Running Fun 说明文挡

戴秉璋（3120103469）-求是科学班（物理）
陈雪儿（3130100668）-计算机科学与技术（1302）

一. App 功能分析

Running Fun 是一款运动辅助类 APP，可以帮助用户记录全天候的运动轨迹，并且支持用户在记录中添加心情描述、照片等信息。它的功能有 GPS 定位、显示运动轨迹、记录并显示特定时间点的运动信息比如心情状态、照片等、下载离线地图。

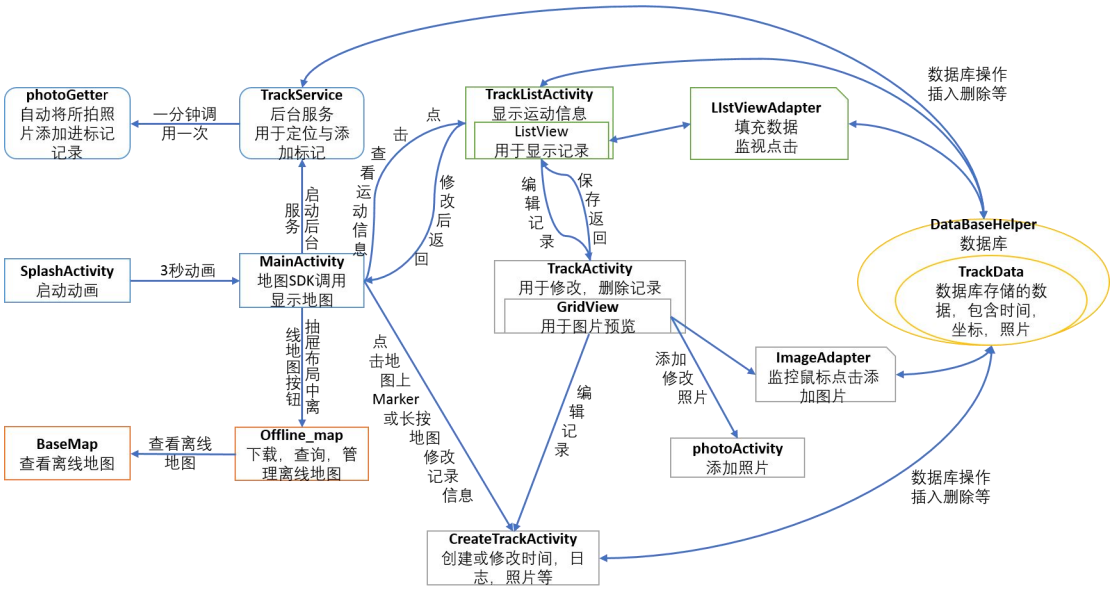
二. 系统设计

基于 Baidu Android 地图 SDK，用于 Android 系统移动设备的地图应用，通过调用地图 SDK 接口，访问百度地图服务和数据。

1. 首先使用地图展示和地图操作功能，联网显示实时交通图。地图可通过接口或手势控制来实现地图的点击、双击、长按、缩放、旋转、改变视角等操作。
2. 使用百度定位类实现后台位置跟踪，记录每时每刻的地理位置并自动在地图上标注。打开 App 后就可以从地图上看到自己一天的行程。
3. 后台服务在预定定位时间到达的时候就把这一段时间所有的照片的 uri 记录在这个定位点上，所以当打开 app 就可以知道自己在哪段时间拍下了那些照片（也可以自己手动编辑每个点上的描述和图片）。
4. 用户可以在线下载离线包接口，下载离线地图包，使用离线地图可节省用户流量，提供更好的地图展示效果。

三. 总体架构

下面流程图仅包含主要功能部分，anim 动画效果和一些太过简单的显示功能没有包括在流程图里（可能需要放大查看）。



四. 部分功能介绍

1. SplashActivity 会使用 Handler 的 postDelayed 方法，3 秒后执行跳转到 MainActivity。（图 1）
2. MainActivity 启动定位，启动 TrackService 服务，显示地图，监视地图上 marker 的拖动点击等事件。（图 2）
3. MainActivity 中实现了 DrawerLayout，可以点击离线地图，运动信息等。（图 3）
4. TrackService 自动定位，检测用户坐标是否发生变化，若是，则在地图上定一个 marker，一个 marker 对应用户的一项运动记录，调用 PhotoGetter 手机照片。（图 2）
5. PhotoGetter 根据时间间隔，将这段时间拍下的照片添加到对应记录中。
6. TrackListActivity，显示运动信息列表，ListView 的 adapter 是 ListViewAdapter.java。（图 6）
7. offline_map 管理，下载离线地图。（图 8）
8. BaseMap 查看离线视图，直接显示地图。
9. TrackActivity 编辑，删除记录信息。图片预览布局是 GridView，adapter 是 ImageViewAdapter.java。（图 5）
10. PhotoActivity 添加照片，更新视图。（图 7）
11. CreateTrackActivity 创建，删除，编辑记录信息。（图 7）
12. DatabaseHelper 后台数据库封装，用于在数据库中插入，删除，修改信息。
13. TrackData 各个视图和数据库中交换的数据，存储有时间，坐标，图片等信息。

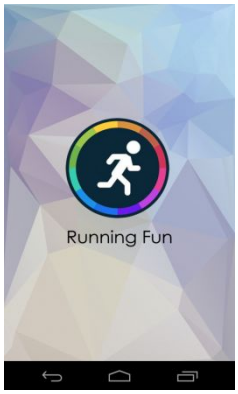


图 1

图 2

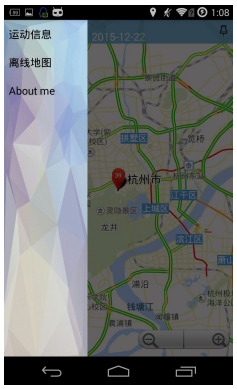


图 3

图 4

图 5

图 6

图 7

图 8

主要功能截图

五. 关键数据结构与难点

- **ListView**

因为在我们的 app 中，很多信息数量都是可变的，比如离线地图，每天的运动信息的记录，在每个记录中所存放的照片数量等等，所以我们需要 ListView 来展现可变数量的内容。很多信息都涉及到数据库的操作，所以数据库也是一个难点，数据库相关的类之后会具体介绍。在 MainActivity 与 TrackListActivity 中，大量用到了 ListView，以列表的形式展示具体内容，并且能够根据数据的长度自适应显示。下面以 MainActivity 中的 drawerlayout 中用到的 ListView 为例来讲解。

```
private DrawerLayout mDrawerLayout;
private ListView mDrawerList;
private ArrayList<String> menuLists;
private ArrayAdapter<String> adapter;

protected void onCreate(Bundle savedInstanceState) {
  /* come code here */
    mDrawerLayout = (DrawerLayout)findViewById(R.id.drawer_layout);
    mDrawerList = (ListView)findViewById(R.id.left_drawer);
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
menuLists);
    mDrawerList.setAdapter(adapter);
    mDrawerList.setOnItemClickListener(this);
    /* some code here */
}

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
      /* some code here */
   }
```

我们先往 Array<String>中填充数据，这里是字符串，同时在后面获得资源的时候把前面的数据填充进到布局中，之后进行 ListView 和其 adapter 绑定，ListView 主要实现两个作用：

1. 将数据填充到布局。
2. 处理用户的选择点击等操作。

onItemClick 中可以处理点击时间，比如开始另一个 activity 等。

- **database 的处理：**

首先，各个 activity 都有可能对记录数据进行增删改查，前面的流程图里可以清楚看见和数据库有交互的 activity 有哪些。数据库返回类型，其增删改查操作都要进行重载，故我们需要进行包装，将一个一个的数据包装成其他 activity 中所需要的内容，这里是 trackdata 类：

```
public class TrackData implements Serializable{
    public final static String LATITUDE_STRING = "latitude";
```

```
    public final static String PLACENAME_STRING = "place_name";
    public final static String DATE_STRING = "date";
    public final static String HOUR_STRING = "hour";
    /* 一些数据和set(...)，get(...)函数 */
}
```

database 中要把 trackdata 中的数据取出放数据库，查询时要取出数据并且构造 trackdata 类返回。

databasehelper 类：

```
public class DatabaseHelper extends SQLiteOpenHelper {
    @Override
    public void onUpgrade(SQLiteDatabase arg0, int arg1, int arg2)；
    public void insertData(TrackData data)；
    public ArrayList<TrackData> selectData(String date);
    public void deleteData(int id)；
    public void deleteData(TrackData data)；
    public void updateData(TrackData data)；
    /* 很多其他操作 */
}
```

在其他类中使用数据库：

```
DatabaseHelper db = new DatabaseHelper(context);
TrackData data = datas.get(position);
db.deleteData(data);
/* other operations */
db.close();
```

- **后台服务**

我们的 app 需要支持后台不断的定位，不断的新拍的照片添加到每一条的记录中。在 MainActivity 中我们开始一项服务：

```
startService(new Intent(getApplicationContext(), TrackService.class));
```

```
public class TrackService extends Service {
    private GeoCoder mSearch = GeoCoder.newInstance();
    private LocationListener locationListener = new LocationListener()

    /* some code here */

    Runnable task = new Runnable() {
        @Override
```

```
        public void run() {
          /* some code here */
        };
}
```

在 service 中需要调用百度 sdk 的接口，因为我们需要实时定位，并且在位置发生变化时在地图上打一个 marker，这样用户可以单机 marker 编辑记录信息。
run()函数中每 1 分钟调用 photoGetter 类，用于查看当前时间段内用户是否拍照，并且自动把拍照的照片添加进记录中去。

六. 整体评价

- **亮点：**

  基于地图应用，地图显示和操作友好，支持离线地图节省流量，后台自动记录轨迹，自动添加照片记录，用户也可以自定义轨迹信息。

- **缺点：**

  应用太过简单，没有实现计算里程，只是记录信息，没有为用户提供运动建议。

- **future work：**

  针对缺点，我们可以根据记录的坐标信息，计算用户运动量，并根据运动量动态为用户提供一些建议。另外，界面美观度也不太够，以后会学习使用一些高级的布局。

七. 小组分工与感悟

戴秉璋：报告书写，数据库类的继承封装，后台 service 的实现，主要基于 Baidu 地图 SDK 实现离线地图，自动定位，记录编辑，添加照片相关的 activity 的实现。
感悟：因为不是计算机系学生，所以编程基础可能略差一些。第一次写 android 程序，java 课也是这学期才学,也遇到了一些困难,通过向助教请教和上网查询都解决了。通过 android app 开发，熟悉了 android 开发的一些基本流程，activity 的生命周期等相关概念，也对 java 程序设计有了提高，对于 java 内部匿名类，多线程等有了进一步了解与熟悉。

android 程序最主要的就是生命周期的管理，startAvtivity，startActivityForResult 的区别，他们对于 activity 之间的跳转的区别，也花了一定时间才完全明白区别。

对于 context 的理解是一个难点，“极客学院”的课程为我提供了很大帮助。在不同的 activity 之间如何用 Intent 传递参数，对于可变数量的记录，如何用 ListView 正确展现数据，如何设置 adapter 等等，这些对于新手都是困难的，所以不停地试错是无法避免的。不过这些错误为后来相似的 activity 的设计提供了很好的借鉴。

最后，baidu SDK 的帮助文档很详细，提供的接口大多都有例子，所以在地图应用中得到了很好的体验。阅读官方文档也是在这项大作业中得到的锻炼之一。

陈雪儿：整体布局、UI 设计、APP 欢迎页面、录制视频
感悟：过对大量 APP 的分析，我对 UI 设计有了更深的理解，好的 UI 应该形成统一协调的风格：色彩不宜过多，否则看起来会显得凌乱；也不能过于鲜艳耀眼，否则会让用户感到不适。此外，图标的风格也应该一致，要么扁平化要么立体化，不能混搭，否则会不伦不类。因此，出于用户至上的设计原则，我决定使用淡蓝色和淡紫色作为 APP 的主色调，选择扁

平化的黑白图标。这些都只是一些设计理念，而实际编程上我还是遇到了一些问题，所幸在与队友的交流和上网查阅资料的过程中，我总算克服了这些困难。

APP 中每个页面的布局、动作、绘制的图案都是用 resource files 实现的，我一开始眼中只有整个页面，没有将布局、动作的概念分开，加上对 Android 的布局属性不够熟悉，一开始做有点无从下手。我就觉得 Android 插入图片、使用颜色的用法都挺奇怪的，非要用一个 value 的资源文件来定义颜色的值，后来我就习惯了，这样做其实挺像宏定义的，而且可读性也比较好。通过查阅资料我对 anim、drawable、layout、menu、values 这些 resource file 的使用有了一定了解之后，在实现起来就得心应手多了。而且 Android Studio 对 layout resource file 还提供了 palette，这个东西就像画图板一样，需要什么样的布局或者部件可以直接拖拽到模拟界面上；除此之外，value resource file 也像画图板一样可以直接在颜色编辑器上取色。这些功能无疑大大便利了开发者，Android Studio 虽然经常被我吐槽资源占用多、运行速度慢，在这一点上倒是挺人性化挺方便的。

在开发过程中我还遇到过制作的欢迎页面不跳转、APP 下载到真机上出现了两个图标等等小问题，最后都通过与队友的交流沟通或者自己上网查阅资料解决了。