系统总体设计与实现文档

界面实现完成人：张楚楚、杨鹏、张明皓

功能模块完成人：付自建、樊成帅

1 系统设计与实现

1.1 系统设计

1.1.1 架构设计

本系统利用第三方SDK集成，整体采用MVP模式进行了架构的设计，将程序分为了三层：Model模型层、View视图层、Presenter业务逻辑层，工具上采用Android Studio和Android SDK结合开发完成。

View视图层即UI界面，它包含了.XML文件、Resource资源文件和一些View接口，负责所有界面的布局和实现，最终在Activity中使用setContentView（）进行显示。View层是用户与系统交互的重要途径，主要有注册界面、登陆界面、会话界面和聊天界面。视图层中的界面进行数据展示时，它可以通知Presenter层需要加载数据，然后Presenter层操控Model层调取数据，传递到View层。

Presenter层是MVP模式中最为重要的部分，它主要包括各个模块的Presenter接口类，其中实现了各种各样的功能方法。作为View和Model之间交互的桥梁，它定义并实现了用于请求Model层中数据调用的接口，完成View和Model之间的数据交换，同时也降低了代码的耦合度。Presenter不仅负责系统业务逻辑的实现，还提供了封装好的功能接口，供View层调用，比如登陆模块中的Presenter需要审查账号和密码是否正确，调用自身中的验证方法并返回登陆成功或者失败的提示给登陆界面。它实现了Model层中定义的接口，并使用Model层中的逻辑，操纵相应的View进行数据变更。

Model即模型层，它包括了网络数据和SQLite数据库等，该层定义了响应数据请求和一些数据操作的接口，主要负责程序中应用数据逻辑的处理，比如数据的请求、解析或者过滤等操作。在一些功能模块中，可能没有Model层。

1.1.2 界面设计

本系统在界面设计上以Flat UI，即扁平化风格为主，整体色调以深蓝色、天蓝色和绿色为主，扁平化界面设计可以使系统的按钮和选项变的较少，突出信息和系统的工作方式，UI视图更加简洁美观，给用户带来较好的体验。系统的扁平化界面设计甚至能够减少待机时间、降低系统消耗和提高运算速度。依据大多数人的操作习惯和即时通讯系统的特点，界面主要分为注册界面、登陆界面、主界面、会话界面、通讯录界面、添加好友界面、联系人界面、聊天界面、语音通话界面和视频通话界面等。所有界面的操作设计流程图如图1所示。

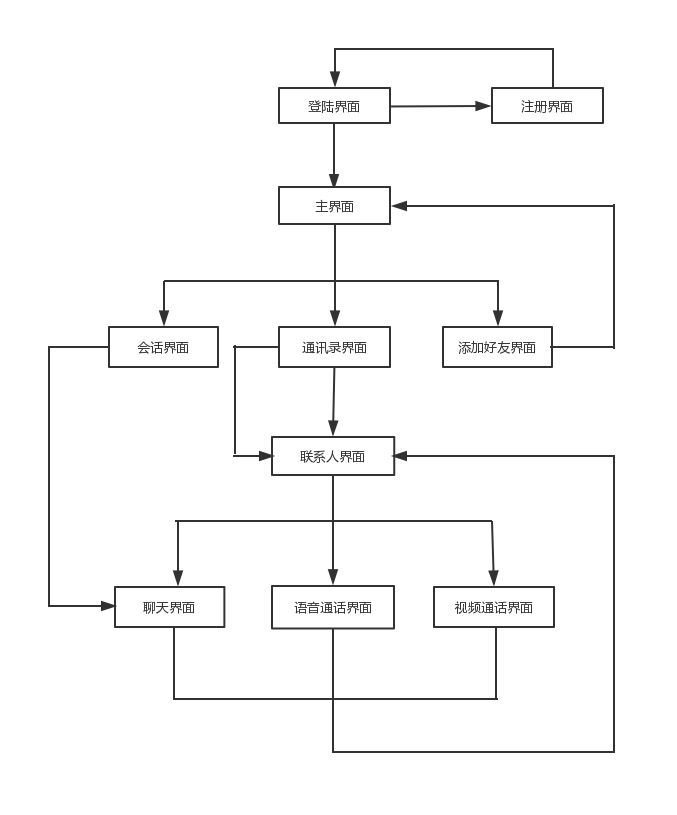


图1 界面操作流程图

界面的详细设计如下安排：

1. 注册界面：做为新用户必须要使用的界面，为了清楚明了和操作方便，注册界面中只包含了手机号和密码的两个输入文本框和一个注册按钮，整体布局和谐，按钮使用绿色更加醒目。
2. 登录界面：登录界面是进入系统的第一个页面，所以一定要有跳转到注册界面的按钮，最终设计将按钮布置到页面右上角，这样更加引人注意。布局中有两个文本输入框分别需要输入已经注册的账号和密码，下方是一个绿色登录按钮。效果如图2所示。



图2 登录界面

1. 主界面：主界面是系统登录成功后跳转到的界面，也是整个即时通讯软件的主要展示界面，包括了顶部右上角的添加好友图标，底部的会话和通讯录两个导航。
2. 会话界面：会话界面主要是存放聊天之后和每个好友的聊天记录，呈碎片化管理，包含在主界面中，是主界面的默认导航。
3. 通讯录界面：通讯录界面是主界面的第二个导航，其中主要包括好友请求的通知和每个已经添加的好友，页面的右上角有一个添加好友的图标，引导用户快速找到添加好友的功能，每个好友显示包含头像、昵称和手机号。
4. 添加好友界面：该界面是由主界面右上角的提示图标进入，主要负责搜索想要添加的好友账号，然后发出请求，通知对方，所以其中包含了一个输入文本框用于输入账号和一个发送请求按钮。
5. 联系人界面：该界面由通讯录界面进入，其中包括了联系人的头像、昵称信息以及对好友进行通讯的方式选择，为了使目的清晰明确，整体分上下布局，其中上方包含了联系人的昵称、头像；下方底部分布三个聊天方式的选择按钮。效果如图3所示。

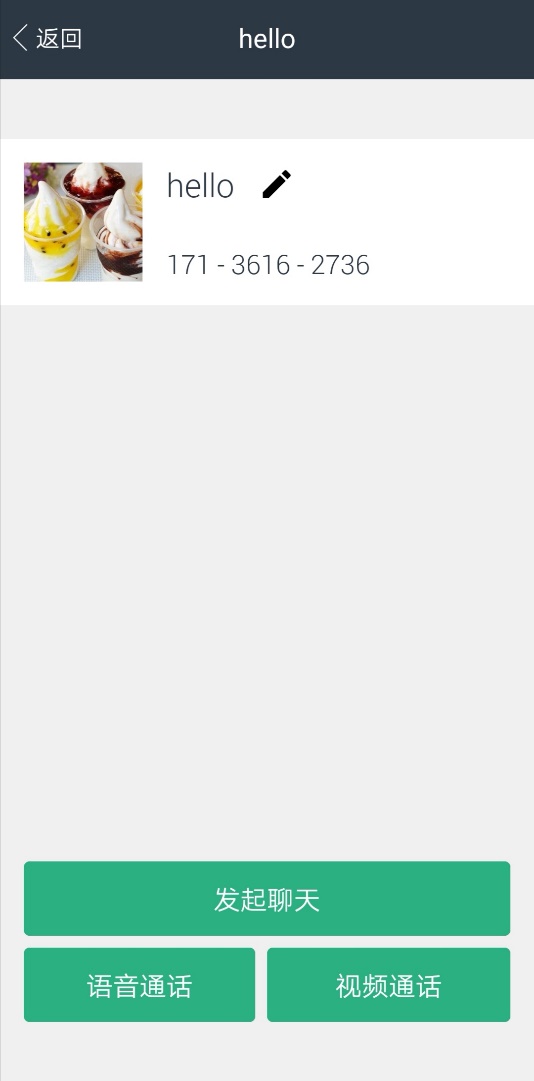


图3 联系人界面

1. 聊天界面：聊天界面可以由会话界面直接点击进入到曾经的历史聊天记录或者从联系人界面点击发起聊天按钮进入，主要负责显示双方的最近聊天信息和时间，整体布局就是以聊天内容为主，底部有一个聊天输入框和一个发送按钮。效果如图4所示。

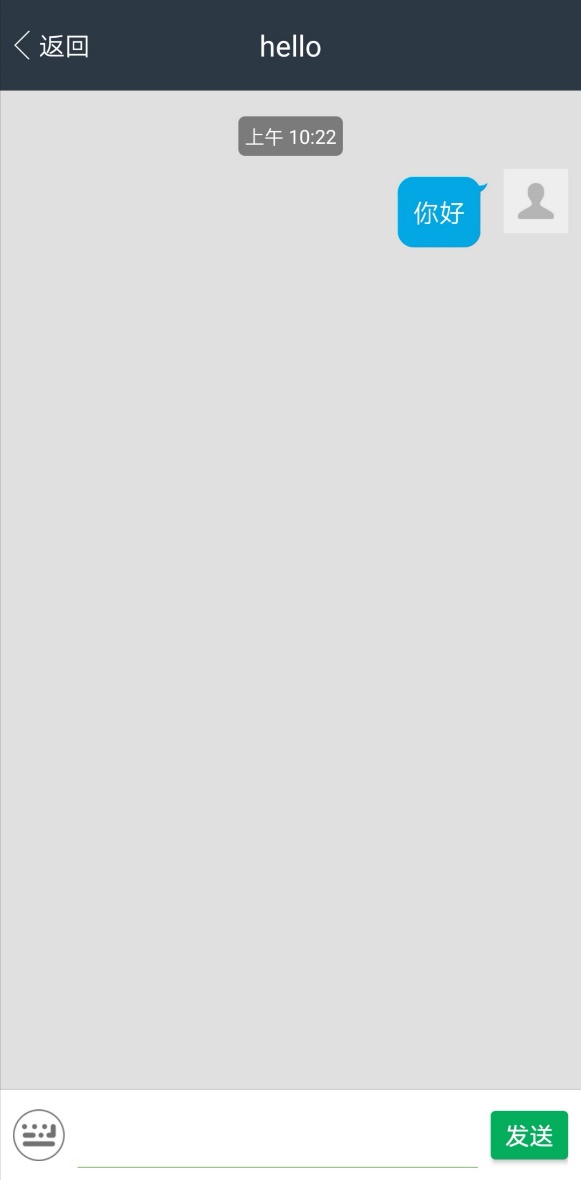


图4 聊天界面

1. 音视频通话界面：这两个界面由联系人界面选择语音通话或视频通话按钮进入，主要负责拨打后的等待情况和对方接听后的内容显示，可以分为拨打时的界面和接通后的界面。拨打时的整体布局为中间显示对方的头像和下方的三个控制按钮，分别为静音、挂断和免提；接通后语音通话界面无大的变化，视频通话界面主要显示对方的摄像内容，右上角为自己的摄像内容。

1.2 系统主要功能

1.2.1 总体实现

即时通讯系统是利用第三方即时通讯云环信SDK和Android SDK，在Android Studio工具中开发完成，其中环信SDK包含两个jar包，一个用于开发带有语音和视频通话功能，一个不带有语音和视频通话功能。系统总体的实现步骤是先注册一个环信即时通讯云的账号得到一个APP key，然后在Android Studio中新建一个工程，在工程的依赖包libs中导入带有语音和视频通话的jar包，在AndroidManifest.xml文件中配置注册得到的APP key和相关权限。接下来就是利用Android Studio编程实现系统的功能。

1.2.2 登陆模块

登录功能的实现是为“登陆”按钮添加点击事件监听器onClick（）方法，在该方法中调用startLogin（Context context，String phone，String password）方法，该方法有三个参数，context是一个上下文对象，phone和password是用户输入的账号和密码，传递给Login（）方法进行验证，Login（）方法是环信SDK提供的登陆方法，判断账号密码的正确性以及是否注册过。在startLogin（）方法中，还有检验账号信息的两个方法isEmpty（）和isMobileNumber（），分别检查账号密码是否为空和账号是否符合手机号的规范。

1.2.3 注册模块

注册功能的实现是为“注册”按钮添加点击事件监听器onClick（）方法，在该方法中调用regist（String phone，String password）方法，该方法有两个参数，分别是phone和password，即用户输入的账号和密码，方法内有检验账号信息的两个方法isEmpty（）和isMobileNumber（），检查账号密码是否为空和账号是否符合手机号的规范，检查符合要求后把phone和password参数传递给环信SDK提供的regist（）方法进行注册和登记。

1.2.4 聊天模块

文字消息聊天功能是在chatPresenter类中的sendTextMessage（String contactID,String content）,两个参数分别代表要发送消息的好友ID以及发送消息的内容，在该方法中首先是利用环信SDK提供createTextMessage（）方法创建一个文本消息对象IMessage，然后用sendMessage（IMessage）将文字消息发送出去。

1.2.5 语音和视频通话模块

语音和视频通话主要需求实现发送通话请求以及回应功能。在callPresenter中有startVoiceCall（phone）、startVideoCall（phone）、answerCall（phone）、endCall（phone）和rejectCall（phone）这五个方法，分别调用了SDK提供的相应的音视频通话方法，实现了语音通话请求、视频通话请求、接听、挂断和拒绝这五个功能。在界面的图片按钮中添加点击事件监听器，再调用相应的功能方法，从而完整实现了语音和视频通话功能。