应用基础实践二

课程设计

题目名称： 微博

姓 名： 杨志轩

学 号： 8208180813

专 业： 计算机科学与技术

班 级： 计科1804

指导教师： 张伟

编写日期： 2021/1/17

目录

[第一章 开发工具及环境说明 3](#_Toc28274)

[第二章 微博应用业务分析 5](#_Toc20317)

[第三章 微博应用系统设计 6](#_Toc5019)

[2.1 系统功能定义 6](#_Toc12566)

[2.2 关键问题设计 6](#_Toc31857)

[第四章 微博应用源代码清单 8](#_Toc18342)

[第五章 微博应用运行结果与测试分析 6](#_Toc21546)0

[第六章 滴滴物流业务分析 6](#_Toc17001)6

[第七章 滴滴物流系统设计 67](#_Toc25020)

[2.1 系统功能定义 67](#_Toc5442)

[2.2 关键问题设计 67](#_Toc5920)

[第八章 滴滴物流源代码清单 68](#_Toc2401)

[第九章 滴滴物流运行结果与测试分析 113](#_Toc9084)

[第十章 结论与心得 119](#_Toc14657)

第一章 开发工具及环境说明  
手机端：Android Studio + SDK

服务器端：idea+TomCat

1.JDK安装与配置

Step 1:下载JDK 可以到官网进行下载：[Jdk官方下载](http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html) 也可到笔者网盘下载：[笔者网盘](http://pan.baidu.com/s/1o6LxZHC) PS:这个随便下一个都可以，关系不大，32位的只能下32位哦！

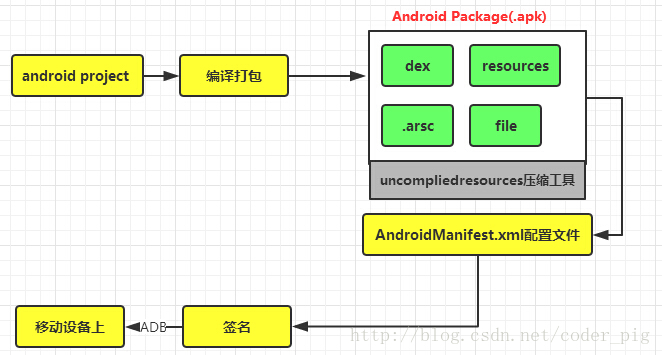
Step 2:JDK安装 傻瓜式的下一步即可！

Step 3:环境变量的配置 配置环境变量是为了方便我们一些命令行的操作，后续会用到！ 右键我的电脑——>高级——>环境变量

新建JAVA\_HOME  修改PATH变量，别把原本的东西删掉！！！！  新建CLASSPATH  验证环境是否配置完成 打开电脑的cmd(命令行)，win键 + R输入cmd，然后在命令行依次javac和java.

2.选择Android Studio + SDK开发工具

3.APP程序打包与安装的流程：



第二章 微博应用业务分析

微博应用应实现基本的客户登录功能，其次客户可发表微博，查看微博列表，并且能对微博列表中的微博点赞，评论，系统需要数据库支持，需要服务器处理客户请求，客户端要有基本的逻辑功能和优美的UI设计,客户端登陆后，可以查看个人用户中心，包括头像和一个用videoview实现的mv视频

与传统的博客相比，微博发布更便利、传播更迅速，发布字数限制在140字之内，方便用户通过电脑、手机等多平台浏览发布，所发布信息是传达，并可一键转发。微博相比传统博客那种需要考虑文题、组织语言修辞来叙述的长篇大论，以“短、灵、快”为特点的“微博”几乎不需要很高成本，无论你是用电脑还是手机，只需三言两语，就可记录下自己某刻的心情、某一瞬的感悟，或者某条可供分享和收藏的信息，这样的即时表述显然更加迎合我们快节奏的生活。微博微博客草根性更强，且广泛分布在桌面、浏览器、移动终端等多个平台上，有多种商业模式并存，或形成多个垂直细分领域的可能。微博更符合现在人的生活节奏和习惯。而新技术的运用使得用户更容易对访问者者留言进行回复，从而形成良好的互动关系。

第三章 微博应用系统设计

2.1 系统功能定义

①用Android开发微博客户端；

②用JSP或PHP开发微博Web服务器；

③用MySQL做Web服务器的后台数据库；

④可以发表微博、查看微博及微博列表等功能；

⑤可以发表评论功能；

⑥可以实现用户客户端登陆功能。

2.2 关键问题设计

①客户端登陆  
客户端登陆，用户需输入账户密码，前台判断数据是否为空，并给出相应提示，然后将数据包装成json格式，传到服务器，服务器解析数据，并查询数据库，查询成功，跳转登陆界面，否则，给出相应错误提示

②微博列表  
微博列表采用scrollview+LinearLayout实现，每一条微博，使用textview实现，发表微博时，采用对话弹框，获取输入内容，传到数据库中，并动态添加textview。每次启动时都会从数据库中加载数据，其次还提供刷新和欢迎用户功能

③点赞评论功能  
为textview添加点击事件，点击时会跳转页面到评论点赞页面，点赞会弹出爱心动画，评论会弹出对话框，并在下方动态添加评论内容，使用textview实现

④微博列表提供首页和我的功能，首页会返回到微博列表页面，点击我的会跳转到用户个人界面，用户界面拥有头像和一个用videoview实现的MV，用户点击爱心可以播放MV或者取消MV。

第四章 微博应用源代码清单

客户端  
Mainactivity

**package** com.example.myvb;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Looper;  
**import** android.text.TextUtils;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **public final static** String ***MyURL*** = **new** ur().**MyURL1**;  
  
 **private** TextView **textView3**;  
 **private** TextView **textView2**;  
 **private** Button **button2**;  
 **private** ImageView **imageView2**;  
 **private** EditText **editText**;  
 **private** EditText **editText2**;  
 **private** JSONObject **object**;  
 Handler **handler**;  
 **private** TextView **t1**;  
 **private** TextView **t2**;  
 @Override  
 **public** <T **extends** View> T findViewById(**int** id) {  
 **return super**.findViewById(id);  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// t1 = (TextView) findViewById(R.id.textView3);  
 //t2= (TextView) findViewById(R.id.textView2);* **button2** = (Button) findViewById(R.id.***button2***);  
 **imageView2** = (ImageView) findViewById(R.id.***imageView2***);  
 **editText** = (EditText) findViewById(R.id.***editText***);*//用户名* **editText2** = (EditText) findViewById(R.id.***editText2***);*//密码* **new** ur().*setLightAni*(**imageView2**);  
 *// t1.setBackground(t1.getContext().getDrawable(R.drawable.h2));  
 //t1.getBackground().setAlpha(100);  
 //t2.setBackground(t2.getContext().getDrawable(R.drawable.h2));  
//t2.getBackground().setAlpha(100);* **button2**.setBackground(**button2**.getContext().getDrawable(R.drawable.***h2***));  
 **button2**.getBackground().setAlpha(100);  
 **button2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 login( );  
 }  
 });  
  
 }  
**public void** login( ){  
  
 **final** String qq=**editText**.getText().toString().trim();  
 **final** String pwd=**editText2**.getText().toString().trim();  
 **if**(TextUtils.*isEmpty*(qq)){  
 Toast.*makeText*(**this**,**"用户名为空登录失败"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **if**(TextUtils.*isEmpty*(pwd)){  
 Toast.*makeText*(**this**,**"密码为空登陆失败"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 *//包装数据* **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**, qq);  
 **object**.put(**"pass"**,pwd);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 *//传输数据* **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 *//把你上面那段网络访问的代码放在这里* **try** {  
 URL url = **new** URL(***MyURL***);  
 */\* HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 ;  
 conn.setRequestMethod("POST");  
 conn.setRequestMethod("POST");  
 conn.setRequestProperty("ser-Agent", "Fiddler");  
 conn.setRequestProperty("Content-Type", "application/json");  
 conn.setConnectTimeout(5 \* 1000);  
\*/* HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
  
  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 **final** JSONObject objectT = **new** JSONObject(jsonStr);  
 *// Log.i("objectT", objectT.getString("server"));* **if**(objectT.optString(**"message"**).equals(**"返回成功"**)){  
 Intent it = **new** Intent(MainActivity.**this**, activity2.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, qq);  
  
 it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);  
 }  
 **else**{  
 Looper.*prepare*();  
 Toast.*makeText*(MainActivity.**this**,**"用户名或密码错误"**, Toast.***LENGTH\_SHORT***).show();  
  
 Looper.*loop*();  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 */\*handler.post(new Runnable() {  
 @Override  
 public void run() {  
  
 Toast.makeText(MainActivity.this, "连接超时", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
 });\*/* }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
  
}  
  
  
  
  
  
  
  
}

Activity2  
**package** com.example.myvb;  
  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Looper;  
**import** android.os.Message;  
**import** android.util.TypedValue;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ImageButton;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
  
**public class** activity2 **extends** AppCompatActivity {  
 **public final static** String ***MyURL*** = **new** ur().**MyURL**;  
  
 **private** TextView **text6**;  
 **private** ImageButton **i1**;  
 **private** Button **b2**;  
 **private** Button **b1**;  
 **public** String **text1**=**""**;  
 String **hehe**;  
 String **name**;  
 *// private Handler handler;* JSONObject **object**;  
 **private** ImageButton **imageButton**;  
 String **result**;  
 Handler **handler**=**new** Handler(){  
 @Override  
 **public void** handleMessage(@NonNull Message msg) {  
 **super**.handleMessage(msg);  
 **switch**(msg.**what**) {  
  
 **case** 1:  
 review();  
 **break**;  
  
 }  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_activity2***);  
  
 **imageButton** = (ImageButton) findViewById(R.id.***imageButton***);  
  
 **text6**= (TextView) findViewById(R.id.***text6***);  
 Bundle bundle=getIntent().getExtras();  
 **name**=bundle.getString(**"user"**);  
 **text6**.setText(**name**);  
 **i1** = (ImageButton) findViewById(R.id.***i1***);*//发表微博* **new** ur().*setLightExpendAni*(**i1**);  
  
 **b2** = (Button) findViewById(R.id.***b2***);*//我的* **b1** = (Button) findViewById(R.id.***b1***);*//首页* **i1**.setBackground(**i1**.getContext().getDrawable(R.drawable.***h2***));  
 **i1**.getBackground().setAlpha(100);  
 **i1**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 go1();  
  
  
 }  
 });  
 **imageButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
  
 **public void** onClick(**final** View view) {  
  
  
 re3();  
  
 }  
  
  
 });  
  
  
  
 re3();  
 **b1**.setBackground(**b1**.getContext().getDrawable(R.drawable.***h2***));  
 **b1**.getBackground().setAlpha(200);  
 **b2**.setBackground(**b2**.getContext().getDrawable(R.drawable.***h2***));  
 **b2**.getBackground().setAlpha(200);  
 **b1**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
  
 Intent it = **new** Intent(activity2.**this**, activity2.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
  
 it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);\*/* startActivity(it);  
  
 }  
 });  
 **b2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 *// go1();* Intent it = **new** Intent(activity2.**this**, avtivity4.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
  
 it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);\*/* startActivity(it);  
  
 }  
 });  
  
 }  
  
**public void** re3(){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
  
 **object**.put(**"id"**,**"2"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **result** = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 System.***out***.println(**result**);  
 *// review();* in.close();  
 conn.disconnect();  
 **handler**.sendEmptyMessage(1);  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 **handler**.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 Toast.*makeText*(activity2.**this**, **"连接超时"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 });  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
}  
**public void** go1(){  
  
 **final** EditText inputServer = **new** EditText(activity2.**this**);  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(activity2.**this**);  
 builder.setTitle(**"请输入内容"**).setIcon(android.R.drawable.***ic\_dialog\_info***).setView(inputServer)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
  
 builder.setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **text1** = inputServer.getText().toString();  
 Date now = **new** Date();  
 SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"yyyy/MM/dd HH:mm:ss"**);*//可以方便地修改日期格式* **hehe** = dateFormat.format( now );  
 **if**(!**text1**.equals(**""**)){  
  
 LinearLayout linear=(LinearLayout) findViewById(R.id.***lay1***);  
 *//添加文本,this代表当前项目* TextView tv=**new** TextView(activity2.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackgroundResource(R.drawable.***h2***);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setText(**"\r\n"**+**" "**+**name** + **":"** + **" "**+**hehe**+**"\r\n"**+**"\r\n"** +**" "**+ **text1**);  
 tv.setId(View.*generateViewId*());  
 send1(**name**,**text1**);  
  
 *//tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使用* tv.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 **int** a2= view.getId();  
 TextView a1=(TextView) findViewById(a2);  
 String a3=a1.getText().toString();  
 go2( a3);  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
  
 linear.addView(tv,0);  
  
  
 **text1**=**""**;  
 }  
  
 }}  
 );  
 builder.show();  
  
  
}  
**public void** go2(String a1){  
  
  
 Intent it = **new** Intent(activity2.**this**, activity3.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, a1);  
 bundle.putString(**"user1"**, **name**);  
  
 it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);\*/* startActivity(it);  
  
}  
  
  
**public void** send1( String name,String vb){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**, name);  
 **object**.put(**"pass"**,vb);  
 **object**.put(**"id"**,**"1"**);  
  
 **object**.put(**"time"**,**hehe**);  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 JSONArray jsonArray = **new** JSONArray(jsonStr);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
  
  
 **if**(name.equals(**"增加成功"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(activity2.**this**,**"增加成功"**, Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 **handler**.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 Toast.*makeText*(activity2.**this**, **"连接超时"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 });  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
  
  
  
}  
**public void** review (){  
 **try** {  
 *//解码* JSONArray jsonArray = **new** JSONArray(**result**);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
 **if**(name.equals(**"失败"**)){  
  
 Looper.*prepare*();  
 Toast.*makeText*(activity2.**this**, **"数据库为空"**, Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
 }  
 **else** {  
  
 *// Log.i("objectT", objectT.getString("server"));* LinearLayout linear1= (LinearLayout) findViewById(R.id.***lay1***);  
 linear1.removeAllViews();  
  
 **for** (**int** i = 0; i < jsonArray.length(); i++) {  
 *// JSON数组里面的具体-JSON对象* JSONObject jsonObject1 = jsonArray.getJSONObject(i);  
 String name1 = jsonObject1.optString(**"user"**);  
  
 String pass1 = jsonObject1.optString(**"pass"**);  
 String time = jsonObject1.optString(**"time"**);  
  
  
 TextView tv = **new** TextView(activity2.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setText(**"\r\n"**+**" "**+name1 + **":"** + **" "**+time+**"\r\n"**+**"\r\n"**+**" "**+ pass1);  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setId(View.*generateViewId*());  
 *// send1(name,text1);  
 //tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使用* tv.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 **int** a2 = view.getId();  
 TextView a1 = (TextView) findViewById(a2);  
 String a3 = a1.getText().toString();  
 go2(a3);  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
  
 linear1.addView(tv, 0);  
 }  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
}

Activity3  
**package** com.example.myvb;  
  
**import** android.content.DialogInterface;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Looper;  
**import** android.os.Message;  
**import** android.util.TypedValue;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageButton;  
**import** android.widget.LinearLayout;  
**import** android.widget.PopupWindow;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.appcompat.widget.AppCompatTextView;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
  
**public class** activity3 **extends** AppCompatActivity {  
 **private** FlowLikeView **likeViewLayout**;  
 **private** TextView **text9**;  
 PopupWindow **mPopupWindow**;  
 JSONObject **object**;  
 String **result**;  
 *// String hehe;* AppCompatTextView **tvAnimation** ;  
 **private** ImageButton **imageButton2**;  
 **private** ImageButton **imageButton9**;  
 **public final static** String ***MyURL*** = **new** ur().**MyURL**;  
 String **text1**;  
 String **name**;  
 String **name1**;  
 Handler **handler**=**new** Handler(){  
 @Override  
 **public void** handleMessage(@NonNull Message msg) {  
 **super**.handleMessage(msg);  
 **switch**(msg.**what**) {  
  
 **case** 1:  
 review();  
 **break**;  
  
 }  
 }  
 };  
 **private** ImageButton **imageButton4**;  
 @Override  
  
 **protected void** onCreate(Bundle savedInstanceState) {  
  
  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_activity3***);  
  
  
  
 **imageButton4** = (ImageButton) findViewById(R.id.***imageButton2***);  
  
 **text9**= (TextView) findViewById(R.id.***t9***);  
 Bundle bundle=getIntent().getExtras();  
 **name**=bundle.getString(**"user"**);  
 **name1**=bundle.getString(**"user1"**);  
  
 **text9**.setBackground(**text9**.getContext().getDrawable(R.drawable.***h2***));  
 **text9**.getBackground().setAlpha(100);  
 **text9**.setTextColor(Color.*rgb*(6,207,160));  
 **text9**.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 **text9**.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 **text9**.setText(**name**);  
 initView();  
 re3();  
 **imageButton2** = (ImageButton) findViewById(R.id.***imageButton4***);  
 **imageButton9** = (ImageButton) findViewById(R.id.***imageButton3***);  
  
 **imageButton2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 addLikeView(view);  
  
  
 }  
 });  
  
 **imageButton9**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 *// Toast.makeText(activity3.this, "执行中", Toast.LENGTH\_SHORT).show();* go1();  
  
 }  
 });  
  
 **imageButton4**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 re3();  
  
  
 }  
 });  
  
 }  
  
  
 **private void** initView() {  
 **likeViewLayout** = (FlowLikeView) findViewById(R.id.***flowLikeView***);  
 }  
  
 **public void** addLikeView(View view) {  
 **likeViewLayout**.addLikeView();  
 }  
  
  
 **public void** go1(){  
  
 **final** EditText inputServer = **new** EditText(activity3.**this**);  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(activity3.**this**);  
 builder.setTitle(**"请输入回复内容"**).setIcon(android.R.drawable.***ic\_dialog\_info***).setView(inputServer)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
  
 builder.setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **text1** = inputServer.getText().toString();  
 **if**(!**text1**.equals(**""**)){  
  
 LinearLayout linear=(LinearLayout) findViewById(R.id.***lay2***);  
 *//添加文本,this代表当前项目* TextView tv=**new** TextView(activity3.**this**);  
 *//tv.setText(name1+" : "+"\r\n"+text1);* LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setText(**"\r\n"**+**" "**+**name1**+**" : "**+**"\r\n"**+**"\r\n"**+**" "**+**text1**);  
 send1(**name1**,**text1**);  
 *//tv.setId(View.generateViewId());  
 //tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使* linear.addView(tv,0);  
 **text1**=**""**;  
 }  
  
 }}  
 );  
 builder.show();  
  
  
 }  
 **public void** re3(){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"name"**,**name**);  
  
 **object**.put(**"id"**,**"4"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **result** = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 System.***out***.println(**result**);  
 *// review();* in.close();  
 conn.disconnect();  
 **handler**.sendEmptyMessage(1);  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 */\* handler.post(new Runnable() {  
 @Override  
 public void run() {  
 e.printStackTrace();  
  
 Toast.makeText(activity3.this, "连接超时", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
 });  
\*/* }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
 }  
  
  
 **public void** review () {  
 **try** {  
 *//解码* JSONArray jsonArray = **new** JSONArray(**result**);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
 **if** (name.equals(**"失败"**)) {  
  
  
 Toast.*makeText*(activity3.**this**, **"数据库为空"**, Toast.***LENGTH\_SHORT***).show();  
  
 } **else** {  
  
 *// Log.i("objectT", objectT.getString("server"));* LinearLayout linear1 = (LinearLayout) findViewById(R.id.***lay2***);  
 linear1.removeAllViews();  
  
 **for** (**int** i = 0; i < jsonArray.length(); i++) {  
 *// JSON数组里面的具体-JSON对象* JSONObject jsonObject1 = jsonArray.getJSONObject(i);  
 String name1 = jsonObject1.optString(**"user"**);  
  
 String pass1 = jsonObject1.optString(**"pass"**);  
 *// String time = jsonObject1.optString("time");* TextView tv = **new** TextView(activity3.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0, 0, 0, ((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 *// tv.setBackgroundResource(R.drawable.h2);* tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
 tv.setText(**"\r\n"**+**" "**+name1 + **":"** + **"\r\n"**+**"\r\n"** +**" "**+ pass1);  
  
 linear1.addView(tv, 0);  
 }  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 **public void** send1( String namex,String vb){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**, namex);  
 **object**.put(**"pass"**,vb);  
 **object**.put(**"id"**,**"3"**);  
  
 **object**.put(**"name"**,**name**);  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 JSONArray jsonArray = **new** JSONArray(jsonStr);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
  
  
 **if**(name.equals(**"增加成功"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(activity3.**this**,**"增加成功"**, Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 **handler**.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
  
 Toast.*makeText*(activity3.**this**, **"连接超时"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 });  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
  
  
  
 }  
  
  
  
  
}

Avtivity4  
**package** com.example.myvb;  
  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.MediaController;  
**import** android.widget.RelativeLayout;  
**import** android.widget.VideoView;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**public class** avtivity4 **extends** AppCompatActivity {  
 **private** VideoView **videoView2**;  
 **private** RelativeLayout **lay3**;  
  
 **private** ImageView **imageView7**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_avtivity4***);  
  
 **lay3** = (RelativeLayout) findViewById(R.id.***lay3***);  
  
 **lay3**.setBackground(**lay3**.getContext().getDrawable(R.drawable.***q2***));  
 *// lay3.getBackground().setAlpha(200);* **videoView2** = (VideoView) findViewById(R.id.***videoView2***);  
 **imageView7** = (ImageView) findViewById(R.id.***imageView7***);  
 **videoView2** .setVideoPath(**"android.resource://"**+getPackageName()+**"/"**+R.raw.***r2***)  
 ;  
 *//实例化媒体控制器* MediaController mediaController=**new** MediaController(**this**);  
 mediaController.setMediaPlayer( **videoView2** );  
 mediaController.show(5);  
 **videoView2** .setMediaController(mediaController);  
 **imageView7**.setOnClickListener(**new** View.OnClickListener() {  
  
 @Override  
 **public void** onClick(**final** View view) {  
 **if** (**videoView2**.getVisibility()==View.***VISIBLE***) {  
 **videoView2**.setVisibility(View.***INVISIBLE***);  
 **videoView2**.start();  
 }  
 **else**{  
 **videoView2**.setVisibility(View.***VISIBLE***);  
 **videoView2**.pause();  
 }  
  
 }  
 });  
  
 }  
}

Flowlikeview

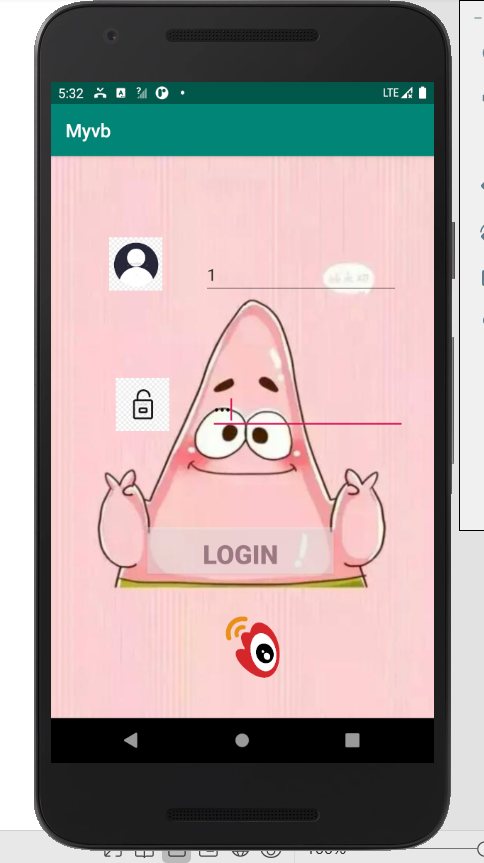
**package** com.example.myvb;  
  
**import** android.content.Context;  
**import** android.graphics.PointF;  
**import** android.graphics.drawable.Drawable;  
**import** androidx.core.content.ContextCompat;  
**import** androidx.core.view.ViewCompat;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.RelativeLayout;  
  
**import** com.nineoldandroids.animation.Animator;  
**import** com.nineoldandroids.animation.AnimatorListenerAdapter;  
**import** com.nineoldandroids.animation.AnimatorSet;  
**import** com.nineoldandroids.animation.ObjectAnimator;  
**import** com.nineoldandroids.animation.TypeEvaluator;  
**import** com.nineoldandroids.animation.ValueAnimator;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
**import** java.util.Random;  
  
*/\*\*  
 \* Created by Anenn on 6/10/16.  
 \*/***public class** FlowLikeView **extends** RelativeLayout {  
  
 **private** List<Drawable> **mLikeDrawables**; *// 图片的集合* **private** LayoutParams **mLayoutParams**; *// 用于设置动画对象的位置参数* **private** Random **mRandom**; *// 用于产生随机数,如生成随机图片* **private int mViewWidth**; *// 控件的宽度* **private int mViewHeight**; *// 控件的高度* **private int mPicWidth**; *// 图片的宽度* **private int mPicHeight**; *// 图片的高度* **private int mChildViewHeight**; *// 在 XML 布局文件中添加的子View的总高度* **public** FlowLikeView(Context context) {  
 **this**(context, **null**);  
 }  
  
 **public** FlowLikeView(Context context, AttributeSet attrs) {  
 **this**(context, attrs, 0);  
 }  
  
 **public** FlowLikeView(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
  
 initParams();  
 }  
  
 **private void** initParams() {  
 **mLikeDrawables** = **new** ArrayList<>();  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart0***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart1***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart2***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart3***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart4***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart5***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart6***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart7***));  
 **mLikeDrawables**.add(generateDrawable(R.drawable.***heart8***));  
  
 *// 获取图片的宽高, 由于图片大小一致,故直接获取第一张图片的宽高* **mPicWidth** = **mLikeDrawables**.get(0).getIntrinsicWidth();  
 **mPicHeight** = **mLikeDrawables**.get(0).getIntrinsicHeight();  
  
 *// 初始化布局参数* **mLayoutParams** = **new** LayoutParams(**mPicWidth**, **mPicHeight**);  
 **mLayoutParams**.addRule(***CENTER\_HORIZONTAL***);  
 **mLayoutParams**.addRule(***ALIGN\_PARENT\_BOTTOM***);  
  
 **mRandom** = **new** Random();  
 }  
  
 **private** Drawable generateDrawable(**int** resID) {  
 **return** ContextCompat.*getDrawable*(getContext(), resID);  
 }  
  
 @Override  
 **protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  
 **super**.onMeasure(widthMeasureSpec, heightMeasureSpec);  
  
 **if** (**mChildViewHeight** <= 0) {  
 **for** (**int** i = 0, size = getChildCount(); i < size; i++) {  
 View childView = getChildAt(i);  
 measureChild(childView, widthMeasureSpec, heightMeasureSpec);  
 **mChildViewHeight** += childView.getMeasuredHeight();  
 }  
  
 *// 设置底部间距* **mLayoutParams**.**bottomMargin** = **mChildViewHeight**;  
 }  
 }  
  
 @Override  
 **protected void** onSizeChanged(**int** w, **int** h, **int** oldw, **int** oldh) {  
 **super**.onSizeChanged(w, h, oldw, oldh);  
  
 **mViewWidth** = getWidth();  
 **mViewHeight** = getHeight();  
 }  
  
 */\*\*  
 \* 动态添加 FlowView  
 \*/* **public void** addLikeView() {  
 ImageView likeView = **new** ImageView(getContext());  
 likeView.setImageDrawable(**mLikeDrawables**.get(**mRandom**.nextInt(**mLikeDrawables**.size())));  
 likeView.setLayoutParams(**mLayoutParams**);  
  
 addView(likeView);  
 startAnimation(likeView);  
 }  
  
 **private void** startAnimation(View target) {  
 *// 设置进入动画* AnimatorSet enterAnimator = generateEnterAnimation(target);  
 *// 设置路径动画* ValueAnimator curveAnimator = generateCurveAnimation(target);  
  
 *// 设置动画集合, 先执行进入动画,最后再执行运动曲线动画* AnimatorSet finalAnimatorSet = **new** AnimatorSet();  
 finalAnimatorSet.setTarget(target);  
 finalAnimatorSet.playSequentially(enterAnimator, curveAnimator);  
 finalAnimatorSet.addListener(**new** AnimationEndListener(target));  
 finalAnimatorSet.start();  
 }  
  
 */\*\*  
 \* 生成进入动画  
 \*  
 \** ***@return*** *动画集合  
 \*/* **private** AnimatorSet generateEnterAnimation(View target) {  
 ObjectAnimator alpha = ObjectAnimator.*ofFloat*(target, **"alpha"**, 0.2f, 1f);  
 ObjectAnimator scaleX = ObjectAnimator.*ofFloat*(target, **"scaleX"**, 0.5f, 1f);  
 ObjectAnimator scaleY = ObjectAnimator.*ofFloat*(target, **"scaleY"**, 0.5f, 1f);  
 AnimatorSet enterAnimation = **new** AnimatorSet();  
 enterAnimation.playTogether(alpha, scaleX, scaleY);  
 enterAnimation.setDuration(150);  
 enterAnimation.setTarget(target);  
 **return** enterAnimation;  
 }  
  
 */\*\*  
 \* 生成曲线运动动画  
 \*  
 \** ***@return*** *动画集合  
 \*/* **private** ValueAnimator generateCurveAnimation(View target) {  
 CurveEvaluator evaluator = **new** CurveEvaluator(generateCTRLPointF(1), generateCTRLPointF(2));  
 ValueAnimator valueAnimator = ValueAnimator.*ofObject*(evaluator,  
 **new** PointF((**mViewWidth** - **mPicWidth**) / 2, **mViewHeight** - **mChildViewHeight** - **mPicHeight**),  
 **new** PointF((**mViewWidth**) / 2 + (**mRandom**.nextBoolean() ? 1 : -1) \* **mRandom**.nextInt(100), 0));  
 valueAnimator.setDuration(3000);  
 valueAnimator.addUpdateListener(**new** CurveUpdateLister(target));  
 valueAnimator.setTarget(target);  
  
 **return** valueAnimator;  
 }  
  
 */\*\*  
 \* 生成贝塞儿曲线的控制点  
 \*  
 \** ***@param value*** *设置控制点 y 轴上取值区域  
 \** ***@return*** *控制点的 x y 坐标  
 \*/* **private** PointF generateCTRLPointF(**int** value) {  
 PointF pointF = **new** PointF();  
 pointF.**x** = **mViewWidth** / 2 - **mRandom**.nextInt(100);  
 pointF.**y** = **mRandom**.nextInt(**mViewHeight** / value);  
  
 **return** pointF;  
 }  
  
 */\*\*  
 \* 自定义估值算法, 计算对象当前运动的具体位置 Point  
 \*/* **private class** CurveEvaluator **implements** TypeEvaluator<PointF> {  
  
 *// 由于这里使用的是三阶的贝塞儿曲线, 所以我们要定义两个控制点* **private** PointF **ctrlPointF1**;  
 **private** PointF **ctrlPointF2**;  
  
 **public** CurveEvaluator(PointF ctrlPointF1, PointF ctrlPointF2) {  
 **this**.**ctrlPointF1** = ctrlPointF1;  
 **this**.**ctrlPointF2** = ctrlPointF2;  
 }  
  
 @Override  
 **public** PointF evaluate(**float** fraction, PointF startValue, PointF endValue) {  
  
 *// 这里运用了三阶贝塞儿曲线的公式, 请自行上网查阅* **float** leftTime = 1.0f - fraction;  
 PointF resultPointF = **new** PointF();  
  
 *// 三阶贝塞儿曲线* resultPointF.**x** = (**float**) Math.*pow*(leftTime, 3) \* startValue.**x** + 3 \* (**float**) Math.*pow*(leftTime, 2) \* fraction \* **ctrlPointF1**.**x** + 3 \* leftTime \* (**float**) Math.*pow*(fraction, 2) \* **ctrlPointF2**.**x** + (**float**) Math.*pow*(fraction, 3) \* endValue.**x**;  
 resultPointF.**y** = (**float**) Math.*pow*(leftTime, 3) \* startValue.**y** + 3 \* (**float**) Math.*pow*(leftTime, 2) \* fraction \* **ctrlPointF1**.**y** + 3 \* leftTime \* fraction \* fraction \* **ctrlPointF2**.**y** + (**float**) Math.*pow*(fraction, 3) \* endValue.**y**;  
  
 *// 二阶贝塞儿曲线  
// resultPointF.x = (float) Math.pow(leftTime, 2) \* startValue.x + 2 \* fraction \* leftTime \* ctrlPointF1.x  
// + ((float) Math.pow(fraction, 2)) \* endValue.x;  
// resultPointF.y = (float) Math.pow(leftTime, 2) \* startValue.y + 2 \* fraction \* leftTime \* ctrlPointF1.y  
// + ((float) Math.pow(fraction, 2)) \* endValue.y;* **return** resultPointF;  
 }  
 }  
  
 */\*\*  
 \* 动画曲线路径更新监听器, 用于动态更新动画作用对象的位置  
 \*/* **private class** CurveUpdateLister **implements** ValueAnimator.AnimatorUpdateListener {  
 **private** View **target**;  
  
 **public** CurveUpdateLister(View target) {  
 **this**.**target** = target;  
 }  
  
 @Override  
 **public void** onAnimationUpdate(ValueAnimator animation) {  
 *// 获取当前动画运行的状态值, 使得动画作用对象沿着曲线(涉及贝塞儿曲线)运动* PointF pointF = (PointF) animation.getAnimatedValue();  
 ViewCompat.*setX*(**target**, pointF.**x**);  
 ViewCompat.*setY*(**target**, pointF.**y**);  
 *// 改变对象的透明度* ViewCompat.*setAlpha*(**target**, 1 - animation.getAnimatedFraction());  
 }  
 }  
  
 */\*\*  
 \* 动画结束监听器,用于释放无用的资源  
 \*/* **private class** AnimationEndListener **extends** AnimatorListenerAdapter {  
 **private** View **target**;  
  
 **public** AnimationEndListener(View target) {  
 **this**.**target** = target;  
 }  
  
 @Override  
 **public void** onAnimationEnd(Animator animation) {  
 **super**.onAnimationEnd(animation);  
  
 removeView(**target**);  
 }  
 }  
}

Ur

**package** com.example.myvb;  
  
**import** android.view.View;  
**import** android.view.animation.AlphaAnimation;  
**import** android.view.animation.Animation;  
**import** android.view.animation.AnimationSet;  
**import** android.view.animation.LinearInterpolator;  
**import** android.view.animation.RotateAnimation;  
**import** android.view.animation.ScaleAnimation;  
**import** android.view.animation.TranslateAnimation;  
  
**public class** ur {  
 **public** String **MyURL** = **"http://10.0.17.210:8080/andsv/ansv1"**;  
 **public** String **MyURL1** = **"http://10.0.17.210:8080/andsv/ansv"**;  
 **public static void** setLightExpendAni(View V)  
 {  
 AnimationSet aniSet = **new** AnimationSet(**true**);  
 **final int** ANITIME = 1200;  
  
 *// 尺寸变化动画，设置尺寸变化* ScaleAnimation scaleAni = **new** ScaleAnimation(0.98f, 1.1f, 0.98f, 1.24f, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 scaleAni.setDuration(ANITIME); *// 设置动画效果时间* scaleAni.setRepeatMode(Animation.***RESTART***); *// 重新播放* scaleAni.setRepeatCount(Animation.***INFINITE***);*// 循环播放* aniSet.addAnimation(scaleAni); *// 将动画效果添加到动画集中  
  
 // 透明度变化* AlphaAnimation alphaAni = **new** AlphaAnimation(1f, 0.05f);  
 alphaAni.setDuration(ANITIME); *// 设置动画效果时间* alphaAni.setRepeatMode(Animation.***RESTART***); *// 重新播放* alphaAni.setRepeatCount(Animation.***INFINITE***);*// 循环播放* aniSet.addAnimation(alphaAni); *// 将动画效果添加到动画集中* V.startAnimation(aniSet); *// 添加光效动画到控件* }  
 **public static void** AddAlphaAni(View view)  
 {  
 *AddAlphaAni*(view, 0, 1, 1000, Animation.***REVERSE***, Animation.***INFINITE***);  
 }  
  
 */\*\*  
 \* 为View添加透明度变换效果，透明度从fromAlpha变化到toAlpha，变化持续时间durationMillis，重复模式repeatMode  
 \*/* **public static void** AddAlphaAni(View view, **float** fromAlpha, **float** toAlpha, **long** durationMillis, **int** repeatMode, **int** repeatCount)  
 {  
 AlphaAnimation alphaAni = **new** AlphaAnimation(fromAlpha, toAlpha);  
 alphaAni.setDuration(durationMillis); *// 设置动画效果时间* alphaAni.setRepeatMode(repeatMode); *// 重新播放* alphaAni.setRepeatCount(repeatCount); *// 循环播放* view.startAnimation(alphaAni);  
 }  
  
 */\*\*  
 \* 设置位移动画效果，来回移动  
 \*/* **public static void** setTransAni(View view, **int** x0, **int** x1, **int** y0, **int** y1, **long** durationMillis)  
 {  
 TranslateAnimation transAni = **new** TranslateAnimation(x0, x1, y0, y1);  
 transAni.setDuration(durationMillis);  
 transAni.setRepeatMode(Animation.***REVERSE***);  
 transAni.setRepeatCount(Animation.***INFINITE***);  
  
 view.startAnimation(transAni);  
 }  
  
 */\*\*  
 \* 设置旋转动画  
 \*/* **public static void** setRotateAni(View view, **float** fromDegrees, **float** toDegrees, **long** time)  
 {  
 *setRotateAni*(view, fromDegrees, toDegrees, time, Animation.***REVERSE***, Animation.***INFINITE***, **true**);  
 }  
  
 */\*\*  
 \* 设置旋转动画, 从角度fromDegrees旋转到toDegrees，旋转速度durationMillis毫秒，重启模式repeatModes，重启次数repeatCount，是否匀速旋转linear。  
 \* 示例：AnimationEffect.setRotateAni(view, 0, 360, 1000, Animation.RESTART, Animation.INFINITE, true); // 控制view每秒旋转360度  
 \*/* **public static void** setRotateAni(View view, **float** fromDegrees, **float** toDegrees, **long** durationMillis, **int** repeatModes, **int** repeatCount, **boolean** linear)  
 {  
 RotateAnimation rotateAni = **new** RotateAnimation(fromDegrees, toDegrees, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 rotateAni.setDuration(durationMillis);  
 rotateAni.setRepeatMode(repeatModes);  
 rotateAni.setRepeatCount(repeatCount);  
  
 **if**(linear) rotateAni.setInterpolator(**new** LinearInterpolator()); *// 匀速旋转* view.startAnimation(rotateAni);  
 }  
  
 */\*\*  
 \* 为控件添加尺寸渐变动画  
 \*/* **public static void** setScaleAni(View V, **float** fromScale, **float** toScale, **long** ANITIME)  
 {  
 AnimationSet aniSet = **new** AnimationSet(**true**);  
 *// final int ANITIME = 500;  
  
 // 尺寸变化动画，设置尺寸变化* ScaleAnimation scaleAni = **new** ScaleAnimation(fromScale, toScale, fromScale, toScale, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 scaleAni.setDuration(ANITIME); *// 设置动画效果时间* aniSet.addAnimation(scaleAni); *// 将动画效果添加到动画集中* V.startAnimation(aniSet); *// 添加光效动画到控件* }  
  
 **public static void** setLightAni(**final** View view)  
 {  
 Animation.AnimationListener listenser = **new** Animation.AnimationListener()  
 {  
 @Override  
 **public void** onAnimationEnd(Animation animation)  
 {  
 view.clearAnimation();  
 *setRotateAni*(view, 36, 3996, 110000, Animation.***RESTART***, Animation.***INFINITE***, **true**);  
 }  
  
 @Override  
 **public void** onAnimationRepeat(Animation animation)  
 {}  
  
 @Override  
 **public void** onAnimationStart(Animation animation)  
 {}  
 };  
  
 AnimationSet set = **new** AnimationSet(**true**);  
  
 *// 渐现* AlphaAnimation alphaAni = **new** AlphaAnimation(0.0f, 1f);  
 alphaAni.setDuration(1000); *// 设置动画效果时间* set.addAnimation(alphaAni);  
  
 *// 旋转* RotateAnimation rotateAni = **new** RotateAnimation(0, 36, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 rotateAni.setDuration(1000);  
 set.addAnimation(rotateAni);  
  
 *// 尺寸由小变大* ScaleAnimation scaleAni = **new** ScaleAnimation(0.0f, 1.0f, 0.0f, 1.0f, Animation.***RELATIVE\_TO\_SELF***, 0.5f, Animation.***RELATIVE\_TO\_SELF***, 0.5f);  
 scaleAni.setDuration(1000);  
 scaleAni.setAnimationListener(listenser);  
 set.addAnimation(scaleAni);  
  
 view.startAnimation(set);  
 }  
  
}

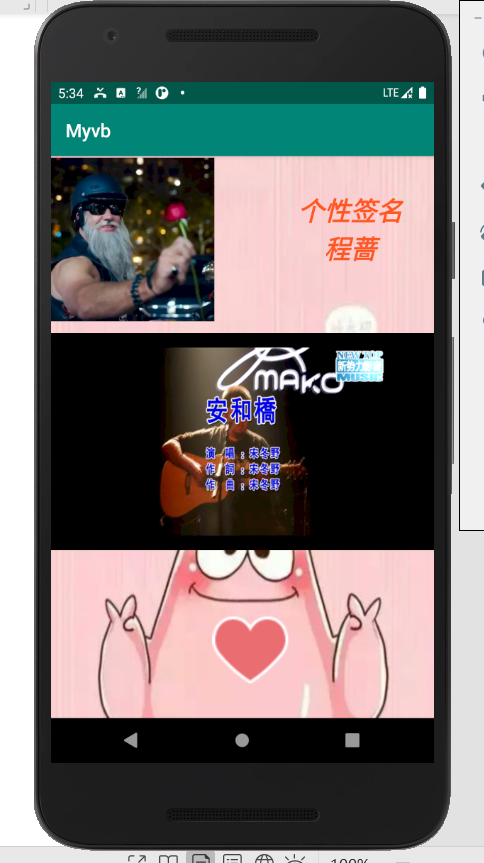
服务器  
sevlet  
package sv;  
  
import net.sf.json.JSONObject;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.PrintWriter;  
import java.net.URLDecoder;  
import java.net.URLEncoder;  
import java.sql.\*;  
  
public class Servlet extends javax.servlet.http.HttpServlet {  
 private static final String *DRIVER* = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 private static final String *URL* = "jdbc:sqlserver://localhost:1433;databasename=";  
 private static final String *DATABASE\_NAME* = "web1";  
 private static final String *USER\_NAME* = "user";  
 private static final String *PASSWORD* = "123456";  
  
 String user,pass,s1;  
 protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {  
  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("application/json;charset=UTF-8");  
 *//获取数据* InputStream inputStream = request.getInputStream();  
 String json = NetUtils.*readString*(inputStream);  
 json = URLDecoder.*decode*(json, "UTF-8");  
 JSONObject j1 = JSONObject.*fromObject*(json);  
 user=j1.optString("user");  
 pass=j1.optString("pass");  
 *// s1=j1.optString("username");* try{  
 Class.*forName*( *DRIVER* );  
 Connection conn = DriverManager.*getConnection*(*URL*+*DATABASE\_NAME*,*USER\_NAME*,*PASSWORD*);  
 *//新建一个查询* String s="select [user],pass from tab\_2 where [user]=? and pass=?";  
 PreparedStatement pst=conn.prepareStatement(s);  
  
 pst.setString(1, user);  
 pst.setString(2, pass);  
 ResultSet rs=pst.executeQuery();  
 if(rs.next()){  
 *//req.setAttribute("username",email);  
 // req.getRequestDispatcher("/index.jsp").forward(req,resp);  
 //resp.sendRedirect("/index.jsp");* JSONObject result = new JSONObject();  
 result.put("message", "返回成功");  
 String serverStr = URLEncoder.*encode*(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
 *// PrintWriter printWriter = response.getWriter();  
 //printWriter.write(serverStr);* }  
 else{  
 *// req.setAttribute("error","用户名或者密码错误，请重新输入");  
 //req.getRequestDispatcher("/").forward(req,resp);* JSONObject result = new JSONObject();  
 result.put("message", "失败");  
 String serverStr = URLEncoder.*encode*(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
 *//PrintWriter printWriter = response.getWriter();  
 //printWriter.write(serverStr);* }  
  
 rs.close();  
 pst.close();  
 conn.close();  
  
 }catch(ClassNotFoundException e){  
 System.*out*.println("驱动问题"+e.getMessage());  
 }catch(SQLException e){  
 System.*out*.println("发生异常:"+e.getMessage());  
 }  
  
  
  
 }  
  
 protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {  
  
 }  
}

Servlet1  
package sv;  
  
import net.sf.json.JSONArray;  
import net.sf.json.JSONObject;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.URLDecoder;  
import java.net.URLEncoder;  
import java.sql.\*;  
  
@WebServlet(name = "Servlet1")  
public class Servlet1 extends HttpServlet {  
 private static final String *DRIVER* = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 private static final String *URL* = "jdbc:sqlserver://localhost:1433;databasename=";  
 private static final String *DATABASE\_NAME* = "web1";  
 private static final String *USER\_NAME* = "user";  
 private static final String *PASSWORD* = "123456";  
 String user,pass,id,time,name;  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("application/json;charset=UTF-8");  
 *//获取数据* InputStream inputStream = request.getInputStream();  
 String json = NetUtils.*readString*(inputStream);  
 json = URLDecoder.*decode*(json, "UTF-8");  
 JSONObject j1 = JSONObject.*fromObject*(json);  
 user = j1.optString("user");  
 pass = j1.optString("pass");  
 id = j1.optString("id");  
 time=j1.optString("time");  
 name=j1.optString("name");  
 if (id .equals("1") ) {  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询* String s;  
  
 s = "insert into tab\_3(zh,mm,column\_3) values(?,?,?)";  
  
  
 PreparedStatement pst = conn.prepareStatement(s);  
  
 pst.setString(1, user);  
 pst.setString(2, pass);  
 pst.setString(3, time);  
 pst.execute();  
 JSONArray arr = new JSONArray();  
 JSONObject result = new JSONObject();  
 result.put("message","增加成功");  
 arr.add(result);  
 response.getWriter().print(arr);  
  
 pst.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
 }  
 if (id .equals("2") ) {  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询* Statement stmt=conn.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,ResultSet.*CONCUR\_READ\_ONLY*);  
 *//查询语句* String s;  
 s = "select zh,mm,column\_3 from tab\_3 ";  
 conn.prepareStatement(s,ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,ResultSet.*CONCUR\_READ\_ONLY*);  
  
 ResultSet rs=stmt.executeQuery(s);  
 JSONArray arr = new JSONArray();  
  
 if(rs.next()) {  
 rs.previous();  
 while (rs.next()) {  
  
 JSONObject result = new JSONObject();  
 result.put("message","成功");  
 result.put("user", rs.getString("zh"));  
 result.put("pass", rs.getString("mm"));  
 result.put("time", rs.getString("column\_3"));  
 arr.add(result);  
  
  
 }  
  
 }  
 else{  
  
 JSONObject result = new JSONObject();  
 result.put("message","失败");  
  
 arr.add(result);  
  
  
 }  
 response.getWriter().print(arr);  
 rs.close();  
 stmt.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
 }  
  
 if (id .equals("3") ) {  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询* String s = "insert into tab\_4(name,pass,com) values(?,?,?)";  
  
  
 PreparedStatement pst = conn.prepareStatement(s);  
  
 pst.setString(1, user);  
 pst.setString(2, pass);  
 pst.setString(3, name);  
  
 pst.execute();  
 JSONArray arr = new JSONArray();  
 JSONObject result = new JSONObject();  
 result.put("message","增加成功");  
 arr.add(result);  
 response.getWriter().print(arr);  
  
 pst.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
 }  
  
  
  
 if (id .equals("4") ) {  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询  
 // Statement stmt=conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 //查询语句* String s;  
 s = "select name,pass from tab\_4 where com= ?";  
 PreparedStatement stmt =conn.prepareStatement(s,ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,ResultSet.*CONCUR\_READ\_ONLY*);  
 stmt.setString(1,name);  
 ResultSet rs=stmt.executeQuery();  
 JSONArray arr = new JSONArray();  
  
 if(rs.next()) {  
 rs.previous();  
 while (rs.next()) {  
  
 JSONObject result = new JSONObject();  
 result.put("message","成功");  
 result.put("user", rs.getString("name"));  
 result.put("pass", rs.getString("pass"));  
  
 arr.add(result);  
  
  
 }  
  
 }  
 else{  
  
 JSONObject result = new JSONObject();  
 result.put("message","失败");  
  
 arr.add(result);  
  
  
 }  
 response.getWriter().print(arr);  
 rs.close();  
 stmt.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
 }  
  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 }  
}

第五章 微博应用运行结果与测试分析  










第六章 滴滴物流业务分析

滴滴物流实现①编写Android客户端应用，具有货主、货车司机注册、登陆以及需求信息发布功能；②对用JSP或PHP开发“滴滴物流”Web服务器；③用MySQL做“滴滴物流”Web服务器的后台数据库；④服务器端货主、货车司机供需匹配功能；⑤客户端供需匹配成功与否显示功能；⑥在线支付功能。

客户端登录选择身份和并输入账号密码，不能不选择身份，或者同时选择多个身份，在登陆时会进行，逻辑判断，货主登陆成功后，从数据库中加载货主的运输单，包括运输中的和已完成的，其次货主可以选择司机寄货，选择完成后需要支付，支付成功则往数据库中加载数据

司机端登陆成功后，加载自己的运输单，和任务量，司机可以选者完成某订单，更新数据库数据

第七章 滴滴物流系统设计

2.1 系统功能定义

①编写Android客户端应用，具有货主、货车司机注册、登陆以及需求信息发布功能；②对用JSP或PHP开发“滴滴物流”Web服务器；③用MySQL做“滴滴物流”Web服务器的后台数据库；④服务器端货主、货车司机供需匹配功能；⑤客户端供需匹配成功与否显示功能；⑥在线支付功能。

2.2 关键问题设计

货主和司机身份，在登陆注册页面设计，并作为标识添加到数据库中

货主和自己的供需匹配，通过数据库中的供需关系表来维持，此表包含，司机，用户，订单，订单时间，订单状态。

货主寄货，选择对应司机，支付功能调用支付宝接口实现，需导入相关的jar，相应代码在支付宝官方文档中可以查到

司机完成订单，目前通过司机手动操作完成订单，并更新数据库，实现供需匹配。

第八章 滴滴物流源代码清单

Mainactivity

**package** com.example.ddwl;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.os.Looper;  
**import** android.text.TextUtils;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** EditText **editText**;  
 **private** EditText **editText2**;  
 **private** Button **button**;  
 **private** Button **button2**;  
 **private** CheckBox **c1**;  
 **private** CheckBox **c2**;  
 **private int num**;  
 **public final static** String ***MyURL*** = **new** ur().**MyURL**;  
 **private** JSONObject **object**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
 **editText** = (EditText) findViewById(R.id.***editText***);  
 **editText2** = (EditText) findViewById(R.id.***editText2***);  
 **button** = (Button) findViewById(R.id.***button***);*//登录* **button2** = (Button) findViewById(R.id.***button2***);*//注册* **c1** = (CheckBox) findViewById(R.id.***c1***);*//货主* **c2** = (CheckBox) findViewById(R.id.***c2***);*//司机* **button2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 login(1);  
 }  
 });  
 **button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 login(2);  
 }  
 });  
  
  
  
 }  
 **public void** login(**int** i){  
  
 **final** String qq=**editText**.getText().toString().trim();  
 **final** String pwd=**editText2**.getText().toString().trim();  
  
  
 **if**(TextUtils.*isEmpty*(qq)){  
 Toast.*makeText*(**this**,**"用户名为空登录失败"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **if**(TextUtils.*isEmpty*(pwd)){  
 Toast.*makeText*(**this**,**"密码为空登陆失败"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **if**(**c1**.isChecked()&&!**c2**.isChecked()){  
  
 **num**=1;*//货主* }  
 **else if**(**c2**.isChecked()&&!**c1**.isChecked()){  
 **num**=2;*//司机* }  
 **else**{  
  
 Toast.*makeText*(**this**,**"请选择您的唯一身份"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**, qq);  
 **object**.put(**"pass"**,pwd);  
 **object**.put(**"id"**,**num**);  
 **object**.put(**"x"**,i);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 *//传输数据* **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 *//把你上面那段网络访问的代码放在这里* **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 **final** JSONObject objectT = **new** JSONObject(jsonStr);  
 *// Log.i("objectT", objectT.getString("server"));* **if**(objectT.optString(**"message"**).equals(**"成功"**)){  
 **if**(**num**==1){  
 Intent it = **new** Intent(MainActivity.**this**, activity2.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, qq);  
 *// bundle.putInt("id", num);* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);}  
 **else**{  
  
 Intent it = **new** Intent(MainActivity.**this**, activity3.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, qq);  
 *// bundle.putInt("id", num);* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);}  
  
  
 }**else if**(objectT.optString(**"message"**).equals(**"al"**)){  
  
 Looper.*prepare*();  
 Toast.*makeText*(MainActivity.**this**,**"用户名已存在"**, Toast.***LENGTH\_SHORT***).show();  
  
 Looper.*loop*();  
  
 }  
  
 **else**{  
 Looper.*prepare*();  
 Toast.*makeText*(MainActivity.**this**,**"身份错误"**, Toast.***LENGTH\_SHORT***).show();  
  
 Looper.*loop*();  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch**(Exception e){  
 e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
 }  
  
  
}

activity2

**package** com.example.ddwl;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.util.TypedValue;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.Button;  
**import** android.widget.ImageButton;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
  
**public class** activity2 **extends** AppCompatActivity {  
 **private** JSONObject **object**;  
 **public final static** String ***MyURL*** = **new** ur().**MyURL1**;  
 **private** Button **button3**;  
 **private** TextView **t4**;  
 **private** LinearLayout **lay1**;  
 **private** String **name**;  
 **private** String **result**;  
 **private** ImageButton **imageButton**;  
  
 Handler **handler**=**new** Handler(){  
 @Override  
 **public void** handleMessage(@NonNull Message msg) {  
 **super**.handleMessage(msg);  
 **switch**(msg.**what**) {  
  
 **case** 1:  
 review();  
 **break**;  
  
 }  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_activity2***);  
  
 **button3** = (Button) findViewById(R.id.***button3***);  
 **t4** = (TextView) findViewById(R.id.***t4***);  
 **lay1** = (LinearLayout) findViewById(R.id.***lay1***);  
  
 **imageButton** = (ImageButton) findViewById(R.id.***imageButton***);  
  
 Bundle bundle=getIntent().getExtras();  
 **name**=bundle.getString(**"user"**);  
 **t4**.setText(**name**);  
 **button3**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 Intent it = **new** Intent(activity2.**this**, activity4.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
 *// bundle.putInt("id", num);* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);  
 }  
 });  
  
 **imageButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 Intent it = **new** Intent(activity2.**this**, activity2.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
 *// bundle.putInt("id", num);* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);  
 }  
 });  
  
  
r1();  
  
  
 }  
  
 **public void** r1(){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**,**name**);  
  
 **object**.put(**"y"**,**"3"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **result** = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 System.***out***.println(**result**);  
 *// review();* in.close();  
 conn.disconnect();  
 **handler**.sendEmptyMessage(1);  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
 }  
  
  
 **public void** review(){  
  
 **try** {  
 *//解码* JSONArray jsonArray = **new** JSONArray(**result**);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
 **if**(name.equals(**"失败"**)){  
  
 *// Looper.prepare();* Toast.*makeText*(activity2.**this**, **"当前没有货物在运输"**, Toast.***LENGTH\_SHORT***).show();  
 *// Looper.loop();* }  
 **else** {  
  
 *// Log.i("objectT", objectT.getString("server"));* LinearLayout linear1= (LinearLayout) findViewById(R.id.***lay1***);  
 linear1.removeAllViews();  
  
 **for** (**int** i = 0; i < jsonArray.length(); i++) {  
 *// JSON数组里面的具体-JSON对象* JSONObject jsonObject1 = jsonArray.getJSONObject(i);  
 String bb = jsonObject1.optString(**"bb"**);  
  
 String com= jsonObject1.optString(**"com"**);  
 String d1=jsonObject1.optString(**"d1"**);  
 *// String time = jsonObject1.optString("time");* TextView tv1= **new** TextView(activity2.**this**);  
 tv1.setBackground(tv1.getContext().getDrawable(R.drawable.***h2***));  
 tv1.getBackground().setAlpha(150);  
 tv1.setTextColor(Color.*rgb*(6,207,160));  
 tv1.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 **if**(d1.equals(**"0"**))  
 { tv1.setText(**"该订单正在运输中"**);}  
 **else**{  
 tv1.setText(**"该订单已完成"**);  
 }  
 TextView tv = **new** TextView(activity2.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setText(com+**"\r\n"**+**"司机："**+bb);  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setId(View.*generateViewId*());  
  
 *// send1(name,text1);  
 //tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使用* tv.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 Toast.*makeText*(activity2.**this**, **"该货物正在运输中"**, Toast.***LENGTH\_SHORT***).show();  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
 tv1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 Toast.*makeText*(activity2.**this**, **"该货物正在运输中"**, Toast.***LENGTH\_SHORT***).show();  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
 linear1.addView(tv, 0);  
 linear1.addView(tv1, 0);  
 }  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
  
  
  
  
  
}

activity3

**package** com.example.ddwl;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Looper;  
**import** android.os.Message;  
**import** android.util.TypedValue;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageButton;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
  
**public class** activity3 **extends** AppCompatActivity {  
 **public final static** String ***MyURL*** = **new** ur().**MyURL1**;  
 **private** TextView **textView5**;  
 String **name**;  
 **private** JSONObject **object**;  
 **private** ImageButton **imageButton3**;  
 **private** String **result**;  
 **private** String **a5**;  
  
 Handler **handler**=**new** Handler(){  
 @Override  
 **public void** handleMessage(@NonNull Message msg) {  
 **super**.handleMessage(msg);  
 **switch**(msg.**what**) {  
  
 **case** 1:  
 review();  
 **break**;  
  
 }  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_activity3***);  
  
 **textView5** = (TextView) findViewById(R.id.***textView5***);  
  
 **imageButton3** = (ImageButton) findViewById(R.id.***imageButton3***);  
 **imageButton3**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 Intent it = **new** Intent(activity3.**this**, activity3.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
 *// bundle.putInt("id", num);/* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);  
 }  
 });  
  
 Bundle bundle=getIntent().getExtras();  
 **name**=bundle.getString(**"user"**);  
 **textView5**.setText(**name**);  
r1();  
 }  
  
 **public void** r1(){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"bb"**,**name**);  
  
 **object**.put(**"y"**,**"4"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **result** = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 System.***out***.println(**result**);  
 *// review();* in.close();  
 conn.disconnect();  
 **handler**.sendEmptyMessage(1);  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
 }  
  
  
 **public void** review(){  
  
 **try** {  
 *//解码* JSONArray jsonArray = **new** JSONArray(**result**);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
 **if**(name.equals(**"失败"**)){  
  
  
 Toast.*makeText*(activity3.**this**, **"当前没有用户请求"**, Toast.***LENGTH\_SHORT***).show();  
  
 }  
 **else** {  
  
 *// Log.i("objectT", objectT.getString("server"));* LinearLayout linear1= (LinearLayout) findViewById(R.id.***lay3***);  
 linear1.removeAllViews();  
  
 **for** (**int** i = 0; i < jsonArray.length(); i++) {  
 *// JSON数组里面的具体-JSON对象* JSONObject jsonObject1 = jsonArray.getJSONObject(i);  
 String user= jsonObject1.optString(**"user"**);  
  
 String com= jsonObject1.optString(**"com"**);  
 *// String time = jsonObject1.optString("time");* TextView tv = **new** TextView(activity3.**this**);  
 TextView tv1 = **new** TextView(activity3.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv1.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv1.getBackground().setAlpha(100);  
 tv1.setTextColor(Color.*rgb*(6,207,160));  
 tv1.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setText(com);  
 tv1.setText(**"客户："**+user+**" 运输中"**);  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setId(View.*generateViewId*());  
  
 *// send1(name,text1);  
 //tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使用* tv.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 *// Toast.makeText(activity2.this, "该货物正在运输中", Toast.LENGTH\_SHORT).show();  
 // Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* **int** a2 = view.getId();  
 TextView a1 = (TextView) findViewById(a2);  
 **a5** = a1.getText().toString();  
  
 go2(**a5**);  
 }  
 });  
 tv1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 Toast.*makeText*(activity3.**this**, **"该货物正在运输中"**, Toast.***LENGTH\_SHORT***).show();  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
 linear1.addView(tv, 0);  
 linear1.addView(tv1, 0);  
  
 }  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
  
  
  
  
 **public void** go2(String a5){  
 *// final EditText inputServer = new EditText(activity3.this);* AlertDialog.Builder builder = **new** AlertDialog.Builder(activity3.**this**);  
 builder.setTitle(**"点击确认完成该订单？"**).setIcon(android.R.drawable.***ic\_dialog\_info***)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
  
 builder.setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
  
  
 send();  
  
  
  
 }}  
 );  
 builder.show();  
  
  
  
  
 }  
  
 **public void** send(){  
 *// String bb=a3.substring(0, a3.indexOf("司机"));* **try** {  
  
 **object** = **new** JSONObject();  
  
 **object**.put(**"com"**,**a5**);  
 **object**.put(**"user"**,**name**);  
  
 **object**.put(**"y"**,**"5"**);  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 JSONArray jsonArray = **new** JSONArray(jsonStr);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
  
  
 **if**(name.equals(**"成功"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(activity3.**this**,**"成功完成"**, Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
  
  
  
  
 }  
  
  
  
}

activity4

**package** com.example.ddwl;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Looper;  
**import** android.os.Message;  
**import** android.text.TextUtils;  
**import** android.util.TypedValue;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.EditText;  
**import** android.widget.ImageButton;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.alipay.sdk.app.PayResultActivity;  
**import** com.alipay.sdk.app.PayTask;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.MalformedURLException;  
**import** java.net.ProtocolException;  
**import** java.net.SocketTimeoutException;  
**import** java.net.URL;  
**import** java.net.URLDecoder;  
**import** java.net.URLEncoder;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
**import** java.util.Map;  
  
**public class** activity4 **extends** AppCompatActivity {  
  
 **public static final** String ***PARTNER*** = **""**;  
 *// 商户收款账号* **public static final** String ***SELLER*** = **""**;  
 *// 商户私钥，pkcs8格式* **public static final** String ***RSA\_PRIVATE*** = **""**;  
 **private static final int *SDK\_PAY\_FLAG*** = 1;  
 **public final static** String ***MyURL*** = **new** ur().**MyURL1**;  
 **private** JSONObject **object**;  
 **private** String **name**;  
 **private** LinearLayout **lay2**;  
 **private** String **result**;  
 **private** String **a3**;  
 **private** ImageButton **imageButton2**;  
 **final** String **orderInfo** = getOrderInfo(**"测试商品"**, **"商品描述"**, **"0.1"**);  
 Handler **handler**=**new** Handler(){  
 @Override  
 **public void** handleMessage(@NonNull Message msg) {  
 **super**.handleMessage(msg);  
 **switch**(msg.**what**) {  
  
 **case** 1:  
 review();  
 **break**;  
  
 }  
 }  
 };  
  
 **private** Handler **mHandler** = **new** Handler() {  
 **public void** handleMessage(Message msg) {  
 **switch** (msg.**what**) {  
 **case *SDK\_PAY\_FLAG***: {  
 PayResult payResult = **new** PayResult((Map<String, String>) msg.**obj**);  
 */\*\*  
 \* 同步返回的结果必须放置到服务端进行验证（验证的规则请看https://doc.open.alipay.com/doc2/  
 \* detail.htm?spm=0.0.0.0.xdvAU6&treeId=59&articleId=103665&  
 \* docType=1) 建议商户依赖异步通知  
 \*/* String resultInfo = payResult.getResult();*// 同步返回需要验证的信息* String resultStatus = payResult.getResultStatus();  
 *// 判断resultStatus 为“9000”则代表支付成功，具体状态码代表含义可参考接口文档* **if** (TextUtils.*equals*(resultStatus, **"9000"**)) {  
 Toast.*makeText*(activity4.**this**, **"支付成功"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 *// 判断resultStatus 为非"9000"则代表可能支付失败  
 // "8000"代表支付结果因为支付渠道原因或者系统原因还在等待支付结果确认，最终交易是否成功以服务端异步通知为准（小概率状态）* **if** (TextUtils.*equals*(resultStatus, **"8000"**)) {  
 Toast.*makeText*(activity4.**this**, **"支付结果确认中"**, Toast.***LENGTH\_SHORT***).show();  
  
  
 } **else** {  
 *// 其他值就可以判断为支付失败，包括用户主动取消支付，或者系统返回的错误* Toast.*makeText*(activity4.**this**, **"支付失败"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 **break**;  
 }  
 **default**:  
 **break**;  
 }  
 };  
 };  
  
 **private** String getOrderInfo(String subject, String body, String price) {  
  
  
 *// 签约合作者身份ID* String orderInfo = **"partner="** + **"\""** + ***PARTNER*** + **"\""**;  
  
  
 *// 签约卖家支付宝账号* orderInfo += **"&seller\_id="** + **"\""** + ***SELLER*** + **"\""**;  
  
  
 *// 商户网站唯一订单号* orderInfo += **"&out\_trade\_no="** + **"\""** + **"\""**;  
  
  
 *// 商品名称* orderInfo += **"&subject="** + **"\""** + subject + **"\""**;  
  
  
 *// 商品详情* orderInfo += **"&body="** + **"\""** + body + **"\""**;  
  
  
 *// 商品金额* orderInfo += **"&total\_fee="** + **"\""** + price + **"\""**;  
  
  
 *// 服务器异步通知页面路径* orderInfo += **"&notify\_url="** + **"\""** + **"http://notify.msp.hk/notify.htm"** + **"\""**;  
  
  
 *// 服务接口名称， 固定值* orderInfo += **"&service=\"mobile.securitypay.pay\""**;  
  
  
 *// 支付类型， 固定值* orderInfo += **"&payment\_type=\"1\""**;  
  
  
 *// 参数编码， 固定值* orderInfo += **"&\_input\_charset=\"utf-8\""**;  
  
  
 *// 设置未付款交易的超时时间  
 // 默认30分钟，一旦超时，该笔交易就会自动被关闭。  
 // 取值范围：1m～15d。  
 // m-分钟，h-小时，d-天，1c-当天（无论交易何时创建，都在0点关闭）。  
 // 该参数数值不接受小数点，如1.5h，可转换为90m。* orderInfo += **"&it\_b\_pay=\"30m\""**;  
  
  
 *// extern\_token为经过快登授权获取到的alipay\_open\_id,带上此参数用户将使用授权的账户进行支付  
 // orderInfo += "&extern\_token=" + "\"" + extern\_token + "\"";  
  
  
 // 支付宝处理完请求后，当前页面跳转到商户指定页面的路径，可空* orderInfo += **"&return\_url=\"m.alipay.com\""**;  
  
  
 *// 调用银行卡支付，需配置此参数，参与签名， 固定值 （需要签约《无线银行卡快捷支付》才能使用）  
 // orderInfo += "&paymethod=\"expressGateway\"";* **return** orderInfo;  
 }  
 **public void** pay(){  
  
  
 Runnable payRunnable = **new** Runnable() {  
  
 @Override  
 **public void** run() {  
 *// PayResult payResult = new PayResult((String) msg.obj);* PayTask alipay = **new** PayTask(activity4.**this**);  
 Map<String, String> result = alipay.payV2(**orderInfo**, **true**);  
  
 Message msg = **new** Message();  
 msg.**what** = ***SDK\_PAY\_FLAG***;  
 msg.**obj** = result;  
 **mHandler**.sendMessage(msg);  
 }  
 };  
 *// 必须异步调用* Thread payThread = **new** Thread(payRunnable);  
 payThread.start();  
  
  
  
 }  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_activity4***);  
 r1();  
  
  
 **imageButton2** = (ImageButton) findViewById(R.id.***imageButton2***);  
 **imageButton2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
  
 Intent it = **new** Intent(activity4.**this**, activity4.**class**);  
  
 Bundle bundle=**new** Bundle();  
  
 bundle.putString(**"user"**, **name**);  
 *// bundle.putInt("id", num);* it.putExtras(bundle); *// it.putExtra(“test”, "shuju”);* startActivity(it);  
 }  
 });  
  
 Bundle bundle=getIntent().getExtras();  
 **name**=bundle.getString(**"user"**);  
 **lay2** = (LinearLayout) findViewById(R.id.***lay2***);}  
  
  
 **public void** r1(){  
  
 **try** {  
  
 **object** = **new** JSONObject();  
  
 **object**.put(**"y"**,**"1"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **result** = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 System.***out***.println(**result**);  
 *// review();* in.close();  
 conn.disconnect();  
 **handler**.sendEmptyMessage(1);  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
 }  
  
  
 **public void** review(){  
  
 **try** {  
 *//解码* JSONArray jsonArray = **new** JSONArray(**result**);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
 **if**(name.equals(**"失败"**)){  
  
 *// Looper.prepare();* Toast.*makeText*(activity4.**this**, **"当前没有司机"**, Toast.***LENGTH\_SHORT***).show();  
 *// Looper.loop();* }  
 **else** {  
  
 *// Log.i("objectT", objectT.getString("server"));* LinearLayout linear1= (LinearLayout) findViewById(R.id.***lay2***);  
 linear1.removeAllViews();  
  
 **for** (**int** i = 0; i < jsonArray.length(); i++) {  
 *// JSON数组里面的具体-JSON对象* JSONObject jsonObject1 = jsonArray.getJSONObject(i);  
 String user = jsonObject1.optString(**"user"**);  
  
 String rw = jsonObject1.optString(**"rw"**);  
 *// String time = jsonObject1.optString("time");* TextView tv = **new** TextView(activity4.**this**);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.setMargins(0,0,0,((**int**) TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_DIP***, 10, getResources().getDisplayMetrics())));  
 tv.setLayoutParams(layoutParams);  
 tv.setBackground(tv.getContext().getDrawable(R.drawable.***h2***));  
 tv.getBackground().setAlpha(100);  
 tv.setTextColor(Color.*rgb*(6,207,160));  
 tv.setTextSize(TypedValue.*applyDimension*(TypedValue.***COMPLEX\_UNIT\_SP***, 8, getResources().getDisplayMetrics()));  
 tv.setText(user+**"司机"**+**"\r\n"**+**"当前运输量："**+rw);  
 tv.setTypeface(Typeface.*defaultFromStyle*(Typeface.***BOLD***));  
  
 tv.setId(View.*generateViewId*());  
  
 *// send1(name,text1);  
 //tv.setId(1);//设置ID，可有可无，也可以在R文件中添加字符串，然后在这里使用引用的方式使用* tv.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View view) {  
 **int** a2 = view.getId();  
 TextView a1 = (TextView) findViewById(a2);  
 **a3** = a1.getText().toString();  
 go1();  
 *// Toast.makeText(activity2.this, "连接超时", Toast.LENGTH\_SHORT).show();* }  
 });  
  
 linear1.addView(tv, 0);  
 }  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
  
 }  
 **public void** go1(){  
  
 **final** EditText inputServer = **new** EditText(activity4.**this**);  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(activity4.**this**);  
 builder.setTitle(**"请输入货物及联系电话"**).setIcon(android.R.drawable.***ic\_dialog\_info***).setView(inputServer)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.dismiss();  
 }  
 });  
  
 builder.setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String text1 = inputServer.getText().toString();  
 Date now = **new** Date();  
 SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"yyyy/MM/dd HH:mm:ss"**);*//可以方便地修改日期格式* String hehe = dateFormat.format( now );  
 String com=text1+**"\r\n"**+hehe;  
  
 **if**(!text1.equals(**""**)){  
  
 pay();  
 send(**name**,com);  
  
 }  
  
 }}  
 );  
 builder.show();  
  
  
 }  
  
  
  
 **public void** send(String user,String com){  
 String bb=**a3**.substring(0, **a3**.indexOf(**"司机"**));  
  
 **try** {  
  
 **object** = **new** JSONObject();  
 **object**.put(**"user"**, user);  
 **object**.put(**"com"**,com);  
 **object**.put(**"bb"**,bb);  
  
 **object**.put(**"y"**,**"2"**);  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **new** Thread(){  
 @Override  
 **public void** run()  
 {  
 **try** {  
 URL url = **new** URL(***MyURL***);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 *//设置超时时间。* conn.setConnectTimeout(30000);  
 conn.setRequestMethod(**"POST"**);*// 大写* conn.setRequestProperty(**"User-Agent"**, **"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)"**);  
 conn.setRequestProperty(**"Content-Type"**, **"application/json"**);*//请求的类型 表单数据//区别4： 记得设置把数据写给服务器。* conn.setDoOutput(**true**);*//设置向服务器写数据。* conn.setDoInput(**true**);  
 conn.setUseCaches(**false**);  
 OutputStream outputStream = conn.getOutputStream();  
 outputStream.write(URLEncoder.*encode*(**object**.toString(), **"UTF-8"**).getBytes());  
 **int** code = conn.getResponseCode();  
 *// 如果请求响应码是200，则表示成功* **if** (code == HttpURLConnection.***HTTP\_OK***) {  
  
 *//获取服务器上的数据* BufferedReader in = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream(), **"utf-8"**));  
 **try** {  
 *//解码* String jsonStr = URLDecoder.*decode*(in.readLine(), **"UTF-8"**);  
 JSONArray jsonArray = **new** JSONArray(jsonStr);  
 JSONObject jsonObject = jsonArray.getJSONObject(0);  
 String name = jsonObject.optString(**"message"**);  
  
  
  
 **if**(name.equals(**"增加成功"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(activity4.**this**,**"订单增加成功"**, Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 in.close();  
 conn.disconnect();  
 }  
 } **catch** (MalformedURLException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 } **catch** (ProtocolException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }**catch** (SocketTimeoutException e) {  
  
  
  
 }**catch** (IOException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
  
  
 }  
 }.start();  
  
  
  
  
  
  
  
  
 }  
  
  
}

PayResult

**package** com.example.ddwl;  
  
**import** android.text.TextUtils;  
  
**import** java.util.Map;  
  
**public class** PayResult {  
  
 **private** String **resultStatus**;  
 **private** String **result**;  
 **private** String **memo**;  
  
 **public** PayResult(Map<String, String> rawResult) {  
 **if** (rawResult == **null**) {  
 **return**;  
 }  
  
 **for** (String key : rawResult.keySet()) {  
 **if** (TextUtils.*equals*(key, **"resultStatus"**)) {  
 **resultStatus** = rawResult.get(key);  
 } **else if** (TextUtils.*equals*(key, **"result"**)) {  
 **result** = rawResult.get(key);  
 } **else if** (TextUtils.*equals*(key, **"memo"**)) {  
 **memo** = rawResult.get(key);  
 }  
 }  
 }  
  
 @Override  
 **public** String toString() {  
 **return "resultStatus={"** + **resultStatus** + **"};memo={"** + **memo** + **"};result={"** + **result** + **"}"**;  
 }  
  
 */\*\*  
 \** ***@return*** *the resultStatus  
 \*/* **public** String getResultStatus() {  
 **return resultStatus**;  
 }  
  
 */\*\*  
 \** ***@return*** *the memo  
 \*/* **public** String getMemo() {  
 **return memo**;  
 }  
  
 */\*\*  
 \** ***@return*** *the result  
 \*/* **public** String getResult() {  
 **return result**;  
 }  
}

url

**package** com.example.ddwl;  
  
**public class** ur {  
 **public** String **MyURL** = **"http://10.0.0.91:8080/ddwl/s1"**;  
 **public** String **MyURL1** = **"http://10.0.0.91:8080/ddwl/s2"**;  
}

Servlet

package sample;  
  
import net.sf.json.JSONObject;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.URLDecoder;  
import java.net.URLEncoder;  
import java.sql.\*;  
  
public class s1 extends javax.servlet.http.HttpServlet {  
 private static final String DRIVER = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 private static final String URL = "jdbc:sqlserver://localhost:1433;databasename=";  
 private static final String DATABASE\_NAME = "web1";  
 private static final String USER\_NAME = "user";  
 private static final String PASSWORD = "123456";  
 private String user,pass,id,x;  
 protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {  
  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("application/json;charset=UTF-8");  
 *//获取数据* InputStream inputStream = request.getInputStream();  
 String json = NetUtils.readString(inputStream);  
 json = URLDecoder.decode(json, "UTF-8");  
 JSONObject j1 = JSONObject.fromObject(json);  
 user = j1.optString("user");  
 pass = j1.optString("pass");  
 id = j1.optString("id");  
 x=j1.optString("x");  
 if(x.equals("1")){  
 try{  
 Class.forName( DRIVER );  
 Connection conn = DriverManager.getConnection(URL+DATABASE\_NAME,USER\_NAME,PASSWORD);  
 *//新建一个查询* String s1="select [user] from tab\_5 where [user]=? ";  
 PreparedStatement pst1=conn.prepareStatement(s1);  
  
 pst1.setString(1, user);  
  
 ResultSet rs=pst1.executeQuery();  
 if(rs.next()){  
  
 JSONObject result = new JSONObject();  
 result.put("message", "al");  
 String serverStr = URLEncoder.encode(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
  
 }  
 else{  
 String s="insert into tab\_5([user],pass,id) values (?,?,?)";  
 PreparedStatement pst=conn.prepareStatement(s);  
  
 pst.setString(1, user);  
 pst.setString(2, pass);  
 pst.setString(3, id);  
 pst.execute();  
  
  
 JSONObject result = new JSONObject();  
 result.put("message", "成功");  
 String serverStr = URLEncoder.*encode*(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
 pst.close();  
 pst1.close();  
 conn.close();}  
  
 }catch(ClassNotFoundException e){  
 System.out.println("驱动问题"+e.getMessage());  
 }catch(SQLException e){  
 System.out.println("发生异常:"+e.getMessage());  
 }  
  
  
  
  
 }  
 if(x.equals("2")){  
  
  
 try{  
 Class.forName( DRIVER );  
 Connection conn = DriverManager.getConnection(URL+DATABASE\_NAME,USER\_NAME,PASSWORD);  
 *//新建一个查询* String s="select [user],pass,id from tab\_5 where [user]=? and pass=? and id=?";  
 PreparedStatement pst=conn.prepareStatement(s);  
  
 pst.setString(1, user);  
 pst.setString(2, pass);  
 pst.setString(3, id);  
 ResultSet rs=pst.executeQuery();  
 if(rs.next()){  
  
 JSONObject result = new JSONObject();  
 result.put("message", "成功");  
 String serverStr = URLEncoder.encode(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
  
 }  
 else{  
  
 JSONObject result = new JSONObject();  
 result.put("message", "失败");  
 String serverStr = URLEncoder.encode(result.toString(), "UTF-8");  
 response.getWriter().print(result);  
  
 }  
  
 rs.close();  
 pst.close();  
 conn.close();  
 }catch(ClassNotFoundException e){  
 System.out.println("驱动问题"+e.getMessage());  
 }catch(SQLException e){  
 System.out.println("发生异常:"+e.getMessage());  
 }  
  
  
  
  
  
 }  
  
  
  
  
  
  
  
  
 }  
  
 protected void doGet(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException, IOException {  
  
 }  
}

servlet2

package sample;  
  
import net.sf.json.JSONArray;  
import net.sf.json.JSONObject;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.URLDecoder;  
import java.sql.\*;  
  
@WebServlet(name = "s2")  
public class s2 extends HttpServlet {  
 private static final String *DRIVER* = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  
 private static final String *URL* = "jdbc:sqlserver://localhost:1433;databasename=";  
 private static final String *DATABASE\_NAME* = "web1";  
 private static final String *USER\_NAME* = "user";  
 private static final String *PASSWORD* = "123456";  
 private String user,pass,id,x,y,com,bb;  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
  
 request.setCharacterEncoding("utf-8");  
 response.setContentType("application/json;charset=UTF-8");  
 *//获取数据* InputStream inputStream = request.getInputStream();  
 String json = NetUtils.*readString*(inputStream);  
 json = URLDecoder.*decode*(json, "UTF-8");  
 JSONObject j1 = JSONObject.*fromObject*(json);  
 user = j1.optString("user");  
 pass = j1.optString("pass");  
 id = j1.optString("id");  
 x=j1.optString("x");  
 y=j1.optString("y");  
 com=j1.optString("com");  
 bb=j1.optString("bb");  
  
 if(y.equals("1")){  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询* Statement stmt=conn.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,ResultSet.*CONCUR\_READ\_ONLY*);  
 *//查询语句* String s;  
 s = "select [user],rw from tab\_5 where id=2 ";  
 conn.prepareStatement(s,ResultSet.*TYPE\_SCROLL\_INSENSITIVE*,ResultSet.*CONCUR\_READ\_ONLY*);  
  
 ResultSet rs=stmt.executeQuery(s);  
 JSONArray arr = new JSONArray();  
  
 if(rs.next()) {  
 rs.previous();  
 while (rs.next()) {  
  
 JSONObject result = new JSONObject();  
 result.put("message","成功");  
 result.put("user", rs.getString("user"));  
 result.put("rw", rs.getString("rw"));  
 *// result.put("time", rs.getString("column\_3"));* arr.add(result);  
  
  
 }  
  
 }  
 else{  
  
 JSONObject result = new JSONObject();  
 result.put("message","失败");  
  
 arr.add(result);  
  
  
 }  
 response.getWriter().print(arr);  
 rs.close();  
 stmt.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
  
  
  
  
 }  
  
 if(y.equals("2")){  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询* String s = "insert into tab\_6(a1,b1,c1) values(?,?,?)";  
 String s1 ="update tab\_5 set rw+=1 where [user]=? or [user]=?";  
 PreparedStatement pst1=conn.prepareStatement(s1);  
 pst1.setString(1, bb);  
 pst1.setString(2, user);  
  
 pst1.execute();  
  
  
 PreparedStatement pst = conn.prepareStatement(s);  
  
 pst.setString(1, bb);  
 pst.setString(2, user);  
 pst.setString(3, com);  
  
 pst.execute();  
 JSONArray arr = new JSONArray();  
 JSONObject result = new JSONObject();  
 result.put("message","增加成功");  
 arr.add(result);  
 response.getWriter().print(arr);  
  
 pst.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
  
  
  
 }  
  
  
 if(y.equals("3")){  
  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询  
  
  
  
  
 // Statement stmt=conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 //查询语句* String s,s1;  
 s = "select a1,c1,d1 from tab\_6 where b1=? ";  
  
 PreparedStatement pst1=conn.prepareStatement(s);  
  
 pst1.setString(1, user);  
  
 *// conn.prepareStatement(s,ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 // conn.prepareStatement(s1,ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);* ResultSet rs=pst1.executeQuery();  
 JSONArray arr = new JSONArray();  
  
  
 JSONObject result = new JSONObject();  
 result.put("message","失败");  
  
 *// arr.add(result);* while (rs.next()) {  
  
 *// JSONObject result = new JSONObject();* result.put("message","成功");  
 result.put("bb", rs.getString("a1"));  
 result.put("com", rs.getString("c1"));  
 result.put("d1", rs.getString("d1"));  
 *// result.put("time", rs.getString("column\_3"));* arr.add(result);  
  
  
 }  
  
  
  
  
  
 response.getWriter().print(arr);  
 rs.close();  
 pst1.close();  
  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
  
  
  
 }  
 if(y.equals("4")){  
  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询  
  
  
  
  
 //Statement stmt=conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 //查询语句* String s,s1;  
 s = "select b1,c1 from tab\_6 where a1=? and d1=? ";  
  
 PreparedStatement pst=conn.prepareStatement(s);  
  
 *//s1 ="update tab\_5 set rw+=1 where user=? or ?";  
 // PreparedStatement pst=conn.prepareStatement(s1);* pst.setString(1,bb);  
 pst.setString(2,"0");  
  
 ResultSet rs=pst.executeQuery();  
 JSONArray arr = new JSONArray();  
  
  
 JSONObject result = new JSONObject();  
 result.put("message","失败");  
  
 *// arr.add(result);* while (rs.next()) {  
  
  
 result.put("message","成功");  
 result.put("user", rs.getString("b1"));  
 result.put("com", rs.getString("c1"));  
 *// result.put("time", rs.getString("column\_3"));* arr.add(result);  
  
  
 }  
  
  
  
  
  
 response.getWriter().print(arr);  
 rs.close();  
 pst.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
  
  
  
 }  
 if(y.equals("5")){  
  
  
 try {  
 Class.*forName*(*DRIVER*);  
 Connection conn = DriverManager.*getConnection*(*URL* + *DATABASE\_NAME*, *USER\_NAME*, *PASSWORD*);  
 *//新建一个查询  
  
  
  
  
 //Statement stmt=conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 //查询语句* String s,s1;  
 s = "update tab\_6 set d1=?";  
 s1="update tab\_5 set rw-=1 where [user]=?";  
 PreparedStatement pst=conn.prepareStatement(s);  
 PreparedStatement pst1=conn.prepareStatement(s1);  
 pst1.setString(1,user);  
 *//s1 ="update tab\_5 set rw+=1 where user=? or ?";  
 // PreparedStatement pst=conn.prepareStatement(s1);* pst.setString(1,"1");  
  
 *// pst.setString(1, bb);  
 // pst.setString(2, user);  
  
 // pst.execute();  
 // conn.prepareStatement(s,ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
 // conn.prepareStatement(s1,ResultSet.TYPE\_SCROLL\_INSENSITIVE,ResultSet.CONCUR\_READ\_ONLY);*pst.execute();pst1.execute();  
 *// ResultSet rs=pst.executeQuery(s);* JSONArray arr = new JSONArray();  
 JSONObject result = new JSONObject();  
 result.put("message","成功");  
 arr.add(result);  
 response.getWriter().print(arr);  
  
 pst.close();  
 conn.close();  
  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("驱动问题" + e.getMessage());  
 } catch (SQLException e) {  
 System.*out*.println("发生异常:" + e.getMessage());  
 }  
  
  
  
 }  
  
  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 }  
}

第九章 滴滴物流运行结果与测试分析



第十章 结论与心得  
通过本次上机，综合运用所学过的《计算机程序设计基础》、《数据结构》、《算法》、《Java语言与系统设计》、《数据库》、《Web技术》、《移动应用开发》等课程的基础知识，熟练掌握了开发市面上比较流行的移动应用开发的基本技能，设计和开发出具有一定规模的Android客户端应用+JSP（PHP）服务器端编程+MySQL数据库的典型移动互联网应用。

在知识上，了解HTML、HTML5、CSS、JavaScript、JSP、Android等开发技术在实际移动互联网应用开发过程中的基本用法；了解移动互联网应用开发从需求分析、系统设计、模块设计、开发以及调试的各个环节。掌握典型移动互联网应用各个环节的技术以及具体运用方式；了解移动互联网典型应用开发是如何结合Web相关技术（HTML、HTML5、CSS、JavaScript、JSP等）、移动客户端开发技术（Android为例）以及数据库技术（以MySQL为例）进行开发的。

在能力上：通过从无到有动手完成至少两个基于Web+Android+MySQL移动应用开发的案例的设计、编码和调试工作，将对移动互联网应用开发的理性认识转变为感性认识，初步体会和理解计算机相关学科的软件项目工程的具体概念，体会工程的复杂性；学会将理论知识与实际生产结合，学会用工程的角度去提出问题、分析问题和解决问题；初步建立移动互联网应用软件工程开发项目需求分析、总体设计、模块设计、编码实现和调试的完整概念，提高实际动手能力，为具备大型移动互联网应用开发软件系统从设计到开发的能力打下坚实基础。通过实际动手设计和开发多个完整的移动互联网应用实例，培养理论知识的综合运用能力和动手编程开发的能力，培养软件工程开发的流程化管理观念；在实习中理解并遵守软件工程开发的职业道德和规范，履行责任，提高职了业规范素质

本人代码能力和debug能力得到了巨大的提升，同时对以往学过的课程有了新的认知和理解，对androdi,web,数据库开发，有了一定的基础，同时也学到了很多新的东西，开阔了视野，以后还会再接再厉，继续努力学习，做一个优秀的程序员，成为IT精英,第一个微博实验是入门实验，从百度到b站，本人查阅了大量的学习资料，学习了很多的东西，从0开始学android studio以及android开发，服务器搭建有了WEB的基础，要好做一些。第二个实验相对来说就比较好些了，有了基本的逻辑，框架就很好搭了，第二个实验的业务逻辑要比第一个复杂，数据库设计，逻辑处理，供需匹配经过一定的思考和组合，还要引入支付宝支付，需要支付宝审核，以及代码的学习，也遇到了很多问题，总之来说，IT，我还在路上，我还会继续努力，提高个人能力，提高代码能力。