项目设计报告

* **需求分析**

在本项目中，需要有一个主页面，显示所有的功能，包括对于学生的相关信息进行录入，删除，修改，查询等功能。

为了增加本项目的创新性，需要在本项目中加入了背景轻音乐，可以让使用者在枯燥的信息管理工作中，由于轻音乐的加入，改善一下工作时的心情，提升工作效率，此功能通过后台服务功能获取。

同时为了增加APP的鲁棒性，防止因为意外导致的数据丢失，需要在本项目中使用了两种方法来实现对于信息的管理功能，分别用到了动态数组以及SQLite数据库。

为了增加本项目的安全性，需要用到NDK实现对于数据的加密。

为了提高本项目的辨识度，需要在本项目中加入我们校区的定位，通过百度地图功能实现。

为了提高本软件的宣传度，为了更好的推广应用，需要在APP中加入抽奖活动，灵感来自于拼多多发红包，可以以最短的时间吸引最多的用户。通过使用线程实现。

最后为了提高软件的呈现率，需要写一个桌面小组件，用户放在桌面上之后，每次打开手机都可以看到，并且可以快速启动，通过widget组件开发实现。

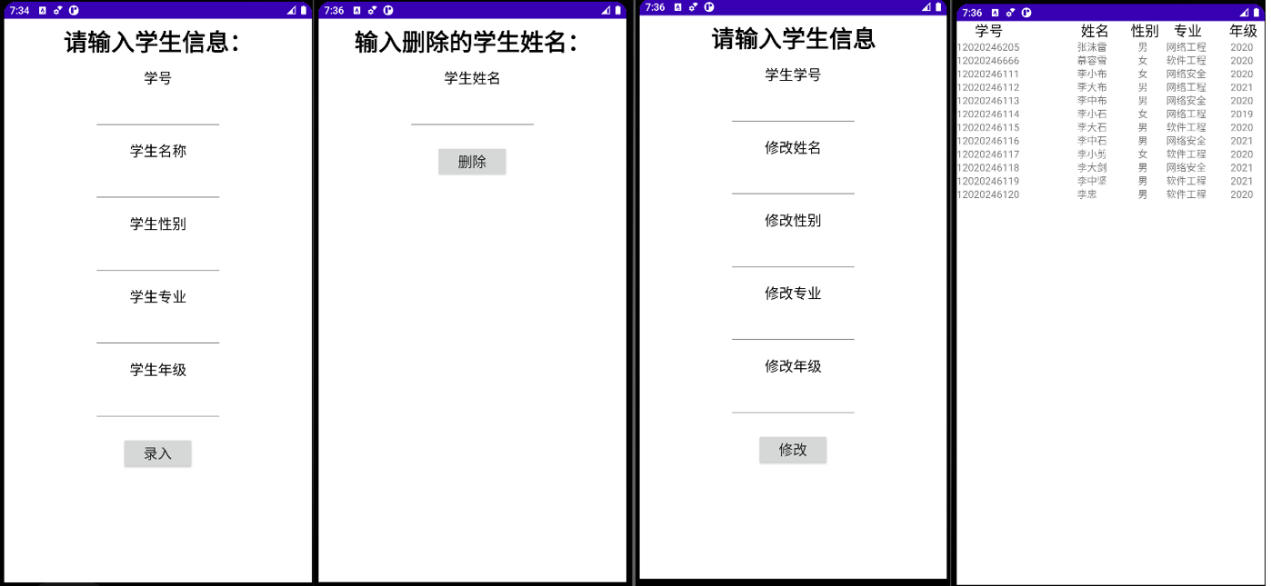
* **程序设计**
* **界面设计**

通过以上的需求分析，应用程序应包含个应用界面。

在首页面中，显示本应用的所有功能



最上面的四个按钮使用了动态数组的技术，实现本应用基础的最简单的增删改查功能，界面分别为：



使用SQLite数据库操作界面：



使用NDK对数据进行加密的界面：



后台播放音乐控制界面：



百度地图定位中卫校区的界面：

抽奖界面：

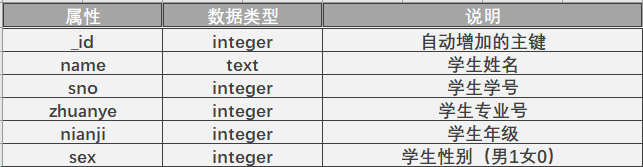


小组件：

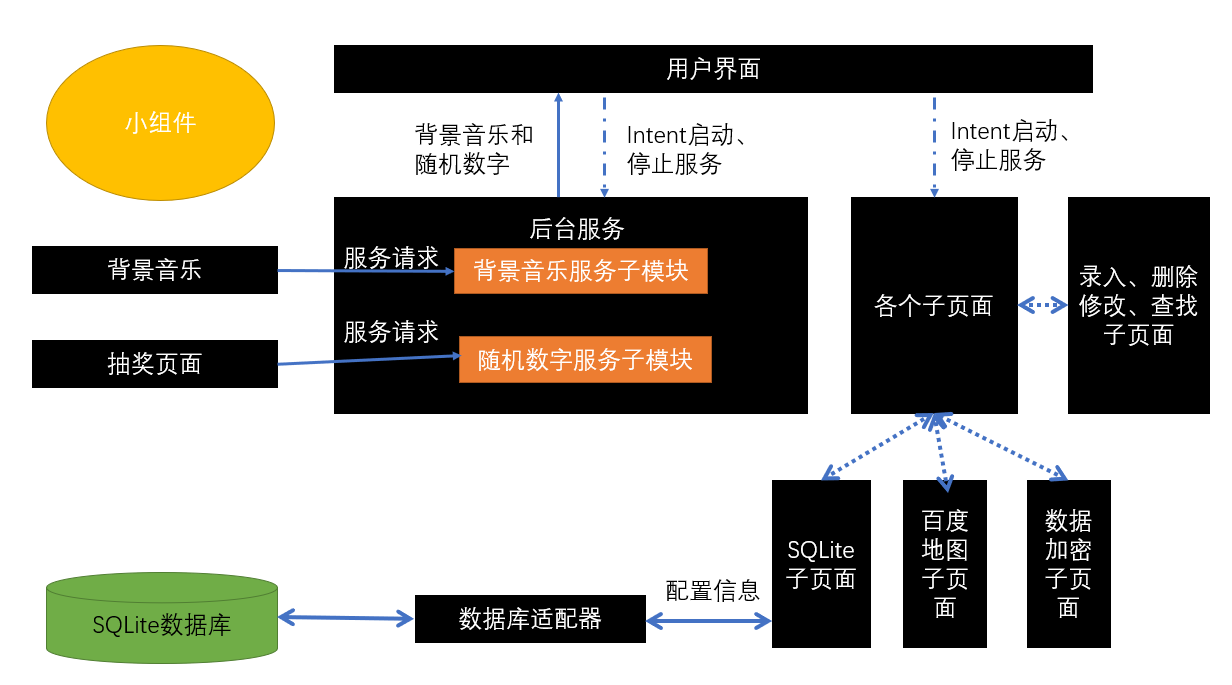


* **数据库设计**

配置信息的数据库表结构

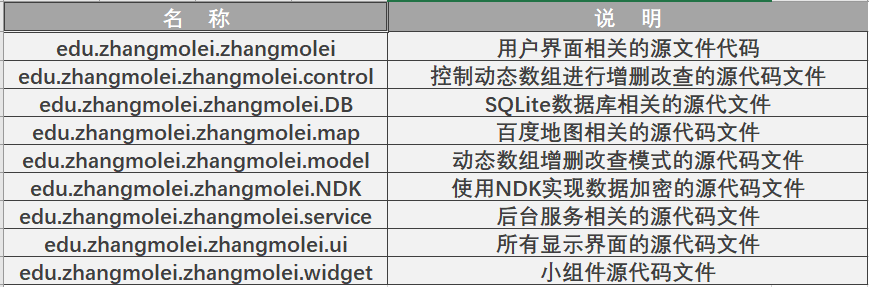
****

* **模块结构设计**

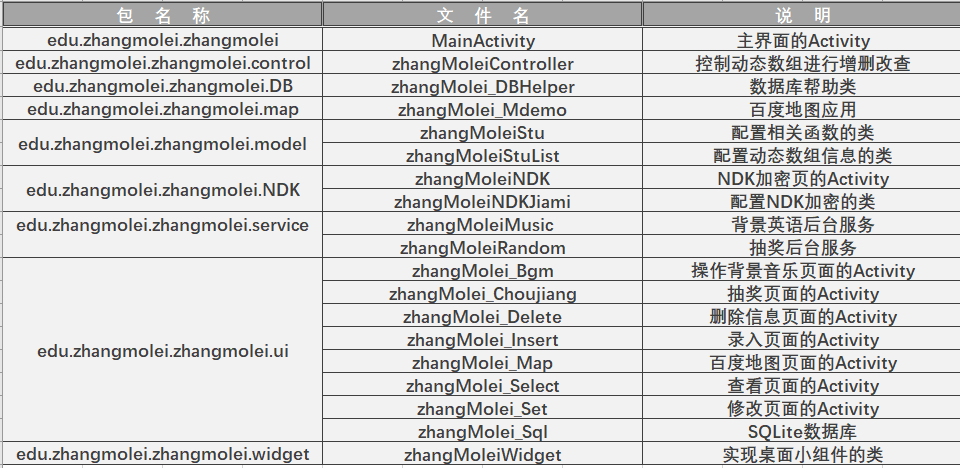
****

* **程序开发**
* **功能报告**

本项目的命名空间：



本项目源代码文件用途说明：



本项目资源文件名称与用途：



* **开发功能的核心代码**

对动态数组进行控制的“zhangMoleiController”类增删改查的核心代码如下：

public boolean addStu(String sno,String name,String sex,String zhuanye,String nianji) {  
 zhangMoleiStuList zhangMoleiStuList = edu.zhangmolei.zhangmolei.model.zhangMoleiStuList.*getZhangMoleiStuList*();  
 int i = 0;  
 for (; i < zhangMoleiStuList.size(); ++i) {  
  
 zhangMoleiStu zhangMoleiStu = zhangMoleiStuList.get(i);  
 String bid = zhangMoleiStu.getSno();  
 if (bid.equals(sno)) {  
 break;  
 }  
 }  
 if (i == zhangMoleiStuList.size()) {  
 zhangMoleiStu zhangMoleiStu = new zhangMoleiStu(sno, name, sex,zhuanye,nianji);  
 zhangMoleiStuList.add(zhangMoleiStu);  
 return true;  
  
 }  
 return false;  
}  
  
public zhangMoleiStuList searchStu() {  
 zhangMoleiStuList zhangMoleiStuList = edu.zhangmolei.zhangmolei.model.zhangMoleiStuList.*getZhangMoleiStuList*();  
 return zhangMoleiStuList;  
}  
  
public boolean deleteStu(String name)  
{  
 zhangMoleiStuList zhangMoleiStuList = edu.zhangmolei.zhangmolei.model.zhangMoleiStuList.*getZhangMoleiStuList*();  
 for (int i=0; i<zhangMoleiStuList.size(); ++i)  
 {  
 zhangMoleiStu zhangMoleiStu = zhangMoleiStuList.get(i);  
 if(zhangMoleiStu.getName().equals(name))  
 {  
 zhangMoleiStuList.remove(i);  
 return true;  
 }  
 }  
 return false;  
}  
public boolean setStu(String sno,String name,String sex,String zhuanye,String nianji)  
{  
 zhangMoleiStuList zhangMoleiStuList = edu.zhangmolei.zhangmolei.model.zhangMoleiStuList.*getZhangMoleiStuList*();  
 for(int i=0;i<zhangMoleiStuList.size();++i)  
 {  
 zhangMoleiStu zhangMoleiStu = zhangMoleiStuList.get(i);  
 if(zhangMoleiStu.getSno().equals(sno))  
 {  
 zhangMoleiStu zhangMoleiStu1 = new zhangMoleiStu(sno, name, sex,zhuanye,nianji);  
 zhangMoleiStuList.set(i,zhangMoleiStu1);  
 return true;  
 }  
 }  
 return false;  
  
}

“zhangMolei\_DBHelper”的核心代码如下

public zhangMolei\_DBHelper(Context context){super(context,"zmldemo.db",null,1);}  
  
public void onCreate(SQLiteDatabase db) {  
 db.execSQL("CREATE TABLE information(\_id INTEGER PRIMARY KEY AUTOINCREMENT,name TEXT,sno INTEGER,zhuanye INTEGER,nianji INTGER,sex INTEGER)");  
}

创建了一个存放数据的表，并且定义了相应表项。

“zhangMolei\_Mdemo”的核心代码如下

SDKInitializer.*initialize*(getApplicationContext());

SDKInitializer.*setCoordType*(CoordType.*BD09LL*);

初始化了SDK，并且定义了坐标类型。

“zhangMoleiStu”的核心代码如下

public zhangMoleiStu(String sno, String name, String sex,String zhuangye,String nianji) {  
 this.sno = sno;  
 this.name = name;  
 this.sex = sex;  
 this.zhuanye = zhuangye;  
 this.nianji = nianji;  
}

定义了需要用到的属性。

“zhangMoleiStuList”的核心代码如下

private static zhangMoleiStuList *zhangMoleiStuList* = null;

防止外部调用构造函数。

“zhangMoleiNDK”的核心代码如下

public void encode(View view) {  
 //获取edittext的文字  
 String word = et.getText().toString();  
 //传递c加密处理  
 word = jni.encodeInC(word);  
 //把处理后的结果显示到edittext上  
 et.setText(word);  
}  
  
public void decode(View view) {  
 //获取edittext的文字  
 String word = et.getText().toString();  
 //传递c加密处理  
 word = jni.decodeInC(word);  
 //把处理后的结果显示到edittext上  
 et.setText(word);  
}

定义了加密以及解密两个按钮的功能。

“zhangMoleiNDKJiami”的核心代码如下

static {  
 System.*loadLibrary*("zmlencrypt");  
}

使用静态加载方式，在类加载时加载共享库。

“zhangMoleiMusic”的核心代码如下

public void onCreate(){//创建后台服务  
 super.onCreate();  
 player =MediaPlayer.*create*(this,R.raw.*summer*);//创建媒体播放对象  
 }  
public int onStartCommand(Intent intent,int flags,int startId){//启动后台服务  
 super.onStartCommand(intent,flags,startId);

}

以上为该文件内的部分主要代码，创建了后台服务。

“zhangMoleiRandom”的核心代码如下

public void onCreate() {  
 super.onCreate();  
 workThread = new Thread(null,backgroudWork,"WorkThread");  
}

public void run() {  
 try {  
 while(!Thread.*interrupted*()){  
 Random rand = new Random();  
 int random = rand.nextInt(100);  
 zhangMolei\_Choujiang.*UpdateGUI*(random);  
 Thread.*sleep*(100);  
 }  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
}

以上为利用线程实现随机函数的主要代码，在后台服务。

“zhangMolei\_Bgm”的核心代码如下

if(view==btn){//开始  
 intent.putExtra("play",1);  
 zhangMolei\_Bgm.this.startService(intent);  
 myConnection=new MyConnection();  
 bindService(intent,myConnection,*BIND\_AUTO\_CREATE*);  
}  
else if(view==btn2){//暂停  
 intent.putExtra("play",2);  
 zhangMolei\_Bgm.this.startService(intent);  
 System.*out*.println(2);  
  
}

开始和暂停背景音乐的核心代码。

“zhangMolei\_Map”的核心代码如下

baiduMap = mapView.getMap();

获取百度地图。

Double lng = 105.172915;  
Double lat = 37.497521;  
LatLng point = new LatLng(lat, lng);  
  
MapStatus mMapStatus = new MapStatus.Builder()  
 .target(point)  
 .zoom(11)  
 .build();

地图预定显示中心点的经纬度，以及地图缩放等级。

“zhangMolei\_Sql”的核心代码如下

final zhangMolei\_DBHelper dbHelper=new zhangMolei\_DBHelper(zhangMolei\_Sql.this);

SQLiteDatabase db=dbHelper.getReadableDatabase();

SQLiteDatabase db=dbHelper.getWritableDatabase();

long id =db.insert("information",null,values);

db.update("information",values,"name=?",new String[]{name.getText().toString()});

Cursor cursor = db.rawQuery("select \* from information where name = ? ", new String[]{name1});

while (cursor.moveToNext())

往下移动一条数据

db.delete("information","name=?",new String[]{name.getText().toString()});

Cursor cursor = db.rawQuery("select \* from information",null);

以上为SQLite数据库中实现增删改查的一小部分核心代码，完整代码在Android Studio中查看。

“zhangMoleiWidget”的核心代码如下

public void onUpdate(Context context, AppWidgetManager appWidgetManager,  
 int[] appWidgetIds) {  
 super.onUpdate(context, appWidgetManager, appWidgetIds);  
 Intent intent = new Intent(context,MainActivity.class);  
 PendingIntent Pfullintent=PendingIntent.*getActivity*(context, 0, intent, 0);  
 RemoteViews views= new RemoteViews(context.getPackageName(), R.layout.*zhangmolei\_widget*);  
 views.setOnClickPendingIntent(R.id.*imageView1*,Pfullintent);  
 appWidgetManager.updateAppWidget(appWidgetIds, views);  
  
}

Android.mk文件中的核心代码

LOCAL\_MODULE := zmlencrypt  
LOCAL\_SRC\_FILES := zmlencrypt.c

声明了模块名称，以及编译模块所需使用的C/C++文件列表。

“zmlencrypt.c”文件中的核心代码

char \*\_JString2CStr(JNIEnv \*env, jstring jstr) {  
 char \*rtn = NULL;  
 jclass clsstring = (\*env)->FindClass(env, "java/lang/String");  
 jstring strencode = (\*env)->NewStringUTF(env, "GB2312");  
 jmethodID mid = (\*env)->GetMethodID(env, clsstring, "getBytes", "(Ljava/lang/String;)");  
 jbyteArray barr = (jbyteArray) (\*env)->CallObjectMethod(env, jstr, mid,strencode);  
 jsize alen = (\*env)->GetArrayLength(env, barr);  
 jbyte \*ba = (\*env)->GetByteArrayElements(env, barr, JNI\_FALSE);  
 if (alen > 0) {  
 rtn = (char \*) malloc(alen + 1);  
 memcpy(rtn, ba, alen);  
 rtn[alen] = 0;  
 }  
 (\*env)->ReleaseByteArrayElements(env, barr, ba, 0);  
 return rtn;  
}  
  
JNIEXPORT jstring JNICALL Java\_com\_fly\_demo5\_JNI\_encodeInC(JNIEnv \*env, jobject clazz, jstring jstr) {  
 char \*cstr = \_JString2CStr(env, jstr);  
 int length = strlen(cstr);  
 int i;  
 for (i = 0; i < length; i++) {  
 \*(cstr + i) += 2;  
 }  
 return (\*env)->NewStringUTF(env, cstr);  
}  
  
  
JNIEXPORT jstring JNICALL Java\_com\_fly\_demo5\_JNI\_decodeInC(JNIEnv \*env, jobject clazz, jstring jstr) {  
 char \*cstr = \_JString2CStr(env, jstr);  
 int length = strlen(cstr);  
 int i;  
 for (i = 0; i < length; i++) {  
 \*(cstr + i) -= 2;  
 }  
 return (\*env)->NewStringUTF(env, cstr);  
}

实现了把一个jstring转换成一个c语言的char\* 类型.实现加密功能，通过还原实现解密功能。

“zml\_widget\_template.xml”文件的核心代码如下

<appwidget-provider  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:minWidth="110dp"  
 android:minHeight="110dp"  
 android:updatePeriodMillis="86400000"  
 android:initialLayout="@layout/zhangmolei\_widget"  
 >  
</appwidget-provider>

“build.gradle”文件的核心代码如下

sourceSets **{** *main* **{** jniLibs.srcDir 'libs'  
 **}  
}**

implementation files('libs\\BaiduLBS\_Android.jar')

为了实现百度地图功能，以及NDK功能所需的代码。实现对开发包的导入。

“AndroidManifest.xml”中值得注意的代码（Activity和Service的就不展示了）：

<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

以上是地图相关权限许可

<meta-data  
 android:name="com.baidu.lbsapi.API\_KEY"  
 android:value="4qrSt2BAsoUYV5tYcLYbudIiSlOIsKB9" />

以上是百度地图的密钥，在百度开发者平台申请。

<receiver android:name=".widget.zhangMoleiWidget">  
 <intent-filter>  
 <action android:name="android.appwidget.action.APPWIDGET\_UPDATE"/>  
 </intent-filter>  
 <meta-data android:name="android.appwidget.provider"  
 android:resource="@xml/zml\_widget\_template"/>  
</receiver>

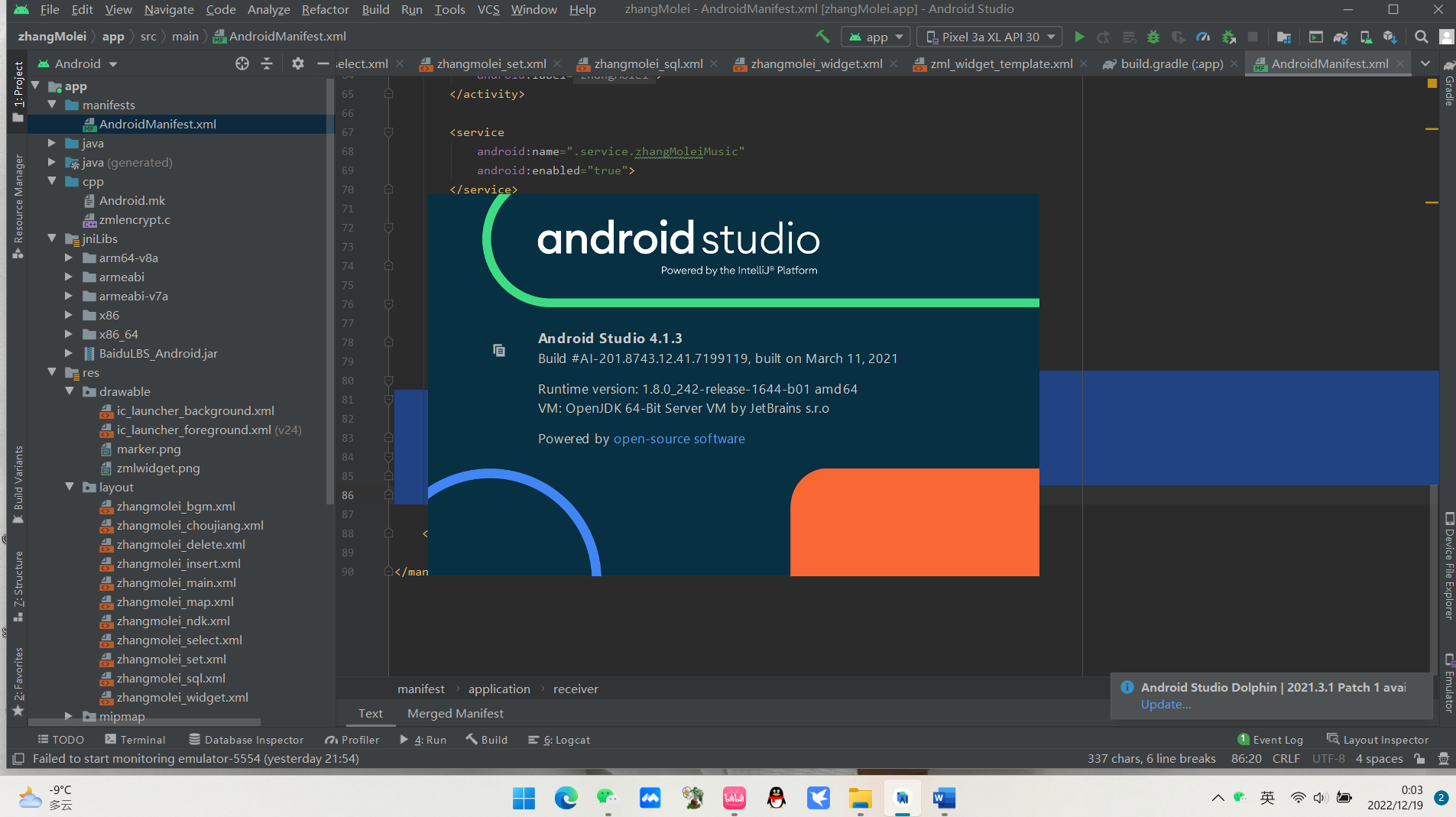
声明了receiver标签，定义了小组件相关的属性和数据。

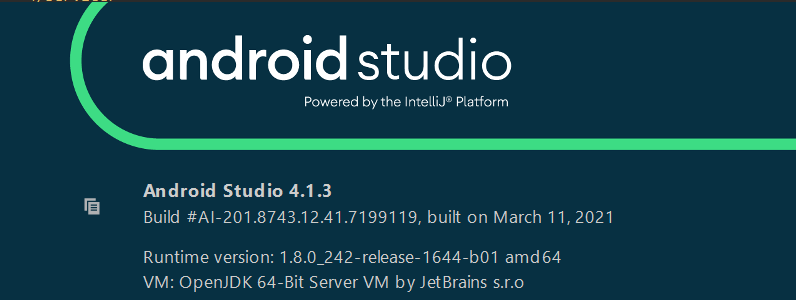
以上为我认为本项目的核心代码，不完整，具体的代码还请在Android Studio里面查看，除了NDK加密那个界面可能闪退（目前在努力找原因中），其他功能都可以正常使用。

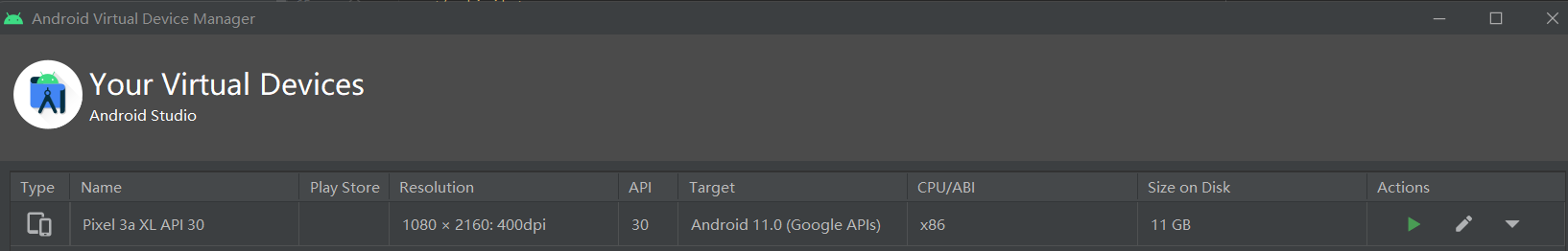
为确保可以正常使用，建议使用和我一样的Android Studio环境配置。

Android Studio 4.1.3

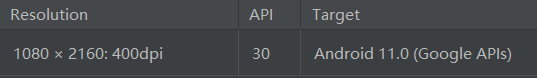
API 30 1080\*2160最佳。











感谢老师的浏览。

**网络工程一班**

**张沫雷**

**12020246205**