**江苏科技大学**

**本 科 毕 业 设 计（论文）**

学 院 电信学院\_\_\_\_\_\_\_\_\_\_

专 业 软件工程\_\_\_\_\_\_\_\_\_\_

学生姓名 刘菲炫\_\_\_\_\_\_\_\_\_\_\_\_

班级学号 1345536215\_\_\_\_\_\_\_\_

指导教师 丁玉萍\_\_\_\_\_\_\_\_\_\_\_\_

二零一七年六月

江苏科技大学本科毕业论文

基于Android的OA类产品开发

A product development based on office automation

# 摘 要

现如今企业的数量越来越多，规模也越来越大，其内部的业务流程也就变得更加复杂。譬如，就单单一张审批表，就需要跑到不同的部门，找到相关负责人签字盖章。不过如果运气不好，负责人不在，审批表单将会一拖再拖，严重影响办公效率。因此，很多做OA办公软件公司相继问世，例如用友、金蝶等，这些PC端的办公软件很大程度上省去了不必要的流程。随着移动互联网概念逐渐被大众所接受，阿里旗下的钉钉也逐渐崭露头角。然而这毕竟是第三方的平台，公司的数据得不到有效的保证，所以，我设计这一款为公司量身打造的移动端OA，将数据实时掌握在自己手中!

该系统主要分为：聊天通讯、公司组织架构搭建、审批、签到等。前端使用Android Studio 2.2 开发，后台使用MyEclipse 开发。

管理者实用该App可以构建公司的等级架构，公司下面设置子部门，子部门下面仍然可以设置子部门，无限制的允许管理员设置下去，并且可以设置部门的领导，方便审批的时候确定审批人。用户可以和用户进行聊天，并且可以创建讨论组进行多人聊天。取消传统纸质审批，在该App中，通过选择不同的审批类型进行填写详细内容，系统会自动选择该部门的领导以及父领导作为审批人。管理员设置好签到信息后，用户则可以进行签到，系统会自动判断签到是否成功。

OA系统逻辑复杂，需要与公司的实际业务逻辑进行结合，然后将复杂的业务流程化繁为简正是该系统的一大特色之一。无论使用者在哪里，只需在手机上进行触控，就能实现办公，极大的方便决策者使用，提高办公效率。服务器搭建在自己公司内部，牢牢将数据掌握在自己手中，在一定程度上避免的数据的泄露。充分将移动互联网这个概念实例化，为中国互联网贡献自己一份薄利！

**关键字**：Android；Java；OA；移动互联网；及时通讯；

# Abstract

Now the number of enterprises more and more, the scale is also growing, its internal business processes will become more complex.For example, on a single Approval form, you need to go to different departments, find the relevant person in charge of signature and seal. But if the bad luck, the person in charge is not, the Approval form will be delayed, seriously affecting the efficiency of office. Therefore, many OA class office software have come out, the famous, UF, Kingdee, etc., these PC-side office software to a large extent eliminates the unnecessary process. However, the concept of mobile Internet gradually accepted by the public, Ali's nail is also gradually cut a striking figure. And then this is, after all, a third party platform, the company's data can not be effectively guaranteed, so I designed this one tailored for the company's mobile side OA, the data in real time in their own hands!

The system is divided into: chat communication, the company structure to build, Approval, sign and so on. Front end using Android Studio 2.2 development, the background using MyEclipse development.

The administrator can use the App can build the company's hierarchical structure, the company below the sub-sector, sub-department can still set sub-department below, unlimited to allow administrators to set down, and can set the department's leadership to facilitate the approval of the time to determine the approval The Users can chat with the user, and can create discussion groups for more than chat. Cancel the traditional paper approval, in the App, by selecting a different type of approval to fill in the details, the system will automatically select the leadership of the department and the parent leadership as the approver. The administrator set up a good sign information, the user can sign, the system will automatically determine whether the success of the sign.

OA system logic is complex, need to be combined with the company's actual business logic, and then complex business processes into the simplicity of the system is one of the major features. No matter where the user, just touch on the phone, you can achieve office, a great convenience to decision makers to use, improve office efficiency. The server built in their own company, firmly in the hands of the data in their own hands, to some extent to avoid the leakage of data. Fully the concept of mobile Internet instantiation of the Chinese Internet to contribute their own a small profit!

.

**Keywords:** J2EE; Spring; Mybatis; Ajax; 4SShop

**目　录**

[**第一章 绪论** 1](#_Toc483170535)

[1.1 研究背景和意义 1](#_Toc483170536)

[1.2 国内外研究现状 1](#_Toc483170537)

[1.3 本文的主要研究内容 2](#_Toc483170538)

[**第二章 关键技术简介** 3](#_Toc483170539)

[2.1 Android 简介 3](#_Toc483170540)

[2.2 Android 四大组件 3](#_Toc483170541)

[2.3 View 6](#_Toc483170542)

[2.4 动画 8](#_Toc483170543)

[**第三章 系统总体分析** 10](#_Toc483170544)

[3.1项目开发可行性分析 10](#_Toc483170545)

[3.1.1 项目经济可行性分析 10](#_Toc483170546)

[3.1.2 项目技术可行性分析 10](#_Toc483170547)

[3.2项目功能需求分析 10](#_Toc483170548)

[3.2.1 登录注册模块 11](#_Toc483170549)

[3.2.2 个人中心功能模块 11](#_Toc483170550)

[3.2.3 聊天模块 12](#_Toc483170551)

[3.2.4 审批模块 12](#_Toc483170552)

[3.2.5团队管理模块 13](#_Toc483170553)

[3.2.6签到模块 13](#_Toc483170554)

[**第四章 程序架构设计** 15](#_Toc483170555)

[4.1 用户界面设计 15](#_Toc483170556)

[4.1.1 登录/注册界面 15](#_Toc483170557)

[4.1.2 主界面 15](#_Toc483170558)

[4.1.3 团队管理界面 17](#_Toc483170559)

[4.1.4审批 18](#_Toc483170560)

[4.1.5 签到界面 18](#_Toc483170561)

[4.2 数据库设计 19](#_Toc483170562)

[4.3 接口设计 22](#_Toc483170563)

[4.3.1 登录 22](#_Toc483172930)

[4.3.2 注册 22](#_Toc483170564)

[4.3.3 获取公司信息 22](#_Toc483170565)

[4.3.4 获取用户所在部门信息 22](#_Toc483170566)

[4.3.5 将签到的设置存到数据库 23](#_Toc483170567)

[4.3.6添加好友/添加子部门，修改部门信息 23](#_Toc483170568)

[**第五章 程序代码开发** 24](#_Toc483170569)

[5.1 Android 端代码分析 24](#_Toc483170570)

[5.2 服务器端代码分析 28](#_Toc483170571)

[**结 语** 30](#_Toc483170572)

[**致 谢** 31](#_Toc483170573)

[**参考文献** 32](#_Toc483170574)

# 第一章 绪论

## 1.1 研究背景和意义

OA是指办公自动化，将传统的线下办公改进为以先进的网络互联为基础,共享资源,交流信息,极大的提高工作效率,减少沟通成本,降低劳动强度。为了更加方便办公，将办公移动化一直被企业所追求，于是我设计这种移动办公化的软件-基于Android的OA类的产品开发。

该产品是和企业的业务紧密结合，如企业内部沟通，审批，查询，签到等传统的线下功能全部移动化，用户只需轻轻一点，就可得到想要的结果。对于员工来说，省去层层递交材料的不必要时间，快捷。对于企业领导者来说，不论自己身在何方，拿起手机便可实现异地审批，办公更加高效，极大的方便企业领导者的管理和决策。

## 1.2 国内外研究现状

OA软件的产品发展可以说随着个人电脑的产生而产生，一共经历的三个发展阶段。

第一代：个人电脑的出现和应用。这个时候电脑并不是很普及，所以这个阶段发展最为艰难。这个阶段金山软件WPS等类似的办公文件系统取代传统的纸质操作，在很大的程度上提高了办公效率。

第二代：个人电脑的大量普及。联网技术的成熟与普及Internet和Intranet的快速发展。除办公软件外，IBM公司Lotus Domino和微软公司Exchange 群件技术的引入。这种优点是第二代是以网络为中心，或工作流为主要存储和处理对象，以非结构化数据的信息流、以文档数据库管理、E-mail、复制、群组协同工作目录服务、群组协同工作等技术作支撑，方便群体操作，彻底打破了第一代办公OA的”信息孤岛”，是真正意义的”办公自动化”的开始。缺点是难以实现、移动的办公，随时随地的办公，所以不易于实现企业资源的延展，企业资源不能得到最大利用。系统开发和操作投资昂贵、使用复杂，无法得到充分的推广。

第三代：与真实工作环境结合，前两代都是以文件为基础，而第三代系统不仅模拟和实现了工作流的自动化，更模拟和实现了工作流中每一个单元和每一个工作人员运用知识的过程。在此期间，国内众多专注OA软件系统崛起，其中已用友、金蝶软件已经很成熟。而在此期间移动端的产品市场上不是很多，因此，接下来OA将会进入第四代，办公移动化。目前主流OA仍然处于第三代PC端的使用。从阿里的钉钉在市场上愈发火热，然而毕竟是第三方的软件，在信息安全上存在或多或少的泄露隐患。所以我开发的这款产品旨在为公司定制软件，一切的信息由自己掌控。

## 1.3 本文的主要研究内容

针对企业的实际需求，共有一下几点研究内容：

聊天。为了方便企业进行内部上的沟通，打开软件，找到相应的公司成员即可实现聊天，查看对方基本信息。也可以加入多人讨论。

团队管理。为了对公司整个架构的搭建，管理员通过团队管理设置部门，部门负责人，移动该部门等操作。界面简洁，操作简单

签到打卡。管理员根据公司的实际情况，可以设置签到打卡时间，允许设定签到位置，签到弹性时间。用户签到根据当前定位位置，允许微调位置，不再签到允许范围内，签到视为失败。管理员还可以查看所有员工的打卡签到情况。

审批。系统会根据公司实际的报销单，来设计审批表，以及审批层级领导。用户填完审批信息，根据审批类型自动将信息传达给下一个审批者，直至最后一个审批者完成审批。审批者可以查看当前的审批进度，取消审批单等多种操作。

# 

# 第二章 关键技术简介

## 2.1 Android 简介

Android 系统是一款基于Linux内核的移动端操作系统，由Google公司开发(被Google收购)，目前支持手机、平板、手表、电视等，未来可能会大面积的应用于智能家居。截至目前Android 7.1 nougat 是最新的版本。并且根据最新的数据显示，Android 以0.02%的微弱优势超越Windows成为全球占有率第一的操作系统，而在五年前，Android 仅占全球2.4%的使用，可以说这是Android 的里程碑。

移动端Android 一直与IOS相抗衡，尽管Android 的市场份额超过IOS，但是由于底层的设计的不同导致Android 在性能方面不及IOS。并且随着HTML5的发展，Android原生应用与H5的结合已经越来越多。React Native 就是一个JavaScript的框架，可以无缝的运行在IOS和Android 端，目前已经被越来越多的公司使用，原生Android 开发也随之减少。

## 2.2 Android 四大组件

**2.2.1 Activity**

Activity是负责界面的展示，负责与用户进行直接的交互。是四大组件中唯一可以被用户感知的。几乎我们App中每个页面都对应一个Activity，界面的呈现、点击事件的监听、数据的获取等都是在Activity中进行操作。只有了解其生命周期，我们才能在应用程序中更好的完成复杂的操作。

onCreate(): 周期的开始。在此期间段我们需要让Activity 进行布局的填充，setContentView(R.layout.Activity\_main)指明我们让该页面显示Activity\_main这个xml布局。同时还需要对控件和资源文件进行初始化操作。

onStart:此时界面处于可见状态，但是并没有在前台显示，所以无法与用户进行交互操作。

onResume:此时界面不仅处于可见状态，还可以与用户进行操作。一般情况下如果该页面切换到另一个页面在切换回来，此方法就会被调用。因此我们可以在该方法中进行资源的再次初始化。比如，为了数据的实时性，当页面切换回来时，我们需要刷新页面，就可以在该方法内进行网络的加载，从而达到数据刷新的效果。

onPause:该方法会在页面我们依然能看见，但是用户不可以进行操作，比如，该页面突然弹出一个对话框，此时我们并不能对该页面进行操作，但是依然可以看见。因此，在这个方法中我们可以对资源进行回收，因为Android 内存有限，如果编码规范不注意的会引起资源的泄露，导致程序异常退出。所以我们需要对一些不必要的资源进行回收。

onStop:此时页面已经看不见了，仅在后台运行，该方法紧接上面方法执行。

onRestart:该页面由不可见变为可见状态，调用场景：当我们使用App时从我们的界面切换到另一个App界面，并在切换回来时，该方法就会调用。

onDestory:一旦该方法被调用，那么该Activity的生命周期就会结束，页面销毁，在该方法中我们一般处理一些回收工作。

谈及Activity就不能不提其启动模式。因为在Android 中，我们需要任务栈Task来放置Activity实例，其实现原理就是就是以栈的形式存储，默认是先进后出。例如用户按下返回键，当前的Activity就会从栈顶弹出，而在它下面的Activity就会显示出来。 系统给我们提供了四种启动模式：standard、singleTop、singleTask、singleInstance。

standard:每次启动一个新的Activity，该Activity就会被压入栈中，并处于栈顶。如果在该页面再次请求跳转该Activity，那么又会有一个新的实例压入栈中，此时栈中会存在两个Activity。当我们点击返回按钮时候，会发现得到新的界面和就是刚才的界面，再次按返回键，当前页面才会消失。所以standard会造成多个Activity实例存在。

singleTop:这种模式下，可以说是栈顶复用模式，也就是一个新的Activity准备入栈，发现之前同样一个Activity已经存在并且已经位于栈顶，那么系统就会复用已经存在栈顶的Activity，所以此况下只会存在一个实例。

singleTask:栈内复用模式。此模式下该Activity只会存在一次，并且每次跳转到该Activity,就会把在它上面所有其他Activity清除。使用场景：每个App都有主界面，在App的任何一个地方跳转主界面，都应该把主界面以上的Activity都清除，并且让主界面所在的Activity位于栈顶，只有这样我们按返回键才会回到桌面。

singleInstance:单一实例模式，这种模式下的Activity存在一个独立的任务栈中，并且该任务栈只存在它一个实例，不同应用打开都是共享该Activity。使用情景：呼叫来电界面。

只有了解Activity的生命周期和启动模式，在写程序的时候才会得心应手。

**2.2.2 Service**

Service运行在后台，不像Activity拥有自己的界面，Service并不提供用户界面。并且一旦Service被开启，除非系统因为内存不足回收资源，否则系统不会停止或者销毁Service。因为服务一直在后台运行，所以我们可以执行一些比较耗时的任务，比如，文件下载，音乐的播放等。然后Service也是在主线程中，所以如果在主线程中执行耗时操作，就会出现ANR(Android 系统规定，如果程序操作界面，界面在五秒钟之内没有反应，就会报异常)。所以一定要重新开启新线程执行耗时操作。

**2.2.3 BroadcastReceiver**

广播(Broadcast)是组件之间传播数据的之中机制，可以在不同的进程中进行通信。BroadcastReceiver是对广播出来的信息就行过滤，拦截的组件，这种信息封装在Intent这个对象的Action和category中。所有已注册的广播都会都会检查注册时的IntentFilter是否与发送的Intent相匹配，若匹配则会效用其onReceiver方法。BroadcastReceiver的生命周期很短，从onReceiver执行开始，到其内部方法调用结束。往往我们在设计过程中，采用Service+BroadcastReceiver这种方式，Service因为其生命周期比较长，所以可以定时的去监听某个事件，一旦接收成功，通过BroadcastReceiver去通知主线程更新UI。在Android 系统中，只有主线程才能更新UI操作，子线程执行更新UI操作，会抛出CalledFromWrongThreadException异常。

**2.2.4 ContentProvider**

内容提供者(ContentProvider)相对于其他三大组件，使用的频率远远没有它们高，它的的功能主要为了程序之间数据共享。他也有类似数据库的CRUD(新增，读取，更新，删除)操作，只不过它是用来管理手机中的音频，图片库，联系人等数据。基本上每个程序都会调用本地，就是通过ContentProvider的query方法查询。重写ContentProvider的方法时，每个方法都是有Uri类型的参数。Uri代表了要操作的数据，Uri主要包含两个信息：1、需要操作的ContentProvider 2、对ContentProvider中什么数据进行操作。

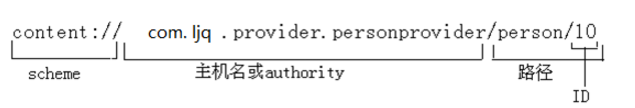


图2-1：Uri结构

ContentProvider的scheme 可以是http、file、content，如果没有指定它，那么其他URI属性会无效。

其次中间部分，是唯一的标识一个特定的ContentProvider,该部分是其所在的package来命名，因为App的package是唯一的。

路径表示的是请求的资源类型，这部分可以是可选的，ID是资源id,表示请求的是一个特定的资源，这部分是由数字构成。忽略这两部分，请求的有可能是某种资源下的所有实例，相当于目录。

总结：四大组件实例之前必须需要在AndroidManifest.xml中声明如图2-2所示。每个组件都有其相应的生命周期，我们必须通过了解其生命周期，才能在相应的回掉方法中处理逻辑。

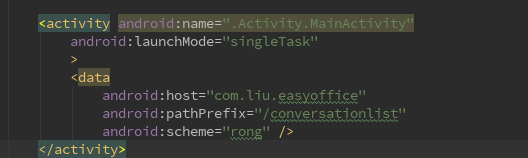


图2-2： Activity的注册

## 2.3 View

View可以作为Android 的第五大组件，其作用并不比四大组件少。Android 布局中的界面都是直接或者间接的继承自View,常见的Button，EditText，TextView，ImageView等都是View的子类。界面排布、动画效果都是通过View来设置，然后大部分情况下系统所提供的控件并不能满足效果，这种情况下就需要开发者自己自定义View。

理解View的绘制流程是对于自定View有着重要的作用。其绘制流程分为三步骤,Measure测量、layout放置、draw画。在Measure中控件会不断的测量它自己，以及它的子View，以此来确定子View需要的宽度和高度。Layout就负责将View摆放在界面中何处位置。最终Draw则负责将View绘制出来。

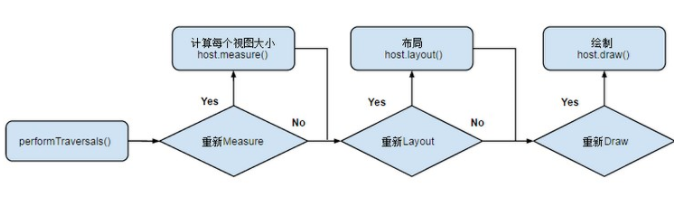


图2-3 ：View的绘制流程

然后自定义View中我们只能通过继承View或者ViewGroup来重写onMeasure、onLayout、onDraw方法，这三者均在其父类中上面三种方法中调用。这就是回调函数，是Java特种的一中机制。父类不知道具体该怎么实现，自己空实现，让代码中其他地方调用这个方法。子类继承过后就会可以重写该方法，做具体的实现。这样当代码执行下去的时候，就会调用子类实现的方法。

View的事件分发机制，在实际应用中也非常重要。由于View是由一个个控件组成的界面并且是一层层嵌套，必然在触摸过程中需要得到响应。具体哪些控件要有响应，哪些控件不需要响应，这就需要通过事件的分发来决定是否处理。图4是View的结构，而事件分发也正是由这种结构由从上向下逐渐分发。首先事件从Activity开始分发，会调用Activity的dispatchTouchEvent方法向下分发到ViewGroup,同样ViewGroup的dispatchToucheEvent方法也会调用，此时如果返回false则意味ViewGroup不允许往下分发，那么Activity的onToucEvnet将会调用（交上级消费）。而ViewGroup自己会有个onInterceptTouchEvnet方法，负责拦截该事件，如果发回true,则交自己的onTouchEvent处理，否则的继续往下传递，最后会分发到最后一个View,而View不像ViewGroup,尽管ViewGroup是View的子类，然后View却不像ViewGroup那样，View自己没有拦截事件的方法，它自己只能选择是否分发，也就是同样调用dispachTouchEvent方法，返回false，则将事件交给其父布局进行消费，否则的调用自己的onTouchEvnet方法。这一系列就是View的事件分发。每一层都可以选择是否消费当前事件。实际开发过程中我们经常能遇到用事件分发解决的问题，在滚动的页面（ScrollView）嵌套一个滚动的页面（ListView或者GridView）,如何滚动ListView时候，不去触碰ScrollView。这就是需要ListView自己消费一系列的触碰事件。

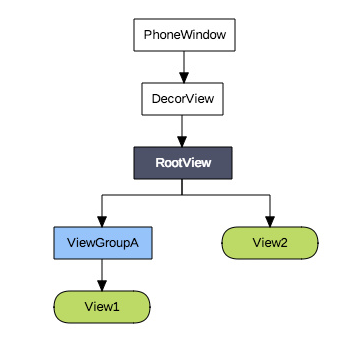


图2-4：View的结构

## 2.4 动画

一款炫酷的App，通常都是由里面的动画效果支撑。App要做到与用户良好交互，就不能让界面看起来死板，多点动画操作，加强用户对整体设计的好感。自从Android 5.0以来，Google调整Android整体布局，界面更加优美，而且通过多样的动画效果，让这个系统更加炫酷。

Android基本上的可以分成三种类型的动画，View动画和属性动画还有一个逐帧动画。View动画结合上面讲述的各种各样的View控件做TranslateAnimation (平移动画)、ScaleAnimation(缩放动画)、RotateAnmation(旋转动画)、AlphaAnmation(透明变化)，这四种动画均可指定执行的时间，以及速度。并且四者可结合起来，先后执行也可同时触发。逐帧动画则是通过一张张静态图片，连接成一张动态的效果。类似gif图片。比如App中常用的加载中的动画，往往是通过逐帧动画拼接而成。这种方式严重影响性能，因为需要加载多张图片，加大App的内存占有量，所以这种技术比较落后。属性动画基本上结合了上面两种的优点，打造出几乎无所不能的动画框架。属性动画改变的是属性！上面的View动画，通过平移改变的只是在界面上重新绘制而已，如果想要对它进行点击操作，就会发现根本不会其作用，因为其本身还是在原来的位置，只是界面上把他绘制在其他地方而已。而通过属性动画，做平移操作，就可以解决View动画存在的问题，因为属性动画，改变的不仅仅是视图，还有它的属性。所以属性动画是对所有对象都起作用。其中最常用的两个属性动画类ValueAnimator和ObjectAnimator，都可以将多个动画结合起来实现各种样式的效果。

动画与数学结合对于可以实现更复杂的效果，贝塞尔曲线由起点、多个控制点和终点共同决定的曲线。其路径就是我们数学中的函数连起来的点构成的，其中t代表时间，P0是起始点，Pi是控制点，Pn是终点。一阶贝塞尔曲其实就是一次函数，其路径就是一条直线。二阶贝塞尔曲线是一个二次函数，其路径就是一个抛物线。三阶届贝塞尔曲线则是三次函数，以此类推还有高阶贝塞尔曲线，其函数就更加复杂。都是需要多个控制点去控制曲线的走向，所以其效果将会越来越复杂。基本上项目中如果不会由太复杂的动画，前三阶足以够用。

# 第三章 系统总体分析

## 3.1项目开发可行性分析

### 3.1.1 项目经济可行性分析

软件这种虚拟化的产品，和其他产品最大的不同之处在于，软件并不需要昂贵的费用。因为虚拟化，只需要给我一能够搭载运行它的机器，就可以运行。它的生成只需要在相应的编译软件上进行编译即可生成apk，然后打包运行在手机上。该系统在Android Studio 2.1进行开发，Android Studio 是一款为Android 平台开发编译软件，并且供开发者免费使用。所以开发过程中并不需要消耗任何费用。要是支持软件运行，必须安装在Android 平台上，可以是手机、平板等。如今一款Android 机器三百块钱就可以买到一款不错的，并且大部分人都在使用Android 手机，所以最多会产生三百元的费用。因为App需要在有网络的情况下进行操作，所以我需要将服务器代码，和数据库相关的，放置腾讯云服务器上，然而腾讯云这种服务器对学生有更好的优惠政策，只需一元便可以搭建完成。其他费用基本上不会产生，所以在经济上，该系统完成被人们所接受。

### 3.1.2 项目技术可行性分析

该系统主要分为Android 端和服务器端。Android端目前仍在采用Java的相关技术，而在2017年的谷歌IO大会上，谷歌终于推出了自己的Android 语言kotlin，不过该系统仍然采用Java语言。Java 语言是比较简单的，通常跟着视屏教学，或者学校的课程也都由教过，Android 主要代码就是通过Java去构建，只要在稍微学习 Android 的一些相关知识，便能很快上手。而服务器也是通过Java去编写，通过JSP对数据进行处理。数据库采用的MySql，基本的增删改查足以可以用来处理数据，只不过复杂一点就要通过事务去处理，技术上都是大学课堂学过的知识。及时有些地方会淡忘，在这个互联网高速发展的时代，技术上实现不了，还可以通过参考网上的一些素材。所以对于该系统，技术完全可以应用。

## 3.2项目功能需求分析

在需求分析设计上，已经将所有可能存在的模块并且比较重要的模块涉及到了。登录/注册模块，个人中心界面、聊天模块、审批模块、签到模块、成员管理模块这六大模块，然后这五大模块下面又会有众多的小模块，涉及面非常广。

### 3.2.1 登录注册模块

（1）注册

使用本产品查看相关功能，得需要进行注册，必须以用户身份进入，否则是没有权限。本款系统需要用户提供真实的手机号，系统会给其手机发送验证码，然后填入符合本产品密码要求的密码，其中任意一项有误，系统都会拒绝操作。

（2）登录

用户必须使用已注册过的账号进行登录操作，填入正确的手机号和密码即可登录。不过为了更加人性化，如果用户忘记密码，我们也会提供忘记密码功能，会像用户之前注册过的手机发送验证码，重新填写密码就可以继续操作了。

### 3.2.2 个人中心功能模块

（1）创建公司/创建部门

创建公司是在该成员并未加入任何公司的情况下才被允许，如果该成员如果已经加入某公司，则提示不允许在创建。因为该系统是针对某公司指定的App,所以只允许一个人在一个公司。创建部门则是建立在创建好公司的基础上进行创建，管理员可以无限制的创建部门。并在该部门下可以添加用户成员。

（2）查看我的团队

用户可以在个人中心界面更加快速的查看他所在团队的信息，该快捷入口将会引导用户直接进入所在的部门下面。查看成员信息。

（3）设置

用户在设置中可以从本地相册或者拍照上传自己的头像，更改自己的姓名，出生日期，以及出生地等。如果没有头像的话，系统也会又一个默认头像，避免了用户不设置头像空白的情景。当然每个信息都会又默认值，用户自己可以决定要不要去修改相关信息。

（4）意见反馈

为了方便用户联系我们，我们开通意见反馈通道允许用户，对于产品的任何功能给予最真实的反馈。这样我们就会根据收集上来的反馈意见对产品进行相应的修改。

### 3.2.3 聊天模块

（1）聊天界面

该产品允许用户发语音，发文字、发图片，更方便办公。而且允许用户之间私聊，创建讨论组。在讨论组中如果有新用户进来则会提示其他成员，并且讨论组的名称如果改变则也会又提示，这样体现信息的及时共享性。

（2）讨论组列表

为了让用户能够能够在第一时间区分讨论组，从讨论组名称上区分不是一个好的办法，所以该系统设计了讨论组的头像，该头像是由组内成员头像合成，成员少于六人头像按照相应规则全部显示，超过六人，则只显示6人头像。让界面看起来更加美观，用户所有加入的讨论组均会在这个页面显示，点击相应讨论组即可聊天。

（3）讨论组设置

聊天界面左上角都会一标志，如果是单聊的情况下，点击左上角则会显示用户的基本信息界面，如果是讨论组，则会显示讨论组的设置，讨论组的每个人成员均可以更改讨论组名称，添加新成员，退出改讨论组等操作。

### 3.2.4 审批模块

（1）四种审批功能

审批中有四种常规的审批类型，根据公司的实际应用场景，有请假审批、报销、出差、补签四大主要审批类型。请假审批需要用户请假者填入请假类型，请假时间，请假天数，和请假事由，用户也可以添加最多九张图片可以作为请假依据。并且系统会自动设置审批人，发布之后相关审批人就会收到信息，领导层层审批。报销、出差、补签对应相关功能，只是审核人不一样而已。

（2）查看我的申请

用户可以在我的申请页面查看我所有申请的历史记录，并且可以分为尚未结束和已结束两部分，点击每条信息列表跳转到相应的审批单中，查看当前进度，进度会以时间条的顺序排序，不同状态均会在页面上呈现出来。并且还增加搜索条目，用户可以搜索申请相关信息，便可以直接查到。用户如果觉得自己的申请突然没必要了，在审批没有完成之前，可以撤销。

（3）查看我的审批

用户既可以是审批发起人也可以审批人，所以当用查看审批但是，就会列出他待他审批的。用户查看审批，对当前的审批进行状态的改变，同意或者拒绝，如果拒绝的话，填写审批意见，审批单就会被退回，就算该审批单结束。如果同意的话，审批则会继续往下执行。传递给下一个审批人，直至最后一个同意审批。

### 3.2.5团队管理模块

团队管理决定了该公司的整体架构，用户可以在界面直接查看公司的架构，了解功能各个部门之间的关系，以及部门的成员，方便查看信息。该架构由管理员设置，其他成员无权设置。该模块是在上面公司成功创建之后才能生效。

（1）添加子部门

进入该页面会显示该部门下的子部门列表和成员部门列表。管理员可以选择添加子部门，在添加子部门页面设置部门名称，部门上级部门（默认是之前的部门），可以添加成员。

（2）设置部门

有时部门信息需要及时的更新，所以用户可以重新设置部门，包括部门名称，部门上级部门（不能选择自己的子部门，可以选择同级部门，或者以上），和部门中的成员调整。

（3）修改成员信息

管理员有权限对每个用户进行设置，包括更改成员所属的部门，和一些其他属性等。

### 3.2.6签到模块

（1） 管理员对签到的设置

员工签到需要明确签到的地点，签到的弹性时间等一些基本设置，所以我们给管理员一个界面，需要管理对签到的基本信息进行设置。

① 考勤时间：管理员需要设置每周的上班时间，这里会添加一个选项法定节假日自动排休，如果勾选了，则按照国家法定节假日来排班。

② 添加办公地点：改办公地点是通过地图自动定位到当年位置，并允许设置者微调当前的位置。设置成功之后以后员工签到均会按照这个位置，并且与之后设置的有效范围结合，共同决定员工能否签到成功。

③ 设置上班弹性时间：为了人性话的设置，管理员可以添加弹性时间，允许员工上班时间后这个范围之内签到，都算打卡成功。

（2）员工打卡签到

员工签到的地点是定位的，并且可以在这个基础上进行小范围内的微调，但是不能为了防止作弊不能让员工大范围选择签到地点。如果符合管理设置的那些条件，则显示签到成功否则签到失败。

（3）统计页面

在该模块中，可以查看团队考勤的一些情况，对签到情况会进行排序，根据这些数据公司可以对员工的考勤有了一定的了解。

# 

# 第四章 程序架构设计

## 4.1 用户界面设计

该系统设计到模块比较多，所以用户界面也是比较多，大概有三十个页面。主页面之中会将六大模块全部展现出来，而每个模块又会根据功能在划分不同的页面。整体界面风格呈现蓝白调，控件设计也拒绝使用默认样式，充分结合Google自从Android 5.0以后推荐的mertirial design设计风格，使得界面清爽优美

### 4.1.1 登录/注册界面

登录界面其实包括登录、忘记密码、注册三个页面。用户点击不同的操作，该界面仍然存在，只是结合了入场出场的动画，让界面中的布局来回切换。拒绝了传统的静态背景转而使用动态背景。让界面变得颇为生动。



图4-1：登录、忘记密码、注册三个页面

### 4.1.2 主界面

主界面分为四个界面，其中三个页面消息、工作、联系人通过点击下方三个按钮进行切换，另一个页面需要点击左上角菜单按钮，第四个页面会以抽屉样式在页面一半的位置显示。通过抽屉的方式有效的避免了界面内容的冗长。

消息界面展示消息列表，如果没有消息则显示一个默认态。

第二个tab工作页面就是列出了工作时所需要的功能，签到、公告、审批、团队管理。点击它们每一块，都代表一个新的界面。最上方是广告的轮播区域，用户可以手动滑动，或者定时滑动轮播图，该区域可以为以后介入广告或者重要公告提供服务。

联系人页面展示用户所在的群组和组织架构。通过群组按钮可以让用户打开所在的讨论组，而下面的组织架构则会让用户跳转到所在公司的架构界面，其下面是用户所在的部门。

右边的类似抽屉的用户相关信息界面，分为三部分，上面展示用户的基本信息，中间的部分是基本操作的快捷入口方便用户直接调用，底部是设置可以让用户调转到个人中心界面，对自己的基本信息进行设置。

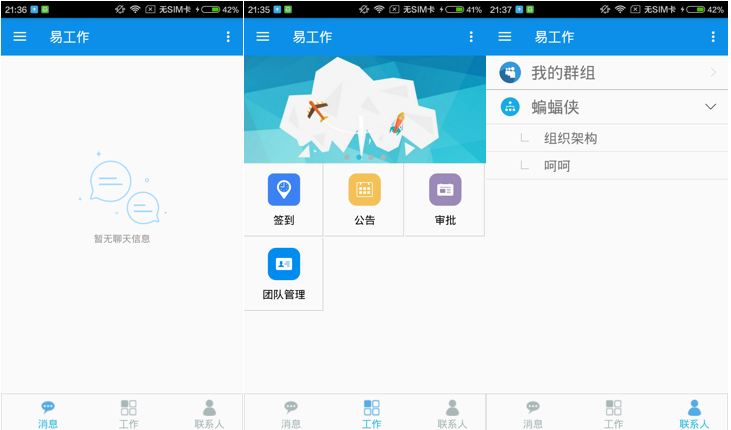


图4-2：消息、工作、联系人界面

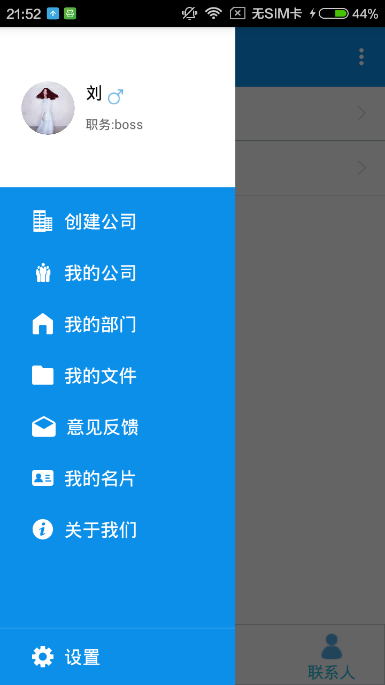


图4-3：左侧个人中心页面

### 4.1.3 团队管理界面

团队管理分为四个部分上面部分表示当前的部门，紧接着的是该部门下的子部门，然后是该部门下的成员，最底下的是设置按钮。

点击添加成员按钮则会进入到搜索联系人界面，根据输入手机号，列表会显示类似的账号并且如果该账号已经在其他的某个部门，则会显示出来，并不会让用户添加。

点击添加子部门，则会进入添加子部门界面填写部门名称，选择上级部门等一些操作，点击完成之后才会请求服务器。



图4-4：组织架构、添加员工、设置部门三个界面

### 4.1.4审批

审批页面中上半部分分为我的申请和我审批的，下面是审批的三种类型，请假、报销、出差。这三种类型界面都差不多都是填一些数据或者选择数据。在我申请的页面中用户可以查看审批的所有形式。该页面也提供了搜索一栏，方便用户找到相应审批单。从列表中可以看出，不同的状态对应的样式也有所不同。点击其中一个条目，进入到查看进度页面上半部分显示申请的基本信息。并且刚进入页面“已通过”标志则会以一种盖章的动画效果呈现出来。底部则是进度的展示，在这里用户可以查看当年审批状态。

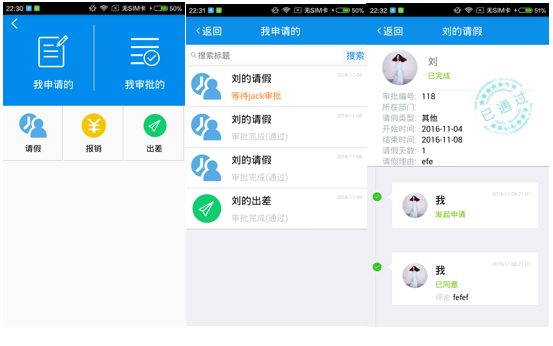


图4-5：审批主页面、我申请的页面、审批详细页面

### 4.1.5 签到界面

签到界面是由同主界面一样包含三个页面。打卡、统计、设置。打卡页面展示的是定位到当前的位置，并在地图上显示出来，下方同样会把当前位置以文字的形式展现出来。最下面是一个自定义View的控件，这样的设计显得更加显目。统计页面会将所有成员的打卡情况进行统计，会以环形比例图，显示打卡与未打卡所占的比例。第三个页面设置页面是让管理员设计相关签到的需求，管理员需要手动设置设置考勤时间、添加办公地点、弹性时间等。



图4-6：打开、统计、设置页面

## 4.2 数据库设计

该系统是根据业务进行模块的划分，所以在数据库的设计上根据模块进行设计。个人基本信息模块包括（用户表、权限表、角色表）、团队管理（公司表、部门表）、聊天模块（讨论组表）、审批模块（审批信息表、审批类型表、审批状态表、请假表、出差表、审批意见表，审批流程表）、签到模块（签到记录表、签到规定表）。其中个人信息模块中和团队各表之间又相互联系（用户权限表、用户角色表、权限角色表、用户部门表等）。

用户表：字段id、姓名、密码、手机号、token、头像、地址等。

权限表：字段id、父权限、权限名称、权限描述

角色表：字段id、父角色、角色名称、角色描述

角色权限表：字段id、权限id、角色id

用户角色表：字段id、用户id、角色id

公司表：字段id、公司名称、公司名称、创建时间、公司类似、公司地点

部门表：字段id、部门名称、父部门id、部门描述、公司id、部门领导。

用户部门表：字段id、用户表、部门id。

讨论组表：讨论组id、讨论组名称、创建时间

用户讨论组表：字段id、用户id、讨论组id

审批表：字段id、审批标题、审批类型、审批创建时间、审批状态、待审批人id

审批类型表：字段id、审批类型描述

申请所有要填写的类型表：字段id、审批类型id、填写的名称、填写的描述、字段类型。

审批详情表：字段id、申请事由、请假类型（出差地点）、扩展1、扩展2、扩展3、请假天数、开始时间、结束时间。

审批状态表：字段id、状态名称。

审批结果表：字段id、结果类型。

审批意见表：字段id、意见描述。

请假表：字段id、请假类型、开始时间、结束时间、请假天数、请假事由、图片id。

签到设置表：字段id、公司id、上班时间、下班时间、经度、纬度、公司地点、签到有效范围、上班打卡最早时间、一周工作时间。

签到表:字段id、用户id、公司id、上班打开时间、下班打卡时间、当天日期、打卡状态。

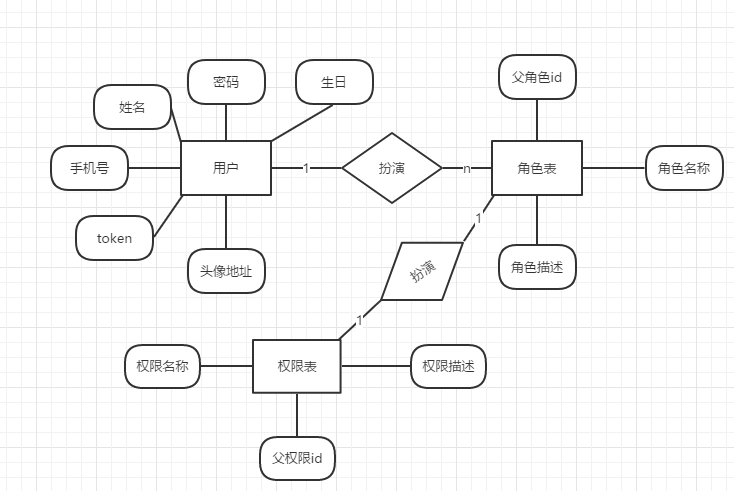


图4-7：用户、角色、权限表E-R图

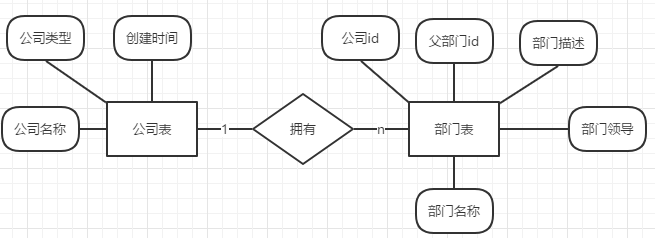


图4-8：公司表、部门表E-R

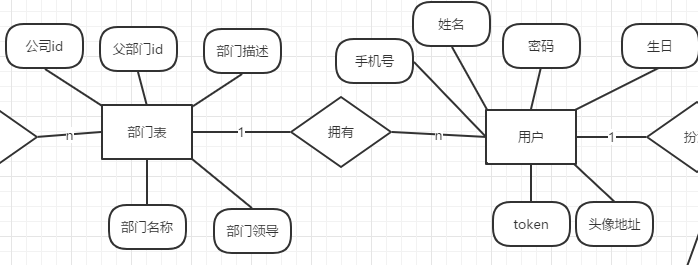


图4-9：部门表、用户表E-R图

## 4.3 接口设计

该系统需要在联网的情况，将客户端的数据传送到服务器端，服务器接收数据进行数据的处理并返回客户端想要的结果。我通过将数据以POST或者GET的请求方式，将数据以参数的形式传到服务器，服务器进行解析参数，并将要返回的数据封装成对象，返回客户端，客户端通过Gson去解析对象，并对返回数据在页面上显示出来。所以在请求之前需要对接口进行设计。接口的设计依旧是根据业务逻辑进行划分。

说明：一下ip地址用“ip“替代。

### 4.3.1 登录

接口：<http://ip:8080/EoApp/login>

参数说明：

（1）userTel 用户手机号

（2）userPwd 用户密码

返回值说明：

（1）status 请求状态 0：成功 1：失败

（2）result 返回以Json格式的用户信息字符串

### 4.3.2 注册

接口：<http://ip:8080/EoApp/regist>

参数说明：

（1）userTel 用户手机号

（2）userPwd 用户密码

（3）requestCode 验证码

返回值说明：

（1）status 请求状态 0：成功 1：失败

（2）result 返回以Json格式的用户信息字符串或者错误消息

### 4.3.3 获取公司信息

接口：<http://ip:8080/EoApp/getcompany>

参数说明：

（1）userId 用户id

返回值说明：

（1）status 请求状态 0：成功 1、失败

（2）result 返回以Json格式的公司信息字符串或者错误消息

### 4.3.4 获取用户所在部门信息

接口：<http://ip:8080/EoApp/getcurrentgroup>

参数说明：

（1）userId 用户id

返回值说明：

（1）status 请求状态 0：成功 1、失败

（2）result 返回以Json格式的部门信息字符串或者错误消息

### 4.3.5 将签到的设置存到数据库

接口：http://ip:8080/EoApp/receiversettinginfo

参数说明：

（1）company 公司id

（2）companySignSettingInfo 以Json格式的签到设置字符串

返回值说明：

（1）status 请求状态 0：成功 1、失败

### 4.3.6添加好友/添加子部门，修改部门信息

接口：http://ip:8080/EoApp/ManageGroup

参数说明：

（1）editMember 是否是编辑成员功能

（2）deleteMember 是否是删除成员功能

（3）addMember 是否是添加成员的功能

（4）user 以Json格式的用户信息字符串

（5）addChild 是否是添加子部门功能

（6）group 以Json格式的部门字符串

（7）update 是否是更新部门信息的功能

（8）delete 是否删除部门的功能

（9）targetGid 要删除的目标id

返回值说明：

（1）status 请求状态 0：成功 1、失败

（2）result 如果错误返回错误的提示信息，成功则不返回该字段

# 第五章 程序代码分析

### 5.1 Android 端代码分析

Android Studio 开发采用逻辑层和视图层分开的模式，该软件再设计就采用MVC的设计模式。我通过XML去绘制每个页面的布局样式，然后通过Java代码将布局显示出来，并去实现相关逻辑。

**5.1.1 整体代码层次结构**

随着模块化编程越来越火热，将部分功能通过模块引入该系统中也是一大特色。该系统引入了三个模块：IMkit融云负责聊天、circlerefresh 下拉刷新样式、timeselecor选择时间的对话框。

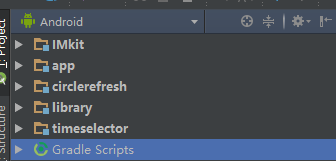


图5-1：整体代码模块结构

在主项目中我们根据Java类的类型去分包。Android 的四大组件也分别在四个包中，Fragment 的功能和Activity效果类似，所以也单独分包。其余的在按照功能分：工具包、自定义View包、JavaBean包等。

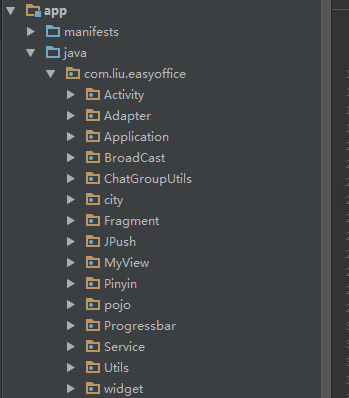


图5-2：主项目结构

资源文件都会在系统自带的res包下，可以分为动画包、颜色资源包、图片资源包、布局文件、values下则是 颜色值和字符串值等。在xml文件我可以通过@drawable 去引入图片资源或者@color引入颜色资源。Java代码中则会通过R.color.xxx或者R.drawable.xxx动态引入。

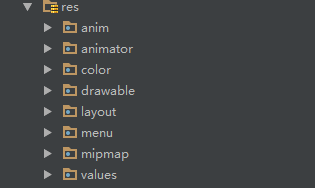


图5-3：资源目录结构

**5.1.2 XML文件分析**

在XML布局文件中通常会用ViewGroup包裹View的样式，一般ViewGroup常用的有RelativeLayout、LinearLayout、FramentLayout这三个最为常用的。虽然官方还推出其他ViewGroup然后做项目中这三者最为常用。RelativeLayout 中的子布局默认都会安放在父布局的左上角，并且其第二个子布局会覆盖在第一个子布局之上。LiearLayout 会根据orientation排列方向自动排列，如果水平排列则子布局们会水平依次排列，竖直则竖直依次排列。总体来说RelativeLayout会比Liearlayout有更少的层次结构。因为层次越深，Java代码解析的时间就会更长，性能会有影响。

如图这是启动页，该布局有两层结构：根布局 RelativeLayout ,两个子布局 ImageView和自定义View。其中ImageView是图片容器，负责展现图片，其中src负责资源的引入，layout\_width和layout\_height 则是为了设置自己在父布局中的大小。这种大小一般分为wrap\_conent自适应、match\_parent(有父布局一样大)、具体的大小这三种。第二个是一个自定义View,该自定义View是让文字以一种动画方式显示。在Relativlayout中要想改变自己在父布局的位置，通常用相对的概念，相对于谁，在谁的哪个位置。alignParentBottom则意思在父布局的底部，marginBottom 是距离底部多少距离，ceterHorizontal=“true”意思水平居中，通过这三个属性，就能指定该控件的具体的位置。为了做到适配效果，Android 中推荐使用以dp为单位进行大小的设置。因为Android 机型很多，若以px为单位，则在不同的机型上会显示不同的效果。dp是每像素多少的点的意思，所以不同机型效果一样。

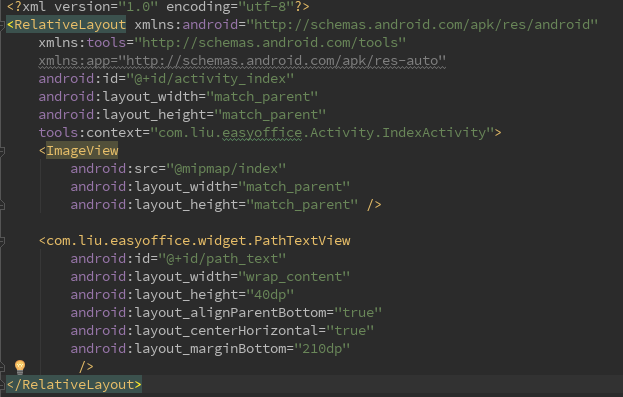


图5-4：引导页面的布局文件

**5.1.3 Java代码的分析**

布局需要加载在Activity才能显示，所以一般我们需要继承Activity重写onCreate等一些方法。一般onCreate里使用setContentView（R.layout.xxx）将写的布局填充进来，紧接者就需要对布局中的要使用到的控件进行初始化，通过findViewById(控件的id)来获得该控件。如果对控件进行点击事件的，需要注册监听器，setOnclickListener(OnclickListener),用户可以使用匿名内部类，或者让该类实现OnclickListener 接口，统一在onClick(View ViewId)实现。但是无论怎样都需要注册监听器。

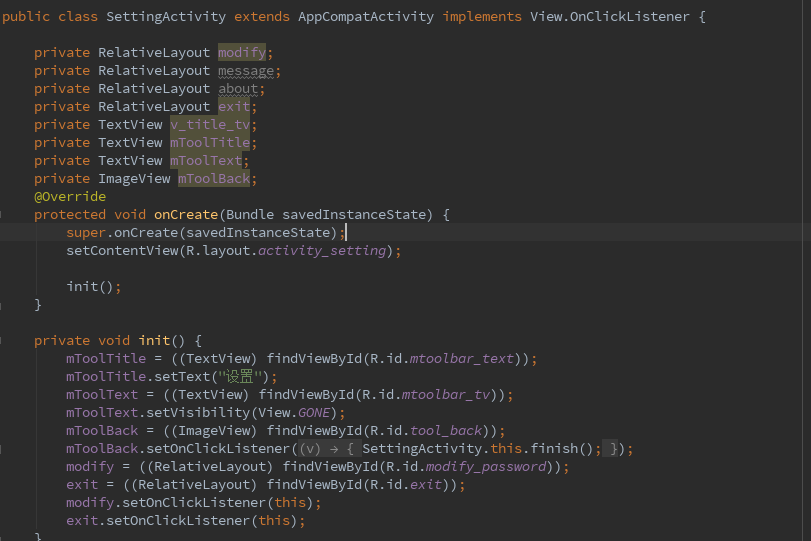


图5-5：SettingActivity的初始化控件的代码部分



图5-6：SettingActivity的所有点击控件的逻辑代码

### 5.2 服务器端代码分析

服务器端代码通过编译软件MyEclipse 来实现，代码主要逻辑都是通过Servlet来编写。整体采用MVC的模式并来分包

**5.2.1 整体架构分析**

整体分为dao层（负责数据库的增删改查操作）、service(负责接口的声明，与dao层紧密联系) 、View层（用来对数据的显示，这里功能实现的主要地方，通常写的类需要继承HttpServlet并实现doGet或者doPost方法，动态显示页面）。再者就是根据功能模块进行分类Chats（聊天模块）、login(登录模块)、shenpi(审批模块)等。

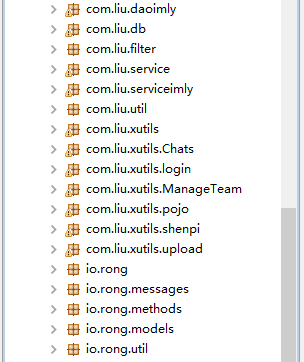


图5-7：后台的整体架构

**5.2.2 部分代码分析**

在LoginCheck中，其目的就是为了对登录界面进行检查，并向页面输入返回的数据。因为该请求是get请求，所以我们需要在doGet进行操作。通过request.getParameter(“xxx”),对客户端传过来的参数进行获取，在调用DbImp的getUserByUserId去数据查询，如果没有查到数据则向页面输出”0”，否则通过Gson 将查询到的对象转为以Json 格式的字符串输出到页面上。像这种继承HttpServlet的需要在web.xml中声明，就像Android 的四大组件需要在AndroidMenifast中声明一样。



图5-8：LoginCheck处理逻辑的代码片段

# 结 语

该系统取代摒弃传统的PC端的OA软件，转而向移动平台，是因为如今是移动平台的一直时代的潮流。就在前不久，Android 最终超过Windows成为目前为止，使用率最多的操作系统，尽管领先的优势很微弱，但毕竟windows在上个世纪九十年代就已经被全球所使用。而Android 则是在2012年才开始步入人们的视野，短短的几年内，能有如此成就，可想前景是多么广阔。

该系统首先在界面上做的很是美观，毕竟一个App能否吸引用户，最重要的还是界面效果。该系统设计之初，就首先定义主题颜色蓝+白。每个页面均是按照这个主题去设计，让用户觉得简洁但又美观，在视觉上能够起到很好的冲击效果。其次系统尽可能让用户少输入文字，多选择，所以需要输入的地方很好，基本上能可以选择的决不让用户填写。动画效果也在该App中多次使用，比如在组织架构模块，用户点击下一个部门，就能看见该页面向左退出，新页面从右边入场。在审批功能中，我是一种流程图样式的审批进度显示，用户很明显就能发现自己的审批单待谁审批。

在功能逻辑上，完全符合实际逻辑，而且功能都已成功实现。扩展上的效果也很好，在创建团队时，允许管理员无限制的创建子部门，在数据库表的设计当中我通过记录当前部门的父部门，就能够确保，子部门无论怎么添加，其顺序始终不会改变。数据真实性：该App一切产生的数据都是从自己的服务器得到，没有本地的假数据、也没有从网上的其他服务器接口直接获取数据，就连验证码也都是结合阿里大于会像注册的手机号发送真实的验证码。系统可用性：该App完全可以投入运营，因为功能都已实现，数据库、服务器只要搭建在公司内部，修改一下ip就能供公司实用。

尽管编写系统的工作量比较大，客户端、服务器、数据库都要实现，因为OA类的逻辑通常比较复杂，这里为了简单化，所以该App只试用于一个公司，也就是如果该公司要试用，需要单独搭建服务器，单独的数据库。所以这是该App的一个缺陷。还有该App只能在Android 机器上运行，IOS系统的用户暂时无法支持。

完成该系统不仅是实现系统功能就行了，其主要目的还是为了自己检测自己大学四年学到的技术。课堂只会授予理论上的知识，而我们的专业在于能否设计出一个优秀的产品出来，将理论知识转换实践是非常重要。通过该系统的设计，我觉得大学四年并没有荒废，除此之外，我们还要不断得汲取新的技术，即使不能引领潮流，但千万也不能被时代所抛弃！

# 致 谢

在选取导师得时候，我毅然选择了丁玉萍老师，因为大二时期她教授的计算机组成原理课程严谨得教学态度让我印象深刻。在选取课题时时候，丁老师在通过分析我的能力，和项目的开发难度对于我设计课题提供了很多的帮助。为了能够让我们能够顺利毕业，及时的告知我们毕设期间一些重要事项。生怕我们当中一个人因为其中某一项延误而影响毕业。

大学这个充满活力的地方，同学之间的友谊、师生之间的情谊，都能够得到最大的展示。感谢母校江科大张家港校区给我提供一个温馨的处所，感谢所有大学四年以来教授我们课程的老师，您们辛苦了！感谢杨子天班主任这四年来的陪伴，13届软件同学们的陪伴，因为有你们所以人生才会时时刻刻精彩！

最终把感谢留给我自己，感谢四年以来没有抛弃自我，没有让自己堕落，为了坚持自己的梦想，付出我认为的所有努力。这些努力也让我在南京找到一份满意的实习工作。坚持努力，相信明天更加精彩！明天，您好！

# 参考文献

[1] Bruce Eckel. Thingking in Java. 陈昊鹏译. 机械工业出版社，2007

[2] 郭霖. 第一行代码 Android. 人民邮电出版社，2014

[3] 王泉. “互联网+”时代下的协同OA管理. 清华大学出版社，2015

[4] 罗鸿. ERP原理·设计·实施. 电子工业出版社，2016

[5] [何红辉](https://book.jd.com/writer/%E4%BD%95%E7%BA%A2%E8%BE%89_1.html)，[关爱民](https://book.jd.com/writer/%E5%85%B3%E7%88%B1%E6%B0%91_1.html). Android 源码设计模式解析与实战. 人民邮电出版社，2015

[6] 任玉刚. Android 开发艺术探索. 电子工业出版社，2015

[7] 李刚. 疯狂Android 讲义. 电子工业出版社，2016

[8] 王珊，萨师煊. 数据库系统概论（第5版）. 高等教育出版社，2016

[9] [唐振明](https://book.jd.com/writer/%E5%94%90%E6%8C%AF%E6%98%8E_1.html)，[王晓华](https://book.jd.com/writer/%E7%8E%8B%E6%99%93%E5%8D%8E_1.html)，[李丹程](https://book.jd.com/writer/%E6%9D%8E%E4%B8%B9%E7%A8%8B_1.html)，[徐洪智](https://book.jd.com/writer/%E5%BE%90%E6%B4%AA%E6%99%BA%E8%91%97_1.html). JavaEE架构与程序设计. 电子工业出版社，2011

[10] 周志明. 深入理解Java虚拟机. 机械工业出版社，2013

[11] 葛一鸣. 实战Java虚拟机. 电子工业出版社，2015

[12] [Dan Pilone](https://www.amazon.cn/s/ref=dp_byline_sr_ebooks_1?ie=UTF8&text=Dan+Pilone&search-alias=digital-text&field-author=Dan+Pilone&sort=relevancerank) . UML 2.0 Pocket Reference: UML Syntax and Usage. O'Reilly Media，2013

[13] [Richard E. Silverman](https://www.amazon.cn/s/ref=dp_byline_sr_ebooks_1?ie=UTF8&text=Richard+E.+Silverman&search-alias=digital-text&field-author=Richard+E.+Silverman&sort=relevancerank).  Git Pocket Guide: A Working Introduction. O'Reilly Media，2013