

LineOnline 软件设计文档

项目成员：彭翌 聂照昌

许伟原 吕朔君

魏毓航 潘桂波

日期： 2015/7/18



1. 概述

LineOnline 是一款智能的游乐园排队取票 APP，旨在将银行取号排队的方式放到游乐园上，使游客在排队的时间可以利用排队的时间去游玩更多的项目。

LineOnline 按游乐设施的接待能力和娱乐时间划分时段，每个时段放出一定空位接受预约，只需在预约时间到达指定地点即可快速通行，最小化游客的排队时间。

析取出软件的用例图如下：

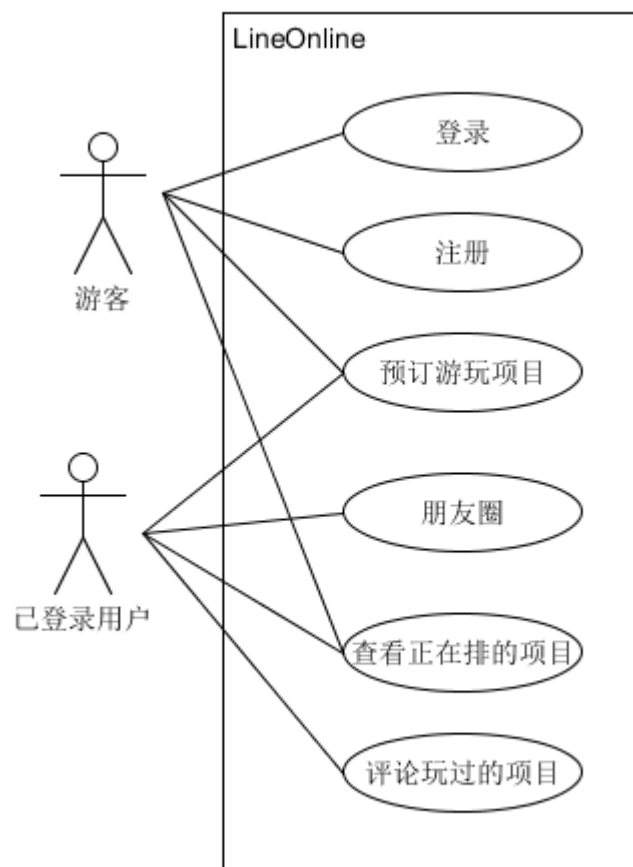


图 1 用例图

2. 技术选型

2.1 前端技术选型

考虑到我们的项目的功能需求和开发平台，前端选择面向对象的编程方法来实现。面向对象技术是目前流行的系统设计开发技术，它包括面向对象分析和面向对象程序设计，它侧重于对象的内部结构及其相互间的联系。

2.2 后台设计模式：

我们采用的设计模式是 Template Method 的设计模式。

其实严格来说，并非我们选择了模块方法的设计模式，而是 Thinkphp 框架选择了这样的设计模式。

以 IndexAction 为例：

```
<?php

class IndexAction extends Action{
    public function index(){

    }

    public function show() { //查询字符串
        $playgrounds = M('playground')->select();

        echo json_encode($playgrounds);
    }

    public function search() {
        $search_string = $_POST['searchName'];
        //echo $search_string;
        $games = M('playground')->select();
        $len = count($games);
        $result=array();
        for($i = 0; $i < $len; $i++) {
            //echo $games[$i]['name'];
            if(substr_count($games[$i]['name'],$search_string) > 0) {

                array_push($result,$games[$i]);
            }
        }
        echo json_encode($result);
    }
}
```

所有的模块 Action 类都集成 Action 这个类。这个类里面有其子类所共同需要的方法，如 CRUD 方法。

在我们本次的开发中 模块与模块之间几乎不存在耦合。所以 我们根本不需要过多的交流，只需要把自己模块的特有的方法（详细见上部分的 API 设计）实现即可。大大加快了开发效率。

3. 项目架构

从技术层面上来看，整个 LineOnline 项目以 C/S 模式构建。

服务器选用 PHP 作为开发语言，依托于强大的 ThinkPHP 开发框架进行敏捷开发，并将服务器部署在了阿里云提供的 ECS 云服务器上，借助阿里云的服务，确保全国各地都能有可观的访问速度。

客户端则选择从移动端出发，首先开发了市场占有率最高的安卓端。安卓端以 HTTP 协议访问服务器、发送请求并接收数据。客户端与服务器之间的数据将以 JSON 格式传输。

整个项目架构大概如下图所示：

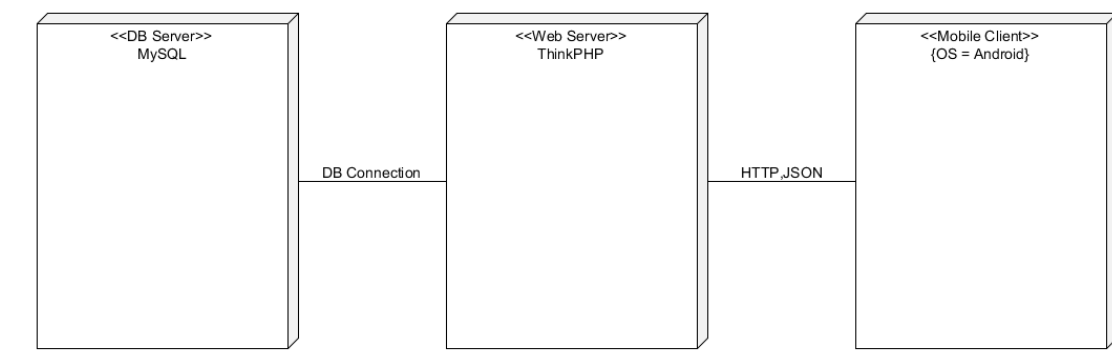


图 2 部署图

4. 模块划分

根据软件的功能需求，设计得到领域模型：

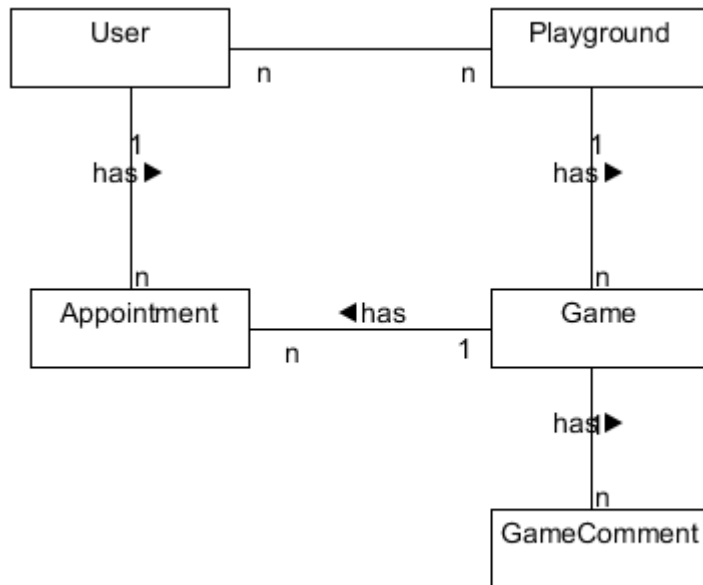


图 3 领域模型

4.1 前端

根据设计需求，可以设计出每个 Activity 类的职务以及跳转关系如下：

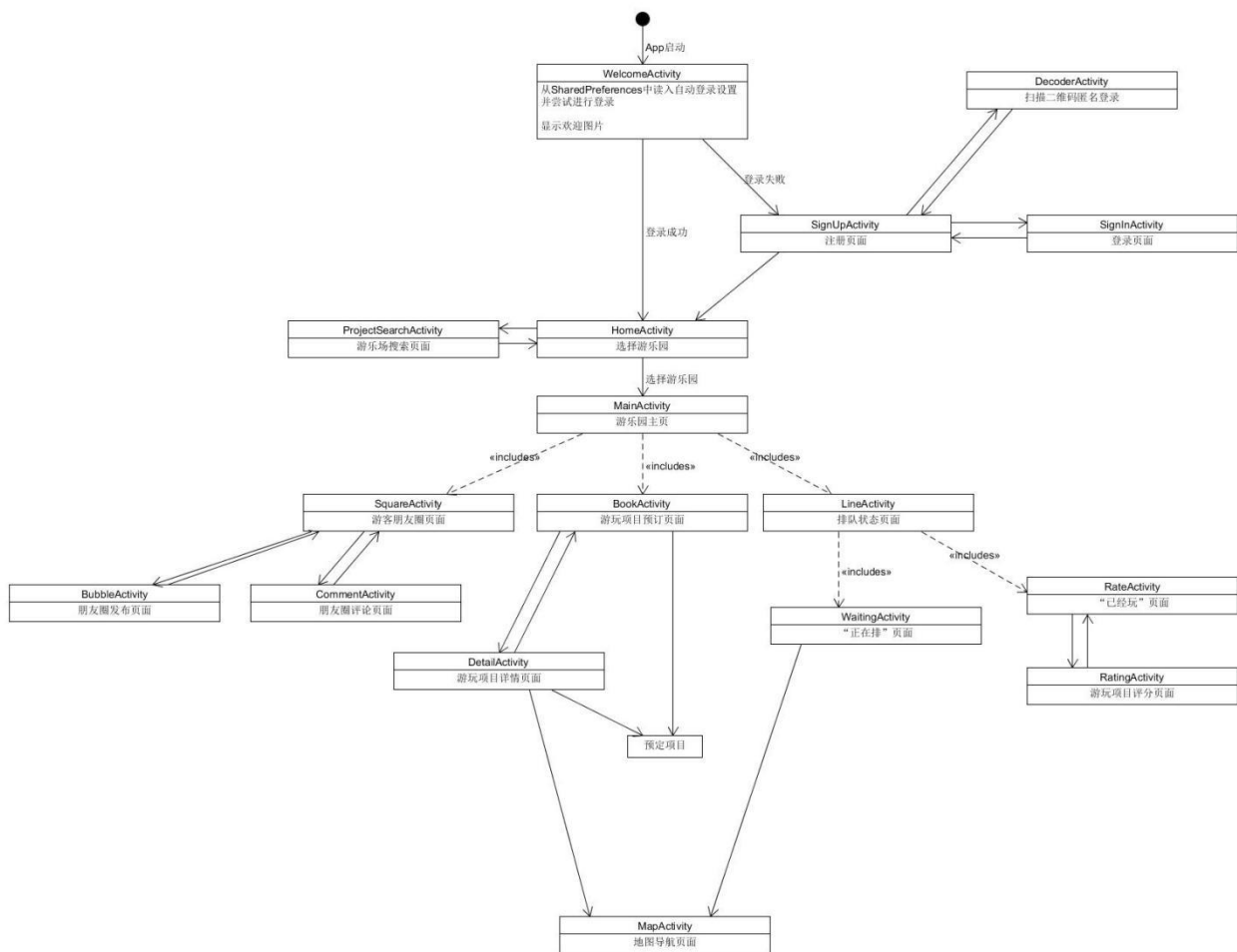


图 4 类图

从安卓的特性上看，每个页面都会是一个独立的 Activity 类，每个页面之间的调用顺序也是简单的压栈/出栈模式，从上图中也能看出 Activity 类之间一定的层级关系。

除了 Activity 类以外，项目还使用了其他的一些功能类。如 util 包中用于与服务器连接的网络通信工具类以及 widget 包中包括可下拉刷新 ListView 与侧滑菜单在内的自定义控件类和组件适配器类。

4.2 后台

- 代码的部署：

计算机 > 文档 (E:) > 大三下 > lineonline >				
名称	修改日期	类型	大小	
app	2015/7/16 20:51	文件夹		
Config	2015/7/16 20:51	文件夹		
Extend	2015/7/16 20:52	文件夹		
Public	2015/7/16 20:52	文件夹		
ThinkPHP	2015/7/16 20:53	文件夹		
.project	2015/7/16 20:52	PROJECT 文件	1 KB	
topic	2015/7/16 20:51	SQL 文件	4 KB	

我们采用的后台框架是 Thinkphp,代码部署的目录如上图。

其中值得注意的是 app 文件夹是程序的入口。

计算机 > 文档 (E:) > 大三下 > lineonline > app >				
名称	修改日期	类型	大小	
Common	2015/7/16 20:51	文件夹		
Conf	2015/7/16 20:51	文件夹		
Lang	2015/7/16 20:51	文件夹		
Lib	2015/7/16 20:51	文件夹		
Runtime	2015/7/16 20:51	文件夹		
Tpl	2015/7/16 20:51	文件夹		
index.php	2015/7/16 20:51	PHP 文件	1 KI	

我们的代码部署在 app 文件夹的 Lib 子文件夹里面。app 文件夹的 Tpl 子文件夹存储静态模板。

计算机 > 文档 (E:) > 大三下 > lineonline > app > Lib >				
名称	修改日期	类型	大小	
Action	2015/7/16 20:51	文件夹		
Behavior	2015/7/16 20:51	文件夹		
Model	2015/7/16 20:51	文件夹		
Widget	2015/7/16 20:51	文件夹		

Lib 文件夹的 Action 类包含了所有的代码。

计算机 > 文档 (E:) > 大三下 > lineonline > app > Lib > Action >				
名称	修改日期	类型	大小	
Book	2015/7/16 20:51	文件夹		
Index	2015/7/16 20:51	文件夹		
Square	2015/7/16 20:51	文件夹		
Wait	2015/7/16 20:51	文件夹		
BookAction.class.php	2015/7/16 20:51	PHP 文件	4 KB	
IndexAction.class.php	2015/7/16 20:51	PHP 文件	2 KB	
PlanAction.class.php	2015/7/16 20:51	PHP 文件	5 KB	
SquareAction.class.php	2015/7/16 20:51	PHP 文件	5 KB	
TestAction.class.php	2015/7/16 20:51	PHP 文件	1 KB	
WaitAction.class.php	2015/7/16 20:51	PHP 文件	3 KB	

每个模板都有一个对应的 Action，处理对应的请求。如游客预定游玩项目的模块的请求用 BookAction.class.php 处理。

● 后台模块设计：

我们将后台请求处理分成四类，对应四个模块。

模块名 Index

函数名	功能	参数	返回值
search(post)	游乐园搜索	searchName	JSONArray，每个元素都是一个游乐场的 JSONObject： pgid – 游乐场 id name – 游乐场名 icon – 游乐场图标 region – 游乐场位置 introduction – 游乐场介绍
show(get)	首页游乐园展示	region	同上

模块名 Book

函数名	功能	参数	返回值
show(get)	列出该游乐场 场内所有游 玩项目	playgroundID	<p>游乐场内所有游玩项目的 JSONArray。每个元素都是一个游玩项目的 JSONObject：</p> <p>gameid – 项目 id</p> <p>title – 项目名</p> <p>photo – 项目图标</p> <p>introduction – 项目介绍</p> <p>caution – 项目注意事项</p> <p>timeterval – JSONArray，每个元素都是一个 JSONObject，表示一个可预约条目：</p> <p>starttime – 开始时间</p> <p>endtime – 结束时间</p> <p>number – 剩余位子</p>
book(post)	根据发送的 预约 id 进行 预约	userID、 gameID startTime、 endTime	True/Exception
select(get)	展示游玩项 目的详细信 息	attractionID	<p>JSONObject，包含该项目的各种信息：</p> <p>gid – 项目 ID</p> <p>pgid – 项目所属游乐场 ID</p> <p>photo – 项目图标 URL</p>

			<p>title – 项目名</p> <p>introduction – 项目简介</p> <p>caution – 注意事项</p> <p>timeterval – JSONArray，所有当前可预约条目，每个元素都是一个 JSONObject，格式同上</p>
--	--	--	--

模块名 Square

函数名	功能	参数	返回值
show(get)	展示所有朋友圈	userID、 playgroundID、 page、 sortMethod	<p>JSONArray，每个元素都是一个 JSONObject，对应每条朋友圈：</p> <p>content – 朋友圈内容</p> <p>good – 点赞数</p> <p>commentnumber – 评论数</p> <p>photo – 图片 url</p> <p>新加入分页功能以及排序功能。</p> <p>服务器根据 page 来返回某一页的数据(暂定每页 20 条朋友圈)。</p> <p>sortMethod 包括 最新 和 最热，分别对应具体值为 0 和 1。</p> <p>数据在返回到客户端前需要服务</p>

			器根据 sortMethod 排序后再返回。
like(post)	点赞	userID、 bubbleID	True/Exception
unLike(post)	取消点赞	userID、 bubbleID	True/Exception
showComment(get)	展示某条朋友圈的所有评论	bubbleID	评论列表
postComment(post)	为某条朋友圈发送评论	bubbleID、 userID、content	True/Exception

模块名 Wait

函数名	功能	参数	返回值
show_1(get)	展示所有该用户已预约的条目	userID	预约项目列表
show_2(get)	展示所有该用户的待评价条目	userID	待评价条目列表

rate(post)	用户发送 评分	userID、attractionID、 rateRank(Integer)	True/Exception
bubblePost(post)	发布朋友 圈	userID、 bubbleContent(String)、	True/Exception

每个模块对应一个 Action :

名称	修改日期	类型	大小
Book	2015/7/16 20:51	文件夹	
Index	2015/7/16 20:51	文件夹	
Square	2015/7/16 20:51	文件夹	
Wait	2015/7/16 20:51	文件夹	
BookAction.class.php	2015/7/16 20:51	PHP 文件	4 KB
IndexAction.class.php	2015/7/16 20:51	PHP 文件	2 KB
PlanAction.class.php	2015/7/16 20:51	PHP 文件	5 KB
SquareAction.class.php	2015/7/16 20:51	PHP 文件	5 KB
TestAction.class.php	2015/7/16 20:51	PHP 文件	1 KB
WaitAction.class.php	2015/7/16 20:51	PHP 文件	3 KB

5. 代码可读性规范

- 格式规范：

规范细则	实现程度
TAB 空格的数量。编辑器上的 TAB 空格数量统一取值为 4	符合规范
换行，每行 120 字符	符合规范
if 语句的嵌套层数 3 层以内	if 语句嵌套最多为 3 层
逻辑上相关序代码与其前后之程序代码间应以空白行加以分隔，在注释段与程序段、以及不同程序段插入空行。	符合规范

● 命名规范：

规范细则	实现程度
包名 必须全部用小写	 > com.playfun.lineonline  > com.playfun.lineonline.util  > com.playfun.lineonline.widget
驼峰命名法	 DetailActivity  attrID  bookListEmergeBtn  bookNowItemList  bookNowListHeight  bookNowListSize  detailInfo
常量名 均全部大写，单词间以'_'隔开	String TAG = "SDK_Sample.Util";
方法 命名采用"动作+属性" 的方法。	private void getValue() { private void setValue() {

● 注释规范：

规范细则	实现程度
JavaDoc 类注释 注明类的功能与开发人员	<pre>/** * 游玩项目预订页面
 * 显示所有可供预订的游玩项目及时间段
 * * @author Robert Peng */ public class BookActivity extends Activity {</pre>
JavaDoc 方法注释 注明方法 的功能、参数说明、返回值说明	<pre>/** * Return the MD5 encode result of the input string * * @param input Input String * @return the MD5 encode result */ public static String MD5(String input) {</pre>
JavaDoc 变量注释 注明变量 的意义	<pre>public class SignUpActivity extends BaseActivity { /** 用于调用二维码扫描页面的request code */ private static final int ANONY_REQUEST_CODE = 1; /** 用于调用登录页面的request code */ private static final int SIGNIN_REQUEST_CODE = 2;</pre>

代码行注释

在必要处解释代码的行为

```
if (selectedText != null && selectedText.equals(mString)) {  
    view.setBackgroundDrawable(selectedDrawable); // 设置选中的背景图片  
} else {  
    view.setBackgroundDrawable(mContext.getResources().getDrawable(  
        normalDrawableId)); // 设置未选中状态背景图片  
}
```

6. 测试用例

测试注册：

```
package com.playfun.lineonline.test;
```

```
import java.net.SocketTimeoutException;
```

```
import junit.framework.TestCase;
```

```
import android.app.Instrumentation;
```

```
import android.content.Intent;
```

```
import android.test.InstrumentationTestCase;
```

```
import com.playfun.lineonline.SignInActivity;
```

```
import com.playfun.lineonline.SignUpActivity;
```

```
import com.playfun.lineonline.util.NetInfoParser;
```

```
import android.widget.EditText;
```

```
public class RegisterTest extends InstrumentationTestCase {
```

```
    public SignUpActivity mact;
```

```
    public EditText email,user,pawd,comfirm;
```

```
    public Instrumentation itc;
```

```
    public String ema,us,pa,comf;
```

```
    /*
```

```
     * (测试环境初始化)
```

```
     * @see android.test.ActivityInstrumentationTestCase2#setUp()
```

```
    */
```

```
    protected void setUp() throws Exception {
```

```
        super.setUp();
```

```
        itc = getInstrumentation();
```

```
        Intent intent = new Intent();
```

```
intent.setClassName("com.playfun.lineonline","com.playfun.lineonline.SignUp  
Activity");
```

```
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
    mact = (SignUpActivity)itc.startActivitySync(intent);
```

```
    //mact = this.getActivity();
```

```
    email =
```

```
(EditText)mact.findViewById(com.playfun.lineonline.R.id.userEmail);
```

```
    user =
```

```
(EditText)mact.findViewById(com.playfun.lineonline.R.id.userName);
```

```

        pawd =
(EditText)mact.findViewById(com.playfun.lineonline.R.id.userPassword);
        comfirm =
(EditText)mact.findViewById(com.playfun.lineonline.R.id.userPasswordConfirm
);
        itc.runOnUiThread(new Runnable(){
            public void run(){
                email.setText("12345678@qq.com");
                user.setText("user1");
                pawd.setText("testpassword2");
                comfirm.setText("testpassword");
            }
        });
    }

    /*
     * 测试用户输入的注册邮箱知否能被控件获取
     */
    public void testEmail(){
        ema="12345678@qq.com";
        assertEquals(ema,email.getText().toString());
    }

    /*
     * 测试用户输入的注册用户名能否被控件获取
     */
    public void testUser(){
        us="user1";
        assertEquals(us,user.getText().toString());
    }

    /*
     * 测试用户输入的注册密码是否能被控件获取
     */
    public void testPassword(){
        pa="testpassword2";
        assertEquals(pa,pawd.getText().toString());
    }

    /*
     * 测试用户的密码跟最后确认的密码是否一致
     */
    public void testComfirm(){

assertEquals(comfirm.getText().toString(),pawd.getText().toString());
    }

    /*
     * 测试确认密码跟用户密码不一致时是否会不通过。

```



```

package com.playfun.lineonline.test;

import java.net.SocketTimeoutException;

import junit.framework.Assert;
import android.app.Instrumentation;
import android.content.Intent;
//import android.test.ActivityInstrumentationTestCase2;
import android.test.InstrumentationTestCase;
import android.text.TextUtils;
import android.util.Log;

import com.playfun.lineonline.util.NetInfoParser;
import com.playfun.lineonline.SignInActivity;
import android.widget.EditText;
public class LoginTest extends InstrumentationTestCase {
    public SignInActivity mactivity;
    public EditText user;
    public EditText paw;
    private String usertest;
    private String pawtest;
    public Instrumentation itc;

    /*
     * 测试环境初始化
     * @see android.test.ActivityInstrumentationTestCase2#setUp()
     */
    protected void setUp() throws Exception {
        super.setUp();
        itc = getInstrumentation();
        Intent intent = new Intent();

intent.setClassName("com.playfun.lineonline", "com.playfun.lineonline.SignIn
Activity");
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        mactivity = (SignInActivity)itc.startActivitySync(intent);
        itc.runOnMainSync(new Runnable(){
            public void run(){
                user =
(EditText)mactivity.findViewById(com.playfun.lineonline.R.id.userName);
                user.setText("xwyyy");
                paw =
(EditText)mactivity.findViewById(com.playfun.lineonline.R.id.userPassword);
                paw.setText("12330355");
            }
        });
    }
}

```

```

        }
    });
}
/*
 * 测试控件是否获取到了用户输入的用户名
 */
public void testUsername(){
    usertest = "xwyyy";
    System.out.println(usertest+" "+user.getText().toString());
    Log.e("user",user.getText().toString());
    Log.e("usertest",usertest);
    Assert.assertEquals("true", usertest, user.getText().toString());
}
/*
 * 测试控件是否获取到了用户输入的密码
 */
public void testPassword(){

    pawtest = "12330355";
    Log.e("paw",paw.getText().toString());
    Log.e("pawtest",pawtest);
    System.out.println(pawtest+" "+paw.getText().toString());
    Assert.assertEquals("true",pawtest,paw.getText().toString());

}
/*
 * 测试客户端将用户名和密码传输给后台是否成功
 */
public void testTransfer(){

    Runnable runnable = new Runnable(){
        public void run(){
            String rel;
            try {
                rel = NetInfoParser.signIn(user.getText().toString(),
paw.getText().toString());
                assertNotNull(rel);
            } catch (SocketTimeoutException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
};

```

```

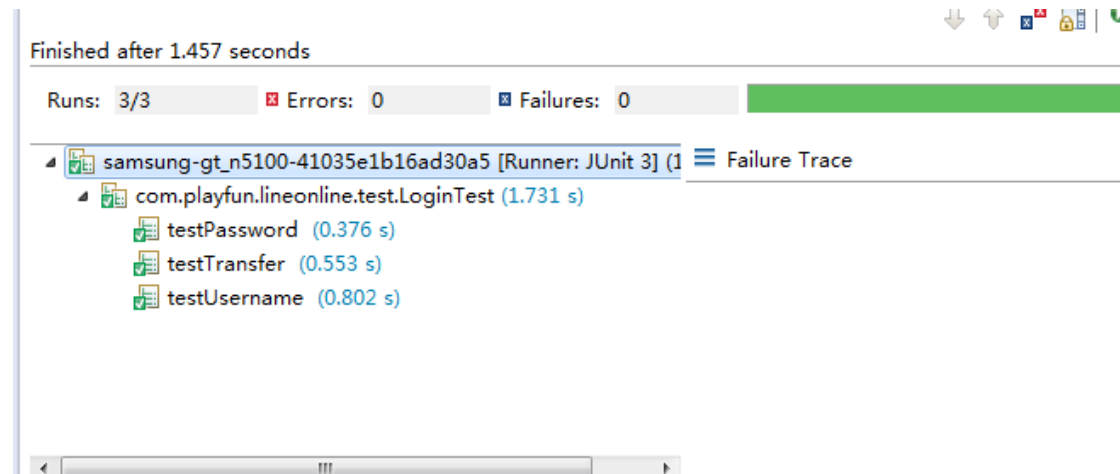
        new Thread(runnable).start();

    }

    protected void tearDown() throws Exception {
        super.tearDown();
        mactivity.finish();
    }

}

```



测试正在排：

```

package com.playfun.lineonline.test;

import java.net.SocketTimeoutException;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import junit.framework.Assert;
import junit.framework.TestCase;
import android.app.Instrumentation;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.test.ActivityInstrumentationTestCase2;
import android.test.InstrumentationTestCase;
import com.playfun.lineonline.SignInActivity;
import com.playfun.lineonline.WaitingActivity;
import com.playfun.lineonline.util.NetInfoParser;

```

```

import android.widget.EditText;
import android.widget.Button;
public class WaitingTest extends InstrumentationTestCase {

    public WaitingActivity wact;
    public Instrumentation itc;
    public String uid = "17";
    public String sthour;
    public String stmin;
    public String endhour;
    public String endmin;
    public Button Tomap;
    /*
     * 向后台请求游玩项目信息
     */
    final Runnable runnable = new Runnable(){
        public void run(){
            try {
                JSONArray getResult = NetInfoParser.showWaitingProject("17");
                Bundle data = new Bundle();
                Assert.assertNotNull(getResult);
                data.putString("rel", getResult.toString());
                Message msg = handler.obtainMessage();
                msg.setData(data);
                handler.sendMessage(msg);

            } catch (SocketTimeoutException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (JSONException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    };
    /*
     * 处理后台返回信息
     */
    final Handler handler = new Handler(){
        public void handleMessage(Message msg){
            try {
                JSONArray rel = new JSONArray(msg.getData().getString("rel"));
                JSONObject iter;
                for (int i = 0; i < rel.length(); i++) {

```

```

        iter = rel.getJSONObject(i);
        //game = iter.getJSONObject("game");

        int distance = 500;
        // TODO should be set according to current position
        int minutes = distance / 100;

        String startTime = iter.getString("starttime");
        String endTime = iter.getString("endtime");
        String startHour = startTime.split(" ")[1].split(":")[0];
        String startMinu = startTime.split(" ")[1].split(":")[1];
        String endHour = endTime.split(" ")[1].split(":")[0];
        String endMinu = endTime.split(" ")[1].split(":")[1];
        int sh = Integer.parseInt(startHour);
        Assert.assertTrue(sh>=0&&24>sh);
        int sm = Integer.parseInt(startMinu);
        Assert.assertFalse(sm<0||sm>=60);
        int eh = Integer.parseInt(endHour);
        Assert.assertTrue(eh>=0&&24>eh);
        int em = Integer.parseInt(endMinu);
        Assert.assertFalse(em<0||em>=60);
    }
} catch (JSONException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

};
/*
 * (初始化测试环境)
 * @see junit.framework.TestCase#setUp()
 */
protected void setUp() throws Exception {
    super.setUp();
    itc = getInstrumentation();
    Intent intent = new Intent();

intent.setClassName("com.playfun.lineonline", "com.playfun.lineonline.Waitin
gActivity");
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    wact = (WaitingActivity)itc.startActivitySync(intent);
    Tomap = (Button)wact.findViewById(com.playfun.lineonline.R.id.go);

```



```



    }
    /*public void testNotNull(){
        Assert.assertNotNull(Tomap);
    }*/
    /*public void testToMap(){
        itc.runOnMainSync(new Runnable(){
            public void run(){
                Tomap.requestFocus();
                Tomap.performClick();
            }
        });
        itc.waitForIdleSync();


    }*/
    /*
    * 测试是否能获取后台信息
    */
    public void testTransfer() throws SocketTimeoutException, JSONException{
        JSONArray getResult = NetInfoParser.showWaitingProject("17");
        Assert.assertNotNull(getResult);
    }
    /*
    * 测试后台返回信息的正确性
    */
    public void testTime(){
        new Thread(runnable).start();
    }
    protected void tearDown() throws Exception {
        super.tearDown();
        wact.finish();
    }
}


```


Finished after 1.086 seconds

Runs: 2/2  Errors: 0  Failures: 0

▲  samsung-gt_n5100-41035e1b16ad30a5 [Runner: JUnit 3] (1)  Failure Trace

▲  com.playfun.lineonline.test.WaitingTest (1.336 s)

 testTime (0.411 s)

 testTransfer (0.925 s)