

## 11 组——软件设计综合实验-软件设计文档

# POKEDEX-精灵宝可梦图鉴

### 技术选型理由

当下手机 app 软件在社会人群中的普及和应用非常广泛，不仅使用方便而且几乎不需要任何因此我们以 Android Studio 为基础，使用 java 和 xml 语言进行编写。

采用以下技术：

- 1、界面框架设计技术，使得本 app 的界面美观，布局合理。
- 2、事件处理的实现技术，使得成功实现点击事件、activity 跳转等功能。
- 3、使用了 SQLite 数据库技术，以对宝可梦资料进行数据储存。
- 4、使用了 Notification 通知栏显示，实现在下拉列表中展示相关信息的功能。
- 5、Broadcast 静态广播通知，实现了进程间的通信。
- 6、Widget 桌面小控件展示，实现在桌面显示相关信息以及快速进入软件的功能。

### 架构设计

简单说明各源码文件作用：

重要的源码文件有以下几个，其余的一些较简单的源码文件如布局全局设置、测试类文件这里不再写出。

AndroidManifest.xml: 是本应用的入口文件，它描述了 package 中暴露的组件（activities, services, 等等），各自的实现类，各种能被处理的数据和启动位置。

main\_activity.java:应用主函数，加载主页面，负责更新数据以及选择对应功能。

pokeinfo\_activity.java:负责在生成的宝可梦页面中更新对应资料和信息。

pokelist\_activity.java:负责宝可梦列表页面的功能实现和页面加载。

pokemonDB.java、pokemonlist.java:负责数据库的实现。

staticReceiver.java:实现静态广播通知效果。

widgetDemo.java:实现 Widget 桌面小控件展示效果。

main\_layout.xml:主页面布局实现。

pokemon\_item.xml、pokemonlist\_layout.xml:宝可梦列表页面布局实现。

pokemon\_layout.xml:宝可梦资料页面布局实现。

widget\_demo.xml:Widget 桌面小控件布局实现。

### 模块划分

可分为两大模块

1. 系统管理：包括数据库加载，功能选择；
2. 图鉴功能管理：包括宝可梦列表页面和宝可梦资料页面中的信息加载和显示，页面跳转，点击后响应的功能实现（如广播通知、Widget 控件等）。

## 软件技术实现

### 1. 界面框架设计技术:

属于图鉴功能管理模块。在源码中的 main\_layout.xml、pokemon\_item.xml、pokemonlist\_layout.xml、pokemon\_layout.xml、widget\_demo.xml 以及 drawable 里的几个 xml 文件中实现。

如 main\_layout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:src="@drawable/backgroundpika"
        android:scaleType="centerCrop"/>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical" android:layout_width="match_parent"
        android:layout_height="match_parent"
    >
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="50dp">
            <ImageView
                android:layout_marginLeft="60dp"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:src="@drawable/log1"
                android:scaleType="centerCrop"
                android:id="@+id/log1"
                android:adjustViewBounds="true" />
        </RelativeLayout>
    </RelativeLayout>
</FrameLayout>
```

### 2. 事件处理的实现技术:

在两个模块中均有运用。主要出现于 main\_activity.java、pokeinfo\_activity.java、pokelist\_activity.java 三个文件里, 实现了点击后进行页面跳转以及其它功能。

如 main\_activity.java 文件中的 54 行, 实现了点击后页面跳转到列表页面的功能:

```
button1.setOnClickListener((view) -> {
    if (buttonDB.getHint().equals("已更新数据")) {
        Intent intent = new Intent(main_activity.this, pokelist_activity.class);
        startActivity(intent);
    } else {
        Toast.makeText(main_activity.this, "请先更新数据", Toast.LENGTH_SHORT).show();
    }
});
```

### 3. 使用了 SQLite 数据库技术

属于系统管理模块。实现代码为 pokemonDB.java、pokemonlist.java。

### 4. 使用了 Notification 通知栏显示

属于图鉴功能管理模块。函数实现在 StaticReceiver.java 里。

```
public class StaticReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals("android.intent.action.staticreceiver")) {
            Bundle bundle = intent.getExtras();
            Bitmap bm = BitmapFactory.decodeResource(Resources.getSystem(), bundle.getInt("fpic"));
            Notification.Builder builder = new Notification.Builder(context);
            Intent intent1 = new Intent(context, pokelist_activity.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent1, 0);
            builder.setContentTitle("POKEDEX")
                .setSmallIcon(bundle.getInt("fpic"))
                .setContentText(bundle.getString("name"))
                .setLargeIcon(bm)
                .setTicker("请详细阅读有关" + bundle.getString("name") + "的介绍")
                .setContentIntent(pendingIntent)
                .setAutoCancel(true);
            NotificationManager manager = (NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
            Notification notify = builder.build();
            manager.notify(0, notify);
        }
    }
}
```

### 5. Broadcast 静态广播通知:

属于图鉴功能管理模块。函数实现在 StaticReceiver.java 里。在 AndroidManifest.xml 里也有声明。

```
<receiver android:name=".StaticReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.staticreceiver" />
    </intent-filter>
</receiver>
<receiver android:name=".WidgetDemo">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="android.intent.action.staticreceiver" />
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_demo_info" />
</receiver>
```

### 6. Widget 桌面小控件展示:

属于图鉴功能管理模块。Widget 控件的布局实现代码为 widget\_demo.xml，函数实现在 WidgetDemo.java 里。在 AndroidManifest.xml 里也有声明。

```
public class WidgetDemo extends AppWidgetProvider {
    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {
        super.onUpdate(context, appWidgetManager, appWidgetIds);
        Intent clickInt = new Intent(context, pokelist_activity.class);
        PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
        RemoteViews rv = new RemoteViews(context.getPackageName(), R.layout.widget_demo);
        rv.setOnClickPendingIntent(R.id.widget_pic, pi);
        appWidgetManager.updateAppWidget(appWidgetIds, rv);
    }
}
```