备注Ps，合同审批意见Comment

其中合同上传者与待审批人均为系统中存在的合法用户的用户名，每次新建、审批合同时需验证这两个属性是否有效；

时间属性：合同生成时间Time

### SocketRequest类

该类负责从用户端传入请求与数据，由用户端建立该类的对象，经通信端传输至服务器端解读，其定义如下：

其中，Op为本次用户端请求的操作名，用来告诉服务器应调用哪个功能模块响应其请求；RequestData是本次请求所需的数据，服务器判定操作后会交由指定的模块进行解读。

[Serializable]

public class SocketRequest{

public string Op;

public object RequestData;

}

### SocketResponse类

与SocketRequest类相反，负责从服务器端传送指令与数据至用户端，由服务器端建立对象传送至用户端进行解读，其定义如下：

其中，Op为服务器传送的指令，ResultData为传送给用户端的数据，特别的，当服务器只需要传送一条简单指令，如同意或反对用户端请求时，ResultData为空。

[Serializable]

public class SocketResponse{

public string Op;

public object ResultData;

}

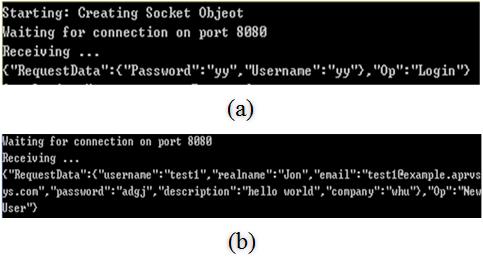
注意到三个类均声明了Serializable，这是为了使该类的对象能够被序列化为JSON对象，以及能被从JSON对象反序列化回原对象起到了保证。而对于后两个数据传输类SocketRequest和SocketResponse，其数据部分均以object类型，而非固定类型定义。这里简要分析下如此设计的理由，首先以SocketRequest类为例，看看用户端发送至服务器端的数据在反序列化后是如下图所示的样子：

图6服务器收到不同请求的格式

可以看到不同操作Op所传来的RequestData格式并不相同，这里没有采用像定义Contract类一样以固定的结构定义RequestData，而只用object类型保存若干对嵌套的<string, object>（类的定义见上）。如图6-(a)中所示收到的登陆请求，其RequestData部分包含两对<string, object>，而对于图6-(b)的创建用户请求，其RequestData部分包含六对<string, object>。其他请求中可能在这里的object又保存为嵌套的<string, object>。

这样的设计既满足了用户端数据格式的灵活性，又使服务器功能模块具有可扩展性。比如将来服务器添加新的模块使之能够处理新的用户请求，系统不需要为该请求数据包的RequestData格式添加新类，而只需要按照<string, object>的格式组织新数据即可。

而发送数据的灵活性给服务器端读取数据带来麻烦。固定的结构如Contract类可以通过直接新建类的对象保存发送数据，而无固定结构的RequestData则只能采用逐个数据读取的办法。设计专门采用了带string的值对结构存储，便于C#端使用字典Dictionary类进行读取。如对于图6-(a)登陆请求的读取，可以采用如下方式：

这样便可将灵活结构的RequestData读出，供后续功能使用。这个例子介绍了数据读取中的一个环节，更完整的传输叙述见3.6.2通信传输部分。

Dictionary<string, object> UserData =

(Dictionary<string, object>)request.RequestData;

string UserName = UserData["Username"].ToString();

string PassWord = UserData["Password"].ToString();

## 服务器端

服务器部分用于在远端管理用户，认证用户，接收合法用户的数据、并处理其合理的请求，建立与维护数据库等功能，其中，主模块作为枢纽，依据不同的Op指令操作调用不同的功能模块。

### 主模块

作为服务器端的主程序，该模块伴随着服务器端程序的整个生命周期，主要负责完成两方面的任务：服务器端套接字的建立与关闭，以及解析用户端请求并调用响应功能模块。

首先，套接字与通信技术的相关概念在3.2节中已经详细地介绍过，而其一系列的操作在C#中有完备的类与方法，在主模块中需要做的便是调用相应的方法执行操作。套接字需要绑定在地址的特定端口上进行监听，所以主模块首先将服务器所在ip地址及本系统监听的端口号存储在字符串中，如下所示：

再通过实例化Socket类的对象listener，调用其Listen方法就可以实现套接字的建立，关于套接字的创建将在3.6通信端详细介绍。

string host = "192.168.1.103";//服务器IP地址

int port = 8080;//服务器监听端口号

主模块第二个功能，解析请求与模块调用，则主要依靠传入对象的取值判断。当新收到用户端的连接后，新建SocketRequest对象request保存通信端反序列化得到的用户端对象，并判断其Op值：

主模块最后调用listener的Close方法关闭服务器套接字，结束程序。

switch (request.Op){

case “Login” 调用登陆模块; break;

case “UpdateAprvResult” 调用更新结果模块; break;

case “Upload” 调用新建合同模块; break;

case “NewUser” 调用新建用户模块; break;

case “EditUser” 调用修改用户信息模块; break;

case “FgtPswd” 调用找回密码模块; break;

}

### 登陆模块

登陆模块主要完成两项工作：向数据库验证用户名密码是否正确，以及正确时向用户返回包括个人信息、待审批合同列表、上传合同列表在内的三类数据。该模块将与数据库频繁接触。

在登陆模块被主模块调用时还会收到其传来的用户数据RequestData，采用3.3节所述的方式读取用户数据后便可开始工作。验证用户信息的工作需要调用用户管理模块中的查询用户功能，通过SQL查询验证信息。使用SQL查询首先需要连接数据库服务器，关键代码如下：

开启连接后，可以通过连接对象dbcon的CreateCommand方法新建并执行相应SQL查询：

public static string SqlConLoc = "Data Source=(localdb)\\Projects;

Initial Catalog=AprvSys\_db;Integrated Security=True";//数据库连接信息

var dbcon = new SqlConnection(SqlConLoc);

dbcon.Open();

其中，UserName和PassWord为用户发送来的待验证信息，当用户信息表UserInfoTable（详细定义见3.7节）中存在用户名Username为UserName的表项，且该表项的密码值Password等于PassWord，则验证成功，否则失败。验证成功后需要得到该用户数据，只需要再执行三次SQL查询，得到个人信息与合同信息即可。查询方法如上段代码，只需要将dbcommand.CommandText中的字符串改为本次查询所需的SQL语句即可。三次查询后得到数据，将其按格式写入SocketResponse的对象：

var dbcommand = dbcon.CreateCommand();

dbcommand.CommandText = "SELECT Password FROM UserInfoTable WHERE Username = '" + UserName + "'";

SqlDataReader dbreader = dbcommand.ExecuteReader();

if (dbreader.Read()){

if (PassWord.Equals(dbreader[0].ToString())){

//验证成功

}

} else {

//验证失败

}

其中loginresp为定义类LoginResponse的对象，由于登陆模块返回数据太多，定义该类方便直观观察。但本质上构造的对象loginresp还是会转换成嵌套<string, object>对的形式（如上采用强制转换object类型的方式），故采用直接构造字符串的方式也可以达到同样的效果。LoginResponse类的定义如下，包含两个合同数组和字符串形式存储的个人信息：

var response = new SocketResponse{

Op = "Correct", //告知用户登录成功

ResultData = (object)loginresp

};

最后一步即通过通信端将response对象返回用户端，则登陆模块任务结束。

[Serializable]

public class LoginResponse{

public Contract[] AprvConList, UpldConList;

public string Email, RealName, Company, Description;

}

### 用户管理模块

用户管理模块分为四部分：创建用户模块，负责验证上传的新用户信息并更新数据库并通知用户端；修改用户信息模块，负责验证所需修改的用户信息并更新数据库；找回密码模块，负责通过注册时预留的邮箱帮助用户重置密码；查询用户模块，负责通过SQL在数据库中查询用户信息。当被主模块调用以后，使用3.3节所述Dictionary类的方法读取到用户数据，再分别开始自己的工作。

查询用户模块的工作在3.4.2中已经描述，即首先建立与数据库服务器的SQL连接，再根据所需查询的内容（如3.4.2登陆模块需要查询用户名是否存在且与密码是否匹配）执行SQL中的SELECT语句，查询结束后关闭SqlDataReader的对象。

创建用户模块首先需要通过调用查询用户模块验证该用户名是否已存在，若返回空则验证成功。再将新用户的完整信息通过SQL写入用户信息表的新表项。注意到3.4.2使用ExecuteReader方法执行SQL查询并返回结果，而这里新增表项语句无需结果返回，故使用ExecuteNonQuery方法：

string sql = "INSERT INTO UserInfoTable VALUES(Value)";

dbcommand.CommandText = sql;

int flag = dbcommand.ExecuteNonQuery();

if (flag > 0) {

//新建表项成功

}

其中Value为存储新用户信息的字符串，当flag返回非零值时表示插入数据行成功，生成答复数据包后通过通信端传送，任务结束。

修改用户信息模块与创建用户类似，同样先通过查询用户模块验证该用户Username是否存在，再通过SQL的UPDATE语句更新指定属性的值即可，这里同样调用ExecuteNonQuery方法执行SQL操作。

找回密码模块依然类似，只是多了一次与用户的通信交互。先向数据库验证用户名是否存在且其输入的邮箱与安全问题和服务器中保存的一致，则服务器告知用户请求被允许，直到用户上传入新的密码，用ExecuteNonQuery方法更新数据库即可。

### 合同管理模块

合同管理模块分为三部分：新建合同模块，用于验证用户端上传的新合同信息并更新数据库；查询合同模块为其他模块提供向数据库查询包含特定信息的合同数据的帮助，并返回给调用其的功能模块；更新结果模块用于验证用户上传的合同审批信息，并更新数据库。

审批系统中一份合同将关联至少两名用户：上传者与待审批者，如图7所示合同1与用户1（上传者）和用户2（待审批者）两名用户关联；同样的，一名用户也关联到两类合同，如图用户2与合同1（待自己审批的）和合同2（自己上传的）均关联。明白这两个关联后不难知道，合同管理模块中几乎每一次对数据库的更改都不仅关系到合同信息表，也关系到用户信息表，且至少涉及到两名用户。

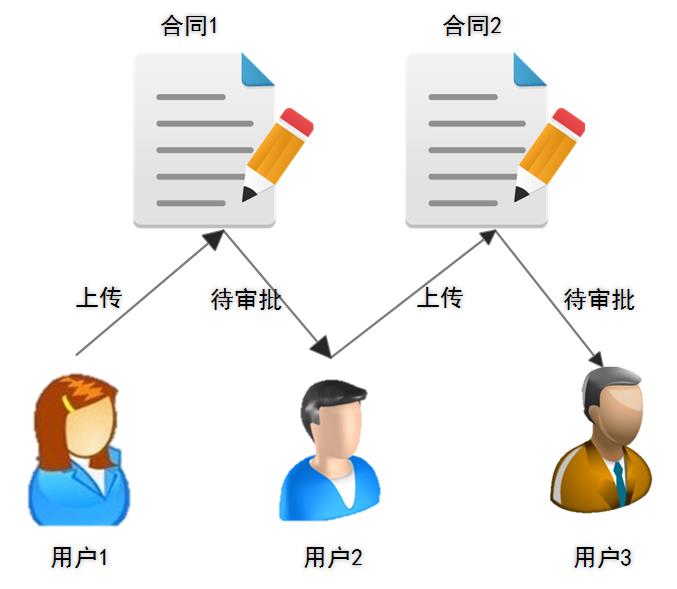


图7合同与用户关联图

查询合同模块与3.4.3中查询用户模块基本类似，与数据库服务器建立连接后通过向ContractTable执行特定SELECT查询语句得到表项，并返回结果，来为其他模块服务。

新建合同模块首先需要通过查询合同模块验证新合同信息是否有效，之后再通过查询用户模块验证合同的待审批人的用户名是否有效，满足这两点才算有效的新合同。通过SQL插入语句新建合同项至数据库：

其中Value为新合同信息。新建合同后需要通过查找用户模块更新用户信息表中上传者表项的上传合同列表属性UpldCon，与待审批者表项的待审批合同列表属性AprvCon。之后给用户回复信息，任务结束。

string sql = "INSERT INTO ContractTableVALUES(Value)";

dbcommand.CommandText = sql;

int flag = dbcommand.ExecuteNonQuery();

if (flag > 0) {

//新建表项成功

}

更新结果模块用于当用户对待审批合同进行批复以后，更新数据库中合同信息表。使用SQL的UPDATE语句更新该合同表项的Status属性和Comment属性即可。该模块不需要更新用户信息，因为更新结果操作并没有影响用户与合同之间的关联属性。

## 用户端

用户端作为运行在使用者安卓手机端的App，完成包括用户在手机上创建用户、管理用户信息、登陆服务器、上传新合同、查看并审批待回复合同等功能，也包括App界面的开发。

### 登陆模块

登陆模块作为打开应用后显示的主界面，其主要功能包括通过通信端上传输入的登陆信息至服务器，并根据服务器返回数据决定调用的模块；也可用来调用创建用户和找回密码模块。

首先，登陆模块MainActivity作为App程序，也即Java程序的入口，在AndroidManifest.xml文件中注册响应启动程序的请求：

登陆界面需要提供给用户输入用户名和密码的文本框，以及分别能跳转至创建用户页面、找回密码页面和功能页面的三个按钮，故设计如图8所示界面。

<activity

android:name=".MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

为“创建用户”和“忘记密码”按钮添加监听器使之具备页面跳转功能，如下所示：

使用Android的Intent类显示地调用并打开一个新的页面，这样便可以实现当用户点击下面两个按钮时跳转至相应的页面。而登陆按钮btn\_login的功能则复杂一些，当用户点击btn\_login时，系统首先验证输入的用户名或密码是否合法，若合法则通过通信端与服务器交互。当服务器返回登陆成功的信息及用户数据后，通过Intent打开功能页面并借助Bundle传递数据，核心代码如下所示：

btn\_newuser = (Button) findViewById(R.id.button\_newuser);

btn\_newuser.setOnClickListener(new ReceiverListener(){

public void onClick(View v){

Intent intent = new Intent(MainActivity.this, NewUserActivity.class);

startActivity(intent); //跳转至创建用户页面

}

});

注意到Bundle只能存储String、int等简单类型而不能存储类的对象等复杂的结构化类型，故在用户端数据依然需要采取序列化的方式进行传输。

Bundle bd = new Bundle();

bd.putString("AprvConList", AprvConListStr);

// 将所有用户数据以字符串形式存到bd中

Intent intent = new Intent(MainActivity.this, ViewContentActivity.class);

intent.putExtras(bd);

startActivity(intent);

### 主模块

系统在用户成功登陆后会自动跳转至主模块页面ViewContentActivity，该页面中包含了用户端系统最重要的三大功能页面：个人信息页面、待审批合同页面和上传合同页面，这样的设计是出于用户界面的简洁性和功能访问的快速性而考虑的。

为了能制作出理想中的页面滑动效果，在参考现有技术并加以改动的情况下，设计使用ViewPager类配合SectionsPagerAdapter适配器类（继承于PagerAdapter类）实现预想的效果。首先将三大功能页面作为三个子视图（View）存储在视图列表ArrayList<View>中，核心代码如下所示：

再用SectionsPagerAdapter适配器将ViewPager与视图列表关联再一起即可。

SectionsPagerAdapter mSectionsPagerAdapter = new SectionsPagerAdapter(mListViews);

mViewPager = (ViewPager) findViewById(R.id.pager);

mViewPager.setAdapter(mSectionsPagerAdapter);

mViewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {

@Override

public void onPageSelected(int position) {

actionBar.setSelectedNavigationItem(position);

}

});

ViewPagermViewPager = (ViewPager) findViewById(R.id.pager);

List<View>mListViews = new ArrayList<View>();

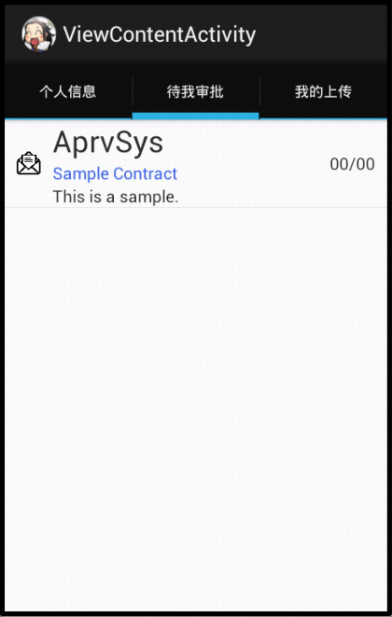
LayoutInflater mInflater = getLayoutInflater();

View\_person\_info = mInflater.inflate(R.layout.view\_person\_info, null);

mListViews.add(View\_person\_info);

// 寻找并添加另外两个View

这样就将三个功能页面与ViewPager的三个视图关联起来，当用户在屏幕做出滑动输入时，SimpleOnPageChangeListener会捕捉到该输入并显示当前位置的页面。对于类SectionsPagerAdapter，只是重写了父类PagerAdapter的部分方法使之能为系统使用的List<View>结构建立适配器。设计好的主界面如下图9所示：

图8登陆界面设计图 图9主界面设计图 图10 创建用户页面图

### 用户管理模块

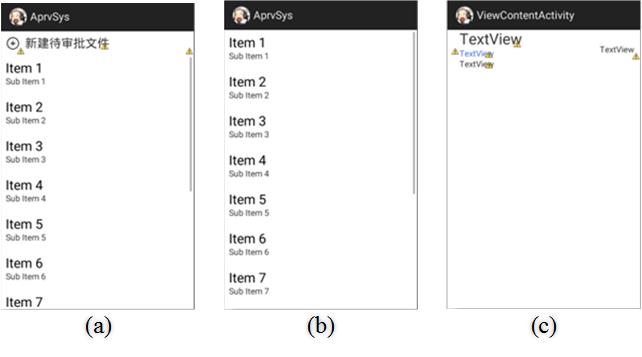
用户管理模块包括三部分：创建用户模块负责通过通信端与服务器交互，验证并上传新建用户信息；找回密码模块负责通过通信端上传邮箱至服务器，以验证是否具有重置密码的资格；个人信息模块负责将服务器传来的用户信息显示给用户，并能交互地修改部分用户信息。三部分功能分别与服务器用户管理模块（3.4.3）相应进行配合，以完成整个系统用户管理的任务。

创建用户模块使用页面NewUserActivity，提供输入框供用户输入注册信息，在提交前确认必填项是否缺失，以及数据格式是否正确，之后通过通信端与服务器交互。其界面设计为如上图10所示。

找回密码页面和个人信息页面界面类似，功能上与3.4.3服务器部分功能完全匹配，完成找回密码和显示个人信息的功能。

### 合同管理模块

合同管理模块包括四部分：待审批模块和上传合同模块将服务器传来的合同数据以列表形式显示给用户，包含其简略信息，并通过与用户交互调用详情模块，设计图见下图11；详情模块则负责显示合同详情，并根据调用模块的不同显示不同的操作；新建合同模块通过通信端与服务器交互，验证并上传新建合同信息。

图11上传合同列表、待审批合同列表与合同简略信息设计图

待审批模块和上传合同模块均以视图的形式存在于主界面上（图9）。为了能显示简略信息首先要将合同数据从登陆界面传来的Bundle中提取出来，并存储在Contract数组中，下面给出以待审批模块为例的核心代码：

至此已得到包含合同数据的列表，如图11-(b)所示。为了能在每个列表项中显示如图11-(c)所示的简略信息，需要重写数组适配器使之能将每个合同的简要信息视图与List<Contract>每一列表项对应。故新建类ContractListAdapter继承ArrayAdapter<Contract>并重写getView等方法，使其支持上文所述界面设计。

List<Contract> contracts = new ArrayList<Contract>();

AprvConList = new JSONArray(getIntent().getStringExtra("AprvConList"));

for(i=0;i<AprvConList.length();i++){

JSONObject aprvobj = AprvConList.getJSONObject(i);

Contract aprvcon = new Contract(aprvobj);

contracts.add(aprvcon);

}

上传合同模块与之类似，只是在列表上方增加一个能够点击的TextView，使之能够调用并跳转至新建合同模块。待审批合同列表和上传合同列表的每个表项均可以被点击，而调用并跳转至该合同的详情模块，该功能通过给ContractListAdapter对象添加setOnItemClickListener即可。在新建setOnItemClickListener对象的onItemClick方法中实现基于Intent的显示页面跳转，并用Bundle传递被点击合同项的数据。

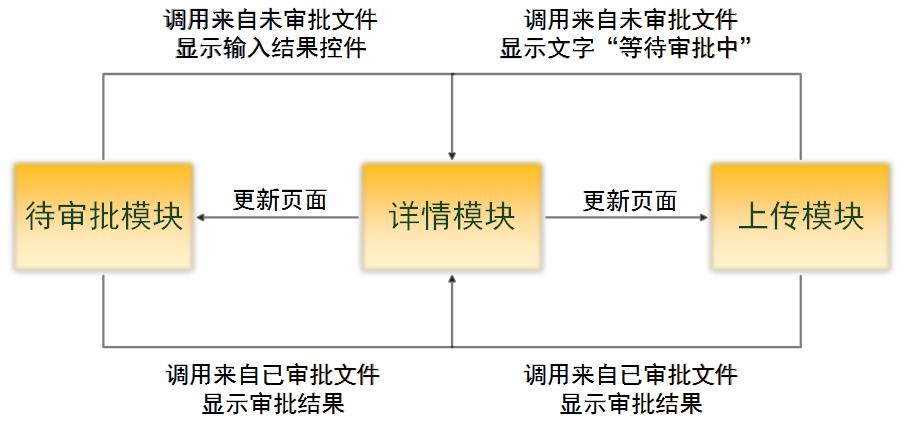
详情模块除了显示传入的合同详细信息外，有两个问题需要解决。一是为了减少代码冗余，待审批模块和上传合同模块将调用同一个详情模块，因为除了微小的差异外其他代码大致相同；二是若详情模块对合同有操作，关闭详情页面返回主页面时应该更新页面中的内容。各页面与详情页面的访问关系如下图所示：

图12详情模块与其他模块间调用关系图

对于第一个问题，只需要在调用时附加的Bundle中增加指示信息，并在详情模块中依据信息略微调整界面即可。对于第二个问题，则需要在使用Intent调用详情模块的时候改变调用方法为startActivityForResult方法而非startActivity方法：

// 在待审批模块为合同项添加setOnItemClickListener时：

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

Intent intent = new Intent(ViewContentActivity.this, ContractDetailsActivity.class);

……

startActivityForResult(intent, ForAprvResult); //系统定义的任意int值

}

相应的，在详情模块需要关闭页面并返回调用前的页面时，用返回的Intent传送需要更新的信息：

这样在调用完返回到主页面时，当收到带有标记ForAprvResult的回复后，便知道是详情页面传回的回复，提取aprvresult中的信息并做相应改变，再调用适配器的notifyDataSetChanged方法更新界面即可。

Intent aprvresult = new Intent();

bd.putString("Comment", t1.getText().toString());

……

aprvresult.putExtras(bd);

setResult(RESULT\_OK, aprvresult);

finish();

本节最后一个模块，新建合同模块与3.5.3中新建用户模块类似，区别在于当从新建合同页面返回主页面时，也需要更新上传合同列表数据，即更新界面，故同上采用startActivityForResult的方法，以不同的标记ForUpldResult调用新建合同页面，返回主页面后做同上处理，更新界面即可。

### 其他

图13 handler对象调用关系图

系统中多次遇到需要更新列表信息，或者弹出提示框等界面上的改动，而同时又因需要通过通信端多次与服务器联络而是用套接字（见3.6节）。从API Level 14以后安卓禁止在主线程内使用套接字等网络通信功能，这使得所有Socket相关操作都需要在线程中执行，而在安卓系统中，只有主线程内可以做出界面上的改动并即时显示出来。这种情况下设计使用Handler类解决界面更新问题，在主线程内调用Handler类并实例化对象handler，当线程需要更新界面时，调用handler的相关方法即可“回到”主线程中执行操作，如上图13所示。

主线程中新建handler对象时通过重写handleMessage方法使其能根据不同需求更新界面，线程内使用时只需要调用其sendMessage方法并传入信息即可，核心代码如下：

private Handler handler = new Handler() {

public void handleMessage(Message msg) {

switch (msg.what){

case ERROR:{

new AlertDialog.Builder(MainActivity.this)

.setTitle("错误")

.setMessage(msg.obj.toString())

.setPositiveButton("确认", null)

.show(); break;

}}

super.handleMessage(msg);

}};

信息msg内包含what属性和obj属性，可以告知handler执行任务需要的信息，如这段代码线程告诉handler需要弹出一个系统提示框，通知用户程序发生了内容为obj的错误。

## 通信端

通信端可以说完成了系统中最重要的两项任务：通信与传输。通信端运行于服务器端与用户端的程序中，主要以Socket及其对象的形式存在。

### 通信连接

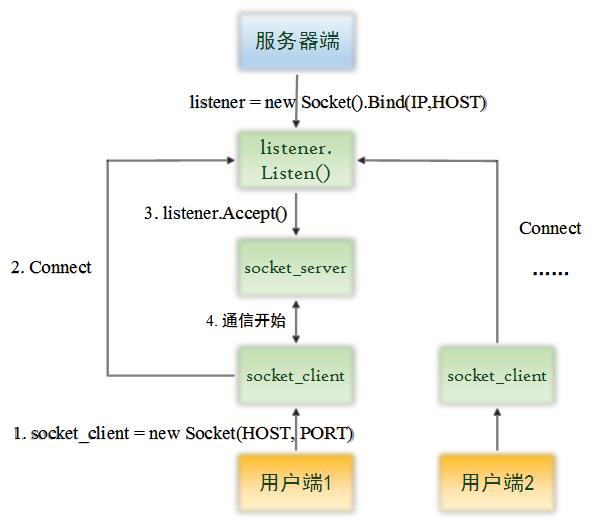
服务器与用户借由Socket通信的关系可以由下图表示：

图14服务器与用户通信关系图

图14和3.2.1对Socket工作原理的介绍已经说明了通信端如何完成的连接与通信工作，下面给出该部分的核心代码：

// 服务器端

Socket listener = new Socket(AddressFamily.InterNetwork,

SocketType.Stream, ProtocolType.Tcp);//创建一个负责监听的套接字对象

listener.Bind(new IPEndPoint(IPAddress.Parse(host), port));//绑定指定端口

listener.Listen(10);//开始监听

Socket socket\_server = listener.Accept();//为新建连接创建新的Socket。

// 用户端

Socket socket\_client = new Socket(HOST, PORT);

这段代码用于建立套接字，服务器在整个工作流程中只需要新建一个Socket对象listener负责监听，重复地处理用户端的连接，而服务器端与用户端每进行一次通信都需要再新建一对套接字对象（socket\_server和socket\_client）。当服务器accept了用户的socket连接后，就可以开始传输了。

### 数据传输

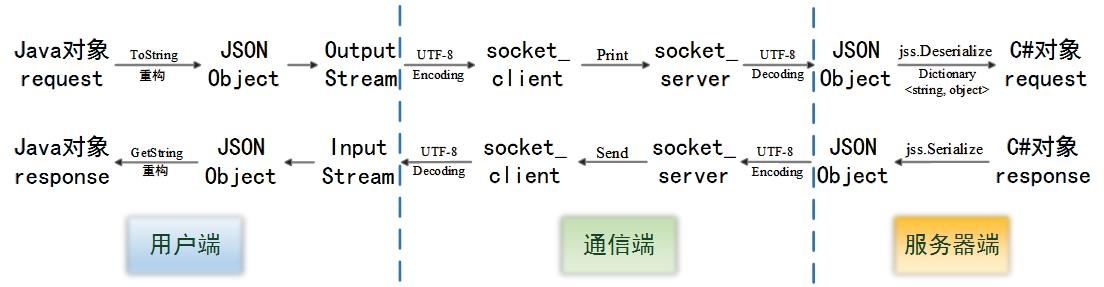
如前所述，由于服务器端与用户端采用编程语言不同，数据的传输成为本系统的难点之一。下图是本系统采用的以JSON为中间格式的数据传输方式：

图15数据传输格式图

图中jss为JavaScriptSerializer类的对象，由图15可看出在整个数据传输过程，多次涉及对于数据结构与编码方式的改变。服务器端，读取JSON对象所用的Dictionary<string, object>方法在3.3节中已经介绍，而在用户端，因为Java可以对JSON对象直接进行操作，故只需要使用字符串的相关函数便可进行数据读取。这里的通信端采用UTF-8格式统一编码传送数据，保证传输的稳定。

## 数据库

移动审批系统的数据库并不复杂，只有一个包含两个表的数据库，分别为用户信息表UserInfoTable和合同信息表ContractTable。合同信息表的设计与3.3.1中Contract类是基本一致的，用VARCHAR存储各个字符串，INT存储数值类型，已每个合同序号Id作为主键。

用户信息表中的非空信息与3.5.3中创建用户页面所规定的一致。除了包含个人信息外，还包括两个合同列表字符串，SQL代码如下：

CREATE TABLE [dbo].[UserInfoTable] (

[Id] INT NOT NULL,

[Username] VARCHAR (20) DEFAULT ('demo') NOT NULL,

[Password] VARCHAR (30) DEFAULT ('demo') NOT NULL,

[AprvCon] VARCHAR (50) NULL,

[UpldCon] VARCHAR (50) NULL,

[RealName] VARCHAR (20) NULL,

[Company] VARCHAR (50) NULL,

[Email] VARCHAR (30) DEFAULT ('example@aprvsys.com')

NOT NULL,

[Description] TEXT NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

);

AprvCon与UpldCon以字符串的形式存放与该用户关联的合同序号，已分号隔开，如用户demo的AprvCon和UpldCon的值分别为“2；6”和“3；4；7”，表示序号为2和6的合同需要demo负责审批，而demo上传了序号为3,4,7的合同等待其他用户审批。

# 系统展示与测评

## 运行环境

由于设备限制，没有合适的应用服务器与数据库服务器可以使用，故本次运行测试的应用与数据库服务器均使用型号为ThinkPad E40的笔记本电脑作为代替，其基本参数为：Intel i5CPU 四核2.67GHz，3GB内存，win8.1专业版32位操作系统。

用户端使用型号为HTC One S的安卓手机，操作系统为Android 4.0.3（API Level 15），高于程序要求的最低API Level 14的要求。为了截图方便，也将系统安装与同参数下的安卓模拟器中，执行同样测试。

## 测试运行

本节将会展示移动审批系统每个功能，从新建用户到登陆入系统，再到新建合同、查看合同、审批合同等一系列功能运行时的数据与程序细节。首先是创建用户功能，用户端与服务器端运行时的截图如下所示：

图16 创建用户功能运行展示图

图16-(a)为用户在手机中输入的信息，点击“立即注册”按钮后将数据提交至服务器，可以从图16-(c)中看到服务器收到和发出数据包的详细信息，此刻收到Op为“NewUser”的操作请求以及新用户的信息，服务器验证后返回指令为“Correct”的回复，用户端收到指令为正确的回复后，通知用户创建用户成功，如图16-(b)所示，通过3.5.5中提到的Handler类创建系统通知弹窗，用户确认信息后返回前一个页面。

图17 登陆功能运行展示图

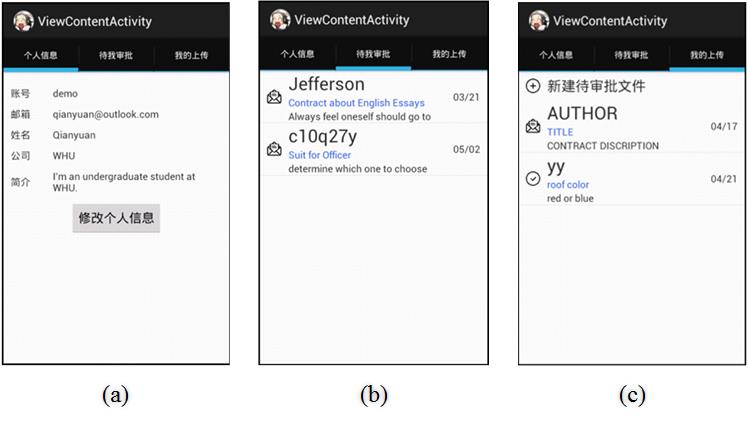


图18 系统主界面运行展示图

上图17显示的为使用用户demo登陆系统的运行图。点击“登陆”按钮后，服务器会收到操作为“Login”的请求，验证通过后会返回如图17-(b)所示的该用户的三大类信息（3.5.1节所述），这里的demo用户共关联有4个合同，故返回信息看起来很长。

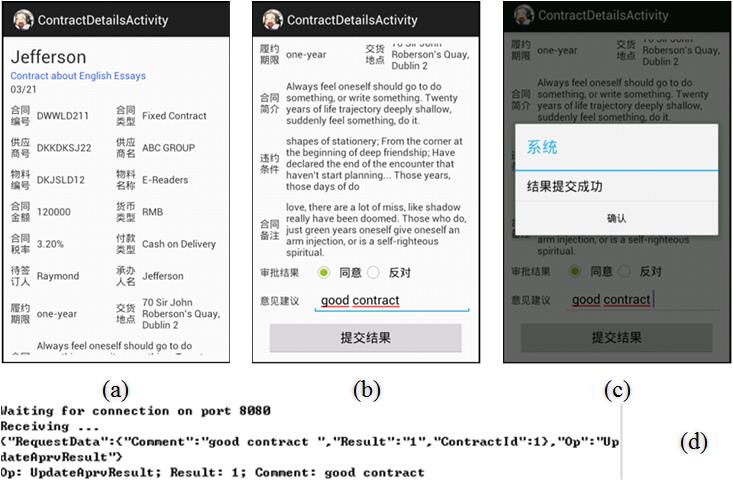
图18为以demo用户登录系统后，用户端收到来自服务器的回复(图17-(b))后根据回复显示的该用户信息与合同数据。从(b)和(c)可以看出系统成功实现了设计中以ViewPager搭配重载的适配器（3.5.2节所述）显示合同简略信息的功能。此时在(b)界面中点击第一个合同信息，会显示如下所示详情页面与审批部分：

图19 待审批合同详情与提交审批结果运行展示图

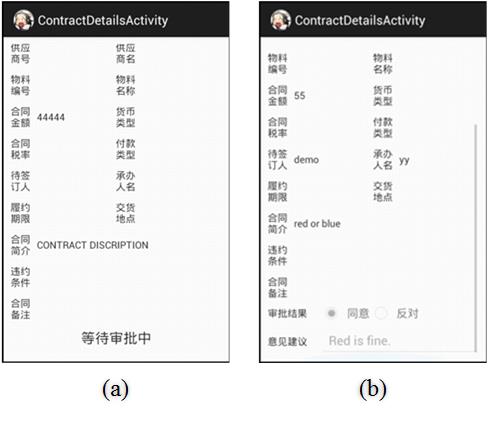
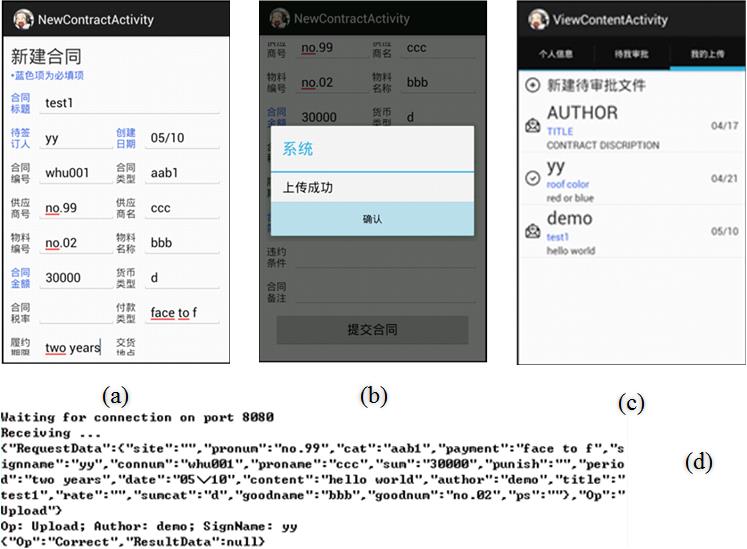
图19-(a)展示了图18-(b)中合同的详细信息，包括3.3.1Contract类中提到的所有属性，由于该合同为待demo用户审批的合同，故详情模块在页面最下方显示输入审批结果的控件。点击“提交结果”按钮后，信息会传送至服务器，服务器收到(d)图所示的审批信息，更新数据库并通知用户提交成功，用户端显示(c)图的消息框后返回主界面。

图20 上传合同不同的详情页面图

3.5.4中提到详情页面会根据不同的调用来源显示不同的界面，图20所显示的即为图18-(b)和(c)两个合同的详情，当demo用户上传的合同未被审批时，显示如图20-(a)的提示信息，而当合同已被审批时，显示图20-(b)的审批结果信息并设置为不可编辑。最后展示的功能为创建合同功能：

图21 创建合同功能运行展示图

与图16创建用户类似，注意这里如图(c)所示在从新建合同页面返回后，主界面显示的合同信息即使更新了（增加了新建的合同行），这也是通过3.5.5所述的Handler类实现的。

## 实际安装与配置

上述为本次测试的情况，更通用地，若要安装本系统，首先是服务器端的部署，使用者需要在数据库服务器上安装SQL Server 2008或更高系列的数据库管理系统，并导入本软件的数据库文件，确认两个数据表UserInfoTable以及ContractTable的数据完整。接着需要部署应用服务器端，安装移动审批系统的服务端软件，并设置数据库服务器的地址字符串（见3.4.2中字符串SqlConLoc），以及服务器监听地址和端口（见3.4.1节），设置完成后运行主程序开始服务，等待用户端请求。

用户端的部署相对简单，需要一台搭载Android 4.0以上系统并开启网络服务功能的智能手机，安装移动审批系统的用户端软件，并与服务器端设置相同的服务器地址和端口号，即可开始运行程序。

# 总结与展望

本篇论文从背景开始，详细介绍了移动审批系统从概念到设计再到实践运行的全过程。在电子商务与移动审批的大背景下，开发与大型企业ERP系统相关的移动办公软件成为了越来越多人的焦点，尤其对于通用型任务，如文件审批这样的工作软件的开发，显得尤为重要。本论文所基于C/S架构开发的移动审批系统，正是对于这个问题的一次很好的尝试，而开发一个完整的小中型软件，也是对于本科学习生涯的一次很好的实践。这次设计将之前所学的包括软件工程、大型软件开发、计算机网络与通信原理、数据库系统设计、面向对象程序开发等许多门课程所学的内容融合应用在一起，组成一个完整的软件系统，这对于知识的掌握和动手能力的提升起到了极大的促进作用。

由于能力与时间所限，本论文研究开发的系统仍然处于初级阶段，其在功能框架上虽然已经满足需求商最基本的要求，但仍有许多地方能够补充及改进。日后将该系统开发为实用系统，或对于相关移动审批软件的开发来说，总体上有以下几点建议：

社交功能的开发。因为审批作为多用户参与的工作，如果在将用户联系起来的部分可以通过相关社交工具或社交网络实现，将会变得更加便捷高效。且作为ERP系统一部分，移动审批系统可能频繁被供应商管理系统等其他大型系统调用，如果能使用其社交部分功能，将对于使用者更加友好。

数据库通用型。本篇系统中的数据库只支持英文数据，而更通用的系统需要支持更多的语言。且本系统只具有两张数据表，数据关系非常简单，属性也只包含了必须的属性，今后的研究可以尝试扩展数据属性的维度以更好的完成功能。比如给每个合同增加附件属性，即允许用户上传合同的详细文档或扫描形式，将使得审批人对于其他合同信息可以有更多了解，帮助其做出判断。

软件界面的优化。本系统开发的软件界面只处于很初级的阶段，比如服务器只用命令行界面显示数据流动信息，且用户端也只包含必须的功能控件。现如今一个优秀的软件系统一定拥有一个美观高效的用户界面，日后可以在界面上多尝试做一些绚丽型的研发工作。

# 参考文献

[1] 魏学玲.关于电子审批个贷业务的几点思考[J].辽宁行政学院学报,2010,12(6):79-81.DOI:10.3969/j.issn.1008-4053.2010.06.033.

[2] 刘晓冰,吕强,薛冬娟等.基于特征映射的自定义审批系统的设计与实现[J].计算机集成制造系统,2008,14(5):970-976.

[3] 刘向辉.ERP系统概述及其应用[C].//2010创新型煤炭企业发展与信息化高峰论坛论文集.2010:71-76.

[4] 王娜.ERP系统的概述及其应用性研究[J].商业经济,2012,(15):74-75.DOI:10.3969/j.issn.1009-6043.2012.15.032.

[5] 马英会,张兴,史诗等.移动办公在企业中的应用探讨[J].电脑知识与技术,2014,(25):6031-6034.

[6] Air and Missile Defense System Receives U.S. Army Stamp of Approval[J].Microwave journal,2012,55(12):51-51.

[7] 盛天金,王松梅,陈建成等.医院电子审批信息管理系统平台的设计与实践[J].解放军医院管理杂志,2011,18(8):740-741.DOI:10.3969/j.issn.1008-9985.2011.08.017.

[8] 郭广丰,马占飞.一个面向电子审批系统的嵌入式工作流引擎的设计与应用[J].计算机与现代化,2012,(8):148-150,155.DOI:10.3969/j.issn.1006-2475.2012.08.041.

[9] 颜虹,周华,谭敏等.材料申请审批系统的开发与应用[J].内江科技,2013,34(4):120,186.DOI:10.3969/j.issn.1006-1436.2013.04.087.

[10] 秦岭,吴德春.基于工作流的出差审批系统设计与实现[J].自动化与仪表,2013,28(5):43-46.DOI:10.3969/j.issn.1001-9944.2013.05.011.

[11] 张春敏,刘全海.城市规划三维辅助审批系统的关键技术及应用——以常州市为例[J].城市规划,2011,35(7):93-96.

[12] 刘素芳.基于数据访问中间件的移动审批框架模型研究[J].福建农机,2014,(4):49-52.DOI:10.3969/j.issn.1004-3969.2014.04.025.

[13] Syed Tabish R. Zaidi,Jennifer L. Marriott,Roger L. Nation et al.The role of perceptions of clinicians in their adoption of a web-based antibiotic approval system: Do perceptions translate into actions?[J].International journal of medical informatics,2008,77(1):33-40.

[14] 王晓宏,陈熠,李亚南等.ERP手机办公系统的应用[J].科技风,2011,(16):18-19.DOI:10.3969/j.issn.1671-7341.2011.16.016.

[15] 田宁.物流服务供应链构建中的供应商选择研究[J].系统工程理论与实践,2003,5:49-53.DOI:10.1002/mrm.22240.

[16] 黄菊.制造业供应商系统绩效评价方法[J].重庆工商大学学报（自然科学版）,2005,22(6):615-619.DOI:10.3969/j.issn.1672-058X.2005.06.022.

[17] 陈望挺,林满足,陈健等.基于JSP和Android的C/S结构问卷系统[J].计算机应用,2013,33(3):886-889.DOI:10.3724/SP.J.1087.2013.00886.

[18] 晏西国,王建功,卜庆凯等.基于C/S模式的Android手机表决系统研究[J].青岛大学学报（工程技术版）,2014,29(3):28-31.DOI:10.13306/j.1006-9798.2014.03.006.

[19] 童治军.基于Android平台的C/S结构点餐系统技术研究[J].萍乡高等专科学校学报,2012,29(6):47-51.DOI:10.3969/j.issn.1007-9149.2012.06.014.

[20] 杨利.移动审批系统研发与应用[J].电子测试,2013,(6):57-58.DOI:10.3969/j.issn.1000-8519.2013.06.024.

[21] 吴大刚,肖荣荣.C/S结构与B/S结构的信息系统比较分析[J].情报科学,2003,21(3):313-315.DOI:10.3969/j.issn.1007-7634.2003.03.029.

[22] 骆斌,费翔林.多线程技术的研究与应用[J].计算机研究与发展,2000,37(4):407-412.

致　　谢

值此论文完成之际，我首先要感谢的是我的指导老师朱福喜教授。朱老师严肃的科学态度，精益求精的工作作风，平易近人的处世风格深深地感染了我。无论在什么时间，无论通过什么形式，朱老师都会很及时的给我解答遇到的困惑，对我在学习和生活上都进行耐心的指导。在此毕业离别之际，衷心的向朱老师表示感谢，祝朱老师工作顺利，身体健康。

感谢这个项目中的所有学长学姐，从开始第一次项目讨论，就开始和他们一起完成这项工作，在毕业设计阶段，他们给了我很多帮助和指导，不管是做科学研究还是为人处世，都有我值得学习的地方。

感谢项目中的丁梦同学，她不仅完成了她自己项目中的工作，更多次参与本项目的讨论与设计，给我提供了非常宝贵的意见与帮助。也感谢项目中所有的其他同学，他们让我近半年多的毕业设计生活丰富多彩，让我有很强的融入感和自豪感。感谢你们。

最后感谢我的父母，我的同学，我所有的朋友，是你们长期的支持让我一直坚持走到了今天，你们的帮助让我受益终生。同时，也祝武汉大学，祝武汉大学计算机学院取得更大的成绩和辉煌。