**基于Android的我们约会吧交友软件V1.1**

北京华清远见科技信息有限公司

**内部资料 请勿外传**

2013年8月19日

[一、 背景 3](#_Toc364843470)

[二、 功能介绍 3](#_Toc364843471)

[2.1客户端功能介绍 3](#_Toc364843472)

[2.1.1 欢迎界面 3](#_Toc364843473)

[2.1.2 用户注册 3](#_Toc364843474)

[2.1.3 用户登录 3](#_Toc364843475)

[2.1.4 发布约会功能 3](#_Toc364843476)

[2.1.5 查询约会功能 3](#_Toc364843477)

[2.1.6查看附近朋友功能 3](#_Toc364843478)

[2.1.7 聊天功能 4](#_Toc364843479)

[2.1.8其他功能 4](#_Toc364843480)

[2.2服务器功能介绍 4](#_Toc364843481)

[2.2.1服务器的功能 4](#_Toc364843482)

[三、 环境搭建 4](#_Toc364843483)

[3.1将工程导入eclipse开发环境。 4](#_Toc364843484)

[3.2 创建服务器使用的数据库 6](#_Toc364843485)

[3.3运行服务器端 8](#_Toc364843486)

[3.4运行客户端 9](#_Toc364843487)

[四、 功能展示 13](#_Toc364843488)

[4.1 欢迎界面 13](#_Toc364843489)

[4.2 登录界面 13](#_Toc364843490)

[4.3 个人中心界面 14](#_Toc364843491)

[4.4 用户修改个人信息界面 14](#_Toc364843492)

[4.5 用户修改密码界面 15](#_Toc364843493)

[4.6 发布约会界面 15](#_Toc364843494)

[4.7 查看约会界面 16](#_Toc364843495)

[4.8 查看最近（朋友）界面 16](#_Toc364843496)

[4.9 即时聊天界面 17](#_Toc364843497)

[五、 程序分析 18](#_Toc364843498)

[5.1 客户端框架分析 18](#_Toc364843499)

[5.2 客户端程序分析 19](#_Toc364843500)

[5.2.1客户端类功能介绍 19](#_Toc364843501)

[5.2.2 界面的跳转 intent介绍 19](#_Toc364843502)

[5.2.3 Android中的消息机制 20](#_Toc364843503)

[5.2.4 个人中心界面代码分析 21](#_Toc364843504)

[5.3 服务器框架分析 23](#_Toc364843505)

[5.4 服务器程序分析 23](#_Toc364843506)

[5.4.1 服务器类功能介绍 23](#_Toc364843507)

[5.4.2 服务器网络相关代码分析 24](#_Toc364843508)

[5.4.3服务器数据库相关代码分析 24](#_Toc364843509)

[六、 总结 25](#_Toc364843510)

# 背景

随着网络的发展，生活节奏的日益提高，人们越来越需要和更多的朋友之间进行交流，因此需要提供更多的交友平台。该软件用于满足宅男宅女们的交友需求，应用于广大宅男宅女，主要用户群体是新兴移动互联网的忠实追随者，寂寞的潮男潮女们能够让他们能够在寂寞时找到快乐。

# 功能介绍

## 2.1客户端功能介绍

### 2.1.1 欢迎界面

1) 用户进入系统后会先出现欢迎界面，欢迎界面保持3秒钟，欢迎界面结束后会跳转到系统登录界面。

### 2.1.2 用户注册

1) 用户可以通过访问客户端，完成注册过程。注册信息支持中文或英文输入。

2) 系统需要验证用户所输入的用户名及密码的注册信息。

3) 注册成功后，系统向客户发送注册成功信息。

### 2.1.3 用户登录

1) 用户可以使用帐号登陆进入交友系统。

2) 支持用户登录成功后跳转。

### 2.1.4 发布约会功能

1) 发布约会信息，约会类型包括四项，分别为吃饭、逛街、打球、其他。

2) 用户发布约会时，可以设置发起约会人的基本信息，包括性别、年龄、距离。

3) 约会发布成功后，可以收到提交成功的提示。

### 2.1.5 查询约会功能

1) 进入查询约会界面，可以查询近期发布的所有约会信息，信息会以列表的形式显示。

2) 每条约会信息上面包含发布者的头像，约会时间，约会地点，类型。

3) 当用户点击一条约会信息后，直接进入聊天界面。

4) 用户可以取消发布过的约会信息。

### 2.1.6查看附近朋友功能

1) 进入查找最近界面，由于程序已经通过使用Gps功能将当前位置发送给服务器,此时通过访问数据库后以列表的形式显示附近的朋友，朋友信息包括姓名、性别、出生年月日等。

2） 点击列表中显示的任何一个附近的朋友，直接进入聊天界面。

### 2.1.7 聊天功能

1) 当用户选择查找最近功能或查看约会功能后，点击列表中显示的任何一个用户进入聊天界面，聊天对话框显示聊天内容。

### 2.1.8其他功能

1) 菜单键功能

关于中显示当前应用程序版本，应用程序logo，客服电话。帮助中显示基本的软件版本和帮助信息。

2） 匿名用户登录功能。

## 2.2服务器功能介绍

### 2.2.1服务器的功能

1）服务器界面有开启和关闭服务器的功能。

2）当用户注册成功后，会将用户的信息上传到服务器，进行验证，并将验证结果返回给客户端。

3）当用户登录系统后，会将用户的gps信息写入数据库。

4）保存用户和好友的聊天记录，并写入数据库。

5）记录发布约会的信息, 并写入数据库。

6）当用户查询约会信息时，服务器访问数据库，返回所有发布的约会信息。

7）记录约会的更新信息（参加约会和取消约会）。

# 环境搭建

## 3.1将工程导入eclipse开发环境。

1)运行eclipse集成开发环境。指定开发android的SDK的目录到eclipse中(注意英文目录下)。然后运行eclipse后，点击window中的preferences后如下配置SDK。如图3.1.1所示。

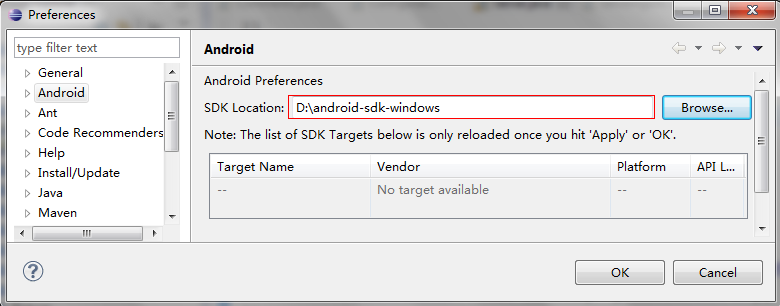


图 3.1.1

在eclipse下，自行创建模拟器相关内容。

2）待步骤1完成好后，点击File选项，在下拉菜单中，点击import按钮。在弹出的窗口中选中General中的Existing Projects into Workspace后点击next。

如图3.1.2所示。

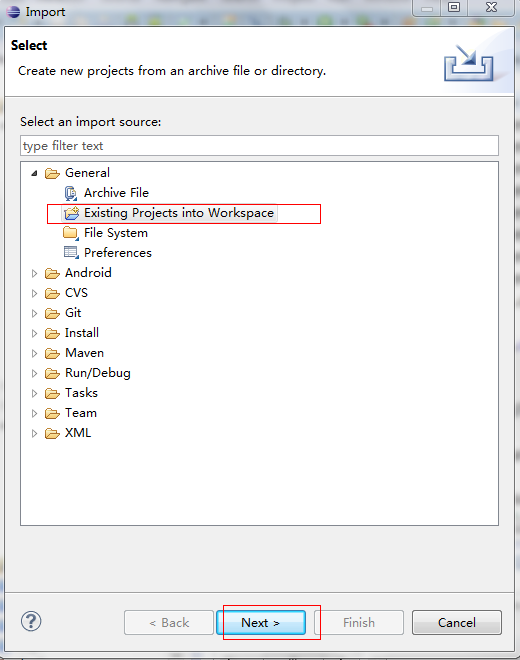


图 3.1.2

3）在随后弹出的窗口中，在Select root directory条目中，点击Browse找到工程文件所在的根目录“源码目录\v1.1”中的date\_Android和date两个文件。finish后导入到eclipse中了。类似操作如图3.1.3所示。

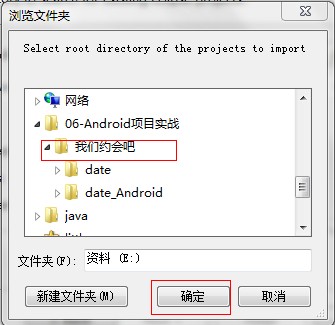


图 3.1.3

4） 工程完成导入如下图3.1.4所示。其中Date工程是java开发的项目服务器。Date\_Adroid工程是android开发的android客户端。

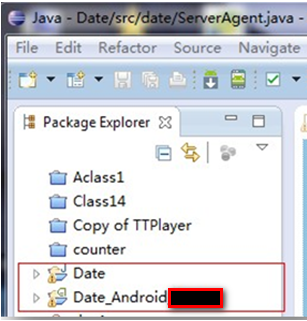


图 3.1.4

## 3.2 创建服务器使用的数据库

1） 工程已经导进来了，现在至少需要两台手机（在聊天和交互时用），其中一台可以用eclipse模拟一台手机。但是在此之前，必须首先在笔记本上安装Mysql数据库，然后在笔记本上将JAVA服务器运行起来。这样Java服务器将根据Android客户端发过来的消息读取操作数据库，并保存用户的相关信息于数据库中。

2） 安装Mysql服务器

参考文档"mysql相关\mysql安装参考手册.doc".

（用户名默认是root，密码请设定为pass）。请直接使用pass作为密码，以便Java服务器可以直接登入（否则将要修改Java服务器代码中的配置文件）。

下图3.2.1是Java服务器代码配置文件database.properties中关于登录Mysql数据库的配置信息。Java服务器通过这个配置文件来登录数据库。

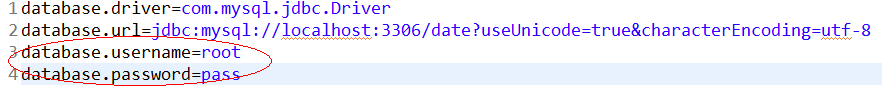


图 3.2.1

3）此时Mysql里面还没有创建需要的数据库和一系列表单。

根据上面的步骤， Mysql就已经配置在环境变量中了，并且密码是pass。

如下图3.2.2所示登录Mysql。



图 3.2.2

4）登入了Mysql中后。接着创建数据库和表单。数据库和表单的名字和Java服务器代码中的名字是对应上的。现在开始创建数据库date。如下图3.2.3。



图 3.2.3

5）通过指令show databases可以查询我们刚创建的数据库。接下来添加表单到date数据库中。先切换当前使用的数据库到date库下。通过指令use date;如下图 3.2.4所示。



图 3.2.4

6）现在开始创建表单，这里面涉及sql的语法请自行参阅相关资料学习。在提供的数据库目录下存在名为data.sql的文件。将其放到D盘下。通过Mysql自带的source命令来导入sql文件来创建表单。如下图 3.2.5所示进行操作。

注意：sql文件来创建表单是“数据库目录\data.sql”。



图 3.2.5

7）执行use date 和show tables显示刚创建的表单。如下图 3.2.6所示。表示数据库和表单都已经成功了，如果出现了错误，可能是mysql没有正确安装。



图 3.2.6

## 3.3运行服务器端

1） 由于之前已经成功的导入了那两个工程目录了，现在在eclipse包浏览器中选中Date工程右键，双击run as选项中的java application来启动服务器端。

在下弹出的窗口中选中Server后点击OK。如下图 3.3.1所示。

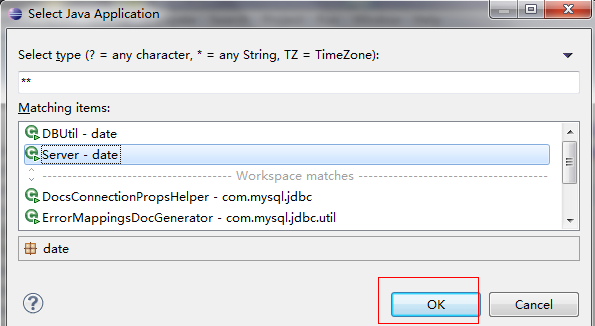


图 3.3.1

2） 此时将启动服务器端，在显示的界面中输入端口号：8888，然后点击连接服务器。如下图 3.3.2所示。

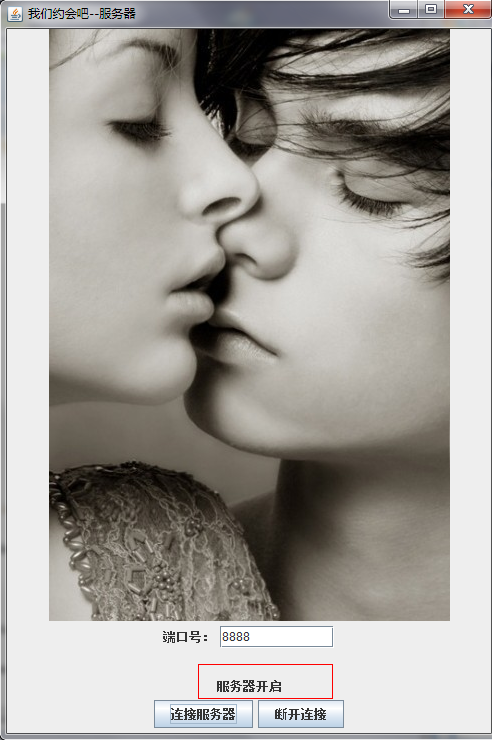


图 3.3.2

3） 在eclipse的控制台中出现Listening表示 《我们约会吧》服务器端已经搭建成功。正在监听有没有客户端的连接。如下图3.3.3所示。

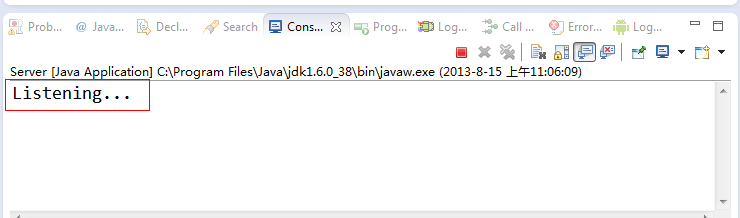


图3.3.3

## 3.4运行客户端

1） 在eclipse中，选中date\_Android工程，点击Run as选项中的android application 。这时将自动的启动模拟器（前提是之前已经配置好了模拟器），出现登陆界面。如下图所示。



图3.4.1

2） 在data.sql文件中保存了测试用的用户名和密码。 （如果没有信息请先点击“用户注册”进行用户信息注册，如用户名test、密码123，系统会返回生成的ID号如2004，通过ID号码进行登录。）

如果输入账号和密码后，仍然不能正常登陆的，是因为服务器的ip地址、端口号和客户端要连接的ip地址、端口号没对应上，就是用来指示客户端去连接局域网中哪个ip地址。所以首先必须保证客户端的本机ip地址要和服务器（运行java的主机）的ip在同一个网段，而且客户端要连接上。所以首先要配置一台无线路由器，让运行java的主机连上这台无线路由器。如下图3.4.2所示。



图3.4.2

3） 右键点击图3.4.2中的farsightyanfa，点击状态。然后会出现下图3.4.3，然后左键单击详细信息。便可得到本机（Java服务器）的地址是192.168.100.103。

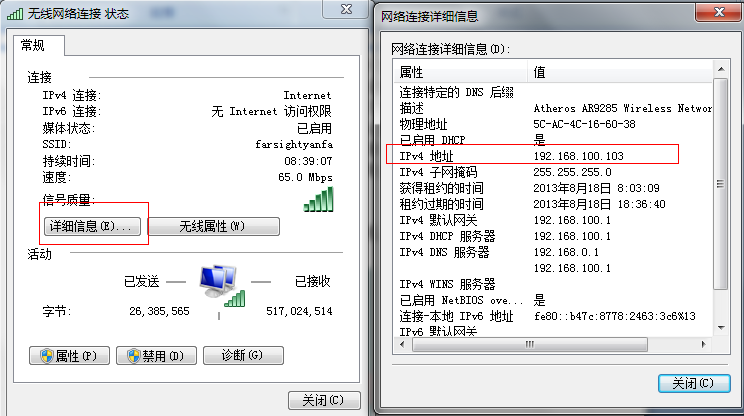


图3.4.3

4） 修改客户端的连接地址为192.168.100.103。进入工程date\_Android 双击ConstantUtil.java文件，其中SERVER\_ADDRESS 为服务器的ip地址，修改为192.168.100.103，SERVER\_PORT为服务器的端口号， 如下图3.4.4所示。

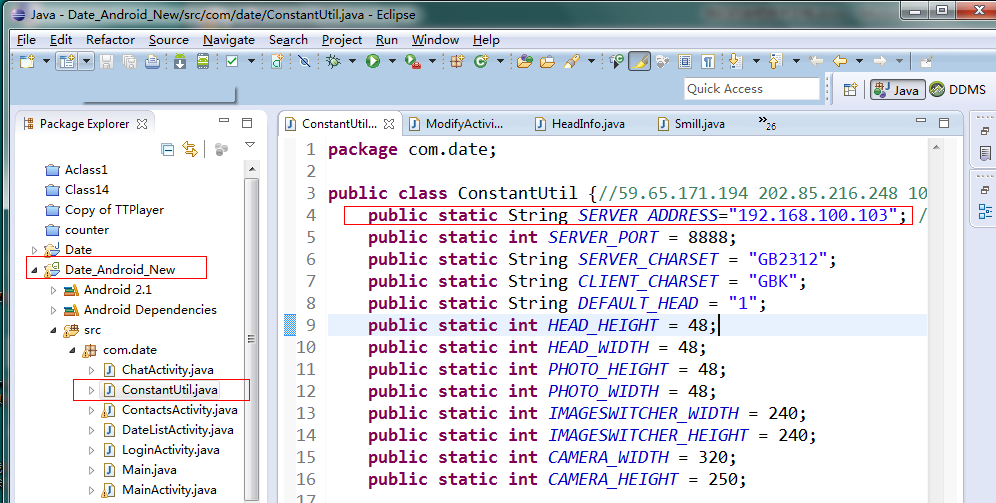


图3.4.4

5） 在eclipse中，选中date\_Android工程，点击Run as选项中的android application 运行Android客户端。输入账号ID 2004，密码123 ，登录服务器。如下图3.4.5所示。



图3.4.5

6）登录成功后将出现个人中心，其中包含个人信息和相关的操作。如图3.4.6所示。



图3.4.6

7） 以上是在模拟器上运行模拟的手机客户端。下面将在实体设备 如手机 上面安装《我们约会吧》客户端程序，首先将手机的usb调试模式打开。（当然可以将android 应用程序工程打包成apk后，通过豌豆荚等工具安装到手机上。）。将手机通过USB线连上电脑。并通过手机wifi，将手机连上farsightyanfa路由器。这样实体手机就和运行java服务器的主机便位于同一个网段了。点击eclipse中的window选项，在下拉菜单中选择open Perspective 然后选择other…中的DDMS(Dalvik Debug Monitor Service)后，将如下图3.4.7显示。

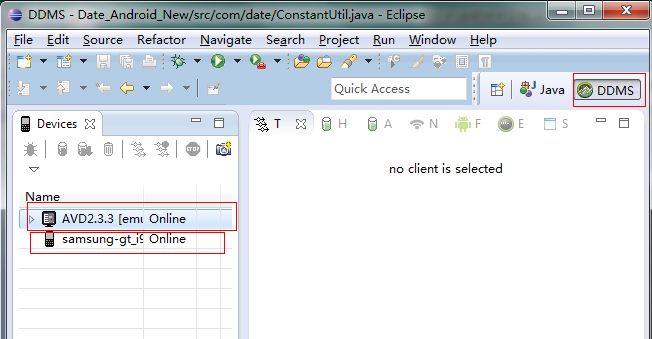


图3.4.7

8） 将客户端安装到手机上，上图3.4.7中存在两个设备，分别是模拟器设备和实体机samsung-gt设备。点击上图的samsung-gt 后选中，然后点击上图中的DDMS左边的Java回到代码界面，选中date\_Android点击Run as选项中的android application。弹出下图3.4.8。

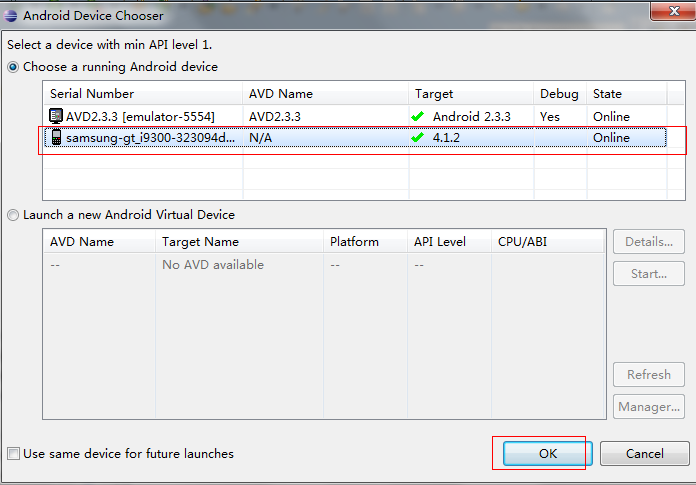


图3.4.8

9） 选中samsung-gt（或其他硬件设备） 点击OK。如下图3.4.9所示。

（如：输入账号ID: 2005 ,密码:123）。



图3.4.9

10）点击登录后如下图3.4.10显示。将客户端安装到多台实体手机上步骤相同。



图3.4.10

# 功能展示

## 4.1 欢迎界面

欢迎界面将默认停留3秒，使用户有一个良好的体验。如下图4.1.1所示。



图4.1.1

## 4.2 登录界面

登录界面是第一个和用户进行交互的界面。支持记住密码、自动登陆、用户注册、匿名登录、退出系统等功能，如下图4.2.1所示。当用户名和密码登陆失败的时候，将给予提示，如下图4.2.2所示。



图4.2.1 图4.2.2

## 4.3 个人中心界面

登陆成功后，将进入个人中心。包括用户图像、ID、昵称、性别、生日和邮箱等信息。支持修改个人信息、修改登陆密码、发布约会、查看约会、查找最近朋友等功能。如下图4.3.1所示。



图4.3.1

## 4.4 用户修改个人信息界面

点击上图4.3.1的修改个人信息将进入修改个人信息界面。如图4.4.1所示。



图4.4.1

## 4.5 用户修改密码界面

如下图图4.5.1和图4.5.2所示。

图4.5.1 图4.5.2

## 4.6 发布约会界面

点击上图4.3.1的发布约会将进入发布约会界面。如下图4.6.1所示。



图4.6.1

## 4.7 查看约会界面

当客户端A发布了约会信息后，在另一台客户端B(另外的帐号登录)上点击图4.3.1中的查看约会，将进入查看约会界面，便可以查看到客户端A发布的约会信息。如下图4.7.1所示。



图4.7.1

## 4.8 查看最近（朋友）界面

点击上图4.3.1的“查看最近”将进入查看最近（朋友）界面。界面将显示附近的好友列表。如下图4.8.1所示。



图4.8.1

## 4.9 即时聊天界面

点击图4.7.1 和 图4.8.1中列表中显示的用户，将直接进入聊天界面。客户端A和客户端B的聊天效果见下图4.9.1和图4.9.2所示。



图4.9.1 图4.9.2

# 程序分析

## 5.1 客户端框架分析

本项目客户端由Android语言开发。程序中的Activity之间的跳转，实现了界面的切换。首先，重点是搞清楚每一个Activity是实现什么功能。关于具体每一个Activity的功能见本节客户端类功能介绍。其次是各个Activity之间的跳转逻辑关系。最后，每一个Activity是怎么实现其具体功能的。下图5.1.1大致给出了andriod客户端应用程序的时序图。下面将结合下图5.1.1分析程序的流程。首先启动软件，在进入LoginActivity显示前，提前启动显示欢迎界面，在WelcomeActivity中启动一个线程用以计时，当3秒时间到的时候，结束欢迎界面，然后显示登陆界面。（由于和登陆过程类似，将注册过程略去）在登陆界面中当用户点击登陆时，首先建立网络连接并返回网络Socket对象。确定连接后，判断用户名和密码是否成功登陆，当网络返回登陆失败的时候，提示错误。否则启动MainActivity进入个人中心。在个人中心所对应的Activity中，总共有5种Activity可以被启动。分别是个人信息修改，密码修改，发布约会，附近好友，查看约会。启动并实现其中的一个的功能和另外4个的过程是一样的，图中只画出了个人信息的修改和密码的修改的过程。下面将以个人信息修改为例介绍实现此功能的过程。在个人中心中，点击修改个人信息进入个人信息修改界面。在个人信息修改界面中，点击返回按钮实现返回并放弃修改。当点击修改按钮时，首先建立网络连接，提交修改的内容并接受服务器端结果，当返回成功时，再次建立网络连接。将已经修改于数据库中用户个人信息读取出，并显示在个人信息修改界面的条目中。关于聊天功能，当点击附近好友界面和查看约会界面中的条目时，将跳转到在线聊天界面中。聊天的内容通过访问数据库中的聊天列表获得，然后显示在聊天界面中。

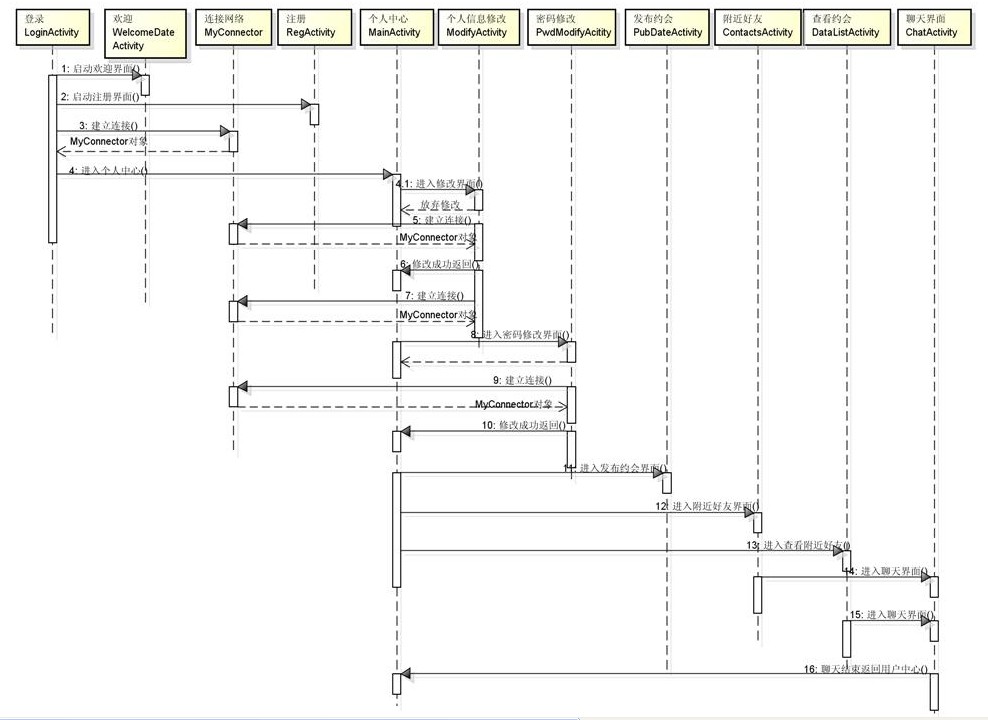


图5.1.1

## 5.2 客户端程序分析

### 5.2.1客户端类功能介绍

1）WelcomeDateActivity类：进入登录窗口前，首先调用其来实现欢迎界面。

2）ChatActivity类：用来实现聊天的功能的类。

3）ConstantUtil类：系统默认的常量，包括固定的服务器ip地址和固定的端口号。

4）ContactsActivity类：实现显示附近好友功能的类，包括好友的姓名，性别，头像等。

5）DateListActivity类： 显示已经发布的约会信息，包括约会类型等。

6）LoginActivity类：实现登录功能的类。

7）Main类：版权声明类。

8）MainActivity类：实现个人中心界面功能的类。

9）ModifyActivity类：实现用户修改个人信息的类。

10）MyConnecter类：建立网络连接类。创建网络输入、输出流。

11）PubDateActivity类：实现发布约会界面功能。

12）PwdModifyActivity类：实现在线修改登陆密码的类。

13）RegActivity类：实现在线注册功能类。

### 5.2.2 界面的跳转 intent介绍

Intent主要是解决Android应用的各项组件之间的通讯。因此，Intent起着一个媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。

在MainActivity.java多个事件处理方法中，实现了两个Activity之间的跳转。

1）实现了个人中心活动向附近好友活动的跳转。

Intent intent = new Intent(MainActivity.this,ContactsActivity.class);

intent.putExtra("uno", uno); //传递键值对

startActivity(intent); //启动intent对象

2）实现了个人中心活动向个人信息修改活动的跳转。

Intent intent = new Intent(MainActivity.this,ModifyActivity.class);

intent.putExtra("uno", uno);

startActivity(intent);

3）实现了个人中心活动向密码修改活动的跳转。

Intent intent = new Intent(MainActivity.this,PwdModifyActivity.class);

intent.putExtra("uno", uno);

startActivity(intent);

4）实现了个人中心活动向发布约会信息活动的跳转。

Intent intent = new Intent(MainActivity.this,PubDateActivity.class);

intent.putExtra("uno", uno);

startActivity(intent);

5）实现了个人中心活动向发布约会信息活动的跳转。

Intent intent = new Intent(MainActivity.this,DateListActivity.class);

intent.putExtra("uno", uno);

startActivity(intent);

### 5.2.3 Android中的消息机制

Looper用于封装了android线程中的消息循环，默认情况下一个非主线程是不存在Looper消息循环的，需要调用Looper.prepare()来给非主线程创建一个消息循环如下图图5.2.3.1所示。调用Looper.loop()来使消息循环起作用，如下图5.2.3.2所示。从消息队列里取消息，处理消息。Android通过Looper、Handler来实现消息循环机制。其中消息循环Looper中维护了一个消息队列。而主线程默认存在一个消息循环Looper和一个消息队列。

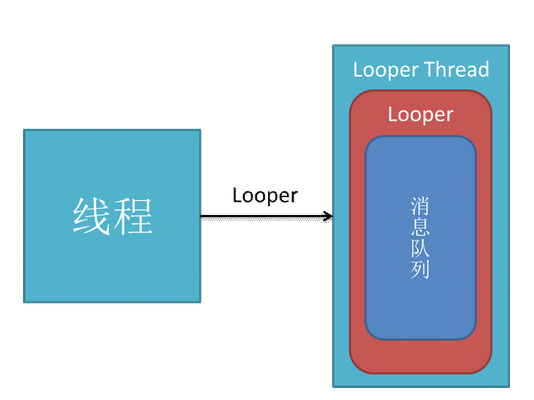


图5.2.3.1

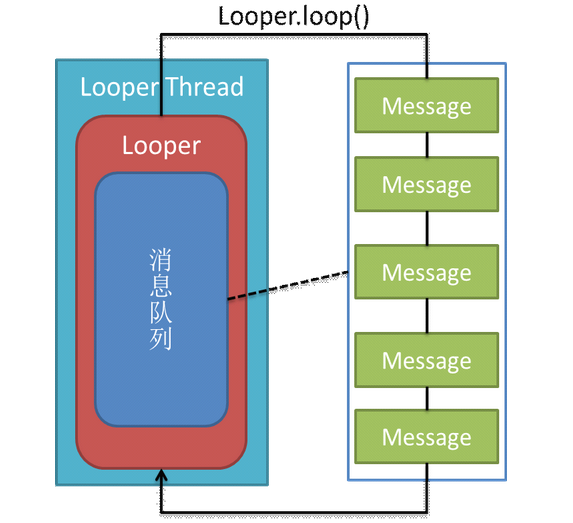


图5.2.3.2

Handler里面封装了一些函数，能很方便的向Looper中的消息队列添加消息，并且最后由Handler来处理消息。由于一个Activity中可以创建多个线程，假如这些线程都将消息添加到主线程消息队列中，那么消息最后将在Activity中的Handler对象中被执行。

使用的方式是在主线程中实例一个Handler对象myHandler，然后通过调用发送消息的接口myHandler.sendEmptyMessage()；此时消息将放入主线程的消息队列中，Handler主线程中调用handlerhandleMessage接口来处理消息。这种消息驱动的机制常见于实现更新UI界面。如下图

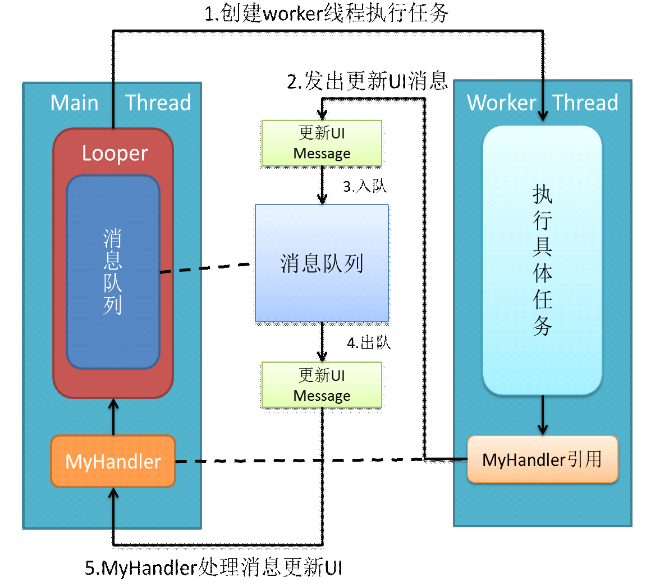


图5.2.3.3

项目中个人信息修改界面使用的更新Ui的方式就类似于上图所显示的方式。下一小节将结合代码介绍个人信息修改界面功能的实现过程。

### 5.2.4 个人中心界面代码分析

1）在代码ModifyActivity中，首先在主线程中定义了两个Handler来处理消息。

Handler myHandler = new Handler(){

public void handleMessage(Message msg) {

//内容省去，用于更新UI

}

super.handleMessage(msg);

}

Handler myHandler2 = new Handler(){

public void handleMessage(Message msg) {

//弹出AlertDialog对话框，提示更新成功

}

super.handleMessage(msg);

}

2）在代码ModifyActivity的public void modify()方法，创建了类似上图的worker线程。

public void modify(){

new Thread(){

public void run(){

Looper.prepare();

//…获得用户输入的数据并进行验证

myHandler2.sendEmptyMessage(0); //用myHandler2给主线程发消息

}

}.start();

}

3）在public void getUser()方法中，同样创建了类似上图的worker线程。

public void getUser(){

new Thread(){

public void run(){

//获得服务器中的数据

myHandler.sendEmptyMessage(0); //用myHandler给主线程发消息

}

}.start();

}

4）在本Activity的onCreate方法的最后先后调用了modify()和getUser()方法。

调用modify()方法的时候，通过myHandler2给主线程的消息队列中发消息。消息的处理由myHandler2的handleMessage()方法来进行处理。在此方法里面弹出AlertDialog对话框，提示更新成功。

同理，调用getUser ()方法的时候，通过myHandler给主线程的消息队列中发消息。消息的处理由myHandler的handleMessage()方法来进行处理。在此方法里面更新UI。

## 5.3 服务器框架分析

本项目服务器由Java语言开发，并发的处理客户端的请求，通过解析客服端的指令来操作数据库。主要使用Socket接收客户端的数据，Java的多线程来实现并发，JDBC（Java Data Base Connectivity）来操作数据库等。下图5.3.1简单的描述了服务器的时序图。图中，服务器管理员点击启动按钮触发启动服务器的事件，实例化Server对象启动线程ServerThread来监听本机端口有没有网络连接，当监听到有连接时返回Socket对象，然后启动ServerAgent线程来解析客户端的数据，这样便实现了多台客户端的同时连接。同时在ServerAgent向DataInputStream（数据输入流）读取网络上客户端的请求，解析请求的内容，分别调用DBUtil中的静态方法来访问、操作数据库并将结果返回。然后将结果通过DataOutputStream（数据输出流）返回给网络客户端。

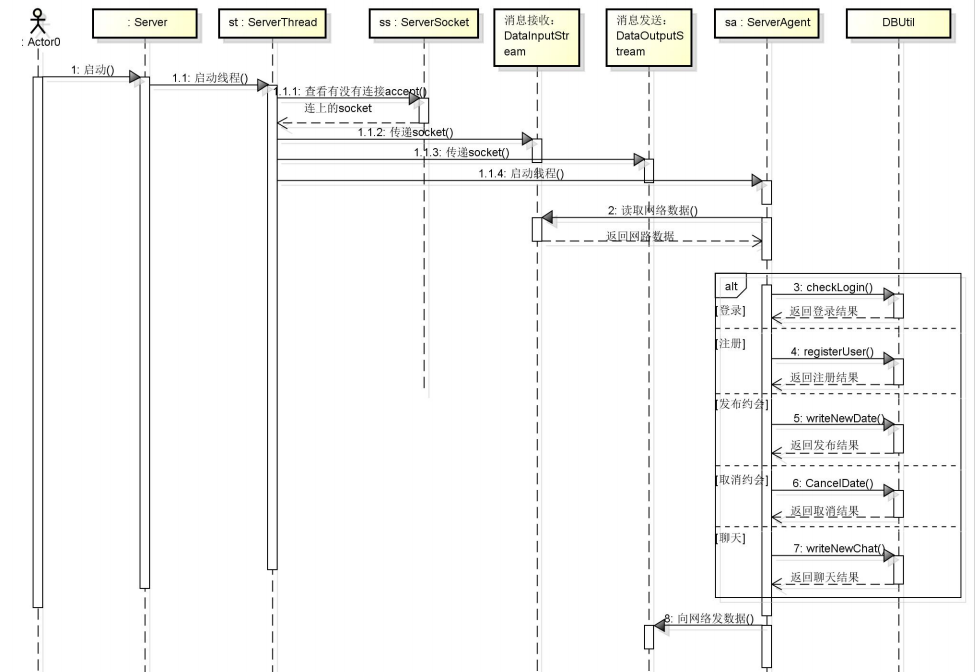


图5.3.1

## 5.4 服务器程序分析

### 5.4.1 服务器类功能介绍

1） Chat类：封装了对话的编号、对话双方的ID和聊天的内容。

2） Comments类 ：封装了评论时间、内容、名称、评论者ID。

3） Configuration类：访问配置文件database.properties，来连接Mysql数据库。

4） ConstantUtil类：封装了程序定义的静态常量。

5） Date类：封装了约会的相关信息。

6） DBUtil类：数据库工具类，提供一系列的静态方法实现对数据库的操作。

7） HeadInfo类：封装了头像的相关信息。

8） Server类：程序的主类，显示服务器的前端界面。

9） ServerThread类：处理端口的连接，来实现多台客户端登录。

10）ServerAgent类：读取网络数据输入源，并解析数据后操作数据库，最后将访问、操作数据库的结果返回给网络数据输出源。

11）User类：封装用户信息。

### 5.4.2 服务器网络相关代码分析

Java类库中Socket类能方便的实现客户端和服务器的连接。它唯一的确定了远端主机上ip地址和端口号的组合。服务器端通过监听某个端口是否有连接请求，同时客户端向服务器端发连接请求，服务器端向客户端返回接受消息，以此来建立连接。主要过程包括：创建，连接，读写，关闭。

1. 在Server.java中创建ServerSocket监听客户请求。

ss = new ServerSocket(Integer.parseInt(port.getText().trim()));

st = new ServerThread(ss);

st.start();

1. 在ServerThread.java中阻塞的等待客户端的请求，建立连接后返回Socket对象。

Socket socket = ss.accept();

1. 在ServerAgent.java中 ，由Socket对象创建输入流、输出流对象。

this.din = new DataInputStream(socket.getInputStream());

this.dout = new DataOutputStream(socket.getOutputStream());

1. 在ServerAgent.java中，线程执行方法run()中，读、写输入输出流。

String msg = din.readUTF();

//对msg的解析，访问数据库。然后将结果写如输出流中。

dout.writeUTF(loginInfo);

5）在ServerAgent.java中，关闭输入、输出流，关闭Socket。

this.din.close();

this.dout.close();

this.socket.close();

### 5.4.3服务器数据库相关代码分析

服务器中对数据库的操作都是通过DBUtil数据库工具类提供一系列的静态方法实现的。下面介绍一下java操作数据库的方式和使用方式。

1） 配置文件database.properties

//驱动程序名

database.driver=com.mysql.jdbc.Driver

//URL指向要访问的数据库名

database.url=jdbc:mysql://localhost:3306/date?useUnicode=true&characterEncoding=utf-8

//数据库的用户名

database.username=root

//数据库的密码

database.password=pass

2）在DBUtil.java中创建一个Configuration对象，拿到配置文件database.properties中的内容。

Configuration rc = new Configuration("database.properties");

driver = rc.getValue("database.driver");

url = rc.getValue("database.url");

userName = rc.getValue("database.username");

passwd = rc.getValue("database.password");

3）在DBUtil.java中，使用JDBC直接访问数据库，返回Connection对象。

Class.forName(driver); //加载驱动程序 con=DriverManager.getConnection(url,userName,passwd); //连接数据库

4） 在DBUtil.java中，使用Statement来执行SQL语句

Statement st = con.createStatement(); //创建一个Statement对象

st.executeUpdate(sql);

# 总结

目的是让学员了解Android社交综合项目知识。