|  |  |
| --- | --- |
| 产品名称 | 密级 |
| WChat | 1 |
| 产品版本 | 共23页 |
| 1.0 |

WChat 详细设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 王秋荣 | 日期 | 2017-04-08 |
| 评审人 |  | 日期 |  |
| 批准 |  | 日期 |  |

西安电子科技大学

版权所有 侵权必究

（仅供内部使用）

修订记录

| 日期 | 修订 版本 | Defect ID CR号 | 修改 章节 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- | --- |
| 2017-04-14 | 1.00 |  |  | 初稿完成 | 王秋荣 |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |
|  |  |  |

分发记录

|  |  |  |
| --- | --- | --- |
| Copy No. | 持有者和角色 | 分发日期 |
| 1 | <RDPDT PDT开发代表> | 2017-04-14 |
| 2 | <Project Manager 项目经理> | 2017-04-14 |
| 3 | <Team members 项目组成员> | 2017-04-14 |
| 4 | <Customer Representative 客户代表> | 2017-04-14 |
| 5 | <Others其它> | 2017-04-14 |

目 录

1 简介 5

1.1 目的 5

1.2 范围 5

2 详细设计 5

2.1 用户管理模块详细设计 6

2.1.1 数据描述 8

2.1.2 函数描述 8

2.2 好友管理模块详细设计 12

2.2.1 数据描述 14

2.2.2 函数描述 15

2.3 消息管理模块详细设计 17

2.3.1 数据描述 19

2.3.2 函数描述 19

2.4 群聊管理模块详细设计 20

2.4.1 数据描述 22

2.4.2 函数描述 22

3 错误处理 23

3.1.1 系统错误 23

3.1.2 接口错误 23

3.1.3 协议错误 23

4 参考资料 23

WChat 详细设计说明书

摘 要：

该软件是一款基于Android的简单聊天通信软件，国内外市场已经有许多的聊天软件。我的这

款软件是基于局域网的Android聊天软件，用户可以使用该软件与好友进行聊天，方便操作，简单

易用。

关键词：android 聊天软件 局域网

缩略语清单：

| 缩略语 | 英文全名 | 中文解释 |
| --- | --- | --- |
| WChat | WChat | 聊天软件名称 |
|  |  |  |
|  |  |  |
|  |  |  |

# 简介

## 目的

这个软件设计文档给 WChat 聊天软件提供了开发时的详细设计，帮助开发人员了解整个系统的设计，WChat聊天软件用于人们日常的聊天，可以单聊或者群聊，实现多形式的聊天。该软件是一款基于Android的聊天软件，包括Android客户端开发和服务器开发。

## 范围

这个文档包含了WChat聊天软件的详细设计。WChat聊天主要使用Android软件通过服务器进行聊天，服务器端进行客户端发送的处理需求，进行处理并进行数据库操作。软件使用Android、Java、Socket、MySQL等开发。

# 详细设计

以下给出了WChat聊天软件的功能模块划分图：



我开发的WChat聊天软件主要包括四个功能模块：用户管理模块、好友管理模块、消息管理模块、群组管理模块。对于用户可以打开软件直接进入登录界面，如果已经有账号输入账号、密码登录，如果没有可以点击“新用户”到注册界面进行注册，成功之后进行登录，进入聊天界面。在聊天界面用户可以选择好友或者群组进行聊天，还可以添加或者删除好友或群组。

## 用户管理模块详细设计

MessageType

数据结构描述：描述客户端与服务器端发送请求消息的类型，根据不同的类型做相应的处理

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | SUCCESS | 表明是否成功 |
| String | FAIL | 表明失败 |
| String | GET\_ONLINE\_FRIENDS | 要求在线好友的包 |
| String | RET\_ONLINE\_FRIENDS | 返回在线好友的包 |
| String | LOGIN | 请求验证登陆 |
| String | COM\_MES | 普通信息包 |
| String | ADD\_BUDDY | 添加好友 |
| String | DEL\_BUDDY | 删除好友 |
| String | GROUP\_MES | 群消息 |
| String | FAIL\_NO\_USER | 登录失败，用户不存在 |
| String | REGISTER\_USER\_EXIST | 注册失败，用户已存在 |
| String | GET\_USER\_INFO | 获取用户信息 |
| String | REGISTER | 请求注册 |
| String | GET\_GROUP\_MEMBER | 获取群成员信息 |
| String | ADD\_GROUP | 请求加群 |
| String | DEL\_GROUP | 请求退出群 |
| String | LOGOUT | 请求登出 |

User

数据结构描述：描述用户及好友信息的类

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | operation | 请求操作类型 |
| String | phone | 手机号（账号） |
| String | password | 密码 |
| String | nick | 昵称 |
| String | avator | 头像 |
| Int | sex | 性别 |
| Int | age | 年龄 |
| Int | isonline | 是否在线 |
| Date | addTime | 注册时间 |

### 数据描述

phone

功能描述：用户账号，该软件使用手机号注册，作为账号

数据定义：String phone；

account

功能描述：登录密码

数据定义：String account

### 函数描述

标识：用户注册模块

类型： 2级设计中的模块

功能概述：用户在Android端输入手机号和密码进行注册，Android首先对手机号和密码的格式进行验证。若格式不符，提示用户重新输入。若格式正确向服务器发送注册请求进行注册。服务器接收到注册请求之后调用相关的函数进行注册，注册结束之后返回响应的结果。客户端根据服务器返回的结果进行响应的处理。

#### 函数1

Function: register()

Description: 根据参数phone、pass进行用户注册

Input: phone、 pass

Return: true or false

Others: 如果返回false，提示用户注册失败。如果返回true，提示用户注册成功，并跳转到登录界面

private void onClickRegister() {

btnRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

final String phone = etPhoneNunmber.getText().toString();

String verifyCode = etVerifyCode.getText().toString();

final String password = etPassword.getText().toString();

String password2 = etConfirmPassword.getText().toString();

if (verifyCode.length() == 0) {

Toast.makeText(RegisterActivity.this, "验证码不能为空", Toast.LENGTH\_SHORT).show();

} else if (password.length() == 0 || password2.length() == 0) {

Toast.makeText(RegisterActivity.this, " 密码不能为空", Toast.LENGTH\_SHORT).show();

} else if (!password.equals(password2)) {

Toast.makeText(RegisterActivity.this, "两次密码输入不一致", Toast.LENGTH\_SHORT).show();

} else {

new Thread(new Runnable() {

@Override

public void run() {

User user = new User();

user.setPhone(phone);

user.setPassword(password);

user.setOperation("register");

boolean b = new WChatClient(RegisterActivity.this).sendRegisterInfo(user);

Message msg = new Message();

if (b) {

//注册成功跳转到登陆

msg.what = 1;

} else {

msg.what = 2;

}

handler.sendMessage(msg);

}

}).start();

}

}

});

标识：用户登录模块

类型： 2级设计中的模块

功能概述：用户在Android端输入手机号和密码进行登录，首先对手机号和密码的格式进行验证。若格式不符，提示用户重新输入。若格式正确向服务器发送登录请求进行登录。服务器接收到登录请求之后调用相关的函数进行登录验证，登录结束之后返回响应的结果。客户端根据服务器返回的结果进行响应的处理。

#### 函数1

Function: login()

Description: 根据参数phone、pass进行用户登录

Input: phone、 pass

Return: true or false

Others: 如果返回false，提示用户登录失败。如果返回true，提示用户登录成功，并跳转到主界面

private void onClickLogin() {

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String phone = etPhone.getText().toString();

String password = etPassword.getText().toString();

if (phone.length() == 0) {

Toast.makeText(LoginActivity.this, "手机号不能为空", Toast.LENGTH\_SHORT).show();

} else if (password.length() == 0) {

Toast.makeText(LoginActivity.this, "密码不能为空", Toast.LENGTH\_SHORT).show();

} else if (!phone.matches("^1[0-9]{10}$")) {

Toast.makeText(LoginActivity.this, "手机号格式不对", Toast.LENGTH\_SHORT).show();

} else {

new Thread(new Runnable() {

public void run() {

boolean b = login(etPhone.getText().toString(), etPassword.getText().toString());

Log.e("Login",b + "====");

Message msg = new Message();

if (b) {

msg.what = 1;

} else {

msg.what = 2;

}

handler.sendMessage(msg);

}

}

).start();

}

}

});

}

## 好友管理模块详细设计

MessageType

数据结构描述：描述客户端与服务器端发送请求消息的类型，根据不同的类型做相应的处理

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | SUCCESS | 表明是否成功 |
| String | FAIL | 表明失败 |
| String | GET\_ONLINE\_FRIENDS | 要求在线好友的包 |
| String | RET\_ONLINE\_FRIENDS | 返回在线好友的包 |
| String | LOGIN | 请求验证登陆 |
| String | COM\_MES | 普通信息包 |
| String | ADD\_BUDDY | 添加好友 |
| String | DEL\_BUDDY | 删除好友 |
| String | GROUP\_MES | 群消息 |
| String | FAIL\_NO\_USER | 登录失败，用户不存在 |
| String | REGISTER\_USER\_EXIST | 注册失败，用户已存在 |
| String | GET\_USER\_INFO | 获取用户信息 |
| String | REGISTER | 请求注册 |
| String | GET\_GROUP\_MEMBER | 获取群成员信息 |
| String | ADD\_GROUP | 请求加群 |
| String | DEL\_GROUP | 请求退出群 |
| String | LOGOUT | 请求登出 |

Message

数据结构描述：客户端与服务端发送消息的类

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | type | 请求操作类型 |
| String | sender | 请求发送者 |
| String | receiver | 请求接收者 |
| String | content | 请求内容 |
| String | sendTime | 请求时间 |

### 数据描述

type

功能描述：请求类型

数据定义：String type

myAccount

功能描述：发送请求的用户账号

数据定义：String myAccount

dfAccount

功能描述：接受请求的用户的账号

数据定义：String dfAccount

content

功能描述：请求内容

数据定义：String content

### 函数描述

标识：用户管理模块

类型： 1级设计中的模块

功能概述：根据用户不同的操作，选择不同的操作类型，添加好友、删除好友、好友信息 。将请求WChatMessage传递给服务器，服务器根据类型进行响应的处理，处理之后返回处理结果，客户端根据返回结果作出相应的反应。

#### 函数1

Function: sendMessage

Description: 根据参数type、myAccount、dfAccount进行好友管理请求

Input: type、myAccount、dfAccount

Return: true or false

Others: 如果返回false，提示用户好友操作失败。如果返回true，提示用户好友操作成功,并进行相应的处理

private View.OnClickListener chatSendOnclickListener = new View.OnClickListener() {

public void onClick(View v) {

// TODO Auto-generated method stub

String contString = etMsg.getText().toString();

if (contString.trim().length() == 0) {

Toast.makeText(ChatRoomActivity.this, "内容不能为空", Toast.LENGTH\_SHORT).show();

} else {

ChatMsgEntity entity = new ChatMsgEntity();

entity.setSender(me.getPhone());

entity.setReceiver(user.getPhone());

entity.setSendTime(getDate());

//entity.setMsgType(false);

//entity.setName(user.getNick());

entity.setContent(contString);

mDataArrays.add(entity);

mAdapter.notifyDataSetChanged();

mListView.setSelection(mListView.getCount() - 1);

messageDB.saveMsg(user.getPhone(), entity);

String type = "";

if (user.getPhone().length() < 11) {

type = MessageType.GROUP\_MES;

} else {

type = MessageType.COM\_MES;

}

SendMessage.sendMes(user.getPhone(), contString, type);

etMsg.setText("");

}

//下面是添加到RecentListView

RecentChatEntity entity1 = new RecentChatEntity(user.getPhone(),

user.getAvatar(), 0, user.getNick(), getDate(),

contString);

ChatFragment.RecentChatList.remove(entity1);

ChatFragment.RecentChatList.addFirst(entity1);

adapter.notifyDataSetChanged();

}

};

## 消息管理模块详细设计

MessageType

数据结构描述：描述客户端与服务器端发送请求消息的类型，根据不同的类型做相应的处理

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | SUCCESS | 表明是否成功 |
| String | FAIL | 表明失败 |
| String | GET\_ONLINE\_FRIENDS | 要求在线好友的包 |
| String | RET\_ONLINE\_FRIENDS | 返回在线好友的包 |
| String | LOGIN | 请求验证登陆 |
| String | COM\_MES | 普通信息包 |
| String | ADD\_BUDDY | 添加好友 |
| String | DEL\_BUDDY | 删除好友 |
| String | GROUP\_MES | 群消息 |
| String | FAIL\_NO\_USER | 登录失败，用户不存在 |
| String | REGISTER\_USER\_EXIST | 注册失败，用户已存在 |
| String | GET\_USER\_INFO | 获取用户信息 |
| String | REGISTER | 请求注册 |
| String | GET\_GROUP\_MEMBER | 获取群成员信息 |
| String | ADD\_GROUP | 请求加群 |
| String | DEL\_GROUP | 请求退出群 |
| String | LOGOUT | 请求登出 |

Message

数据结构描述：客户端与服务端发送消息的类

数据结构定义：

public class WChatMessage implements Serializable {

private static final long serialVersionUID = 2L;

private String type;

private String sender;

private String receiver;

private String content;

private String sendTime;

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | type | 请求操作类型 |
| String | sender | 请求发送者 |
| String | receiver | 请求接收者 |
| String | content | 请求内容 |
| String | sendTime | 请求时间 |

### 数据描述

type

功能描述：请求类型

数据定义：String type

myAccount

功能描述：发送请求的用户账号

数据定义：String myAccount

dfAccount

功能描述：接受请求的用户的账号

数据定义：String dfAccount

content

功能描述：请求内容

数据定义：String content

### 函数描述

标识：发送消息模块

类型： 2级设计中的模块

功能概述：根据用户发送消息请求，传递对应的type、发送者账号、接收者账号、消息内容，给sendMessage函数，函数向服务端发送请求。服务器收到请求之后，将对应的消息发送个对应的客户端，客户端显示消息。

#### 函数1

Function: sendMessage

Description: 根据参数type、myAccount、dfAccount进行好友管理请求

Input: type、myAccount、dfAccount

Return:

## 群聊管理模块详细设计

MessageType

数据结构描述：描述客户端与服务器端发送请求消息的类型，根据不同的类型做相应的处理

数据结构定义：

public class MessageType {

public static final String SUCCESS = "1";//表明是否成功

public static final String FAIL = "2";//表明失败

public static final String COM\_MES = "3";//普通信息包

public static final String GET\_ONLINE\_FRIENDS = "4";//要求在线好友的包

public static final String RET\_ONLINE\_FRIENDS = "5";//返回在线好友的包

public static final String LOGIN = "7";//请求验证登陆

public static final String ADD\_BUDDY = "8";//添加好友

public static final String DEL\_BUDDY = "9";//删除好友

public static final String GROUP\_MES = "10";//群消息

public static final String FAIL\_NO\_USER = "11"; //登录失败，用户不存在

public static final String REGISTER\_USER\_EXIST = "12"; //注册失败，用户已存在

public static final String GET\_USER\_INFO = "13"; //获取用户信息

public static final String REGISTER = "14"; //请求注册

public static final String GET\_GROUP\_MEMBER = "15";//获取群成员信息

public static final String ADD\_GROUP = "16";////请求添加群

public static final String DEL\_GROUP = "17";////请求退出群

public static final String LOGOUT = "18";////请求登出

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | SUCCESS | 表明是否成功 |
| String | FAIL | 表明失败 |
| String | GET\_ONLINE\_FRIENDS | 要求在线好友的包 |
| String | RET\_ONLINE\_FRIENDS | 返回在线好友的包 |
| String | LOGIN | 请求验证登陆 |
| String | COM\_MES | 普通信息包 |
| String | ADD\_BUDDY | 添加好友 |
| String | DEL\_BUDDY | 删除好友 |
| String | GROUP\_MES | 群消息 |
| String | FAIL\_NO\_USER | 登录失败，用户不存在 |
| String | REGISTER\_USER\_EXIST | 注册失败，用户已存在 |
| String | GET\_USER\_INFO | 获取用户信息 |
| String | REGISTER | 请求注册 |
| String | GET\_GROUP\_MEMBER | 获取群成员信息 |
| String | ADD\_GROUP | 请求加群 |
| String | DEL\_GROUP | 请求退出群 |
| String | LOGOUT | 请求登出 |

Message

数据结构描述：客户端与服务端发送消息的类

数据结构定义：

public class WChatMessage implements Serializable {

private static final long serialVersionUID = 2L;

private String type;

private String sender;

private String receiver;

private String content;

private String sendTime;

}

数据项描述：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据项定义 | 数据项描述 |
| String | type | 请求操作类型 |
| String | sender | 请求发送者 |
| String | receiver | 请求接收者 |
| String | content | 请求内容 |
| String | sendTime | 请求时间 |

### 数据描述

type

功能描述：请求类型

数据定义：String type

myAccount

功能描述：发送请求的用户账号

数据定义：String myAccount

dfAccount

功能描述：接受请求的用户的账号

数据定义：String dfAccount

content

功能描述：请求内容

数据定义：String content

### 函数描述

标识：群组管理模块

类型： 1级设计中的模块

功能概述：根据用户不同的操作，选择不同的操作类型，添加群组、删除群组。将请求WChatMessage传递给服务器，服务器根据类型进行响应的处理，处理之后返回处理结果，客户端根据返回结果作出相应的反应。

#### 函数1

Function: sendMessage

Description: 根据参数type、myAccount、dfAccount进行好友管理请求

Input: type、myAccount、dfAccount

Return: true or false

Others: 如果返回false，提示用户群组操作失败。如果返回true，提示用户群组操作成功，并进行相应的处理

# 错误处理

### 系统错误

任务失败会使用Toast设置相应的提示来提示用户。

### 接口错误

|  |  |
| --- | --- |
| **错误码** | **描述** |
| 2 | 操作失败 |
| 11 | 登录失败，不存在 |
| 12 | 注册失败，用户已存在 |

### 协议错误

协议中没有描述的情况下会直接返回FAIL。

# 参考资料

【1】郭霖.第一行行代码[M].人民邮电出版社，2014

【2】Bruce Eckel.Java编程思想[M].机械工业出版社，2007

【3】曾健生.App后台开发运维和架构实践[M].电子工业出版社，2016

【4】Abraham Silberschatz.数据库系统概论[M].机械工业出版社，2012

【5】马志强.基于Android平台即时通信系统的设计与实现[D].北京交通大学，2009