广东工业大学

华立学院

本科毕业设计（论文）

基于即时通讯技术协会管理的 APP 设计

论文题目 _____

学　　部 机电信息与工程

专　　业 计算机科学与技术

班　　级 13 本计算机 1 班

学　　号 511321010110

学生姓名 张松周

指导教师 程东胜

2017 年　4 月　18 日

摘　　要

如今，借助互联网技术的广泛应用，人们之间的联系通讯不也再受到距离的限制。正因如此，人们在各个行业上对沟通交流的依赖和需求也越来越强烈，对需求业务趋向人性化。通过各种的即时通讯软件，人们可以足不出门的情况下，与世界各地的人们进行实时聊天交流，并且通过翻译软件进行语言无障碍沟通。同时即时通讯技术使得人与人之间的交流带来了极大的便捷和大量的经济节省，促进了世界的快速发展。因此，即时通信在人们的生活工作和学习中得到了更广泛的应用。

著名的脸书、推特、qq、微信等知名的即时通讯软件当今的互联网上已得到了广泛的认可，无论在生活上，还是在工作上都是极大方便人们的交流和管理。与此同时，适应各种场合，以及各种的即时通信软件也开始出现在软件市场上，这就包括了跟企业管理相关的通信管理软件，具有企业特色的信息管理的即时通讯软件。

本文介绍基于 Android 的即时通讯系统进行对协会或者企业管理的实现过程。运用 java 语言、javaEE 管理阿里云服务器、Android studio 作为主要的开发平台，开发一个能应用于协会或者企业管理的软件——格局。

该软件的即时聊天功能与微信类似，手机号注册成功登录后，便能添加好友实时聊天，同时用户可以创建一个自己专属的协会，添加并管理成员。不仅如此，内部还拓展了许多的辅助功能。通过这次实践，总结了许多关于安卓开发的所需知识，为自己所在的实习得到更好的施展。

关键字：协会，安卓，即时通讯，管理

Abstract

Now, with the wide application of Internet technology, Contact the communication between people is no longer limited to the distance. For this reason, people in various industries on the demand for communication and dependence is becoming more and more strong, an on demand business to as both land tends to humanization. Through a variety of instant messaging software, people can never go out, chat with people all over the world in real time communication, and barrier-free communication through the translation software language. At the same time instant communication technology makes the communication between people has brought great convenience and save a lot of economy, promote the rapid development of the world.Therefore, im in the work and the study of people's lives more extensive application.

The famous facebook,twitter, qq, WeChat instant messaging software such as today's Internet has been widely recognized, no matter in life, or on the job is greatly convenient people's communication and management. At the same time to adapt to various occasions of real-time communication software also began to appear in the software market, including some instant messenger software, enterprise management related with characteristic of enterprise information management and instant messaging.In this paper, the instant messaging system based on Android to the association or the implementation of enterprise management using the Java language javaEE ali Android cloud server management studio as the main development platform, to develop a can be applied to the association or the software pattern of enterprise management--geju.

Instant messaging functions of the software is similar to WeChat, mobile phone number registered after a successful login, can add buddy live chat, at the same time, the user can create a their own association, add and manage members Not only that, internal also expanded the number of auxiliary functions Through this practice, summarizes the many knowledge about android development, to get better put to good use for their internship.

Key words： Association, Android, Instant communication, Managemen

目　　录

1 绪　论

1.1 开发背景

伴随着互联网技术的快速发展，移动通讯技术已经进入了飞跃性的时代。软件和硬件的技术不断突破，通讯的传输质量和速率有大幅的提高，极大地改善人们的日常生活水平质量。以智能应用，比如手机、电脑、平板等为代表的移动互联终端，变成了人们在日常工作生活中体验移动服务的重要手段。 特别是即时通讯技术，已经融入了日常生活中的各方面。在国内，随着以腾讯为代表的微信和 QQ 软件的流行，即时通讯的相关应用已经成为人们交流的一种重要工具。

随着生活质量的提升，渐渐地人们对文字、语音、图片这种实时通讯类型不感到满足，视频相关的实时通讯越来越受到人们的青睐。经过近几年的迅速的发展，随着通讯的拓展功能不断丰富，它不再是个单一的聊天工具，它已经发展成集交流、资讯、音乐、电视、娱乐游戏、电子商务等为一体的综合化信息平台。

即时通讯技术不同于 E-mail，在于它的交谈时是即时的，大部分的即时通讯服务提供了不少的状态信息特性，例如：联络人是否在线，显示联络人名单，能否与联络人交谈。现在，即时通讯软件，主要是以企业内部办公，建立员工交流平台为基础，通过整合边缘功能，为企业提供一整套的信息沟通和实时协作。由于企业对信息类软件的需求还在探索与尝试阶段，而且现在这些软件也存在许多的安全隐患，但是优越性高始终被选用的重要条件。

1.2 现状

即时通讯软件，几乎是每个企业或者说是每个人上网的必备的联络工具。即时通讯可以分为两种：一种是个人即时通讯，另一种是企业即时通讯。个人即时通讯在中国最有影响的首推腾讯。旗下产品 QQ 和微信应用，不只聊天功能强大，注重保护个人隐私，其次的娱乐性也很强。但是它的竞争对手还是比较多，不同的用途也是一门应用，比如阿里巴巴，是专注商人和客户之间的聊天，这与现在风靡全球的网购有关；飞信是中国移动推出的服务；YY 是专门对网络玩家创办的聊天系统；人人是专注设计给在校生的即时通讯系统，国内还有很多其它 android 软件供于用户使用。综上所述可以看出，人们早已经被即时通讯包围，并适应熟悉使用。未来的安卓即时通讯系统会越来越人性化。同时，协商业的即时通讯功能会更加庞大。

1,3 目的意义

对于现在科学技术的快速发展，掌握了移动互联网就是掌握了未来的发展。通过做这份基于安卓的即时通讯技术是希望本人能更好地掌握计算机互联网知识，同时做商业管理软件附加功能不断完善，有助于对本人 Android 技术水平的提升，更加熟练解决安卓初级常见问题，以及更好了解商业项目的需求，不断完善项目功能，提高用户体验度。

经过这个毕业设计的磨炼，自己对互联网安卓行业有了更深的了解，也有更多的专业知识来充实自己的大脑。集成多媒体的商业管理应用有着更强的吸引力，为用户提供更多个性化的服务的功能，将成为未来移动即时通讯发展的一个趋势。

2 应用相关的技术与理论

2.1 语言

java，C++，SQL，JavaScript+html

2.2 平台

Android Studio (2.3.1),javaee，eclipse

2.3 开发系统

苹果 mini，windows

2.4 调试测试机型

小米 2s（Android5+），华为（Android4.3），三星（Android 4.4），小米 4（Android6.0），以及编译器自带模拟器

2.5 Android 的介绍

Android 是 Google 开发，基于 Linux 平台的开源的手机操作系统。它包括操作系统、用户界面和应用程序等移动电话工作所需的全部软件，而且不存在任何以往阻碍移动产业创新的专有权障碍。Android 采用了 WebKit 浏览器引擎，具备触摸屏、高级图形显示和上网功能，用户能够在手机上搜索网址、查看电子邮件和观看视频节目等，可以说是一种全部 Web 应用的单一平台。

2.6 Android 系统架构

Android 的系统架构和其他操作系统一样，采用了分层的架构。Android 分为四层，从低层到高分别是 Linux 核心层、系统运行库层、应用程序架构层和应用程序层。

2.6.1 应用程序层

Android 会同一系列核心应用程序包一起发布，该应用程序包括 E-mail 客户端、SMS 短信程序、地图、日历、联系人管理程序、浏览器等。所有的应用程序都是使用 java 语言编写。

2.6.2 应用程序架构层

该应用的架构设计简化了所有组件的重用；任何一个应用程序都可以发布它的功能块并且任何其它的应用程序都可以使用其发布的功能块，不过重要的是，这得遵循框架的安全性限制。同样，该应用程序重用机制，可以方便替换程序组

件。

隐藏在每个应用后面的是一系列的系统和服务，其中包括：

* 丰富而又可扩展的视图(Views)，可以用来构建应用程序界面， 它包括列表视图(lists)，网格视图(grids)，文本框视图(text boxes)，按钮视图(buttons)， 甚至可嵌入的 web 浏览器。

* 内容提供器(Content Providers)，可使应用程序可以访问另一个应用程序的数据(如联系人数据库)，或者共享它们自己的应用数据

* 资源管理器(Resource Manager)提供非代码资源的访问，即 xml 文件，如本地字符串，图形，和布局文件(layout files)。

* 通知管理器 (Notification Manager) 使得应用程序可以在状态栏中显示自定义的提示信息，比如推送。

* 活动管理器(Activity Manager) 用来管理应用程序生命周期，包括服务的生命周期，并提供常用的导航回退功能。

2.6.3 系统运行库

2.6.3.1 程序库：

Android 包含一些 C 或 C++库，这些库能被 Android 系统中不同的组件使用，例如多媒体的实现。它们通过 Android 应用程序框架为开发者提供服务。

以下是一些核心库：

* 系统库 C：一个从 BSD 继承来的标准 C 系统函数库(libc)，它是专门为基于 embedded linux 的设备定制的。

* 媒体库：基于 PacketVideo OpenCORE;该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括 MPEG4, H.264, MP3, AAC, AMR, JPG, PNG 。

* Surface Manager：对显示子系统的管理，并且为多个应用程序 提供了 2D 和 3D 图层的无缝融合。

* LibWebCore：一个最新的 web 浏览器引擎，支持 Android 浏览器和一个可嵌入的 web 视图。

* SGL：底层的 2D 图形引擎

* 3D libraries：基于 Ope nGL E S 1.0 APIs 实现;该库可以使用硬 件 3D 加速(如果可用)或者使用高度优化的 3D 软加速。

* FreeType：位图(bitmap)和矢量(vector)字体显示。

* SQLite：一个对于所有应用程序可用，功能强劲的轻型关系型数据库引擎。

2.6.3.2 Android 运行库

Android 包括了一个核心库，该核心库提供了大多数功能的 JAVA 编 程语言核心库。

每一个 Android 应用程序，都拥有一个独立的 Dalvik 虚拟机实例并且都在它自己的进程中运行。Dalvik 被设计为一个设备可以同时高效地运行的同时，并多个虚拟系统。 Dalvik 虚拟机执行(.dex)的 Dalvik 可执行文件，该格式文件针对小内存使用做了优化。同时虚拟机是基于寄存器的，所 有的类都经由 JAVA 编译器编译，然后通过 SDK 中 的 "dx"工具转化成.dex 格式，并由虚拟机执行。

Dalvik 虚拟机依赖于 linux 内核的一些功能，比如底层内存管理机制和线程机制。

2.6.4 Linux 内核

Android 核心系统服务依赖于 Linux 2.6 内核，如内存管理，安全性，，网络协议栈和驱动模型和进程管理。Linux 内核也同时作为硬件和软件栈之间的抽象层。

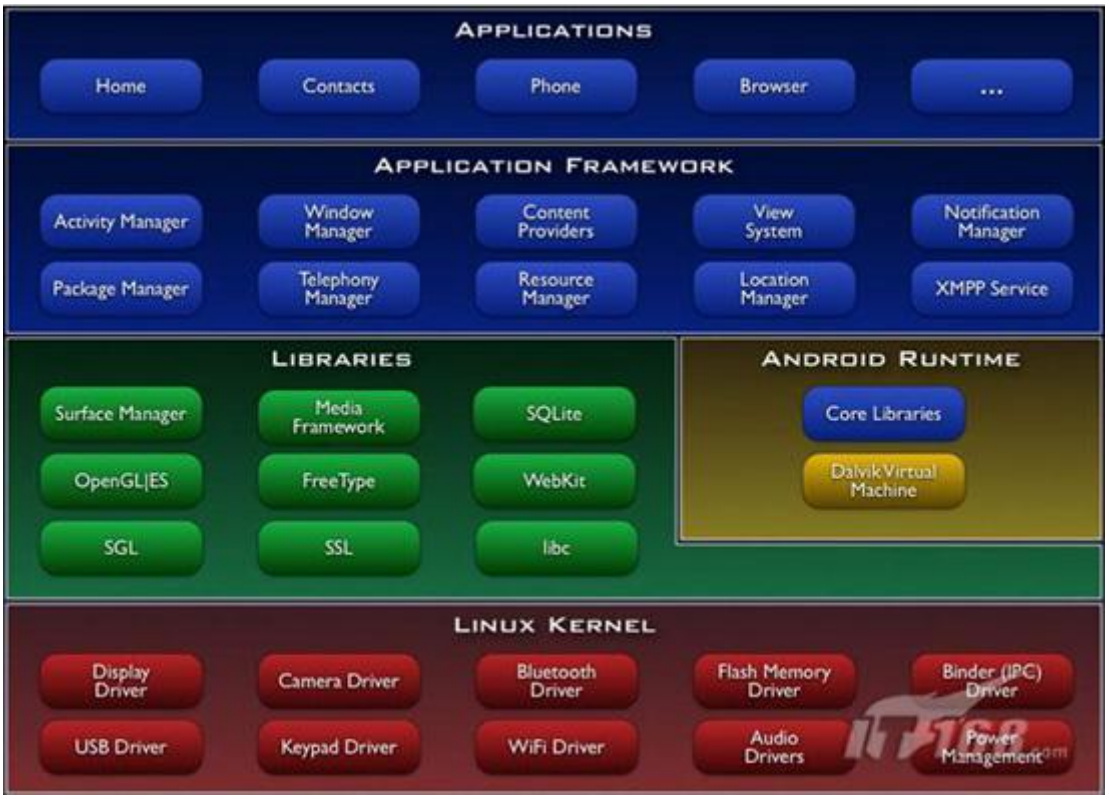

图 2.1Google 提供的架构

2.7 javaEE 后台服务器

javaEE，一种标准 开发架构，是在 JavaS E 基础之上建立起来的。内部提供了一套关于企业应用程序的规范：设计、开发、汇编和部署，来实现企业级应用程序。技术平台的核心思想是：容器+组件

2.7.1 JavaEE 体系结构

Applet JSP + Servlet EJB（重量级框架）。

组件是一个软件单元。它包含一定功能的，有着相关的类和文件一起组成， 并与其他组件进行通信。

EJB 组件：分为会话 EJB、实体 EJB 和消息驱动 EJB；

EJB 设计缺陷：EJB 采用不是面向对象设计的过程设计。

EJB 开发的问题：EJB 开发和测试非常冗长和麻烦。

编辑、编译、调试周期长；

编码冗余、繁琐；

必须编写数据传输对象（DTO）。

面向对象的设计更加容易理解、维护、扩展以及测试。

2.7.2 基于 MVC 的轻量级框架：

2.7.2.1 主流 JavaEE 框架：

表现层框架：SpringMVC 框架，Struts2 框架、JSF 框架、Tapestry 框架、WebWork 框架

业务逻辑层框架：Spring 框架

持久层框架：Hibernate 框架、MyBatis

2.7.2.2 企业级应用需求：

通过将框架集成的应用，可以发挥各框架的最大优势，良好的解决企业级应用的需求本身。

Struts+Spring+Hibernate 框架（SSH 框架）

SpringMVC+Spring+Hibernate 框架

SpringMVC+Spring+MyBatis 框架（SSM 框架）

企业级应用新解决方案：框架集成应用

2.8 应用第三方库

即时通讯和直播：环信 sdk

图片加载：Picasso

支持放大查看图片的控件：photoView

带左滑动菜单 listview：swipemenulistview

网络请求库：okhttp

百度地图 sdk

3 系统的设计与实现

3.1 App 结构介绍

参考了微信 app 的交互流程，实现一个 Android 平台及时通讯系统，实现各类型客户端之间的互操性功能，提供好友的实时状态，随时随地与他人进行即时通讯交流。

主要基于 java 语言在 Android Studio 上调试编译，gradle 项目自动化建构打包，build 脚本编写。

项目整合了 Android 常用的 sdk 为开发基础，参考了商业项目开发流程以及框架技术，根据自身的技术水平，不断完善 app 功能，涉及到 Android 初级到中级的技术水平，界面表现形式主要是 Android 常用控件（TextView、EditeText、ImageView、Button、ListView、RadioButtom 等），期间制作了列表的下拉刷新和上拉加载等的操作；对接 javaee 后台服务器，异步缓存请求网络 json 数据，实时更新界面 UI；三级图片框架缓存；对于缓存数据操作，涉及到 Android 的 sqlite 和 sharepreference 库。

3.2 第三方库介绍

3.2.1 环信

是一家全通讯能力和云服务提供商；该项目用到的版本是 V3.0，包含着新型的通信协议：基于消息同步的私有协议，在不稳定网络环境下，更稳定更省流量，确保消息投递可靠、顺序以及实时性，并具有更高的安全性。同时提供了更好的拓展性，场景将会支持更多的对接和设备同步；全新 sdk：代码上进行全面的重构，将核心通信模块做了更好的封装，主要是运用到了策略模式和工厂模式；简化了接口，结构更清晰，集成更容易，提高了登录速度以及弱网络环境下的可靠性。

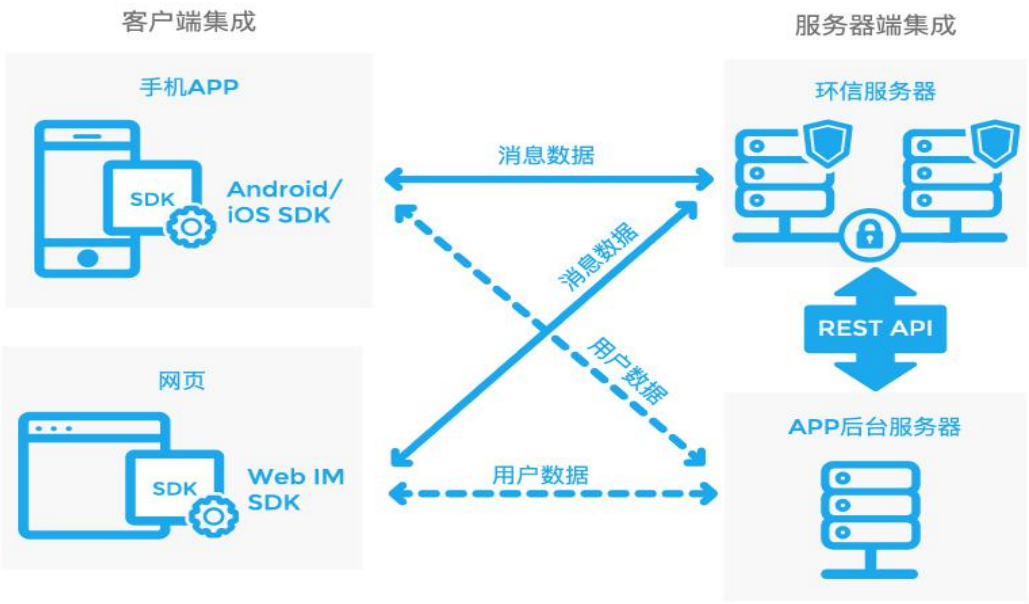

图 3.1 平台架构

3.2.2 OKHttp

HTTP is the way modern applications network. It’s how we exchange data & media. Doing HTTP efficiently makes your stuff load faster and saves bandwidth。这是引用官方的自我介绍。Okhttp 是 Android 一款优秀的 Http 开源框架，支持 get 和 post 请求，支持基于 Http 的 文件上传和下载，支持加载图片，支持下载文件透明的 GZIP 压缩，支持响应缓存，可避免重复的网络请求，支持使用连接池来降低响应延迟问题。

3.2.3 Picasso

Square 公司出品，是一款非常优秀的开源图片加载库，是 Android 开发中目前比较流行的图片加载库之一。解决使用复杂的图片压缩转换来尽可能的减少内存消耗，而且还自带内存和硬盘二级缓存功能。

3.2.4 百度地图 sdk

基于 Android2.3 及 以上版本设备的应用程序接口，适用于 Android 系统移动设备的地图应用，通过调用地图 sdk，可以轻松访问百度地图数据和服务，构建功能丰富、交 互性强的地图类应用程序。

3.2.5 友盟

第三方提供商：全域大数据服务，通过全面覆盖 PC、传感器、手机、无线路由器等多种设备数据。提供全业务链数的应用解决方案，这包括基础统计、数据决 策、运营分析和数据业务等，帮助企业实现数据化管理和运行。

其他第三方库：都是 Github 优秀开源项目。

3.3 app 结构图

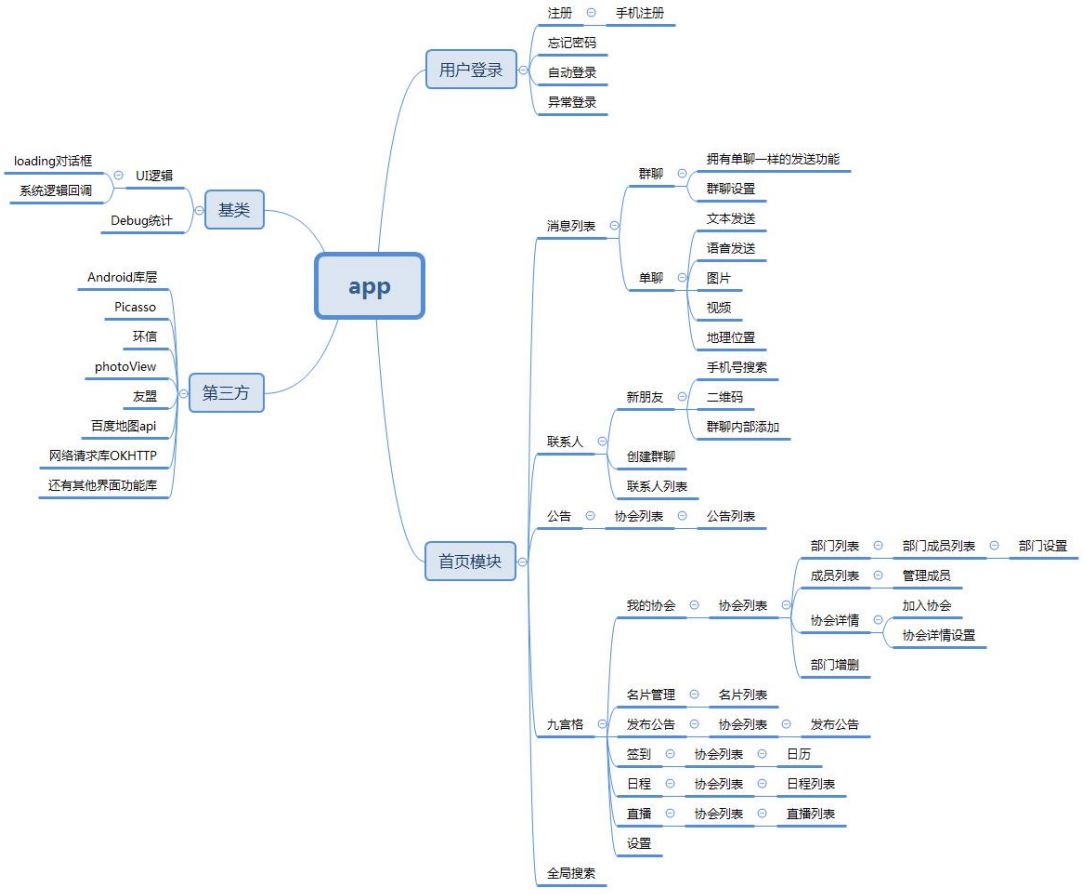

图 3.2App 大体结构

3.4 App 适用 Android 版本

适用 Android 最低版本为 16（ Android 4 . 1 ），最高版本为 22 （ Android 5 . 1 ）

< uses - sdk

Android : minSdkVersion = " 14 "

Android : targetSdkVersion = " 2 2 " / >

3.5 APP 程序入口

程序启动的入口第一个界面：

< activity

Android : name = "app.logic.activity.launch.LaunchActivity"

Android : screenOrientation = " portrait "

Android : theme = " @style / windowallwhite " >

<intent-filter>

<action android : name = " android . Intent . Action . MAIN " / >

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

</intent-filter>

</activity>

3.6 App 交互流程浏览

3.6.1 启动页

使用 Viewpager 装载多张图片来向用户介绍该 app 的大体功能和用户

主要代码：

((TextView)view01.findViewById(R.id.launch_tv01)).setText(″协 作 从格局开始″);

((TextView)view01.findViewById(R.id.launch_tv02)).setText(″发布需求、创建组织、管理部门成员″);

((TextView)view01.findViewById(R.id.launch_tv03)).setText(" 一步到位″);

(TextView) view02.findViewById(R.id.launch_tv01)).setText(" 创建自己的工作圈″);

((TextView)view02.findViewById(R.id.launch_tv02)).setText(″组织成员、客户伙伴各种人脉快速找到″);

((TextView)view02.findViewById(R.id.launch_tv03)).setText(″协作关系更加轻松″);

((TextView)view03.findViewById(R.id.launch_tv01)).setText(″圈内公告″);

((TextView)view03.findViewById(R.id.launch_tv02)).setText(″随时随地查看最新动态，收纳繁琐消息″);

((TextView)view03.findViewById(R.id.launch_tv03)).setText(″不易遗漏错过″);

((TextView)view04.findViewById(R.id.launch_tv01)).setText(″组织管家″);

((TextView)view04.findViewById(R.id.launch_tv02)).setText(″一目了然，随我掌控″);

((TextView)view04.findViewById(R.id.launch_tv03)).setText(″开启智能化管理″);

图 3.3 启动图

3.6.2 登录页

这里是主要的登录方法，多重检查，人性化判断懒惰的操作。

这也是用户信息的缓存操作，单一用户信息全局化

Android 轻量级缓存 share preference，存储用户 id 和 token 信息

```
/**
  * 用户登录方法
  *
  * @param user name
  * @param pass word
  */
 Private void login ( final String username, final String password ) {
      showWai tDialog ( ) ;
      if ( ! Username.equals ( " " )  &&  ! Password .equals (" " ) ) {
          User  ManagerController.Login(LoginActivity.this,  username,
password, new Listener < Integer, UserInfo > ( ) {

              @Override
              public void on CallBack(Integer status, UserInfo reply ) {
```

```
if ( status == 1 ) {
    // 登录成功
    QLConstant.client_id = reply.getWp_member_info_id ( ) ;
    QLConstant.user_picture_url = reply.get Picture_url ( ) ;
    JPushI terface.setAlias ( LoginActivity.this, reply.getWp_member_info_id ( ) , null ) ;
    utils.setUserName ( username ) ; //本地保存用户名
    utils.setPassword(password); //本地保存密码
    // im 聊天登录
    String chartpsw = "wudi#" + username;
    new EMOptions().setAutoLogin ( true ) ;
    EMClient.getInstance ( ) . Login ( username, chartpsw, new EMCallBack ( ) {// 回调
        @Override
        public void onSuccess ( ) {
            Run On UiThread(new Runnable ( ) {
                public void run ( ) {
                    Log.v("hhhh", "im 登陆成功");
                    //从本地加载群组列表
                    EMClient . getInstance ( ).groupManager ( ).getAllGroups ( ) ;
                    EMClient.getInstance ( ) .chatManager ( ) . loadAllConversations ( ) ;
                    // EMGroupManager . getInstance().loadAllGroups ( ) ;
                    / / EMChatManager.getInstance ( ) . loadAllConversations ( ) ;
                    Mhanler . sendEmptyMessage ( 0 0 0 ) ;
```

```
                    }
                });
            }

            @Override
            public void onProgress(int progress, String status )
{

            }

            @Override
            public void onError ( int code, String message ) {
//                Message msg = new Message();
//                msg.what = 111;
//                msg.obj = message;
//                mhanler.sendMessage(msg);
//                Log.v ( "h h h h " , "登陆聊天服务器失败!
" ) ;

                if ( code == 200 ) {
                    runOnUiThread ( n ew Runnable ( ) {
                        public void run() {
                            Log. V ( " h h h h " ,     " i m
登 陆 成 功    "  ) ;
                            //从本地加载群组列表
                            EMClient . getInstance ( ) .
groupManager() . getAllGroups ( ) ;
                            EMClient . getInstance
( ) .chatManager ( ) . l oadAllConversations ( ) ;
//
    EMGroupManager.getInstance().loadAllGroups ( ) ;
//                                EMChatManager .
```

```
getInstance().loadAllConversations ( ) ;
                                                    Mhanler.    sendEmptyMessage
( 00 0 ) ;
                                                        }
                                                    } ) ;
                                                } else {
                                                    Message msg = new Message ( ) ;
                                                    Msg .    what = 111;
                                                    Msg .obj = message ;
                                                    Mhanler .sendMessage ( m s g ) ;
                                                    Log .v("hhhh", "登陆聊天服务器失败!
" ) ;
                                                }
                                            }
                                        } ) ;
                                    } else if ( status == - 1 ) {
                                        dismissWaitDialog (    ) ;
                                        Toast . makeText ( LoginActivity.this,    " 请检查账号
或者密码是否输入正确！", Toast.LENGTH_SHORT ) .show () ;
                                    }
                                }
                            });
                        } else {
                            dismissWaitDialog ( ) ;
                            Toast . makeText ( LoginActivity .this, " 用户名或者密码不能为空",
Toast . LENGTH_SHORT) . Show ( ) ;
                        }
```

}

图 3.4 登录

3. 6. 3 消息列表

获取环信数据列表，通过按照时间戳的排序，最新消息依次置顶，同时回调和统计未读消息的数量，点击对话聊天后，自动将消息置零或不显示。

当消息列表上没有对方消息时，对方发来消息时调用服务器接口，将对方注册在对话列表上。

```
/**
    *获取列表消息
    **/
    private void loadMessageList ( ) {
        sessionList.clear ( ) ;
        UserManagerController . getChatList ( getActivity ( ).getApplicationContext ( ), new Listener < Integer, List < YYChatSessionInfo > > ( ) {
            @Override
            public void onCallBack(Integer status, List <Y YChatSessionInfo > reply ) {
```

```
ArrayList < YYChatSessionInfo >  _ t m p L i s t = new
ArrayList < YYChatSessionInfo > ( );
int unReadCountTemp = 0;
listView.stopLoadMore ( ) ;
listView.stopRefresh ( ) ;
if ( r eply != null && reply.size ( )  > 0 ) {

    for ( YYChatSessionInfo yyChatSessionInfo : reply ) {

        if ( yyChatSessionInfo.getChatroom ()  == null ) { //
单聊
            // 过滤掉和自己的聊天记录
            if ( yyChatSessionInfo. getWp_member_info_id
( )  . Equals (y yChatSessionInfo.getWp_other_info_id ( ) ) )  {
                continue;
            }
            // sessionList.add(yyChatSessionInfo ) ;


            // 计算未读
            EMConversation conversation = EMClient .
getInstance( ).chatManager( ) . getConversation(yyChatSessionInfo .
getPhoneNumber ( ) ) ;
            if (conversation != null )  {
                unReadCountTemp +=
conversation.getUnreadMsgCount ( ) ;
             }
        } else { //群聊
            EMConversation conversation = EMClient .
getInstance() . chatManager().getConversation(yyChatSessionInfo.getChatroom
```

```
( ) .getRoom_id ( ) );
                                    if (conversation ! = null ) {
                                        unReadCountTemp += conversation.
getUnreadMsgCount ( );
                                    }
                                }
                                _ tmpList . add( yyChatSessionInfo );

                            }

                            / / 设置未读数量
                            List< YYChatSessionInfo > sortList =
sortYYChatSessionInfos( setUnreadYYChatSessionInfos( _tmpList ) );

                            sessionList.addAll(sortList);

                            if (sortList == null || sortList.size() < 1) {
                                nullDataView. setVisibility ( View.VISIBLE );
                            } else {
                                nullDataView. setVisibility( View.GONE );
                            }

                            adapter.setDatas ( sortList );

                            // 读完的时候回调下,刷新 Tab

ZSZSingleton.getZSZSingleton( ).getStatusMessageListener( ).callbackStatusUpdata(
 unReadCountTemp );
                        } else {
                            nullDataView.setVisibility( View.VISIBLE );
```

```
                }


            }
        });
    }
// 判断对话列表是否有对方信息
    private void haveMessageList( final String phone, final EMMessage message ) {
        UserManagerController.getChatList( getActivity(), new Listener< Integer, List < YYChatSessionInfo > >() {

            @Override
            public void onCallBack ( Integer status, List < YYChatSessionInfo > reply ) {
                if (reply == null || reply.size( ) < 1) {
                    boolean temp = true;
                    for ( YYChatSessionInfo info : reply ) {
                        if (info.getPhoneNumber( ).equals(phone)) {
                            temp = false;
                            break;
                        }
                    }
                    if ( temp ) {
                        addChart( phone, message );
                    } else {
                        msgHandler.sendEmptyMessage ( kRefrashUI );
                    }
                }

            }
```

```
        });
}
// 获取用户信息，注册回话
    private void addChart ( String phone, final EMMessage msg ) {

        UserManagerController.getPhoneMemerInfo ( mContext, phone, new Listener < Integer, List < UserInfo > > ( ) {

            @Override
            public void onCallBack( Integer status, List < UserInfo >  reply ) {
                if ( reply != null && reply.size( ) > 0) {
                    String latestChart = null;
                    if ( msg != null && msg.getBody( ) != null ) {
                        if (msg.getBody() instanceof EMTextMessageBody ) {
                            latestChart = ( ( EMTextMessageBody ) msg.getBody( ) ).getMessage( );
                        } else if ( msg.getBody( ) instanceof EMLocationMessageBody ) {
                            latestChart = ( ( EMLocationMessageBody ) msg.getBody( ) ).getAddress ( );
                        } else if (msg.getBody( ) instanceof EMImageMessageBody ) {
                            latestChart = " 图 片 [ " + ( ( EMImageMessageBody ) msg.getBody( ) ).getFileName( ) + " ] ";
                        } else if ( msg.getBody( ) instanceof EMVideoMessageBody ) {
                            latestChart = " [ 视频 ] ";
                        } else if ( msg.getBody( ) instanceof EMVoiceMessageBody ) { latestChart = " [ 语音 ] "; } }
                    UserManagerController.addChatWith( getContext( ),
```

```
reply.get( 0 ).getWp_member_info_id( ), latestChart, new Listener < Integer, String >( ) {
                                @ Override
                                public void onCallBack (Integer status, String reply ) {
                                    loadMessageList( );
                                }
                            } );
                        }
                    }
                });
            }
```

图 3.5 消息页面

3.6.4 联系人

包括新建群聊操作、添加好友操作（可以通过手机号查找或者个人信息生成二维码，对方通过扫描解析里面的二维码信息，读取到 json 字段对应的手机号码）、联系人列表展示

对于联系人列表，主要先是对数据列表的按照拼音排序，然后对采取首个字节的拼音首字母进行分组，相同字母进行分块处理，最终在 listView 上多布局显示。

```
//获取联系人列表
 private void loadContacts( ) {
      UserManagerController.getFriendsList( getActivity( ), new Listener < List
< FriendInfo >, List < FriendInfo > > ( ) {
          @Override
          public void onCallBack ( List < FriendInfo >  request, List <
FriendInfo >  reply ) {
              ArrayList < FriendInfo > tmpInfos = new ArrayList <
FriendInfo > ( );
              if ( reply != null && reply.size( ) > 0 ) {
                  String myPhone = UserManagerController.getCurrUserInfo
( ) .getPhone ( ) ;
                  for ( FriendInfo friendInfo : reply ) {
                      if ( friendInfo.getPhone( ) != null
&& !friendInfo.getPhone( ).equals ( myPhone )  ) {
                          if ( friendInfo.isResponse( ) &&
friendInfo.isRequest_accept ( ) ) {
                              friendInfo.setOtherRequest ( false );
                              tmpInfos.add( friendInfo );
                          }
                      }
                  }
              }
              String [ ] data = new String [ tmpInfos.size() ] ;
              for ( int i = 0; i < tmpInfos.size( ); i++) {
                  Data [ i ] = tmpInfos.get ( i ) .getNickName( );
              }
              SourceDateList.clear ( ) ;
              SourceDateList.addAll ( filledDataList ( tmpInfos ) ) ;
```

```
                    //自定义排序
                    Collections.sort ( SourceDateList, pinyinComparator ) ;
                    adapter.notifyDataSetChanged ( ) ;
                    if (SourceDateList.size() > 0 ) {
                        empty_view.setVisibility( View.GONE );
                    } else {
                        empty_view.setVisibility(View.VISIBL E);
                    }
                }
            });
        }
     /*
        填充数据
         */
        private List< SortModel > filledDataList ( List < FriendInfo >    models ) {
            List < SortModel > list = new ArrayList < SortModel > ( ) ;
            for (int i = 0; i < models.size(); i++) {
                SortModel sortModel = new SortModel ( models.get ( i ) ) ;
                String tempString = models.get ( i) .getFriend_name ( )      == null ||
TextUtils.isEmpty(models.get ( i ).     getFriend_name ( ) )       ? Models.get ( i ).
getNickName ( )    : models.get ( i ).getFriend_name ( ) ;
                sortModel.setName ( tempString ) ;
                String pinyinString = characterParser.getSelling ( tempString ) ;
                String sortString = pinyinString.substring ( 0 ,      1 ) . toUpperCase
( ) ;

                if    ( s ortString.matches ( "[ A - Z ] " ) )    {
                    sortModel.setSortLetters ( sortString.toUpperCase ( ) );
                } else {
                    sortModel.setSortLetters ( " # " ) ;
```

```
            }
            List.add ( sortModel ) ;
        }
        return list;
    }
```

3.6.5 公告

协会列表到公告列表，动态排序，即时回调其他发布公告消息，快速刷新未读数，以及点击后刷新状态。也能通过点击通知栏的消息推送，调到指定的公告详情。

```
/**
     * 获取发了公告的组织
     */
    private void getMyOrg() {
//          ( ( T Y Ba s e A c t iv i t y ) context ) . showWaitDialog ( );
OrganizationController.getOrgUnreadNumber ( g e t C on text(), new Listener < Integer, List < OrgUnreadNumberInfo > > ( ) {
@Override
public void onCallBack(Integer status, List<OrgUnreadNumberInfo> reply) {
yyListView.stopRefresh();
//                    ( ( T Y B a seA ctivity ) context).dismissWaitDialog();
                    Datas.clear ( ) ;
                    if ( reply != null || reply.size ( )   > 0) {
                        datas.addAll ( reply ) ;
                    }
                    if( datas.size ( ) > 0 ) {
                        empty_view.setVisibility ( View.GONE ) ;
                    }else{
                        empty_view.setVisibility ( View.VISIBLE ) ;
                    }
```

```
                adapter.setDatas ( datas ) ;
            }
        });
    }
@Override
    @Nullable
    public View onCreateView ( LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState )  {
        view = inflater.inflate ( R.layout.fragment_notice_list, null ) ;
        fragmentTabHost = ( Fra g m e n tT a b H o  s t ) view. findViewById ( R . I d . N o t i c e _f tg ) ;
        fragmentTabHost.setup ( view.getContext ( ) , getChildFragmentManager ( ) , R.id.notic_fragment ) ;
        View unreadView = getIndicatorView ( " 未 读 " , getResources().getDrawable(R.drawable.selector_tabnotice_unread ) ) ;
        fragmentTabHost.addTab ( fragmentTabHost.newTabSpec ( " unread " ) .setIndicator ( unreadView ) , FragmentUnread.class, null ) ;
        View readView = getIndicatorView ( " 已 读 " , getResources ( ).getDrawable ( R.drawable.selector_tabnotice_unread ) ) ;
        fragmentTabHost.addTab ( fragmentTabHost.newTabSpec ( " r ead " ) . setIndicator (r eadView ),  FragmentRead.class, null);
        // getActivity ( ).getResources ( ) .getDimension ( R.dimen.dabc_action_bar_default_height_material ) ;
        // getActivity().getActionBar().getHeight ( ) ;
        // 解决 tabhost 会被压缩问题
        fragmentTabHost.getTabWidget().getChildAt ( 0 ) .getLayoutParams ( ) .height = ( i n t ) getActivity ( ).getResources ( ) .getDimension (R . D i m e n . A c t i o n  b a r _h e  i g h t) ;
        fragmentTabHost.setCurrentTab ( 0 ) ;
        fragmentTabHost.setOnTabChangedListener(new OnTabChangeListener ( )
```

```
{
            @Override
                public void onTabChanged(String tabId) {
                    // for ( i n t i = 0 ; i < fragmentTabHost.getChildCount ( ) ;  i + + ) {
                    // System.out.println ( fragmentTabHost.getTabWidget ( ) .getChildAt ( i ) .getTag ( ) ) ;
                    // }
                }
            });
            return view;
    }
```

图 3.6 公告图

3.6.6 九宫格

九宫格页面包括了个人设置，这里头像的点击是图片放大的操作，点击整个背景是直接进入个人设置。

3.6.6.1 我的格局

这里是管理所有协会的入口，包括创建协会，搜索协会信息，加入协会，管理协会等

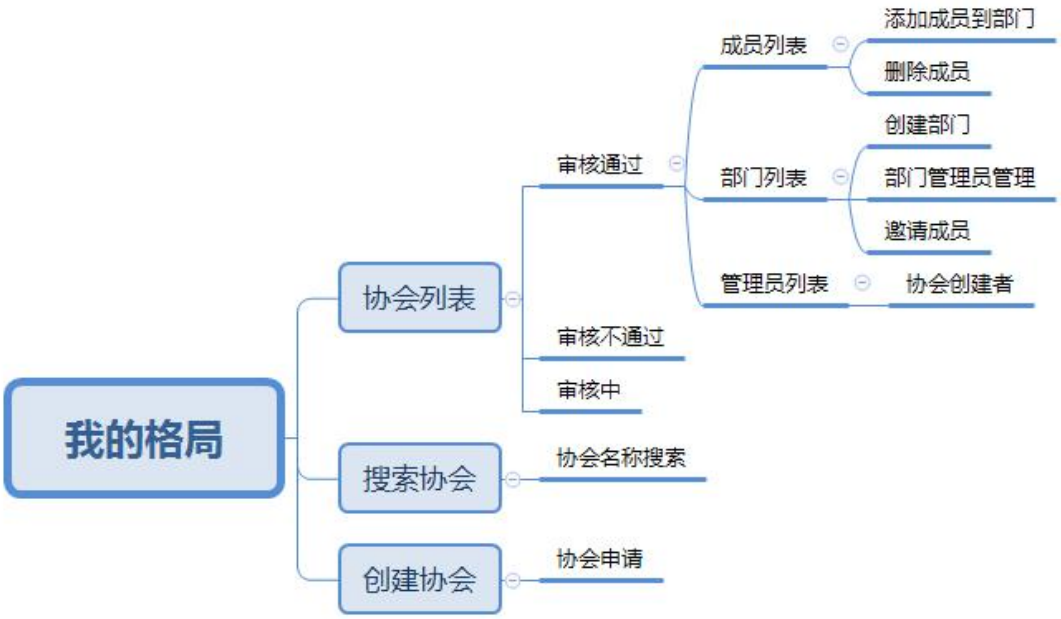

图 3.7 我的格局流程图

主要显示代码如下：

```
FunctionItem item = null;
        String [ ]  itemName = { " 我的格局 " , " 名片管理 " ,  " 发布公告 " ,
" 签到 " ,  " 日程 " ,  " 统计 " ,  " 直播 " , " 视频通话 " ,  " 设置 " } ;
        Int [ ]  itemIconIds = { R . d rawable . my_org_list_icon ,  R . Drawable .
icon_card ,  R . Drawable . icon_notice , R . Drawable . icon_sign , R . Drawable .
icon_day, R . drawable.icon_count ,  R . drawable.icon_tv
                , R . Drawable . icon_video , R . Drawable .i con_set } ;
        ArrayList < FunctionItem > items = new ArrayList <
UserCenterFragment.FunctionItem > ( ) ;
        int idx = 10;
        int index = 0;
        for ( String _name : itemName ) {
            item = new FunctionItem ( ) ;
```

```
            item.itemName = _name ;
            item.itemId = idx ;
            item.iconResId = itemIconIds [ index ] ;
            item.openEnable = true ;
            Idx + + ;
            Items.add ( item ) ;
            Index + + ;
        }
        funcationAdapter.setDatas ( items ) ;
        // functionListView.setAdapter ( funcationAdapter ) ;
        // functionListView.setOnItemClickListener ( th is ) ;
        funtionGridView.setAdapter(funcationAdapter    ) ;
        funtionGridView.setOnItemClickListener(this ) ;
```

3.6.6.2 名片管理

包括创建名片，名片列表，图片主要有两种形式获取，一种是手机本地，另一种调用本地手机拍照；名片列表主要展现当前所有名片信息，可删除，可编辑。

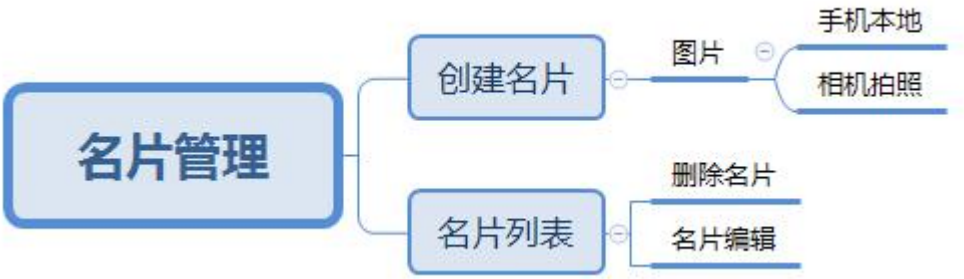

图 3.8 名片管理流程

3.6.6.3 发布公告

经过选择协会列表发布公告，如果没有协会列表则可以点击创建协会，带通过后可发布公告；发布公告可选择图片发布，图片主要以手机本地和手机拍照两种

方式获取，内增加了裁剪框，自动生成缩略图，带编辑文本后可发布公告。发布公告后，会推送一条消息，所属该协会的人便可收到该公告信息。

图 3.9 发送公告流程

3.6.6.4 签到

签到模块用到了百度地图提供的 sdk，打开自动定位，并且可根据移动拖拽的绝对坐标，反地理位置信息。

该功能的主要目的是为了实现一个打卡的功能，对各个协会可进行查看成员签到状态。

图 3.10 签到流程

3.6.6.5 日程

日历用到第三方材料设计的日历，三级缓存日历数据，增加月份滑动切换动画；自定义 dayView；对于装饰者模式，自定义增加已有日程数据的背景

图 3.11 日程流程

3.6.6.6 统计

尚未完成模块。

3.6.6.7 直播

直播功能是环信官网近期新增的功能；

协会的创建者才能开启直播，该属协会成员才能观看该协会的主播；

直播还拥有弹幕和发送礼物的功能

现在的缺陷就是直播卡顿的问题

图 3.12 直播流程

3.6.6.8 视频通话

暂时还没有开放。

3.6.6.9 设置

设置拥有这软件常规的功能：“关于我们”、“帮助”以及“软件更新”

关于我们主要显示安卓基本常用控件显示，内容固定，其中额外增加了隐藏点击的功能，例如点击联系方式的电话可以自动跳转到系统拨号功能，点击官方网址会调用系统浏览器功能访问网址；帮助功能是用户填写想对制作方提出建议问题

或者是反回馈软件体验问题。软件更新，点击后会获取服务器 json 信息来判断软件是否是最新的版本，如果是最新版本则弹出 toast 提醒用户软件意识最新信息，否者会弹出对话框提醒用户，用户可选择性更新，待下载完成后会自动安装，但是技术没有做到热修复的功能。

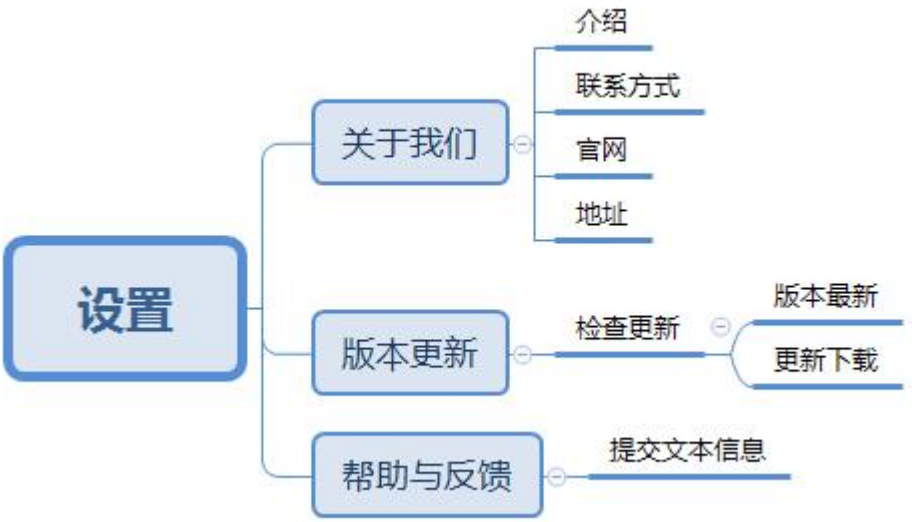

图 3.12 设置流程

3.6.7 全局搜索

全局搜索是对用户信息和服务器数据的全局检索，用户可输入相关信息，代码实时检测文本信息，不断 post 请求获取网络数据，实时更新 UI 信息。
目前能有协会信息和公告信息以及联系人信息。

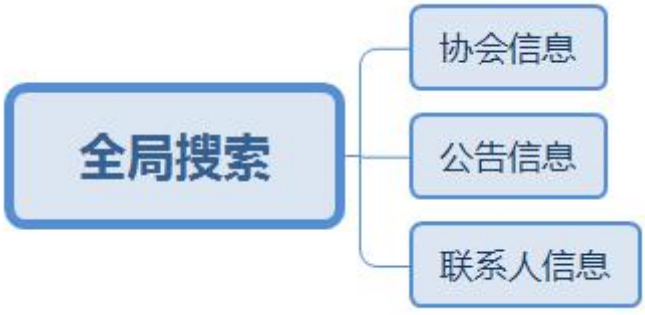

图 3.13 全局搜索流程

4 项目结构模块分析及测试

4.1 工程结构

项目主要分三大模块：代码界面层（YYFramework），项目框架层（QLBundle），第三方引用层（ulive-andoird-sdk）

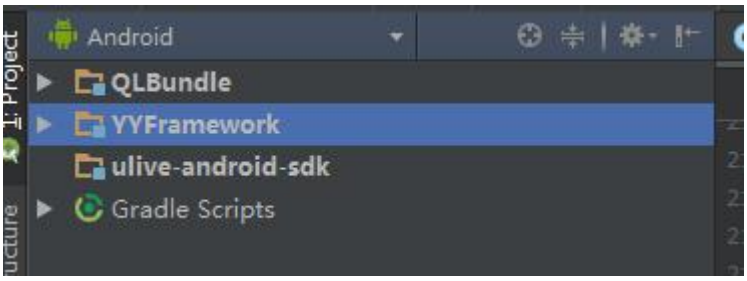

图 4..1 工程结构

4.1.1 代码界面层

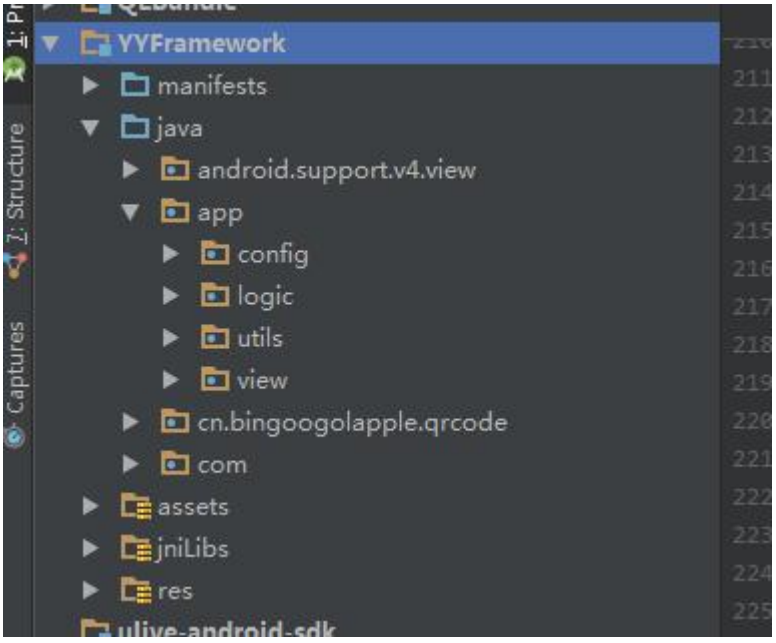

图 4.2 代码界面层

主要是部署显示层的代码，分两种：一种是类的模块，也就是操作业务相关的代码；一种是 xml 模块，也就是实体界面的处理。两种结合在一起就构成的了看到的界面显示。

该层也分许许多多的小模块包，也会包含基类的构造，应用的每个功能点都会

分一个模块包，这样边便于管理每个模块的功能，同时也方便自己测试的时候能够及时找到问题的出处。

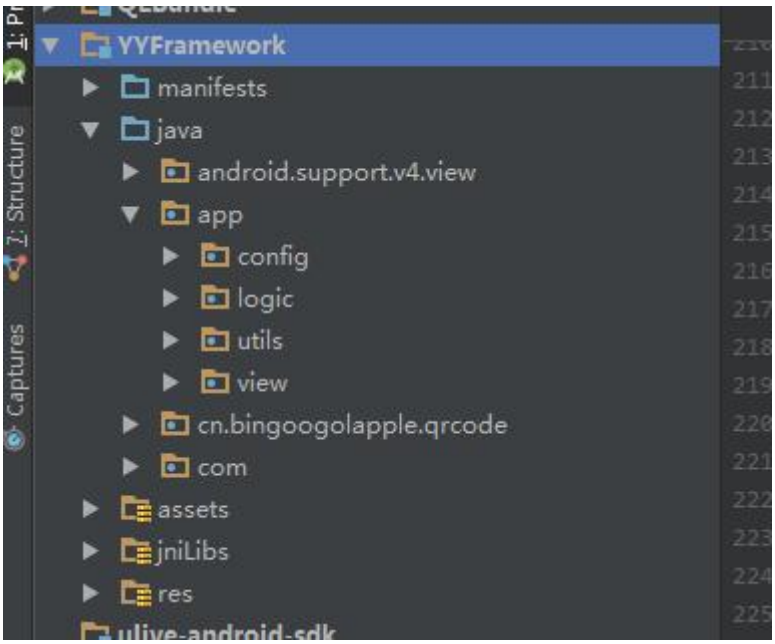

图 4.3 结构包

该层会有个模块包是专门存储和配置用户信息（config 包），这也包括网络所有的 api 接口信息，

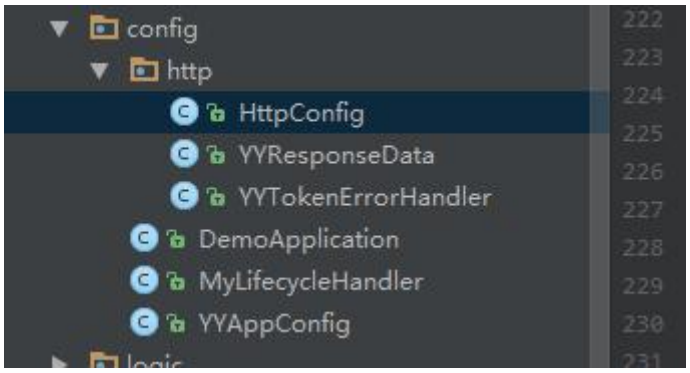

图 4.4Config 包结构

不仅还有 config 模块包而且还有工具包，一些公共的工具，比如日期工具、json 解析工具、数据库操作、debug 统计、对话框控制、线程控制、本地文件控制、二维码工具、图片处理工具、地图信息控制、java 常用方法封装、网络的再次封装、服务类的封装等。

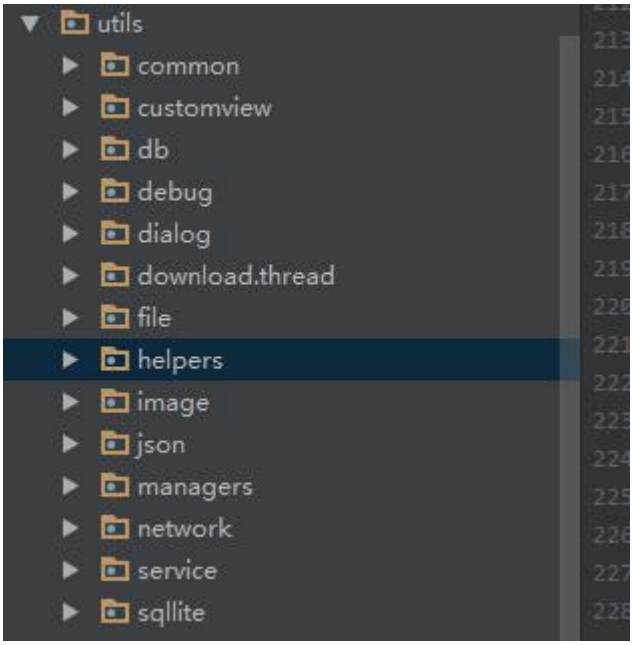

图 4.5 工具包结构

Com 模块是引用 github 的开源代码，常用的安卓控件满足不了需求的时候，为了方便快捷开发，可以引用，也可以自定义 view。

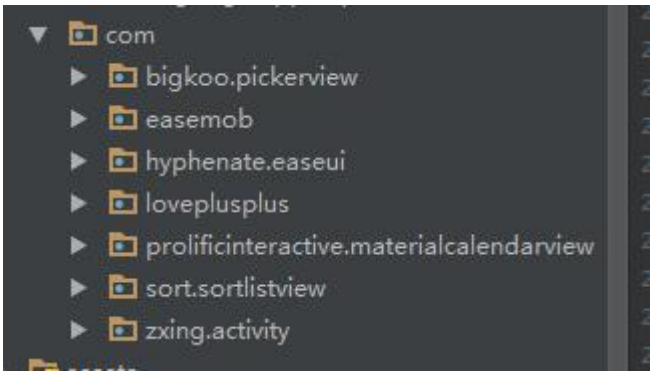

图 4.6Com 包

4.1.2 项目框架层

项目框架层主要是常规的网络请求配置和封装、图片加载的缓存封装等。

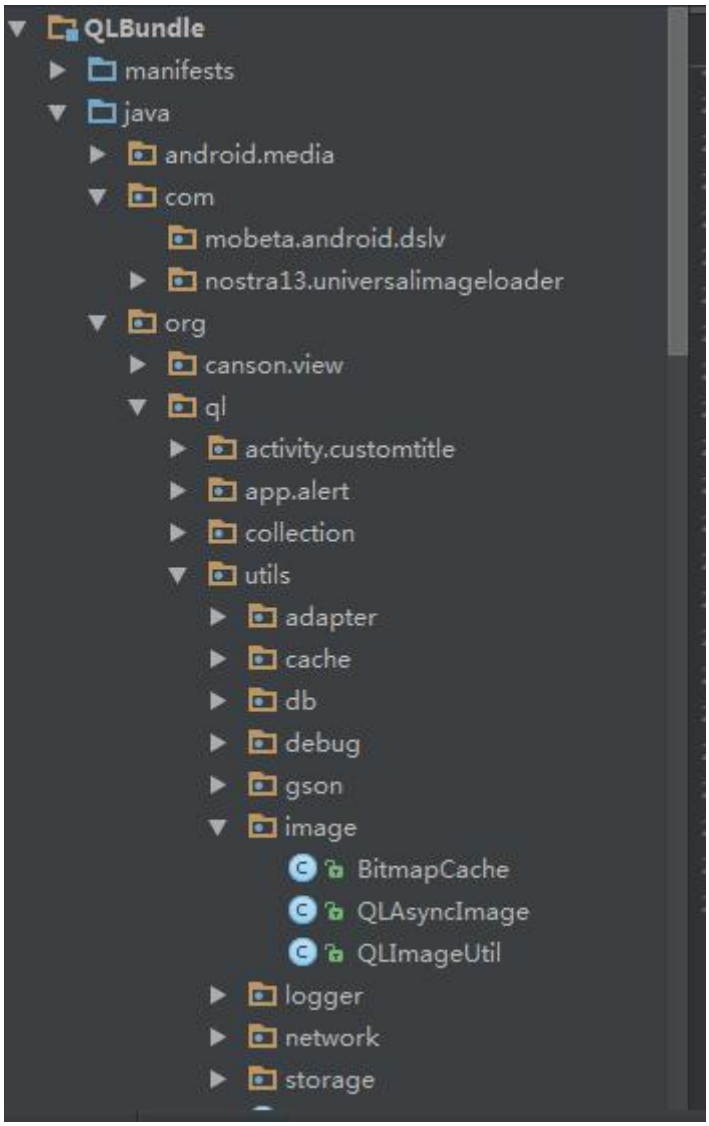

图 4.7 项目框架层

4. 1. 3 第三方库应用层

该层主要是环信的 sdk 应用。

5 测试

5.1 测试对象

已安装在手机的名为“格局”的 APP 应用

测试方案：

本测试是基于手机白盒测试，主要测试该项目所有的业务逻辑能否走通，UI 界面能否正常显示，内存是否溢出；同时分析程序是否存在逻辑性的错误，及时修复发现问题，进一步提高用户体验，

5.2 主要测试流程

5.2.1 登录测试

改手机号是已经注册账号，输入手机号后，点击登录操作；注册流程可以输入手机号，该手机号会收到验证码，然后继续填写信息注册。

图 5.1 登录测试流程

5.2.2 首页测试

登录成功后进入主界面（ MainActivity ）,根据界面显示，已成功登录。

获取环信服务器数据正常；根据时间戳排序正常；未读书显示统一。

图 5.2 首页测试

5.2.3 联系人界面流程测试

已添加联系人能正常显示在列表上，并能根据昵称首字母进行排序分组显示。

图 5.3 联系人测试流程

5.2.4 公告测试

根据界面显示，首先是协会列表的显示，并标志哪个协会有用户未读的公告数量。

图 5.4 公告测试

5.2.5 我的格局测试

填写创建协会的信息提交后，所创建协会和已加入的协会列表。

图 5.5 我的格局测试流程

5.3 测试结果和评价

由于测试界面太多，毕业论文受到限制，这里就不一一展示所有的业务流程。本章主要测试主要的系统部署及测试相关过程，经过测试代码验证，基于安卓的即时通讯软件的聊天功能具有着类似于微信的即时通讯，拥有单聊和群聊功能。手机号注册登录（ 暂不能支持第三方登录 ）图片浏览，聊天语音的播放，小视频的播放等；协会管理部分，修正了扫码加入协会失败的问题，在一定的程度上解决了列表分块的显示不完整的问题。其他的拓展模块如签到、日程、直播、版

本检查等，可以正常使用，至于页面的优化，有待提升；安卓内存控制模块做到了代码什么上的优化简洁，尽量控制 java 函数的合理使用，减少多余变量的声明，UI 绘制过程达到了正常用户的体验效果。日后测试，有待进一步的提升。

6 系统总结

本应用是即时聊天技术模块是基于环信的 sdk 实现；后台服务器数据的使用了阿里云。在开发的过程中，主要在 Android Studio 的平台和 java 语言上开发，即名为“格局”的 APP，在测试过程中，借助了多个安卓牌子手机，以及不同安卓版本上测试，提升 APP 兼容性。

在项目设置和开发工程中，主要经历了以下几个阶段：

1：对于开发平台软件的使用的提升，主要面向对象设计模式的学习，提高安卓技术水平，以及在 java 上能有比较深层的见解和使用；对于 Android 的知识有了进一步的提升，前沿应用框架的了解和计算机网络基础的学习等。

2：对项目进行详细的分析与整体架构的设计，画出项目的框架图，在开发平台上编写相关的 UI 设计。

3：在 Android 现有控件基础上，自定义一些相关的 View 使用，进行分层设计与实现；模块化设计。

4：每完成一个模块的时候并进行详细的测试；完成整个项目时，进行整体的测试。

经果这次的毕业设计的磨炼，对于 Android 互联网上有了更深的了解，也有了更多的知识来填充自己的大脑。了解到一个项目从需求到设计再到代表的编写，最后测试的完成而又复杂的流程，也耗掉了许多时间。同时能够不断进行对 Android 技术的探索，也帮助提高自己在实际的项目开发过程的效率，毕竟现在自己的 Android 技术是少。

“格局”项目能实现基本的功能，少不了自己实习公司技术的人员的建议和技术的培养，而且还要非常感谢老师对毕业项目的进展的合理掌控，以使我在学习过程中少走了不少的弯路，当然自身的知识水平也是最主要的。在解决问题的过程中，搜索互联网知识也使我能够快熟找到解决问题的本身。特别是国外的技术网站，比如 Stack overflow，借鉴了前人许多的开发经验。

参考文献

[1] java 核心设计 卷 1 基础知识(原书第 9 版)/(美)霍斯特曼(Horstmann,C.S.)，科内尔(Cornell,G.)著；周立新译. -北京：机械工程出版社，2013.11

[2] java 核心设计 卷 2 基础知识(原书第 9 版)/(美)霍斯特曼(Horstmann,C.S.)，科内尔(Cornell,G.)著；周立新译. -北京：机械工程出版社，2013.11

[3] William Cheng-Chung Chu,Han-Chieh Chao,Stephen Jenn-Hwa Yang,Chien Hung Liu,Shu-Ling Chen,Huang-Ke Chen. RobotDroid - A Keyword-Driven Testing Tool for Android Applications[M].IOS Press:2015.

[4] App 研发录：架构设计、Crash 分析和竞品技术分析/包建强著. -北京：机械工业出版社，2015.9

[5] Android 源码设计模式解析与实战/何红辉，关爱民著. -北京：人民邮电出版社，2015.11（2015.11 重印）

[6] Android 群英传/徐宜生编著. -北京：电子工业出版社，2015.9

[7] Android 开发实战/软件开发技术联盟编著. -北京：清华大学出版社，2013（软件开发实战）

[8] Sébastien Salva,Stassia R. Zafimiharisoa. Detection of Intent-Based Vulnerabilities in Android Applications[M].Elsevier Inc.:2014.[9]

[9] Android 高级进阶/顾浩鑫著. -北京：电子工业出版社，2016,11

[10] Andrew Hoog. Android software development kit and android debug bridge[M].Elsevier Inc.:2011.