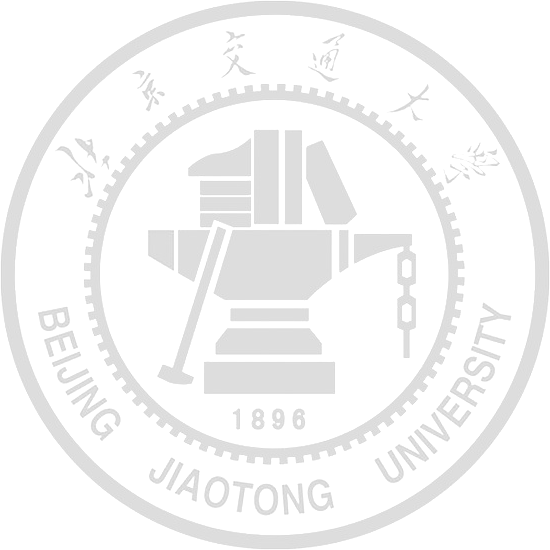


BEIJING JIAOTONG UNIVERSITY



移动应用开发项目

——实现文档

*项目名称****：****话心*

*组长****:*** *杨睿孜*

*组员****:*** *李芮萱、李雅婷、陈增剑、徐恬恬*

*指导老师****：****张顺利*

目录

[1.前端界面实现 3](#_Toc8059)

[1.1 界面实现简介 3](#_Toc12982)

[1.2 界面跳转流程及实现 4](#_Toc20052)

[1.2.1登录与注册 4](#_Toc31291)

[1.2.2用户端首页 7](#_Toc15926)

[1.2.3用户端消息页 12](#_Toc3942)

[1.2.4用户端故事页 13](#_Toc10207)

[1.2.5用户端个人中心 16](#_Toc20487)

[1.2.6咨询师端 19](#_Toc454)

[2. 前端关键技术实现 21](#_Toc12565)

[2.1前端耗时操作与界面更新的分离 21](#_Toc19447)

[2.2前端对服务器的网络传输 22](#_Toc23955)

[2.3前端的图片选择与显示 22](#_Toc11241)

[2.4文字识别功能的实现 24](#_Toc31127)

[2.4.1文字识别功能模块简介 24](#_Toc24558)

[2.4.2文字识别功能模块方法实现 25](#_Toc4726)

[2.4.3文字识别功能模块流程 26](#_Toc3879)

[3. 数据库表结构实现 27](#_Toc19813)

[4. 后端服务器实现 28](#_Toc25130)

[4.1服务器端技术路线 28](#_Toc12303)

[4.1.1技术路线流程图 28](#_Toc29109)

[4.1.2技术路线说明 28](#_Toc31472)

[4.2服务器端核心代码 29](#_Toc17448)

[4.2.1 ApiResponse应答类 29](#_Toc2934)

[4.2.2 Controller中对数据的响应与返回 30](#_Toc6042)

# 1.前端界面实现

## 1.1 界面实现简介

在界面设计上，为达到app界面的美观与清晰，本项目用到了多种布局与控件，如RelativeLayout、LinearLayout、ScrollView、ListView、Imageview；为使消息提醒准确清晰、优化用户选择体验，使用了RadioGroup、Toast、AlertDialog、ProgressDialog、DatePickerDialog、TimePickerDialog等多种组件，且由于界面复杂，控件繁多，本项目使用Databinding绑定控件，简化代码。

由于本软件普通用户端具有多个界面，就需要将UI元素和Activity融合成一个模块。我们决定使用Fragment类来方便的达到这个效果。Fragment类似一个嵌套Activity，可以定义自己的layout和自己的生命周期。多个Fragment可以放在一个Activity中，本软件普通用户端我们将聊首页、消息、故事和个人页组合到MainActivity中，来创建一个多面界面并且可以在多个Activity中重用一个Fragment。

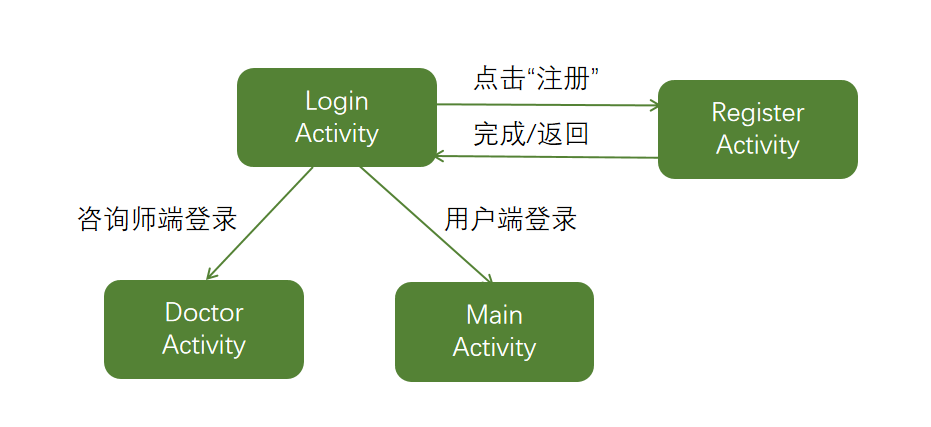
又由于本软件的咨询师端的接待界面中具有“预约”与“消息“两个子界面，因此使用ViewPager组件，将成装View的数组加入到PagerAdapter并设置为ViewPager的Adapter，即可实现子页面的滑动效果。

同时，咨询师列表、聊天界面与故事列表等等界面涉及到较长的滚动窗口，实现该功能使用为ListView设置BaseAdapter实现。其中ListView具有的缓存机制可以节省系统资源；聊天消息的更新与内置数据库同步，BaseAdapter结合ListView可以实现动态更新列表界面。

## 1.2 界面跳转流程及实现

### 1.2.1登录与注册

（1）跳转流程



（2）相关类实现

LoginActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 布局初始化，设置标题为“用户登录”，权限检测。 |
| void onClick(View v) | 根据不同点击事件进行不同操作。 |
| void sendCode() | 在子线程中通过后台发送验证码 |

RegisterActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 布局初始化，设置标题为“用户注册”。 |
| void onClick(View v) | 根据不同点击事件进行不同操作。 |
| void sendCode() | 在子线程中通过后台发送验证码 |
| void onActivityResult(int requestCode, int resultCode, Intent data) | 得到PictureSelector的选择结果，并使用Glide框架加载图片到页面上显示。 |

MainActivity类：

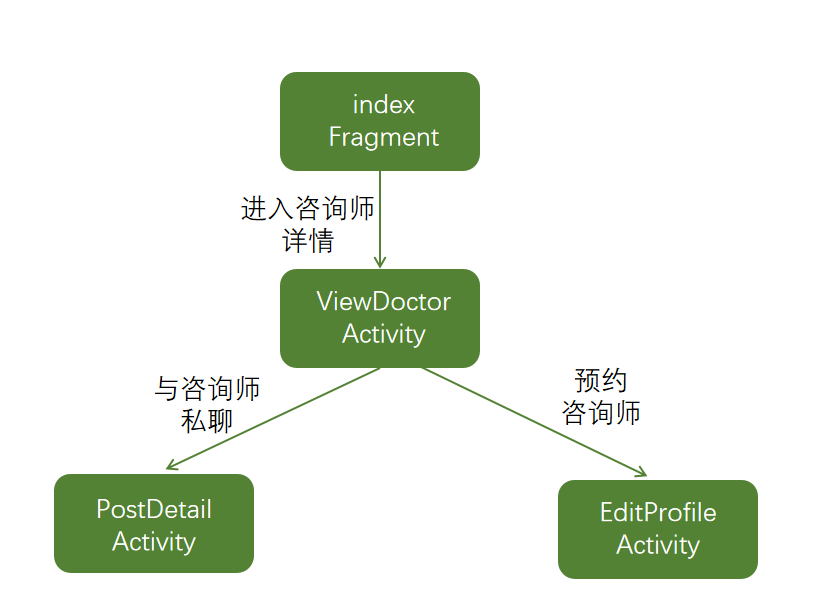
|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 布局初始化，初始化导航按钮。 |
| void onClick(View v) | 将不同按钮的点击事件与以下方法关联起来。 |
| void loadIndex() | 切换页面到首页，使用transaction事务,替换fragment,更改app标题为“首页”。 |
| void loadMessage() | 切换页面到信息页，使用transaction事务,替换fragment,更改app标题为“信息”。 |
| void loadStory() | 切换页面到故事页，使用transaction事务,替换fragment,更改app标题为“故事”。 |
| void loadMine() | 切换页面到个人中心页，使用transaction事务,替换fragment,更改app标题为“我的”。 |
| void changeNavigation(int pos) | 根据当前界面，处理导航按钮选中颜色。 |

DoctorActivity类

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 重写方法。在其中初始化界面、初始化底部导航按钮数组（为了方便选中变色）、设置默认自动加载接待管理界面 |
| void onClick(View v) | 重写方法（该类继承View.OnClickListener）。根据点击的按钮做出相应的显示模块的响应。 |
| void loadAppointmentManage() | 加载预约管理模块，设置title为“接待”；修改接待按钮颜色为选中后的颜色 |
| void loadDoctorMine() | 加载我的模块，设置title为“我的”；修改文档按钮颜色为选中后的颜色 |
| void changeNavigation(int pos) | 改变选中的导航栏按钮的颜色 |

### 1.2.2用户端首页

1. 跳转流程



1. 相关类实现

IndexFragment类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) | 布局初始化，并返回当前View |
| void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) | 调用初始化方法更新界面，为顶部也卡设置切换事件，设置刷新监听，设置咨询师列表点击事件。 |
| void initData() | 从服务器获取数据，初始化页面显示。 |
| void initBanner() | 从后端获取轮播图资源并显示。 |
| void changePoint() | 更新轮播图下方圆点。 |
| void onDestroy() | 重写页面销毁事件。 |
| void changeNavigation(int pos) | 根据当前界面，处理导航按钮选中颜色。 |

ViewDoctorActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 重写方法。在其中初始化界面、设置界面title、获取intent、调用其他成员函数获取咨询师数据 |
| void initData() | 初始化咨询师数据。开启子线程从后台获取相应数据，并将数据显示到界面上 |
| void initFollow() | 初始化关注数据。从后台获取该用户有没有关注该咨询师。如果已关注，按钮显示“取消关注”，如果未关注，按钮显示“关注” |
| void initLikeCount() | 初始化点赞数量。从后台获取该咨询师获赞数以及该用户有没有点赞该咨询师。如果已点赞，按钮显示“取消点赞”，如果未点赞，按钮显示“点赞” |
| void initComments() | 初始化评论列表。开子线程从后台获取评论列表数据，并将相关数据显示到界面上 |
| void onClick(View v) | 监听器。如果点击的是私聊按钮，将咨询师姓名用户名传递并跳转到私聊界面；点击关注或取消关注则开启子线程联系后台更新数据；点击点赞或取消点赞则开启子线程联系后台更新数据；点击评价则跳出对话框并将数据发送后台，并自动刷新界面显示评论；点击预约则跳转预约界面并传递咨询师数据 |

ChatActivity类：

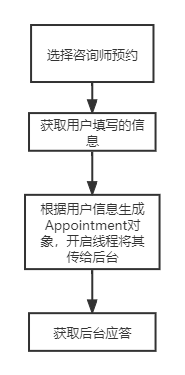
|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreate(Bundle savedInstanceState) | 初始化聊天窗口框架：包括初始化布局与标题 |
| void initData() | 初始化或刷新数据：从服务器中获取聊天记录列表，并为listView设置Adapter，若正在刷新则返回。 |
| void onClick(View v) | 点击按钮，向服务器发送消息并刷新消息列表，更新界面。 |
| void finish() | 重写方法，移除消息的监听任务。 |

AppointmentActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 重写方法。在其中初始化界面、设置界面title、获取intent、设置选择日期焦点事件 |
| void onClick(View v) | 重写方法（该类继承View.OnClickListener）。根据点击的按钮做出相应的响应。 |
| void showDatePicker() | 选择日期的函数，其中使用了Calendar |
| void showTImePicker() | 选择时间的函数，其中使用了Calendar。选择完成后设置选择的日期时间文本到文本框。 |

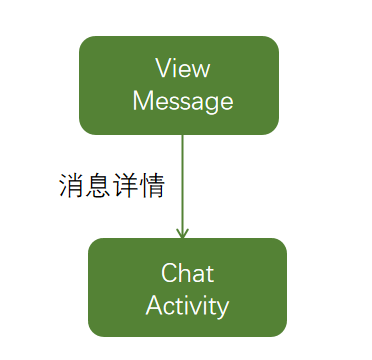
1. 关键模块实现流程

以预约模块实现流程为例：



### 1.2.3用户端消息页

1. 跳转流程



1. 相关类实现

ViewMessage类：

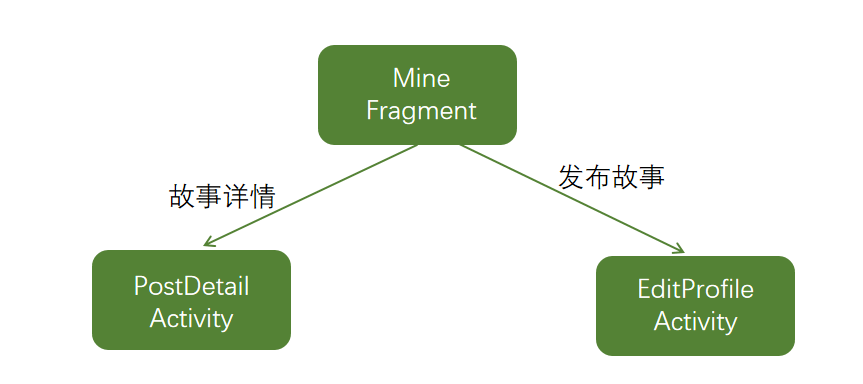
|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| SetOnItemClickListener() | 设置列表点击跳转 |
| setOnItemLongClickListener（） | 设置列表长按删除与取消 |
| initData() | 初始化数据 |
| onResume() | 设置信息的定时刷新线程 |

ChatActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreate(Bundle savedInstanceState) | 初始化聊天窗口框架：包括初始化布局与标题 |
| void initData() | 初始化或刷新数据：从服务器中获取聊天记录列表，并为listView设置Adapter，若正在刷新则返回。 |
| void onClick(View v) | 点击按钮，向服务器发送消息并刷新消息列表，更新界面。 |
| void finish() | 重写方法，移除消息的监听任务。 |

### **1.2.4用户端故事页**

1. 跳转流程



1. 相关类实现

ViewPost类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| ViewPost(Context context) | 构造方法，在其中初始化布局，设置按钮点击事件。 |
| View getRoot() | 返回最外层View。 |
| void initData() | 初始化数据：从后端获取数据并使用adapter显示数据，更新故事列表。 |
| void onClick(View v) | 重写“发布”按钮点击事件，进入PostActivity。 |

PostActivity类：

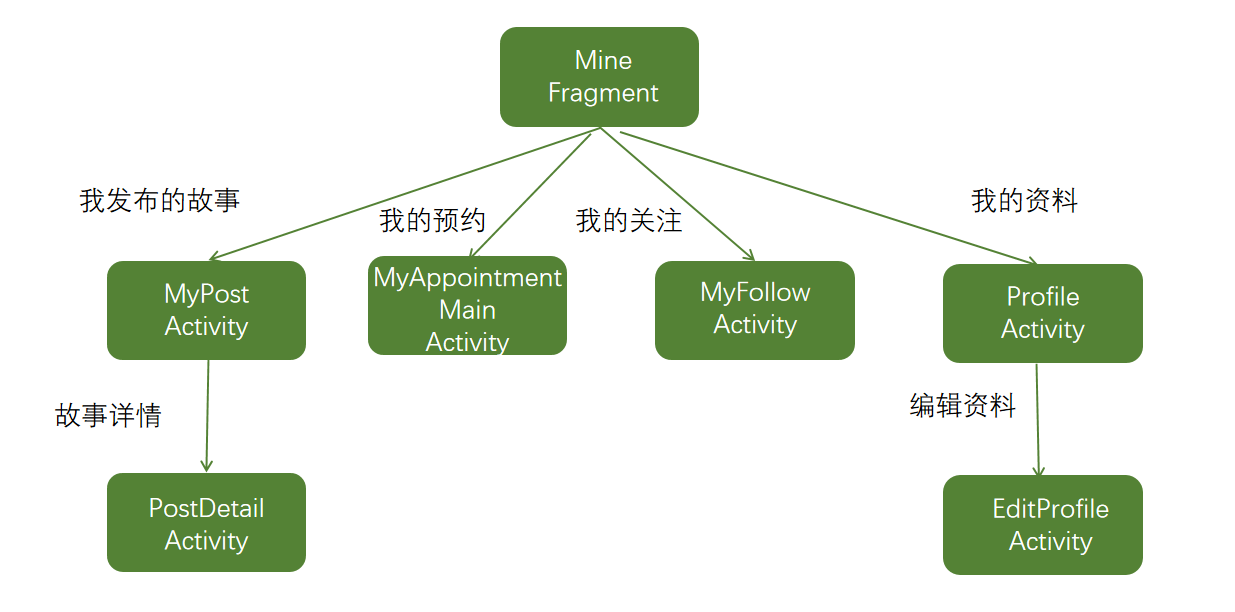
|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 初始化界面并设置标题。 |
| void onClick(View v) | 重写不同按钮的点击事件，构造发布对象并转给服务器，显示提示对话框。 |
| Void addImage(List<LocalMedia> localMedia) | 将selectlist中的图片路径写到images中。 |
| Void showImageList() | 使用Glide框架，显示选择的图片列表。 |
| void onActivityResult(int requestCode,int resultCode,Intent data) | 回调函数，将PictureSelector中选择的结果回调。 |

PostDetailActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| void onCreate(Bundle savedInstanceState) | 初始化布局与数据。 |
| void initData() | 初始化数据，从服务器获取故事详情并填充页面，构造回复列表。 |
| View createView(int i, Reply reply) | 创建回复视图，填充回复详情。 |
| void saveImage(String path) | 保存图片并发送广播更新相册。 |
| void ocr(String path) | 获得图像缓存，调用文字识别模块，并进入新界面，得到识别结果。 |
| void onClick(View v) | 回复按钮的点击事件，显示回复对话框，构造回复数据并传给后台。 |

### **1.2.5用户端个人中心**

1. 跳转流程



（2）相关类实现

MineFragment类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreateView | 绑定视图函数 |
| initData() | 数据获取与绑定 |
| onClick(View v) | 设置点击事件 |

ProfileActivity类：

|  |  |
| --- | --- |
| onCreate(BundlesavedInstanceState) | 个人资料页面绑定 |
| void onResume() | 判断用户类型，显示不同信息 |

EditProfileActivity类：

|  |  |
| --- | --- |
| onCreate(BundlesavedInstanceState) | 修改资料页面绑定，判断用户类型绑定资料 |
| public void onClick(View v) | 点击事件判断，修改信息上传 |

MyAppointmentMainActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| setOnItemClickListener（） | 设置列表点击监听 |
| initData() | 从数据库中读取预约信息，填充到列表 |
| MyAdapter() | 为预约列表设置适配器 |

MyPostActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreate(Bundle savedInstanceState) | 初始化布局，设置按钮监听，实现短按进入故事详情；长按选择“删除”或“取消”。 |
| initData() | 从数据库中读取故事，填充我的故事列表。 |

MyFollow类：

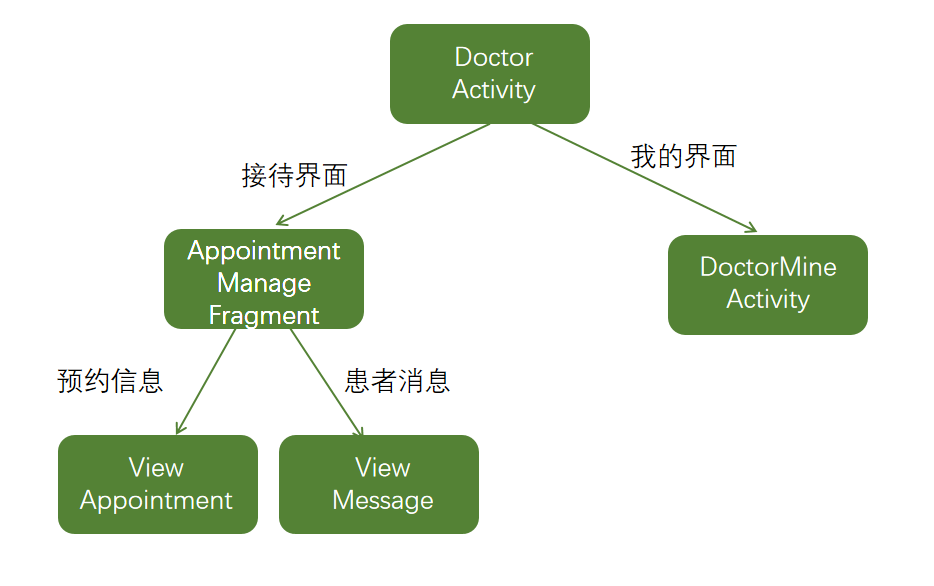
|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreate(Bundle savedInstanceState) | 初始化布局，初始化数据 |
| initData() | 从数据库中读取关注信息，并设置adapter。 |
| class MyAdapter extends BaseAdapter | 内部类，定义关注列表适配器，在其中加载布局，设置取消关注事件，并设置监听使点击进入咨询师详情界面。 |

ChangePasswordActivity类：

|  |  |
| --- | --- |
| 方法名称 | 方法功能 |
| onCreate(Bundle savedInstanceState) | 初始化布局 |
| void onClick(View v) | 重写按钮点击事件，向服务器发送修改密码对象。 |

### 1.2.6咨询师端

（1）跳转流程



1. 相关类实现

DoctorActivity类:

|  |  |
| --- | --- |
| void loadAppointmentManage() | 加载预约管理模块，设置title为“接待”；修改接待按钮颜色为选中后的颜色 |
| void loadDoctorMine() | 加载我的模块，设置title为“我的”；修改文档按钮颜色为选中后的颜色 |
| void changeNavigation(int pos) | 改变选中的导航栏按钮的颜色 |

ViewAppointment类：

|  |  |
| --- | --- |
| void initData() | 开启子线程从后台获取预约数据；如果有预约数据，设置适配器显示预约列表 |

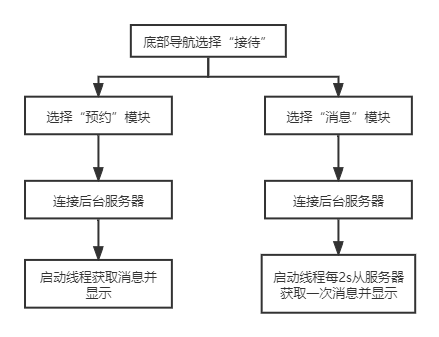
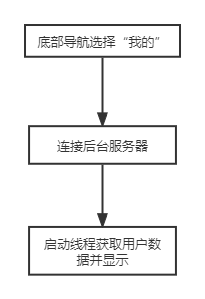
ViewMessage类：

|  |  |
| --- | --- |
| void initData() | 开启子线程从后台获取消息数据；如果有消息，设置适配器显示消息列表 |
| Runnable runnable() | 开启线程，每2秒从后台获取一次消息 |
| boolean onItemLongClick(AdapterView, View, int position, long id) | 监听器方法重写。当长按某个消息时，可选择删除。若选择删除则会开启线程通知后台删除信息并更新消息列表。 |

DoctorMineFragment类：

|  |  |
| --- | --- |
| void initData() | 开启子线程从后台获取用户资料、头像、获赞数量以及粉丝列表 |

（3）模块实现流程



# 前端关键技术实现

## 2.1前端耗时操作与界面更新的分离

当界面中需要网络传输或倒计时等耗时操作时，为保证界面更新及时不卡顿，本项目使用多线程技术、Handler对象及其post方法，实现耗时操作与界面更新的分离。

当前端需要和服务器传输数据时，均使用新线程完成耗时操作，以下图为例，当用户端需要从服务器获取咨询师详情时，用户端开辟新线程，调用工具类与服务器进行http通信，此时，为更新用户端界面，则使用在主线程中创建的handler对象，将主线程需要进行界面更新的代码放入消息队列中，以实现界面的及时更新，而不被耗时操作卡顿。



## 2.2前端对服务器的网络传输

此项目中，前端使用Okhttp框架及相关类，并通过Gson进行序列化与反序列化，从后端获取统一的应答类对象ApiResponse返回，方便信息的传输。

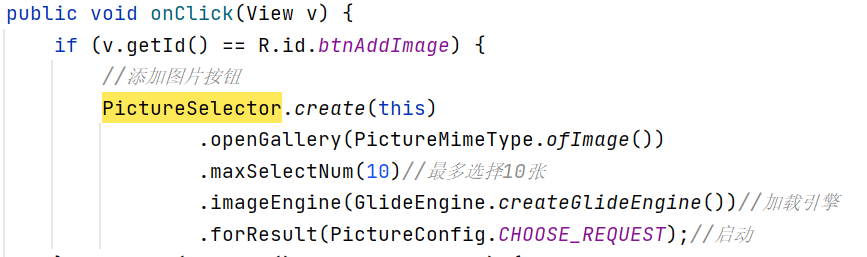
以下图注册时为例，当用户输入信息无误时，前端首先构造一个user类的对象，调用HttpUtil工具类中的register方法，并在此方法内创建OkHttpClient，为发送Post类的请求，首先构建RequestBody类对象，将user对象通过Gson.toJson方法进行序列化，并存入requestBody中；而后构造request对象，在post中放入requestBody；拿到call后用call开始请求，最终从服务器返回response，再次使用Gson进行反序列化，将response中的字符串转化为应答类对象ApiResponse，最终返回给调用此方法的模块取出对象进行界面的更新。



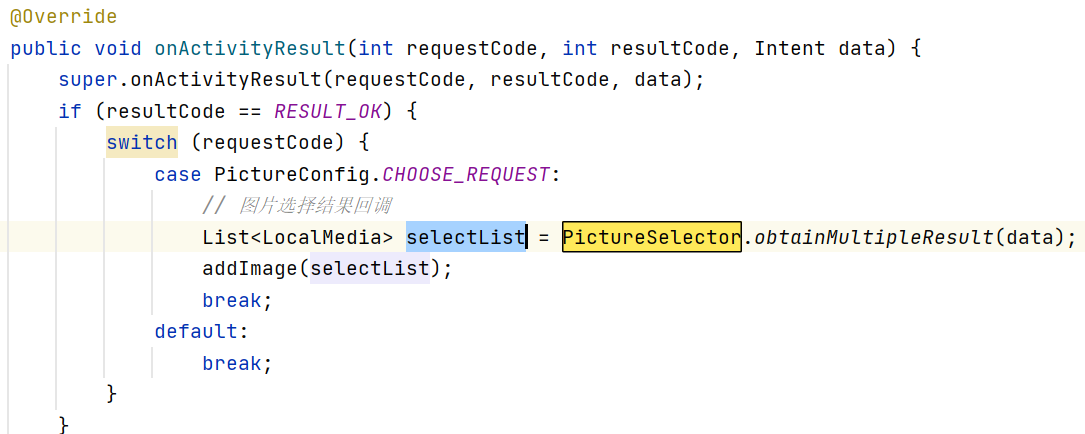
## 2.3前端的图片选择与显示

在此项目中，用户上传头像，发布故事等等功能均需要从手机相册中选择图片，并显示到对应的界面上。本项目采用图片选择框架PictureSelector，图片加载框架Glide完成这一功能。

以用户发布故事为例，在发布故事页面，当用户点击添加图片按钮，即调用PictureSelector.create方法进入相册，设置选择类型，最大图片数量，设置加载引擎，最终启动。



启动后将调用onActivityResult回调函数，在其中可以通过PictureSelector.obtainMultipleResult(data)获取到选择的图片结果，将图片结果传给addImage函数。



在addImage函数中，selectlist中的图片路径被写到images中，并调用showImageList显示选择的图片列表。



在showImageList函数中，从images中循环读取已添加图片的本地路径，并调用Glide框架，根据url显示图片。



## 2.4文字识别功能的实现

### 2.4.1文字识别功能模块简介

当用户在故事会模块长按图片，可选择文本识别。文本识别功能借助百度OCR实现。选择文本识别后，会调用OCRUtil类里的accurateBasic函数。这个函数接收故事会传来的缓存图片路径，然后生成一个临时文件，为了防止图片太大，调用sizeCompress函数将图片压缩后再放入这个临时文件，然后调用工具类AuthService的getAuth函数获取token，然后调用OCR的工具类HttpUtil的post方法把图片传给百度图片识别接口，等待返回结果。结果返回后，跳转界面，将结果显示在新界面，当用户点击复制结果按钮会使用ClipboardManager将文本复制到系统粘贴板。

### 2.4.2文字识别功能模块方法实现

AuthService类：

|  |  |
| --- | --- |
| String getAuth() | 获取API访问token |

FileUtil类：

|  |  |
| --- | --- |
| String readFileAsString(String filePath) | 读取文件内容，作为字符串返回 |
| byte[] readFileByBytes(String filePath) | 根据文件路径读取byte[] 数组 |

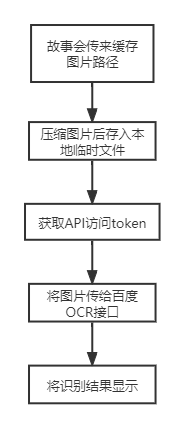
HttpUtil类：

|  |  |
| --- | --- |
| String post(String, String, String) | 向百度图片文本识别接口提交请求 |

OCRUtil类：

|  |  |
| --- | --- |
| OCRResult accurateBasic(String filePath) | 调用上述各工具类的函数，最终产生Gson格式的识别结果 |
| sizeCompress(Bitmap bmp, File file) | 压缩图片，采用尺寸压缩的方法，通过缩放图片像素来减少图片占用的内存 |

2.4.3文字识别功能模块流程



# **数据库表结构实现**

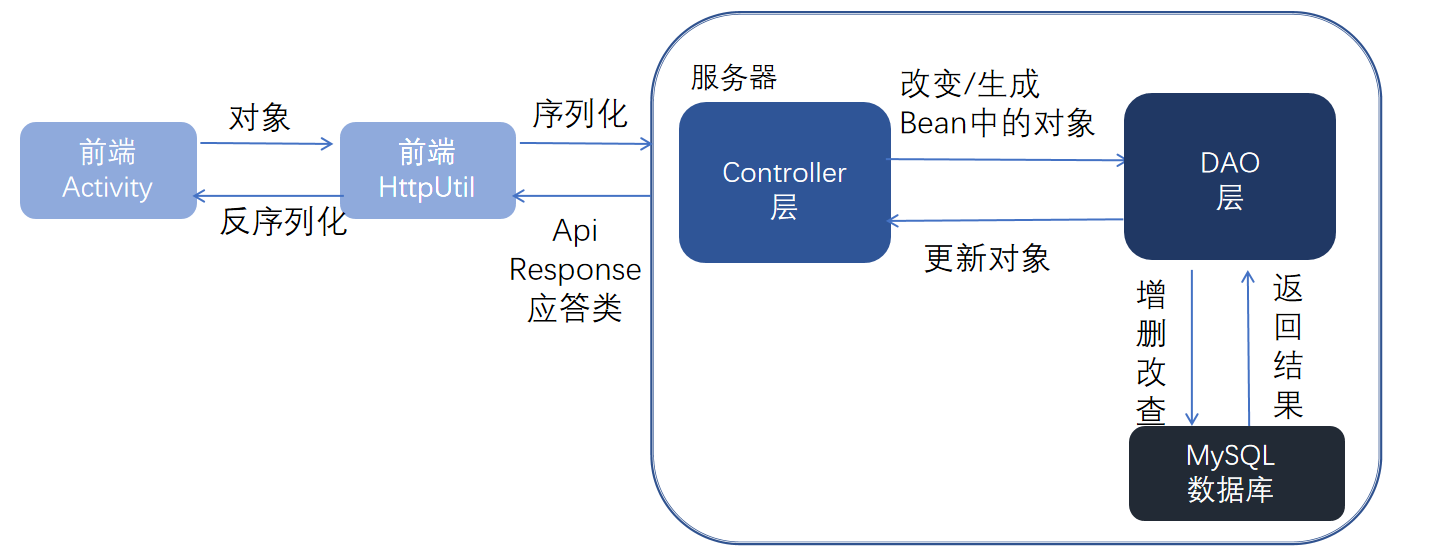
根据设计文档，在mysql中实现了数据库的表结构，其ER图如下：



# **后端服务器实现**

## 4.1服务器端技术路线

### 4.1.1技术路线流程图



### 4.1.2技术路线说明

在后端设计的基础上，后端实现部分明确了后端服务器的架构为Maven搭建springboot项目，通过JPA框架，建立 Java 对象与数据库表之间的映射关系，通过直接操作对象来实现持久化，简化了操作的繁杂度。采用分层实现的思想，使代码结构清晰，便于修改：  
Bean层：存放实体类，使用面向对象的思想简化数据传输，方便数据存储；

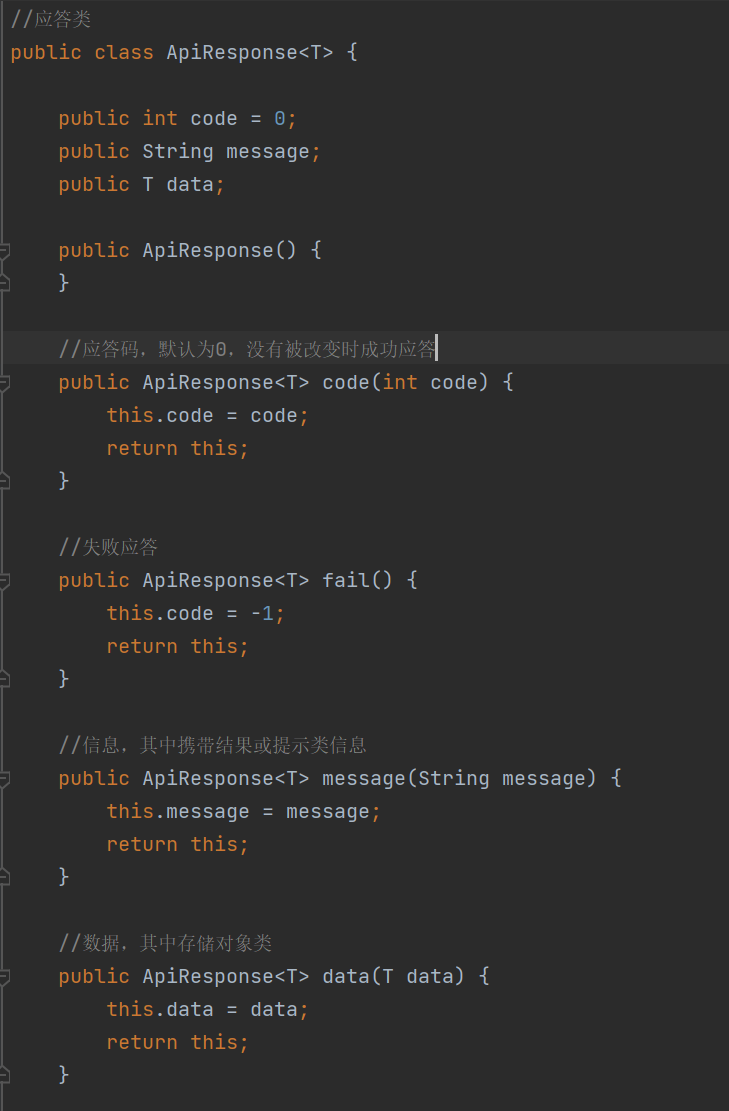
DAO层：通过sql语句与数据库进行交互，实现对数据库的增删改查。

Controller层：接收前端发送的请求，进行一系列处理后将结果封装在ApiResponse这一统一应答中返回给前端。

## 4.2服务器端核心代码

### 4.2.1 ApiResponse应答类

ApiResponse是一个泛型数据，所有与服务器通信需要应答的请求，服务器统一返回此类型数据，可以减少很多数据判断的工作量，其定义如下：



### 4.2.2 Controller中对数据的响应与返回

以UserController中的follow为例，注解@Getmapping用于定义处理浏览器发送的get请求，后面的url用于定义处理请求的相对路径。由于UserController的定义前存在注解@RequestMapping("/user/")，即此处方法用户接收"/user/follow"下的get请求。当服务端收到请求后，创建Follow类型的实体，并将用户名与咨询师名赋予实体中的属性，再使用JpaRepository中的save方法存入数据库中，最后返回给前端一个应答类的对象，提示信息为“关注成功”。  
