【WChat】

详细设计说明书

**目录**

1 引言 3

1.1 编写目的 3

1.2 开发环境 3

2 程序系统的结构 3

3 功能模块设计说明 4

3.1 程序描述 4

3.2 功能 4

3.3 性能 6

3.4 输入项 6

3.5 输出项 7

3.6 算法 7

3.7 接口 7

4 界面设计说明 8

4.1 程序描述 8

4.2 功能 8

4.3 性能 9

4.4 输入项 9

4.5 输出项 9

4.6 页面布局代码 9

5 安全设置设计说明 11

5.1 程序描述 11

5.2 功能 11

5.3 性能 11

5.4 输入项 11

5.5 输出项 11

5.6 算法 12

6 参考资料 14

# 引言

## 编写目的

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。

## 开发环境

软件系统的名称：基于Android的简单聊天通信软件---WChat

项目开发环境：IntelliJ IDEA、Android Studio、MySQL

# 程序系统的结构

WChat软件结构设计如下：



# 功能模块设计说明

## 程序描述

主要功能：用于用户的可视化界面操作，有助于用户快速熟悉软件。

可扩展性：能够适应应用要求的变化和修改，具有灵活的可扩展性。

## 功能

主要有用户管理、好友管理、消息管理、群组管理四个功能模块

1. 用户管理模块介绍及流程图如下所示：

|  |  |  |
| --- | --- | --- |
| 子功能名 | 功能需求 | 程序（界面） |
| 登录 | 用户输入手机号、密码进行登录 | activity\_login.xml |
| 注册 | 用户输入手机号、密码进行注册 | activity.register.xml |

表1 用户管理模块介绍



图1 登录管理模块流程图

用户进入打开应用之后直接进入登陆界面。如果没有账号，点击注册按钮进入注册界面进行注册；如果有账号可以直接输入账号和密码进行登录。登录成功之后直接进入主界面

1. 好友管理模块介绍及流程图如下所示：

|  |  |  |
| --- | --- | --- |
| 子功能名 | 功能需求 | 程序（界面） |
| 好友管理 | 用户可以输入账号添加好友，也可以在好友信息页面点击删除好友 | acyicity\_friend\_info.xml  activity\_friend\_add.xml |

表2 好友管理模块介绍



图2 好友管理模块流程图

用户可以在主界面中点击添加好友按钮进入添加好友界面，输入账号点击确认来添加好友。用户还可以在好友信息界面中直接点击删除好友进行好友删除。

1. 消息管理模块介绍及流程图如下所示：

|  |  |  |
| --- | --- | --- |
| 子功能名 | 功能需求 | 程序（界面） |
| 消息管理 | 用户可以在好友聊天界面输入需要发送的消息，然后点击发送按钮进行消息发送 | activity\_chat.xml |

表3 消息管理模块介绍



图3 消息管理模块流程图

用户在主界面或者最近会话界面中选择要聊天的好友，进入好友聊天界面，输入需要发送的消息，点击发送，就可以将消息发送个好友，通过服务器，好友就可以接收到你发送的消息。

1. 群组管理模块介绍及流程图如下所示：

|  |  |  |
| --- | --- | --- |
| 子功能名 | 功能需求 | 程序（界面） |
| 群组管理 | 用户输入群组账号就可以添加群组，在群组信息界面点击退出群，则退出群租 | activity\_friend\_add.xml  activity\_friend\_info.xml |

表4 群组管理模块介绍



图4 群组管理模块流程图

用户可以在主界面中点击添加群组按钮进入添加群组界面，输入账号点击确认来添加群组。用户还可以在群组信息界面中直接点击退出群组进行群组退出。

## 性能

当用户点击相应的按钮时可以快速的做出响应，发送的消息可以实时到达接收方。

## 输入项

用户管理：

手机号和密码

好友管理：

好友手机号

消息管理：

文字消息

群组管理：

群组账号

## 输出项

用户管理：

提示成功或者失败

好友管理：

提示成功或者失败

消息管理：

无

群组管理：

提示成功或者失败

## 算法

主界面代码：

public class MainActivity extends FragmentActivity {

//用户信息

public static String myInfo = "";

protected static final String TAG = "MainActivity";

private Toolbar toolbar;

private TextView tvTitle;

private Fragment fragment;

private FragmentManager fragmentManager;

private FragmentTransaction fragmentTransaction;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

findView();

init();

}

public void init() {

getSupportFragmentManager().beginTransaction().replace(R.id.content, new ChatFragment()).commit();

}

//按下返回键 直接返回桌面

@Override

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE\_BACK) {

Intent home = new Intent(Intent.ACTION\_MAIN);

home.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

home.addCategory(Intent.CATEGORY\_HOME);

startActivity(home);

return true;

}

return super.onKeyDown(keyCode, event);

}

private void findView() {

toolbar = (Toolbar) findViewById(R.id.toolbar);

tvTitle = (TextView) findViewById(R.id.titlebar\_title);

BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.navigation);

navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);

}

private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener

= new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()) {

case R.id.navigation\_home:

fragment = new ChatFragment();

break;

case R.id.navigation\_dashboard:

fragment = new FriendFragment();

break;

case R.id.navigation\_notifications:

fragment = new MineFragment();

break;

default:

break;

}

if (fragment != null) {

fragmentManager = getSupportFragmentManager();

fragmentTransaction = fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.content, fragment);

fragmentTransaction.commit();

}

return true;

}

};

## 接口

### 内部接口



当用户下载安装此软件之后，程序就被存入手机软件管理库，用户就可以运行此程序，运行之后，程序就进入登陆界面。

### 外部接口

### 

这是外部接口图，开发环境时Android Studio模拟器

# 界面设计说明

## 程序描述

主要功能：用于为用户提供可视化操作界面，增加了用户友好性

可扩展性：能够适应应用程序要求的变化和修改，具有灵活的可扩展性

## 功能

登录界面、注册界面、主界面、最近会话界面、好友界面、我的信息界面、聊天界面、添加好友界面、好友信息界面

## 性能

能够快速的显示用户所需的操作界面

## 输入项

登录界面：

手机号和密码

注册界面：

手机号和密码

聊天界面：

文字信息

添加好友界面：

好有账号

## 输出项

相应的操作界面

## 页面布局代码

聊天界面的代码：

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include layout="@layout/layout\_toolbar"

android:id="@+id/toolbarLinear"/>

<LinearLayout

android:id="@+id/chat\_bottom\_linear"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_alignParentBottom="true"

android:background="@color/whites"

android:orientation="horizontal">

<EditText

android:id="@+id/chat\_bottom\_edit"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="4"

android:layout\_marginLeft="2dp"

android:background="@drawable/acm\_inputbox">

<requestFocus />

</EditText>

<Button

android:id="@+id/chat\_bottom\_send"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_marginLeft="1dp"

android:layout\_marginRight="2dp"

android:layout\_weight="1"

android:background="@drawable/chat\_bottom\_send"

android:text="@string/send"

android:textColor="@color/whites"

android:textSize="18sp" />

</LinearLayout>

<View

android:id="@+id/line"

android:layout\_width="match\_parent"

android:layout\_height="1dp"

android:layout\_above="@id/chat\_bottom\_linear"

android:background="@color/devide\_line" />

<ListView

android:id="@+id/chat\_content\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:divider="@null"

android:layout\_above="@id/chat\_bottom\_linear"

android:layout\_below="@id/toolbarLinear"/>

</RelativeLayout>

# 安全设置设计说明

## 程序描述

主要功能：当用户打开应用时，首先进入登陆界面，用户输入正确的账号和密码之后才能进入应用，正常的使用应用。

可扩展性：能够适应应用要求的变化和修改，具有灵活的可扩展性

## 功能

安全保护流程：



## 性能

能够保证用户输入正确的账号和密码，正常使用应用

## 输入项

用户帐号和密码

## 输出项

应用界面

## 算法

登录逻辑代码：

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();

}

}

});

private boolean login(String phone, String pass) {

final User user = new User();

user.setPhone(phone);

user.setPassword(pass);

user.setOperation("login");

boolean b = new WChatClient(LoginActivity.this).sendLoginInfo(user);

//登陆成功

if (b) {

this.finish();

return true;

} else {

return false;

}

}

public boolean sendLoginInfo(Object obj) {

boolean b = false;

try {

s = new Socket();

try {

s.connect(new InetSocketAddress(IP, PORT), 2000);

} catch (SocketTimeoutException e) {

return false;

}

ObjectOutputStream oos = new ObjectOutputStream(s.getOutputStream());

Log.e("WChat", oos.toString());

oos.writeObject(obj);

ObjectInputStream ois = new ObjectInputStream(s.getInputStream());

WChatMessage ms = (WChatMessage) ois.readObject();

if (ms.getType().equals(MessageType.SUCCESS)) {

String[] returnStr = ms.getContent().split("&");

MainActivity.myInfo = returnStr[0];

String[] buddyRet = returnStr[1].split(",");

FriendFragment.buddyStr = buddyRet[0];

FriendFragment.groupStr = buddyRet[1];

ClientConServerThread ccst = new ClientConServerThread(context, s);

ManageClientConServer.getManager().addClientConServerThread(((User) obj).getPhone(), ccst);

ccst.start();

b = true;

} else if (ms.getType().equals(MessageType.FAIL)) {

b = false;

}

} catch (IOException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

return b;

}

# 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