编码文档

小组负责人：吴庆顺 - 201912172014282

小组成员：郭梦利 - 201912172014278

李璐璐 - 201912172014280

聂茜茜 - 201912172014281

袁　玥 - 201912172014283

姚恩广 - 201912172014291

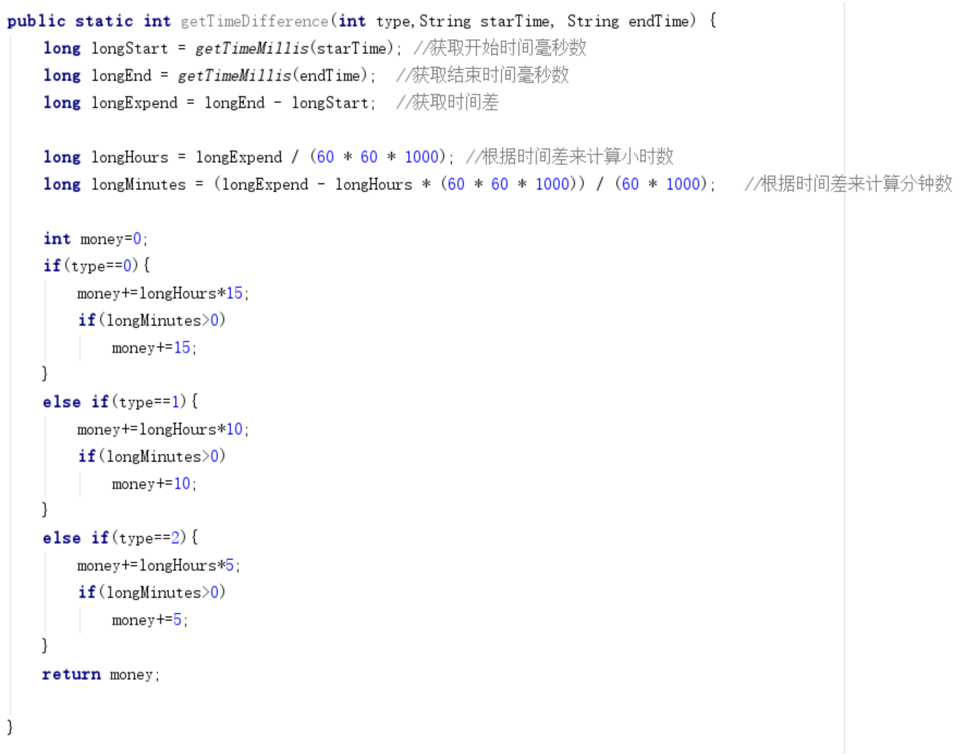
李宜飞 - 201912172014293

一. 语言及环境

- 使用的语言
  - java
- Openjdk 版本
  - openjdk-8-jdk
- 开发软件版本
  - Android Studio 3.2
- 操作系统
  - Window 10
- Android 内核版本
  - 5.0.2
- Android 内核版本
  - 5.0.2
- Android 编译版本
  - 25
- 使用最低 Sdk 版本
  - 15

二. 关键算法

由于本项目主要为在线服务器数据库内容的删改以及百度地图 API 的使用所以与算法部分关系甚微。此处仅介绍停车费计算的算法：



```java
public static int getTimeDifference(int type,String starTime, String endTime) {
    long longStart = getTimeMillis(starTime); //获取开始时间毫秒数
    long longEnd = getTimeMillis(endTime);  //获取结束时间毫秒数
    long longExpend = longEnd - longStart;  //获取时间差

    long longHours = longExpend / (60 * 60 * 1000); //根据时间差来计算小时数
    long longMinutes = (longExpend - longHours * (60 * 60 * 1000)) / (60 * 1000);   //根据时间差来计算分钟数

    int money=0;
    if(type==0){
        money+=longHours*15;
        if(longMinutes>0)
            money+=15;
    }
    else if(type==1){
        money+=longHours*10;
        if(longMinutes>0)
            money+=10;
    }
    else if(type==2){
        money+=longHours*5;
        if(longMinutes>0)
            money+=5;
    }
    return money;

}
```

三. 关键代码（仅显示部分功能的代码，全部代码在项目代码压缩包中）

3.1 周边停车场信息展示界面：

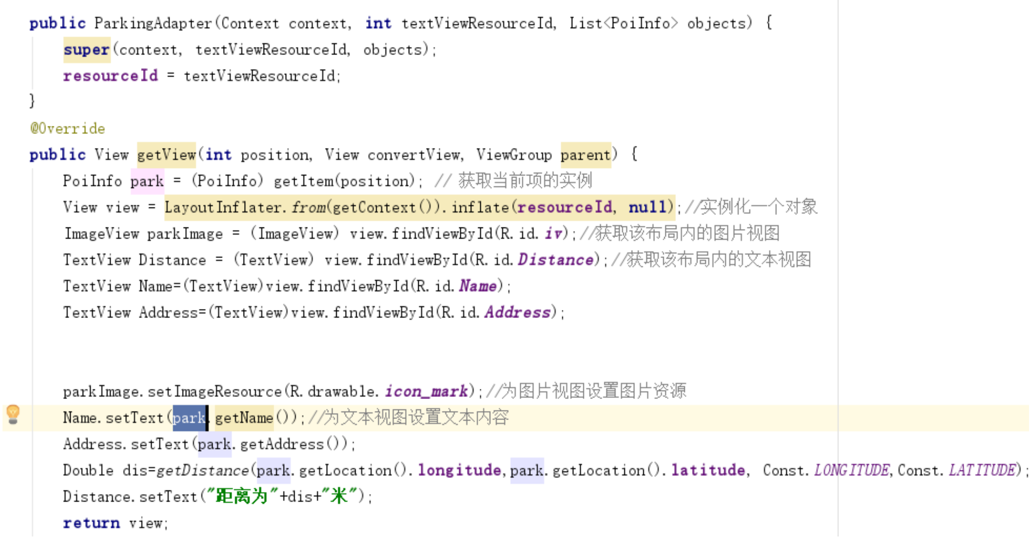
停车场信息展示界面采用 LinearLayout，并在其中布置 Listview 控件，单个停车场信息展示采用自绘制 listitem，具体步骤如下：

1） 定义一个实体类 park，作为 ListView 适配器的适配类型。

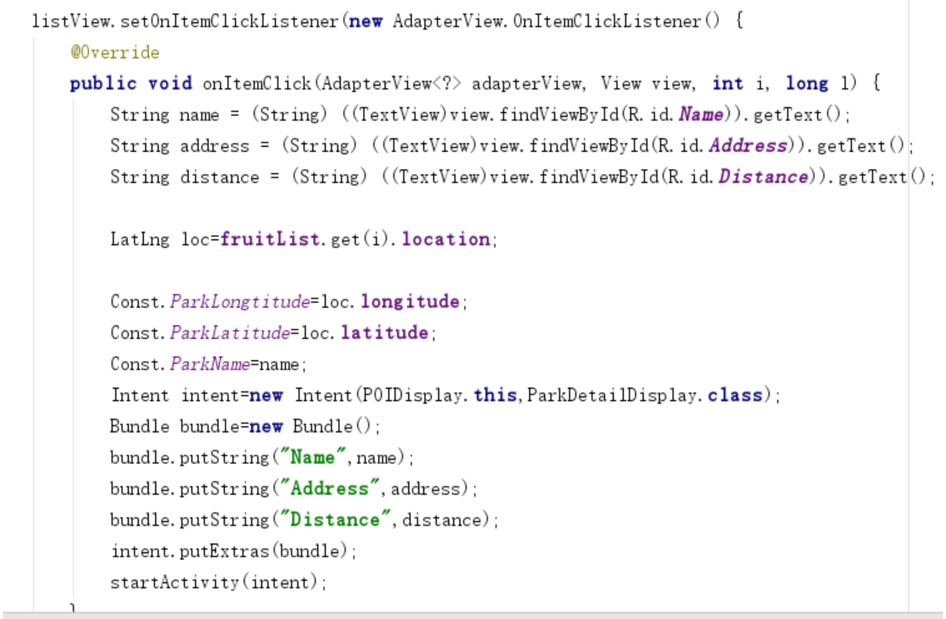
2） 为 ListView 的子项指定一个我们自定义的布局 park_item.xml。

3） 创建一个自定义的适配器 ParkingAdapter ，这个适配器继承自 ArrayAdapter。重写构造方法和 getView 方法。

4)在 MainActivity 中编写，初始化停车场数据



```
public ParkingAdapter(Context context, int textViewResourceId, List<PoiInfo> objects) {
    super(context, textViewResourceId, objects);
    resourceId = textViewResourceId;
}
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    PoiInfo park = (PoiInfo) getItem(position); // 获取当前项的实例
    View view = LayoutInflater.from(getContext()).inflate(resourceId, null);//实例化一个对象
    ImageView parkImage = (ImageView) view.findViewById(R.id.iv);//获取该布局内的图片视图
    TextView Distance = (TextView) view.findViewById(R.id.Distance);//获取该布局内的文本视图
    TextView Name=(TextView)view.findViewById(R.id.Name);
    TextView Address=(TextView)view.findViewById(R.id.Address);


    parkImage.setImageResource(R.drawable.icon_mark);//为图片视图设置图片资源
    Name.setText(park.getName());//为文本视图设置文本内容
    Address.setText(park.getAddress());
    Double dis=getDistance(park.getLocation().longitude,park.getLocation().latitude, Const.LONGITUDE,Const.LATITUDE);
    Distance.setText("距离为"+dis+"米");
    return view;
```



```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        String name = (String) ((TextView)view.findViewById(R.id.Name)).getText();
        String address = (String) ((TextView)view.findViewById(R.id.Address)).getText();
        String distance = (String) ((TextView)view.findViewById(R.id.Distance)).getText();

        LatLng loc=fruitList.get(i).location;

        Const.ParkLongtitude=loc.longitude;
        Const.ParkLatitude=loc.latitude;
        Const.ParkName=name;
        Intent intent=new Intent(POIDisplay.this,ParkDetailDisplay.class);
        Bundle bundle=new Bundle();
        bundle.putString("Name",name);
        bundle.putString("Address",address);
        bundle.putString("Distance",distance);
        intent.putExtras(bundle);
        startActivity(intent);
    }
```

3.2 系统主界面及系统主界面侧滑栏

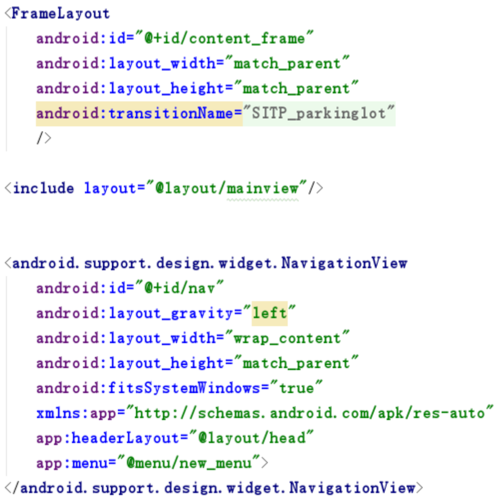
系统主界面包含三个按钮及一个侧滑栏,三个按钮均为圆形按钮,使用自绘制按钮模式,侧滑栏实现步骤:

1. 主布局文件中创建 DrawerLayout 根元素布局

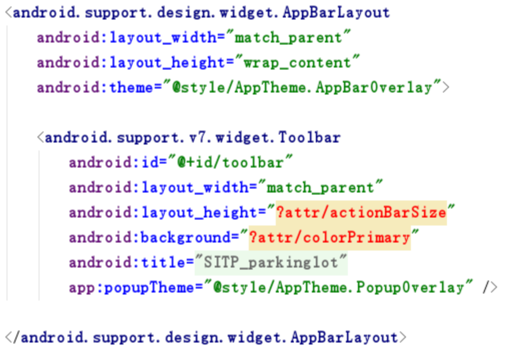
2. 创建侧滑菜单 ListView 的 Adapter

3. 为 DrawerLayout 添加菜单开、关状态监听器，为的是能够动态的改变 ActionBar 中的显示内容

相关代码实现如下:



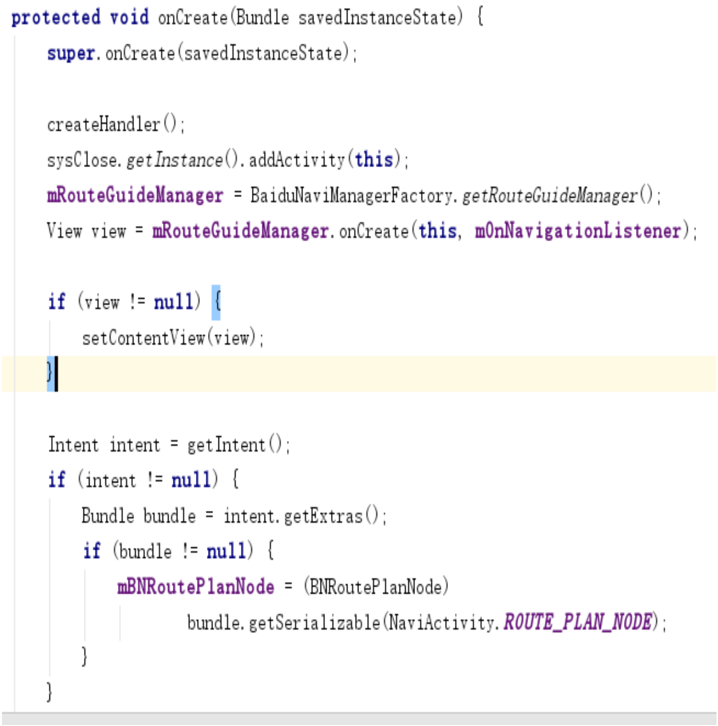
```xml
<FrameLayout
    android:id="@+id/content_frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:transitionName="SITP_parkinglot"
    />

<include layout="@layout/mainview"/>


<android.support.design.widget.NavigationView
    android:id="@+id/nav"
    android:layout_gravity="left"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:headerLayout="@layout/head"
    app:menu="@menu/new_menu">
</android.support.design.widget.NavigationView>
```



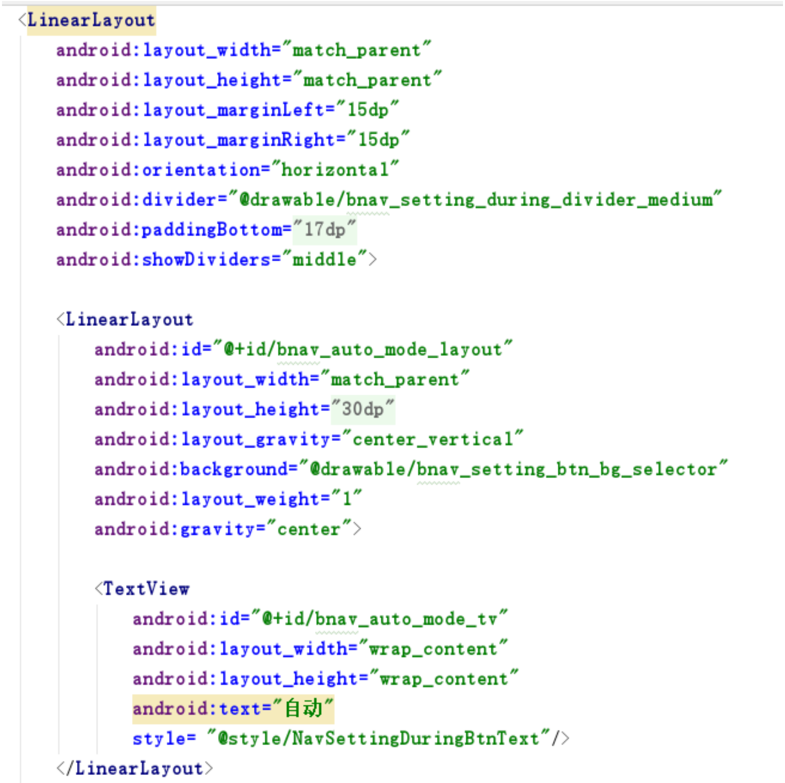
```xml
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:title="SITP_parkinglot"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>
```

3.3 停车场导航界面

停车场导航界面的实现为使用百度地图SDK中的mRouteGuideManager显示导航全景图，创建EventDialog.xml显示路线信息以及行车提示，具体代码(部分)如下：



```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    createHandler();
    sysClose.getInstance().addActivity(this);
    mRouteGuideManager = BaiduNaviManagerFactory.getRouteGuideManager();
    View view = mRouteGuideManager.onCreate(this, mOnNavigationListener);

    if (view != null) {
        setContentView(view);
    }

    Intent intent = getIntent();
    if (intent != null) {
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            mBNRoutePlanNode = (BNRoutePlanNode)
                    bundle.getSerializable(NaviActivity.ROUTE_PLAN_NODE);
        }
    }
```



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:orientation="horizontal"
    android:divider="@drawable/bnav_setting_during_divider_medium"
    android:paddingBottom="17dp"
    android:showDividers="middle">

    <LinearLayout
        android:id="@+id/bnav_auto_mode_layout"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/bnav_setting_btn_bg_selector"
        android:layout_weight="1"
        android:gravity="center">

        <TextView
            android:id="@+id/bnav_auto_mode_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="自动"
            style= "@style/NavSettingDuringBtnText"/>
    </LinearLayout>
```

除此之外导航界面还需要设置功能选择的下端停靠栏，也就是 Event_Dialog 其布局如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="320dp"
    android:layout_height="260dp"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center" >
    <ImageView
        android:id="@+id/enlarge_view_img"
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:visibility="gone"/>
    <LinearLayout
        android:id="@+id/route_guide_ll"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="center">
        <LinearLayout
            android:id="@+id/time_distance_ll"
            android:layout_width="match_parent"
            android:layout_height="30dp"
            android:layout_gravity="center"
            android:orientation="horizontal"
            android:gravity="center">
            <TextView
                android:id="@+id/remain_time_tx"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:layout_marginRight="10dp"
                android:text="明日到达"
                />
            <TextView
                android:id="@+id/remain_distance_tx"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="剩 10 公里"
                />
            <TextView
                android:id="@+id/current_speed_tx"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="100km/h"
                android:textSize="14sp"
                android:layout_marginLeft="10dp"
                android:gravity="center"
                />
        </LinearLayout>
        <RelativeLayout
            android:id="@+id/simple_guide_rl"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            >
            <LinearLayout
                android:id="@+id/simple_go_ll"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_centerHorizontal="true"
                >
                <ImageView
                    android:id="@+id/turn_img"
                    android:layout_width="30dp"
                    android:layout_height="30dp"
                    android:background="@drawable/navi_guide_turn"
                    />
                <TextView
                    android:id="@+id/remain_distance"
                    android:layout_width="wrap_content"
                    android:layout_height="30dp"
                    android:layout_toRightOf="@id/turn_img"
                    android:gravity="center"
                    android:layout_marginLeft="8dp"
                    android:text="100 米"
                    />
                <TextView
                    android:id="@+id/go_tx"
                    android:layout_width="wrap_content"
                    android:layout_height="30dp"
                    android:layout_toRightOf="@id/remain_distance"
                    android:layout_marginLeft="8dp"
                    android:gravity="center"
                    android:text="去往"
                    />
            </LinearLayout>
            <TextView
                android:id="@+id/next_road_tx"
                android:layout_width="wrap_content"
                android:layout_height="20dp"
                android:layout_marginTop="20dp"
                android:layout_centerHorizontal="true"
                android:layout_below="@id/simple_go_ll"
                android:text="上地信息路"
                />
        </RelativeLayout>
        <RelativeLayout
            android:id="@+id/along_rl"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            >
            <LinearLayout
                android:id="@+id/along_meter_ll"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:orientation="horizontal">
                <TextView
                    android:id="@+id/along_wrods_tx"
                    android:layout_width="wrap_content"
                    android:layout_height="30dp"
                    android:text="顺行" />
                <TextView
                    android:id="@+id/along_meters_tx"
                    android:layout_width="wrap_content"
                    android:layout_height="30dp"
                    android:layout_marginLeft="30dp"
                    android:text="600 米"
                    />
            </LinearLayout>
            <TextView
                android:id="@+id/current_road_tx"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerHorizontal="true"
                android:layout_below="@id/along_meter_ll"
                android:text="来广营东路"
                />
        </RelativeLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="10dp"
            android:layout_gravity="center"
            >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14sp"
                android:text="GPS 定位:"
                />
            <TextView
                android:id="@+id/loacte_tx"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="14sp"
                android:text="未定位"
                />
            />
        </LinearLayout>
    </LinearLayout>
```

```
</LinearLayout>
```

导航界面功能选择界面，包含选择“导航视角”“日夜模式”“路况条”“全览小窗”“地图图层选择”：

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  ~ Copyright (C) 2017 Baidu, Inc. All Rights Reserved.
  -->
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    >
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <RelativeLayout
            android:id="@+id/nav_view_select_layout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:layout_gravity="center_vertical"
            android:background="@drawable/nav_settings_middle_selector"
            android:baselineAligned="false"
            android:clickable="false"
            android:gravity="center_vertical">

            <TextView
                android:id="@+id/nav_view_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="12dp"
                android:layout_marginBottom="16dp"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="导航视角"
                />
        </RelativeLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@drawable/nav_settings_middle_selector"
            android:orientation="horizontal">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:orientation="horizontal"
    android:paddingBottom="@dimen/during_setting_btn_bottom_line_margin"
    android:divider="@drawable/bnav_setting_during_divider_medium"
    android:showDividers="middle">

    <LinearLayout
        android:id="@+id/bnav_view_car3d_layout"
        android:layout_width="match_parent"
        android:layout_height="@dimen/during_setting_btn_height"
        android:background="@drawable/bnav_setting_btn_bg_selector"
        android:layout_gravity="center_vertical"
        android:layout_weight="1"
        android:gravity="center">

        <TextView
            android:id="@+id/bnav_view_car3d_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="跟随车头"
            style= "@style/NavSettingDuringBtnText"/>
    </LinearLayout>

    <LinearLayout
        android:id="@+id/bnav_view_north2d_layout"
        android:layout_width="match_parent"
        android:layout_height="@dimen/during_setting_btn_height"
        android:background="@drawable/bnav_setting_btn_bg_selector"
        android:layout_gravity="center_vertical"
        android:layout_weight="1"
        android:gravity="center">

        <TextView
            android:id="@+id/bnav_view_north2d_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:text="正北朝上"
            style= "@style/NavSettingDuringBtnText"
```

```
                    />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>

    <ImageView style="@style/NavSettingDuringDivider"/>

    <RelativeLayout
        android:id="@+id/nav_day_night_mode_select_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:baselineAligned="false"
        android:clickable="false"
        android:gravity="center_vertical">

        <TextView
            android:id="@+id/nav_day_night_mode_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="12dp"
            android:layout_marginBottom="16dp"
            android:layout_marginLeft="15dp"
            android:textAppearance="@style/NavCommonListItemTitleText"
            android:text="日夜模式"
            />
    </RelativeLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/nav_settings_middle_selector"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="15dp"
            android:layout_marginRight="15dp"
            android:orientation="horizontal"
            android:divider="@drawable/bnav_setting_during_divider_medium"
            android:paddingBottom="@dimen/during_setting_btn_bottom_line_margin"
            android:showDividers="middle">
```

```
<LinearLayout
    android:id="@+id/bnav_auto_mode_layout"
    android:layout_width="match_parent"
    android:layout_height="@dimen/during_setting_btn_height"
    android:layout_gravity="center_vertical"
    android:background="@drawable/bnav_setting_btn_bg_selector"
    android:layout_weight="1"
    android:gravity="center">

    <TextView
        android:id="@+id/bnav_auto_mode_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="自动"
        style= "@style/NavSettingDuringBtnText"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/bnav_day_mode_layout"
    android:layout_width="match_parent"
    android:layout_height="@dimen/during_setting_btn_height"
    android:layout_gravity="center_vertical"
    android:background="@drawable/bnav_setting_btn_bg_selector"
    android:layout_weight="1"
    android:gravity="center">

    <TextView
        android:id="@+id/bnav_day_mode_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="白天"
        style= "@style/NavSettingDuringBtnText"/>
</LinearLayout>

<LinearLayout
    android:id="@+id/bnav_night_mode_layout"
    android:layout_width="match_parent"
    android:layout_height="@dimen/during_setting_btn_height"
    android:layout_gravity="center_vertical"
    android:background="@drawable/bnav_setting_btn_bg_selector"
    android:layout_weight="1"
    android:gravity="center">
```

```
            <TextView
                android:id="@+id/bnav_night_mode_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="夜间"
                style= "@style/NavSettingDuringBtnText"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>

<ImageView style="@style/NavSettingDuringDivider"/>

<RelativeLayout
    android:id="@+id/nav_display_select_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:background="@drawable/nav_settings_middle_selector"
    android:baselineAligned="false"
    android:clickable="false"
    android:gravity="center_vertical">

    <TextView
        android:id="@+id/nav_display_mode_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="15dp"
        android:textAppearance="@style/NavCommonListItemTitleText"
        android:text="导航中图面显示"
        />
</RelativeLayout>

<LinearLayout
    android:id="@+id/nav_remaining_road_condition_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center_vertical"
    android:paddingBottom="@dimen/during_setting_btn_bottom_line_margin"
    android:background="@drawable/nav_settings_middle_selector"
    android:gravity="center_vertical">
    <LinearLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="15dp"
        android:layout_marginRight="15dp"
        android:orientation="horizontal"
        android:divider="@drawable/bnav_setting_during_divider_medium"
        android:showDividers="middle">
        <LinearLayout
            android:id="@+id/bnav_display_overview_mode_layout"
            android:layout_width="match_parent"
            android:layout_height="@dimen/during_setting_btn_height"
            android:orientation="vertical"
            android:background="@drawable/bnav_setting_btn_bg_selector"
            android:layout_weight="1"
            android:gravity="center">
            <TextView
                android:id="@+id/nav_display_overview_mode_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                style= "@style/NavSettingDuringBtnText"
                android:gravity="center"
                android:text="全览小窗"/>
        </LinearLayout>
        <LinearLayout
            android:id="@+id/bnav_display_road_cond_mode_layout"
            android:layout_width="match_parent"
            android:layout_height="@dimen/during_setting_btn_height"
            android:orientation="vertical"
            android:layout_weight="1"
            android:background="@drawable/bnav_setting_btn_bg_selector"
            android:gravity="center">
            <TextView
                android:id="@+id/nav_display_road_condition_mode_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                style= "@style/NavSettingDuringBtnText"
                android:gravity="center"
                android:text="路况条"/>

        </LinearLayout>
    </LinearLayout>
</LinearLayout>
<ImageView style="@style/NavSettingDuringDivider"/>
<RelativeLayout
```

```
        android:id="@+id/nav_scale_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="center_vertical"
            >
            <TextView
                android:id="@+id/nav_scale_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="导航比例尺智能缩放"
                />
            <TextView
                android:id="@+id/nav_scale_tips_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:textAppearance="@style/NavCommonListItemTipsText"
                android:paddingLeft="15dp"
                android:gravity="center_vertical"
                android:text="导航中地图自动放大或缩小"
                />
        </LinearLayout>
        <ImageView
            android:id="@+id/nav_scale_cb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_marginRight="15dp"
            android:layout_centerVertical="true"
            android:contentDescription="@null"
            android:src="@drawable/nav_checkout_icon"/>
    </RelativeLayout>
    <RelativeLayout
        android:id="@+id/nav_multi_route_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="center_vertical"
            >
            <TextView
                android:id="@+id/nav_multi_route_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="多路线推荐"
                />

            <TextView
                android:id="@+id/nav_multi_route_tips_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:textAppearance="@style/NavCommonListItemTipsText"
                android:paddingLeft="15dp"
                android:gravity="center_vertical"
                android:text="开启或关闭多路线推荐"
                />
        </LinearLayout>
        <ImageView
            android:id="@+id/nav_multi_route_cb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_marginRight="15dp"
            android:layout_centerVertical="true"
            android:contentDescription="@null"
            android:src="@drawable/nav_checkout_icon"/>
    </RelativeLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:visibility="gone">
        <Button
            android:id="@+id/set_diy_car_logo"
            android:layout_width="0dp"
            android:layout_height="25dp"
            android:textSize="10dp"
            android:text="设置自定义车标"
            android:textColor="@color/bnav_setting_btn_text_selector"
            android:background="@drawable/bnav_setting_btn_bg_selector"
            android:layout_weight="1"/>
        <Button
            android:id="@+id/set_diy_start_logo"
            android:layout_width="0dp"
            android:layout_height="25dp"
            android:textSize="10dp"
            android:text="设置自定义起点"
            android:textColor="@color/bnav_setting_btn_text_selector"
            android:background="@drawable/bnav_setting_btn_bg_selector"
            android:layout_weight="1"/>
        <Button
            android:id="@+id/set_diy_end_logo"
            android:layout_width="0dp"
            android:layout_height="25dp"
            android:textSize="10dp"
            android:text="设置自定义终点"
            android:textColor="@color/bnav_setting_btn_text_selector"
            android:background="@drawable/bnav_setting_btn_bg_selector"
            android:layout_weight="1"/>
    </LinearLayout>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:layout_marginBottom="16dp"
        android:layout_marginLeft="15dp"
        android:textAppearance="@style/NavCommonListItemTitleText"
        android:text="专业导航设置功能入口控制"
        />
    <!-- bottombar open -->
    <RelativeLayout
        android:id="@+id/nav_bottombar_open_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="center_vertical"
            >
            <TextView
                android:id="@+id/nav_bottombar_open_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="底部栏展开控制"
                />
            <TextView
                android:id="@+id/nav_bottombar_open_tips_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:textAppearance="@style/NavCommonListItemTipsText"
                android:paddingLeft="15dp"
                android:gravity="center_vertical"
                android:text="控制底部栏是否可以展开，总控制入口"
                />
        </LinearLayout>
        <ImageView
            android:id="@+id/nav_bottombar_open_cb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_marginRight="15dp"
            android:layout_centerVertical="true"
            android:contentDescription="@null"
            android:src="@drawable/nav_checkout_icon"/>
    </RelativeLayout>
    <!-- more settings enable -->
    <RelativeLayout
        android:id="@+id/nav_more_settings_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="center_vertical"
            >
            <TextView
                android:id="@+id/nav_more_settings_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="更多设置功能入口控制"
                />
            <TextView
                android:id="@+id/nav_more_settings_tips_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:textAppearance="@style/NavCommonListItemTipsText"
                android:paddingLeft="15dp"
                android:gravity="center_vertical"
                android:text="控制底部 bar 展开后是否展示更多设置入口"
                />
        </LinearLayout>
        <ImageView
            android:id="@+id/nav_more_settings_cb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_marginRight="15dp"
            android:layout_centerVertical="true"
            android:contentDescription="@null"
            android:src="@drawable/nav_checkout_icon"/>
    </RelativeLayout>
    <!-- 路线偏好 -->
    <RelativeLayout
        android:id="@+id/nav_route_sort_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="center_vertical"
            >
            <TextView
                android:id="@+id/nav_route_sort_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:textAppearance="@style/NavCommonListItemTitleText"
                android:text="底部栏路线偏好入口控制"
                />

            <TextView
                android:id="@+id/nav_route_sort_tips_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:textAppearance="@style/NavCommonListItemTipsText"
                android:paddingLeft="15dp"
                android:gravity="center_vertical"
                android:text="控制是否展示底部栏的偏好入口"
                />
        </LinearLayout>
        <ImageView
            android:id="@+id/nav_route_sort_cb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_marginRight="15dp"
            android:layout_centerVertical="true"
            android:contentDescription="@null"
            android:src="@drawable/nav_checkout_icon"/>
    </RelativeLayout>
    <!-- 沿途检索 -->
    <RelativeLayout
        android:id="@+id/nav_route_search_layout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_gravity="center_vertical"
        android:background="@drawable/nav_settings_middle_selector"
        android:gravity="center_vertical">
```

```
            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:orientation="vertical"
                android:gravity="center_vertical"
               >
                <TextView
                    android:id="@+id/nav_route_search_tv"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="15dp"
                    android:textAppearance="@style/NavCommonListItemTitleText"
                    android:text="沿途检索功能入口控制"
                    />
                <TextView
                    android:id="@+id/nav_route_search_tips_tv"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:paddingTop="5dp"
                    android:textAppearance="@style/NavCommonListItemTipsText"
                    android:paddingLeft="15dp"
                    android:gravity="center_vertical"
                    android:text="控制是否展示沿途检索功能入口"
                    />
            </LinearLayout>
            <ImageView
                android:id="@+id/nav_route_search_cb"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentRight="true"
                android:layout_marginRight="15dp"
                android:layout_centerVertical="true"
                android:contentDescription="@null"
                android:src="@drawable/nav_checkout_icon"/>
        </RelativeLayout>
    </LinearLayout>
</ScrollView>
```

3.4 预约信息展示界面

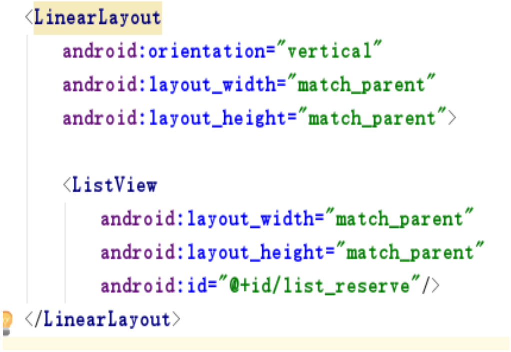
预约信息展示界面采用 LinearLayout，并在其中布置 Listview 控件，单个预约信息展示采用自绘制 listitem，具体步骤如下:

1）定义一个实体类 reserve，作为 ListView 适配器的适配类型。

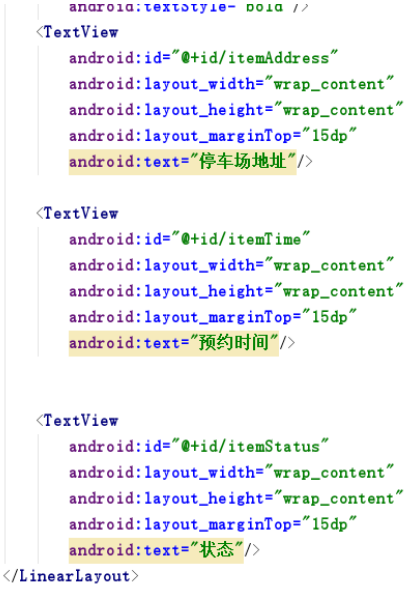
2）为 ListView 的子项指定一个我们自定义的布局 reserve_item.xml。

3）创建一个自定义的适配器 ReserveAdapter ，这个适配器继承自 ArrayAdapter。重写构造方法和 getView 方法。

4)在 MainActivity 中编写，初始化预约数据



```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/list_reserve"/>
</LinearLayout>
```



```
        android:textStyle="bold"/>
    <TextView
        android:id="@+id/itemAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:text="停车场地址"/>

    <TextView
        android:id="@+id/itemTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:text="预约时间"/>


    <TextView
        android:id="@+id/itemStatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:text="状态"/>
</LinearLayout>
```

3.5 模块间调用关系映射

(1)主模块调用周边停车场查询模块、预约停车信息查询模块、取车模块

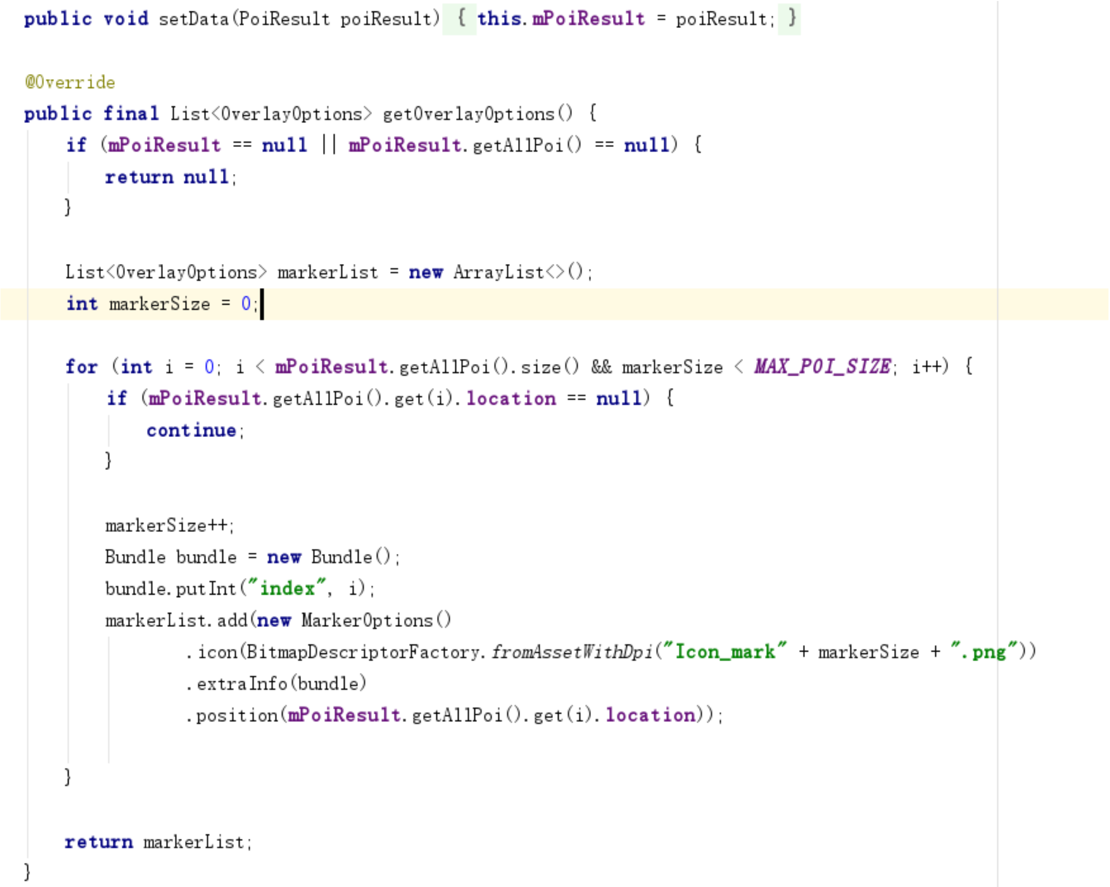
(2)侧滑栏模块调用个人信息查询模块、修改密码模块、账户余额查询模块、反馈意见模块、关于我们模块、注销账号模块

(3)账户余额模块调用账户余额充值模块

(4)周边停车场查询模块调用停车场分布查询模块、停车场导航模块、存车模块

(5)取车模块调用停车信息查询模块、停车场导航模块、支付模块、充值模块

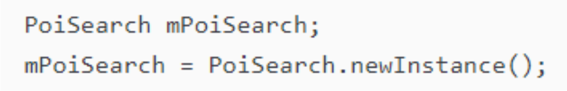
3.6 周边停车场展示功能



```
public void setData(PoiResult poiResult) { this.mPoiResult = poiResult; }

@Override
public final List<OverlayOptions> getOverlayOptions() {
    if (mPoiResult == null || mPoiResult.getAllPoi() == null) {
        return null;
    }

    List<OverlayOptions> markerList = new ArrayList<>();
    int markerSize = 0;

    for (int i = 0; i < mPoiResult.getAllPoi().size() && markerSize < MAX_POI_SIZE; i++) {
        if (mPoiResult.getAllPoi().get(i).location == null) {
            continue;
        }

        markerSize++;
        Bundle bundle = new Bundle();
        bundle.putInt("index", i);
        markerList.add(new MarkerOptions()
                .icon(BitmapDescriptorFactory.fromAssetWithDpi("Icon_mark" + markerSize + ".png"))
                .extraInfo(bundle)
                .position(mPoiResult.getAllPoi().get(i).location));

    }

    return markerList;
}
```

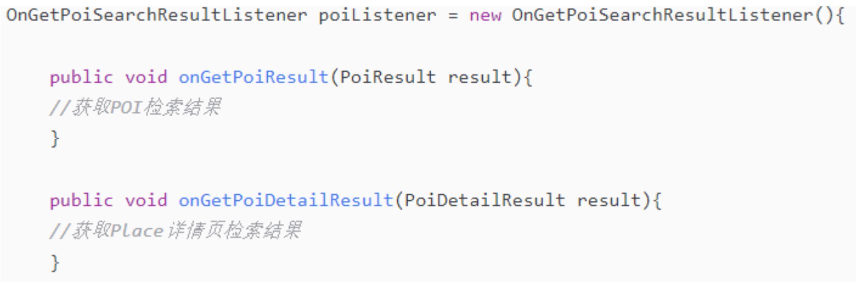
使用百度地图SDK进行区域范围poi检索得到附近停车场数据随后以listwidget展示获取的poi 数据，具体步骤如下:

1.创建POI检索实例



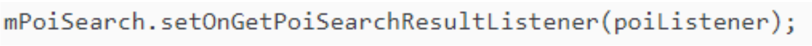
```
PoiSearch mPoiSearch;
mPoiSearch = PoiSearch.newInstance();
```

2.创建POI检索监听者



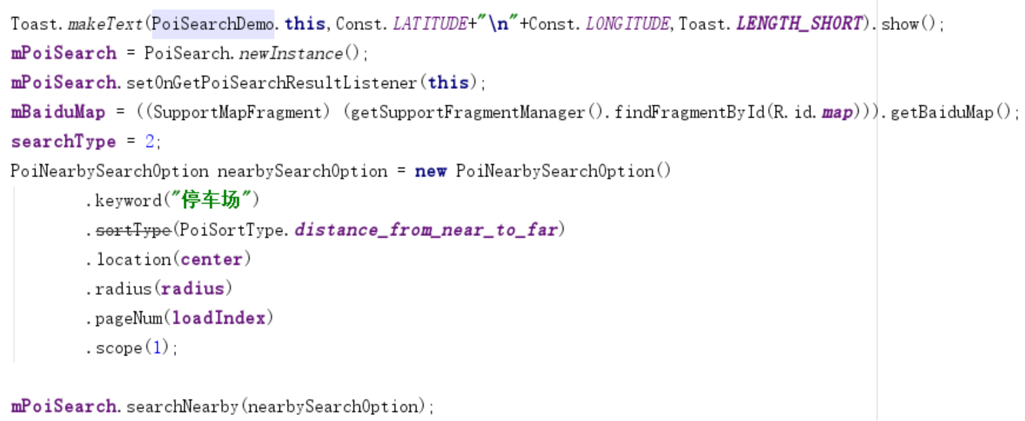
```
OnGetPoiSearchResultListener poiListener = new OnGetPoiSearchResultListener(){

    public void onGetPoiResult(PoiResult result){
    //获取POI检索结果
    }

    public void onGetPoiDetailResult(PoiDetailResult result){
    //获取Place详情页检索结果
    }
```

3.设置POI检索监听者



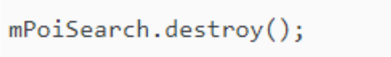
```
mPoiSearch.setOnGetPoiSearchResultListener(poiListener);
```

4.发起检索请求



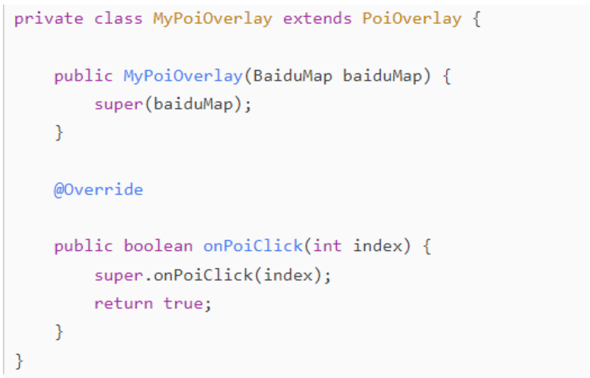
```
Toast.makeText(PoiSearchDemo.this,Const.LATITUDE+"\n"+Const.LONGITUDE,Toast.LENGTH_SHORT).show();
mPoiSearch = PoiSearch.newInstance();
mPoiSearch.setOnGetPoiSearchResultListener(this);
mBaiduMap = ((SupportMapFragment) (getSupportFragmentManager().findFragmentById(R.id.map))).getBaiduMap();
searchType = 2;
PoiNearbySearchOption nearbySearchOption = new PoiNearbySearchOption()
        .keyword("停车场")
        .~~sortType~~(PoiSortType.distance_from_near_to_far)
        .location(center)
        .radius(radius)
        .pageNum(loadIndex)
        .scope(1);

mPoiSearch.searchNearby(nearbySearchOption);
```

5.释放 POI 检索实例



```
mPoiSearch.destroy();
```

6.显示检索结果覆盖物



```
private class MyPoiOverlay extends PoiOverlay {

    public MyPoiOverlay(BaiduMap baiduMap) {
        super(baiduMap);
    }

    @Override

    public boolean onPoiClick(int index) {
        super.onPoiClick(index);
        return true;
    }
}
```

3.7 停车场导航功能

集成百度地图 SDK 进行导航的具体步骤如下:

1.初始化地图

2.开启定位图层，打开 gps 并设置坐标类型

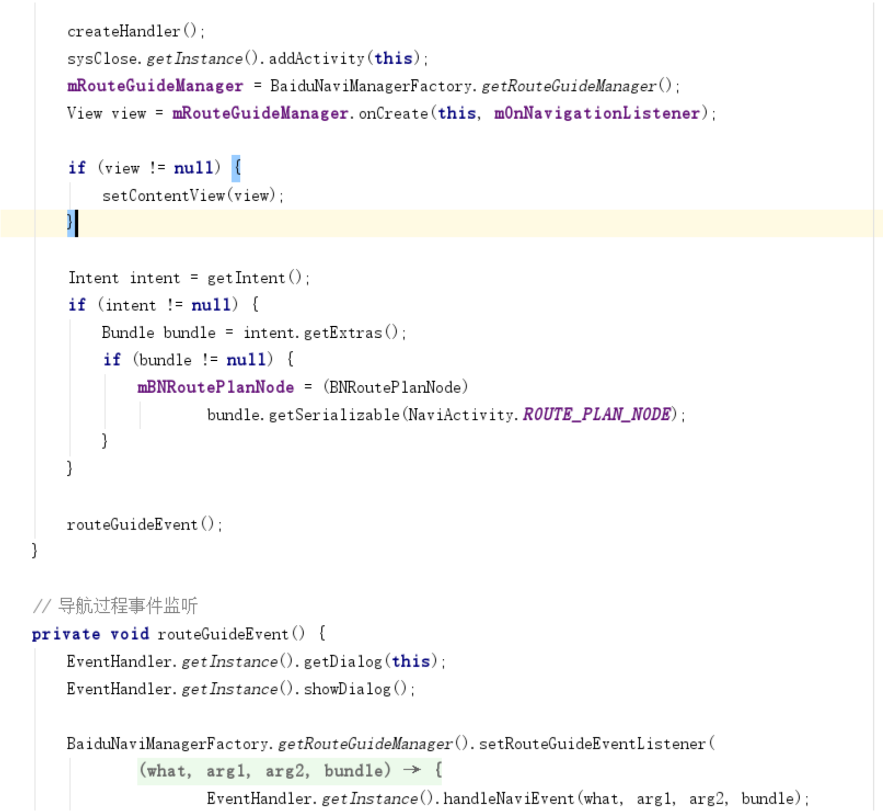
3.重置浏览节点的路线数据，对起点终点城市进行正确的设定

4.设置节点索引，并获取结果信息,定义定位 SDK 监听函数

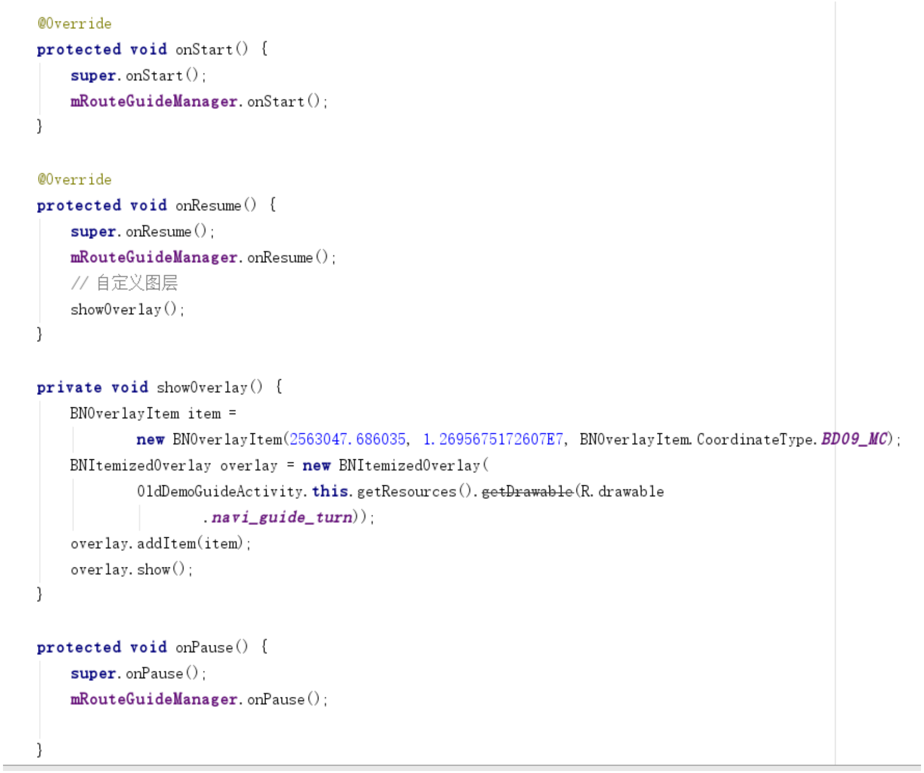
5.切换路线图标，刷新地图使其生效，起终点图标使用中心对齐.

6.起终点或途经点地址有岐义，通过接口获取建议查询信息

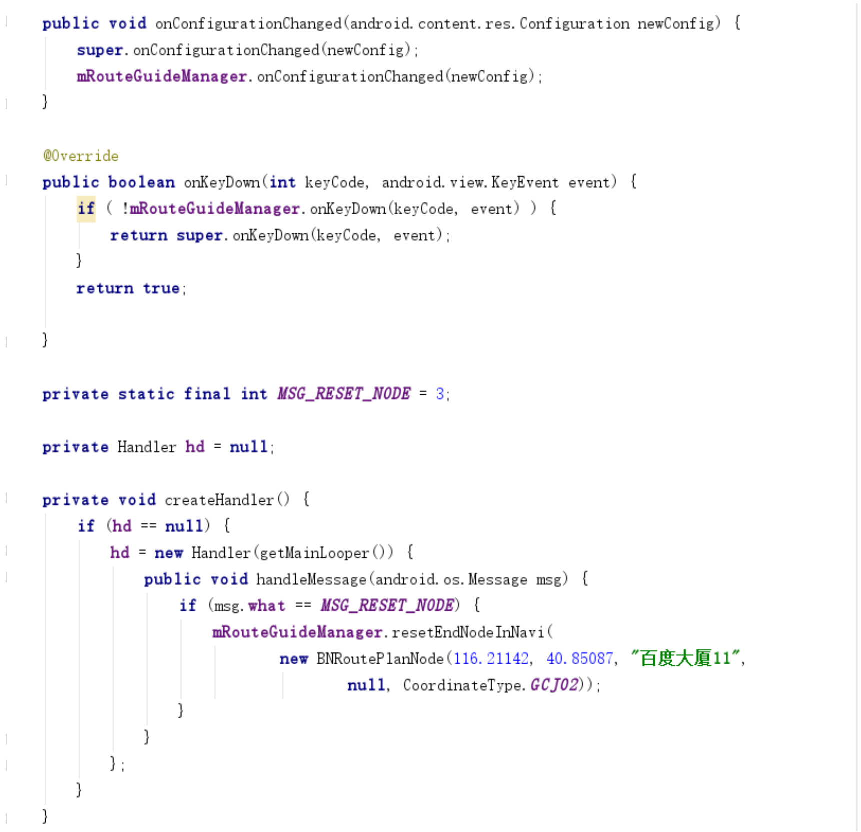
具体实现代码如下:



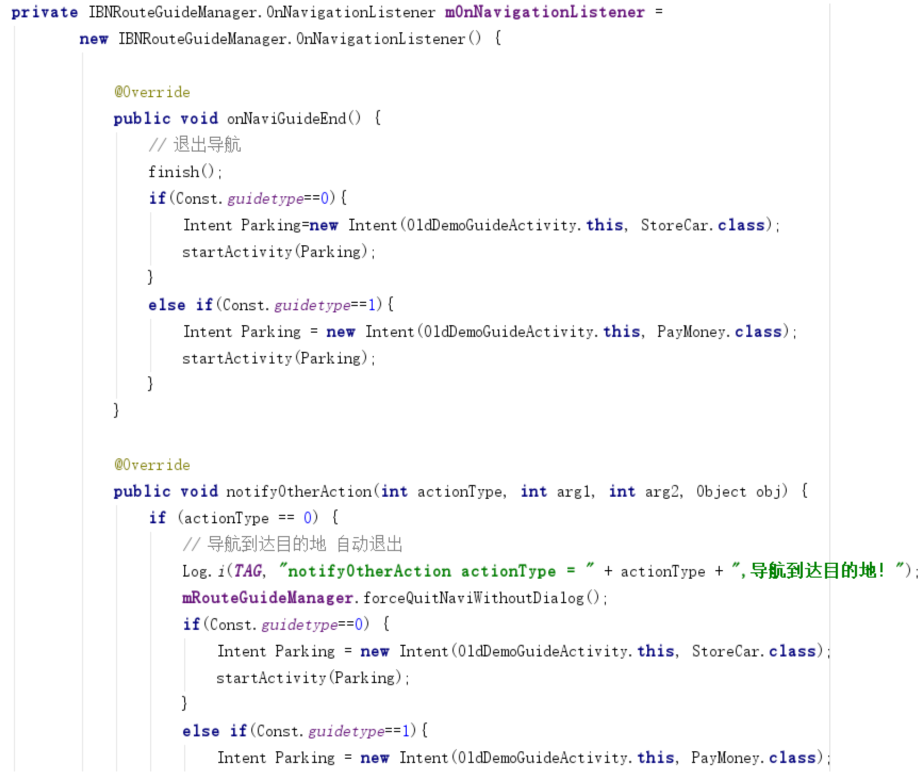
```
    createHandler();
    sysClose.getInstance().addActivity(this);
    mRouteGuideManager = BaiduNaviManagerFactory.getRouteGuideManager();
    View view = mRouteGuideManager.onCreate(this, mOnNavigationListener);

    if (view != null) {
        setContentView(view);
    }

    Intent intent = getIntent();
    if (intent != null) {
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            mBNRoutePlanNode = (BNRoutePlanNode)
                    bundle.getSerializable(NaviActivity.ROUTE_PLAN_NODE);
        }
    }

    routeGuideEvent();
}

// 导航过程事件监听
private void routeGuideEvent() {
    EventHandler.getInstance().getDialog(this);
    EventHandler.getInstance().showDialog();

    BaiduNaviManagerFactory.getRouteGuideManager().setRouteGuideEventListener(
            (what, arg1, arg2, bundle) → {
                EventHandler.getInstance().handleNaviEvent(what, arg1, arg2, bundle);
```



```
@Override
protected void onStart() {
    super.onStart();
    mRouteGuideManager.onStart();
}

@Override
protected void onResume() {
    super.onResume();
    mRouteGuideManager.onResume();
    // 自定义图层
    showOverlay();
}

private void showOverlay() {
    BNOverlayItem item =
            new BNOverlayItem(2563047.686035, 1.2695675172607E7, BNOverlayItem.CoordinateType.BD09_MC);
    BNItemizedOverlay overlay = new BNItemizedOverlay(
            OldDemoGuideActivity.this.getResources().~~getDrawable~~(R.drawable
                    .navi_guide_turn));
    overlay.addItem(item);
    overlay.show();
}

protected void onPause() {
    super.onPause();
    mRouteGuideManager.onPause();

}
```

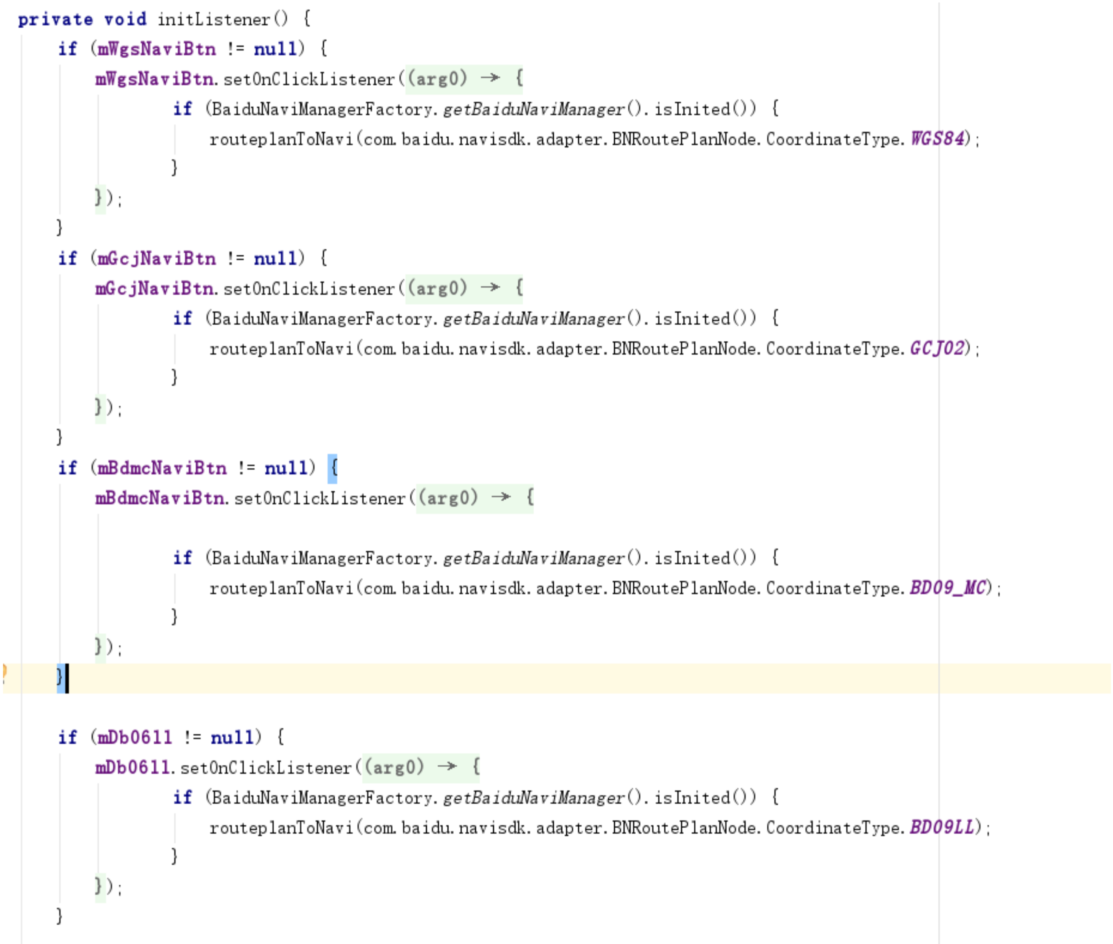


```
public void onConfigurationChanged(android.content.res.Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    mRouteGuideManager.onConfigurationChanged(newConfig);
}

@Override
public boolean onKeyDown(int keyCode, android.view.KeyEvent event) {
    if ( !mRouteGuideManager.onKeyDown(keyCode, event) ) {
        return super.onKeyDown(keyCode, event);
    }
    return true;

}

private static final int MSG_RESET_NODE = 3;

private Handler hd = null;

private void createHandler() {
    if (hd == null) {
        hd = new Handler(getMainLooper()) {
            public void handleMessage(android.os.Message msg) {
                if (msg.what == MSG_RESET_NODE) {
                    mRouteGuideManager.resetEndNodeInNavi(
                            new BNRoutePlanNode(116.21142, 40.85087, "百度大厦11",
                                    null, CoordinateType.GCJ02));
                }
            }
        };
    }
}
```



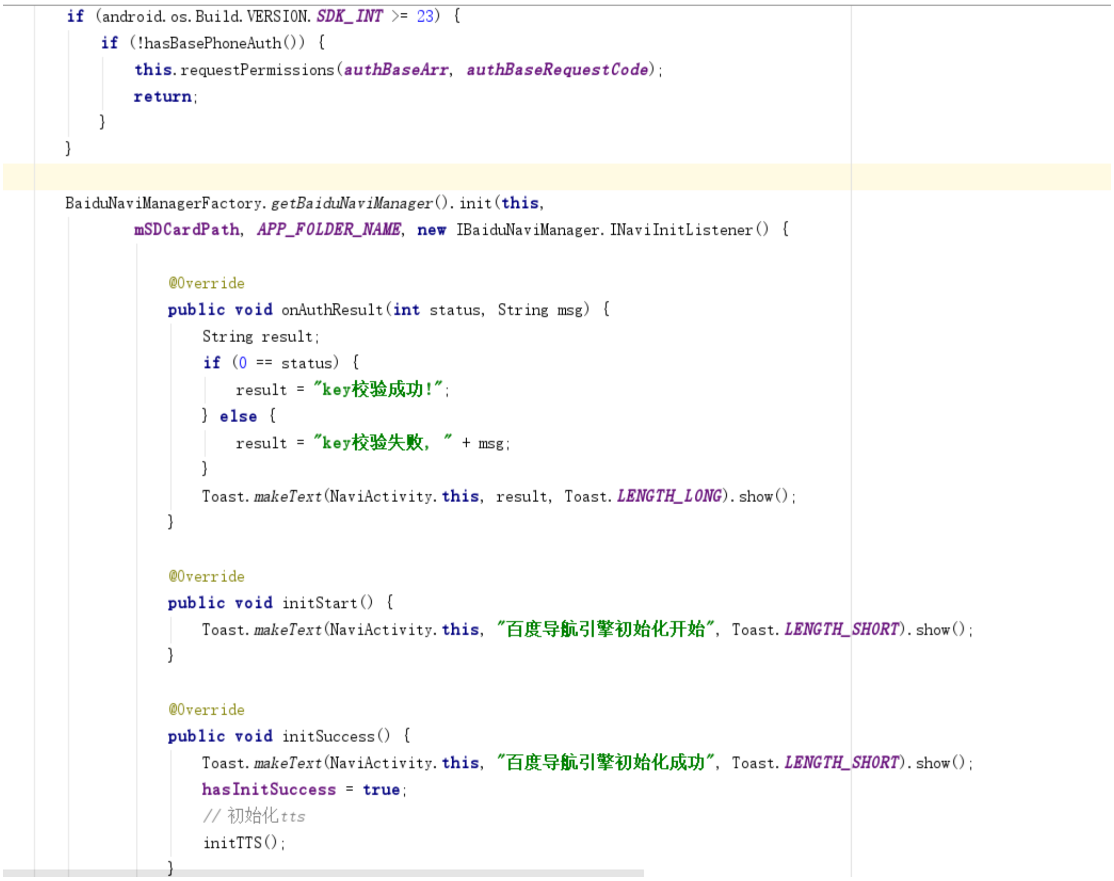
```
private IBNRouteGuideManager.OnNavigationListener mOnNavigationListener =
        new IBNRouteGuideManager.OnNavigationListener() {

            @Override
            public void onNaviGuideEnd() {
                //退出导航
                finish();
                if(Const.guidetype==0){
                    Intent Parking=new Intent(OldDemoGuideActivity.this, StoreCar.class);
                    startActivity(Parking);
                }
                else if(Const.guidetype==1){
                    Intent Parking = new Intent(OldDemoGuideActivity.this, PayMoney.class);
                    startActivity(Parking);
                }
            }

            @Override
            public void notifyOtherAction(int actionType, int arg1, int arg2, Object obj) {
                if (actionType == 0) {
                    //导航到达目的地 自动退出
                    Log.i(TAG, "notifyOtherAction actionType = " + actionType + ",导航到达目的地！");
                    mRouteGuideManager.forceQuitNaviWithoutDialog();
                    if(Const.guidetype==0) {
                        Intent Parking = new Intent(OldDemoGuideActivity.this, StoreCar.class);
                        startActivity(Parking);
                    }
                    else if(Const.guidetype==1){
                        Intent Parking = new Intent(OldDemoGuideActivity.this, PayMoney.class);
```

设置启用的百度地图 API 中的坐标类型：



```
private void initListener() {
    if (mWgsNaviBtn != null) {
        mWgsNaviBtn.setOnClickListener((arg0) -> {
                if (BaiduNaviManagerFactory.getBaiduNaviManager().isInited()) {
                    routeplanToNavi(com.baidu.navisdk.adapter.BNRoutePlanNode.CoordinateType.WGS84);
                }
        });
    }
    if (mGcjNaviBtn != null) {
        mGcjNaviBtn.setOnClickListener((arg0) -> {
                if (BaiduNaviManagerFactory.getBaiduNaviManager().isInited()) {
                    routeplanToNavi(com.baidu.navisdk.adapter.BNRoutePlanNode.CoordinateType.GCJ02);
                }
        });
    }
    if (mBdmcNaviBtn != null) {
        mBdmcNaviBtn.setOnClickListener((arg0) -> {

                if (BaiduNaviManagerFactory.getBaiduNaviManager().isInited()) {
                    routeplanToNavi(com.baidu.navisdk.adapter.BNRoutePlanNode.CoordinateType.BD09_MC);
                }
        });
    }

    if (mDb0611 != null) {
        mDb0611.setOnClickListener((arg0) -> {
                if (BaiduNaviManagerFactory.getBaiduNaviManager().isInited()) {
                    routeplanToNavi(com.baidu.navisdk.adapter.BNRoutePlanNode.CoordinateType.BD09LL);
                }
        });
    }
```

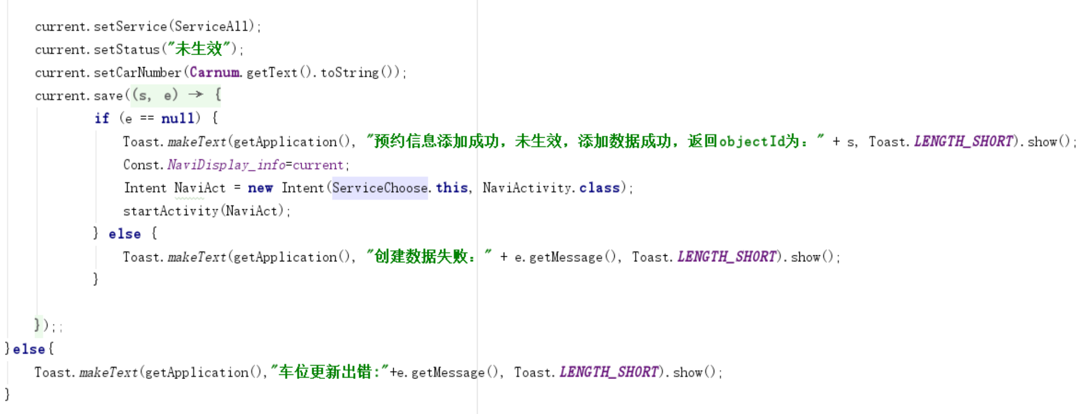
百度地图引擎初始化，进行开发者 Key 校验(本应用申请的百度地图开发者 Key 为 ndendL4vseT20HbHo64DhiltBfYjmlTT)：



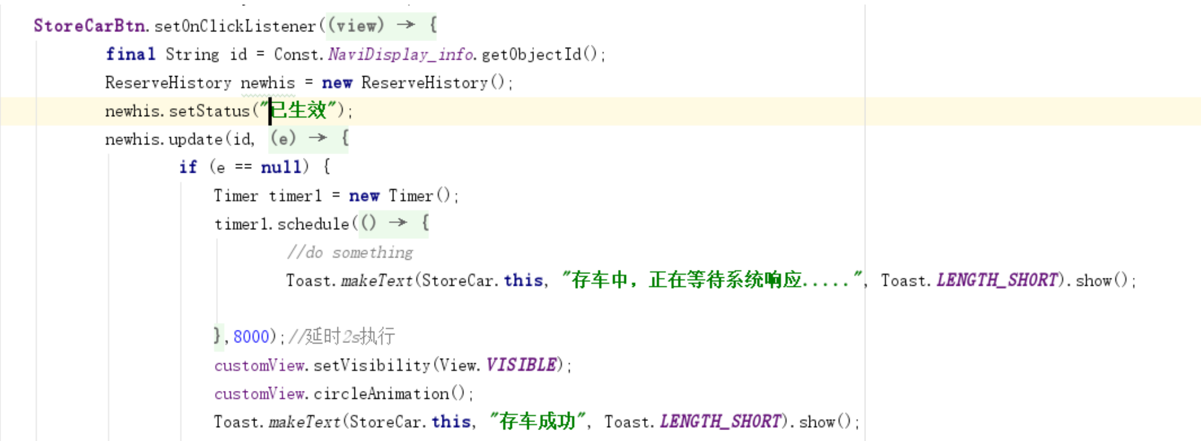
```
    if (android.os.Build.VERSION.SDK_INT >= 23) {
        if (!hasBasePhoneAuth()) {
            this.requestPermissions(authBaseArr, authBaseRequestCode);
            return;
        }
    }

    BaiduNaviManagerFactory.getBaiduNaviManager().init(this,
            mSDCardPath, APP_FOLDER_NAME, new IBaiduNaviManager.INaviInitListener() {

                @Override
                public void onAuthResult(int status, String msg) {
                    String result;
                    if (0 == status) {
                        result = "key校验成功!";
                    } else {
                        result = "key校验失败, " + msg;
                    }
                    Toast.makeText(NaviActivity.this, result, Toast.LENGTH_LONG).show();
                }

                @Override
                public void initStart() {
                    Toast.makeText(NaviActivity.this, "百度导航引擎初始化开始", Toast.LENGTH_SHORT).show();
                }

                @Override
                public void initSuccess() {
                    Toast.makeText(NaviActivity.this, "百度导航引擎初始化成功", Toast.LENGTH_SHORT).show();
                    hasInitSuccess = true;
                    // 初始化tts
                    initTTS();
                }
```

3.8 停车场服务预约功能

停车场服务预约功能为通过创建预约信息对象，并将预约信息对象写入云端数据库并修改对应停车场剩余车位信息，如果车位不足则显示预约失败，车位不足等提示信息，在到达目的地之后，将预约信息修改为“已生效”从而实现预约服务功能，具体实现代码如下：



```
    current.setService(ServiceAll);
    current.setStatus("未生效");
    current.setCarNumber(Carnum.getText().toString());
    current.save((s, e) -> {
            if (e == null) {
                Toast.makeText(getApplication(), "预约信息添加成功，未生效，添加数据成功，返回objectId为：" + s, Toast.LENGTH_SHORT).show();
                Const.NaviDisplay_info=current;
                Intent NaviAct = new Intent(ServiceChoose.this, NaviActivity.class);
                startActivity(NaviAct);
            } else {
                Toast.makeText(getApplication(), "创建数据失败：" + e.getMessage(), Toast.LENGTH_SHORT).show();
            }

    });;
}else{
    Toast.makeText(getApplication(),"车位更新出错:"+e.getMessage(), Toast.LENGTH_SHORT).show();
}
```

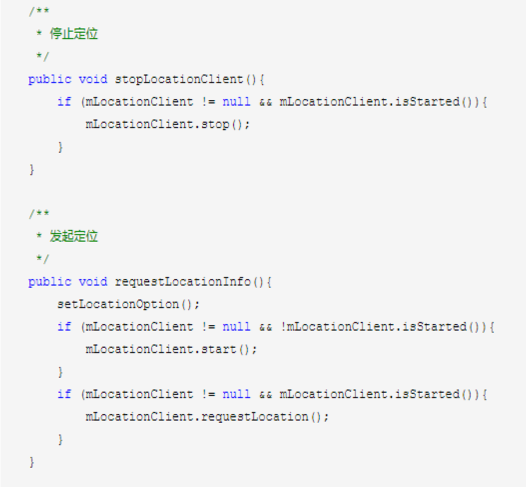


```
StoreCarBtn.setOnClickListener((view) → {
        final String id = Const.NaviDisplay_info.getObjectId();
        ReserveHistory newhis = new ReserveHistory();
        newhis.setStatus("已生效");
        newhis.update(id, (e) → {
                if (e == null) {
                    Timer timer1 = new Timer();
                    timer1.schedule(() → {
                            //do something
                            Toast.makeText(StoreCar.this, "存车中，正在等待系统响应.....", Toast.LENGTH_SHORT).show();

                    },8000);//延时2s执行
                    customView.setVisibility(View.VISIBLE);
                    customView.circleAnimation();
                    Toast.makeText(StoreCar.this, "存车成功", Toast.LENGTH_SHORT).show();
```

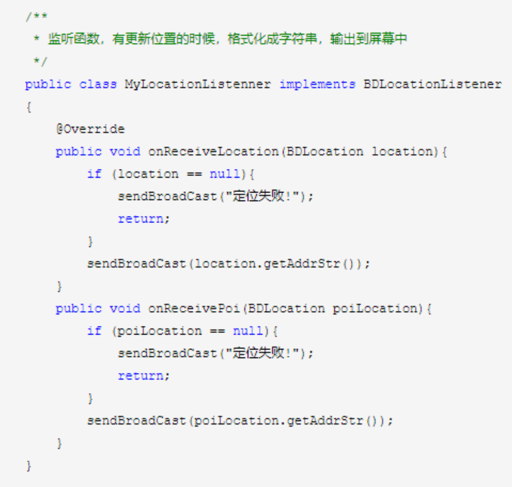
3.9 实时定位功能

集成百度地图进行实时定位获取定位信息，具体步骤如下：

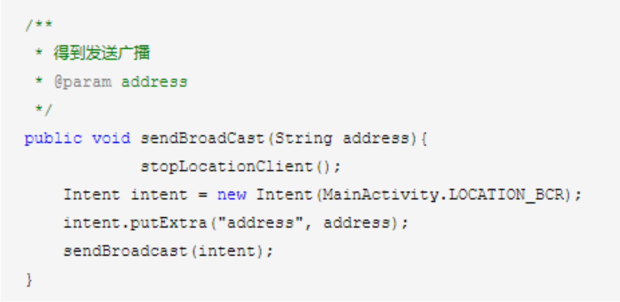
1. 获取定位信息——requestLocationInfo()
2. 通过广播发送位置信息——sendBroadCast()
3. 停止定位——stopLocationClient()



```
/**
 * 停止定位
 */
public void stopLocationClient(){
    if (mLocationClient != null && mLocationClient.isStarted()){
        mLocationClient.stop();
    }
}

/**
 * 发起定位
 */
public void requestLocationInfo(){
    setLocationOption();
    if (mLocationClient != null && !mLocationClient.isStarted()){
        mLocationClient.start();
    }
    if (mLocationClient != null && mLocationClient.isStarted()){
        mLocationClient.requestLocation();
    }
}
```



```
/**
 * 监听函数，有更新位置的时候，格式化成字符串，输出到屏幕中
 */
public class MyLocationListenner implements BDLocationListener
{
    @Override
    public void onReceiveLocation(BDLocation location){
        if (location == null){
            sendBroadCast("定位失败!");
            return;
        }
        sendBroadCast(location.getAddrStr());
    }
    public void onReceivePoi(BDLocation poiLocation){
        if (poiLocation == null){
            sendBroadCast("定位失败!");
            return;
        }
        sendBroadCast(poiLocation.getAddrStr());
    }
}
```

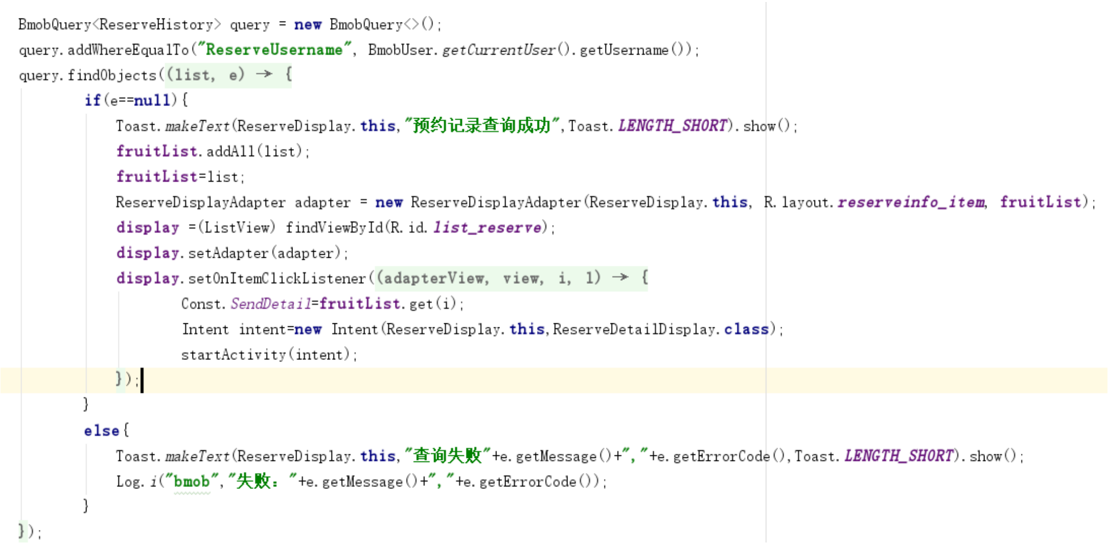


```
/**
 * 得到发送广播
 * @param address
 */
public void sendBroadCast(String address){
            stopLocationClient();
    Intent intent = new Intent(MainActivity.LOCATION_BCR);
    intent.putExtra("address", address);
    sendBroadcast(intent);
}
```

3.10 预约信息查询/个人信息查询/修改密码/充值

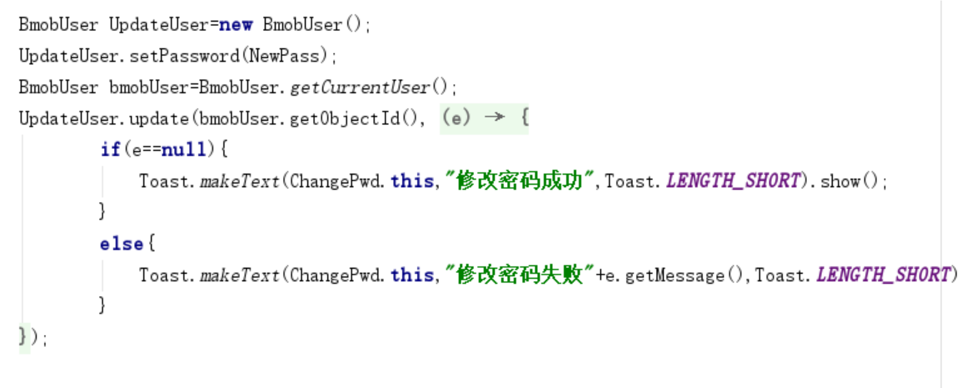
预约信息查询/个人信息查询/修改密码均为使用 Bmob 云端数据库进行数据交互的部分，所以此处合并为一项进行解释：

以预约信息查询为例，首先创建 BmobQuery 对象，以 BmobQuery 中封装的 addWhereEqualTo 函数对预约信息进行筛选从而获取当前登录用户的预约信息，具体代码如下：



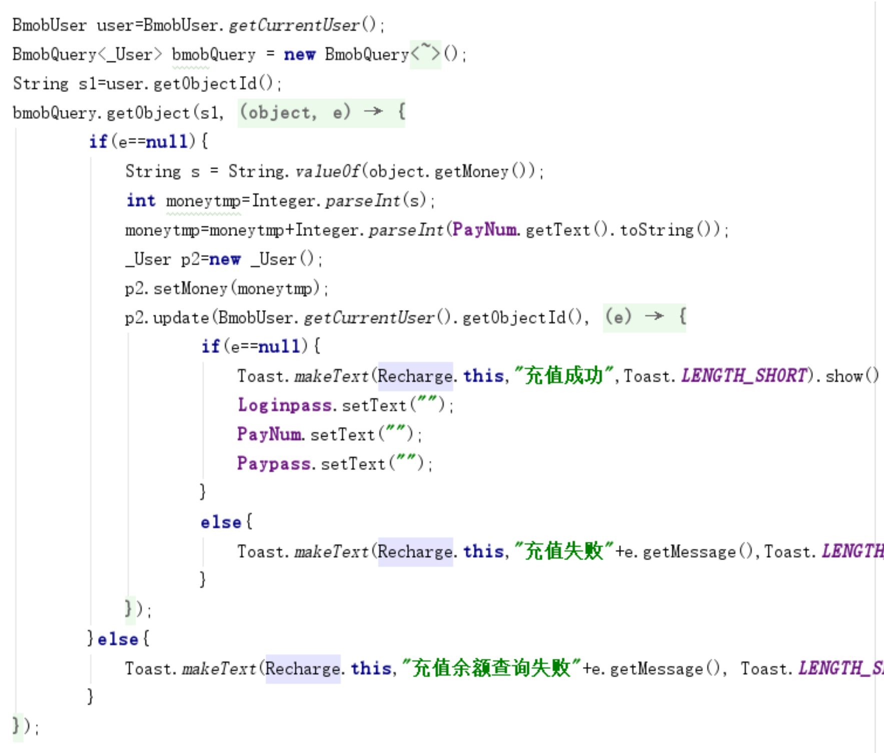
```
BmobQuery<ReserveHistory> query = new BmobQuery<>();
query.addWhereEqualTo("ReserveUsername", BmobUser.getCurrentUser().getUsername());
query.findObjects((list, e) -> {
        if(e==null){
            Toast.makeText(ReserveDisplay.this,"预约记录查询成功",Toast.LENGTH_SHORT).show();
            fruitList.addAll(list);
            fruitList=list;
            ReserveDisplayAdapter adapter = new ReserveDisplayAdapter(ReserveDisplay.this, R.layout.reserveinfo_item, fruitList);
            display =(ListView) findViewById(R.id.list_reserve);
            display.setAdapter(adapter);
            display.setOnItemClickListener((adapterView, view, i, l) -> {
                    Const.SendDetail=fruitList.get(i);
                    Intent intent=new Intent(ReserveDisplay.this,ReserveDetailDisplay.class);
                    startActivity(intent);
            });
        }
        else{
            Toast.makeText(ReserveDisplay.this,"查询失败"+e.getMessage()+","+e.getErrorCode(),Toast.LENGTH_SHORT).show();
            Log.i("bmob","失败："+e.getMessage()+","+e.getErrorCode());
        }
});
```

而修改用户密码则是通过 update(objectId,updateListener)方法来更新



```
BmobUser UpdateUser=new BmobUser();
UpdateUser.setPassword(NewPass);
BmobUser bmobUser=BmobUser.getCurrentUser();
UpdateUser.update(bmobUser.getObjectId(), (e) -> {
        if(e==null){
            Toast.makeText(ChangePwd.this,"修改密码成功",Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(ChangePwd.this,"修改密码失败"+e.getMessage(),Toast.LENGTH_SHORT).show();
        }
});
```

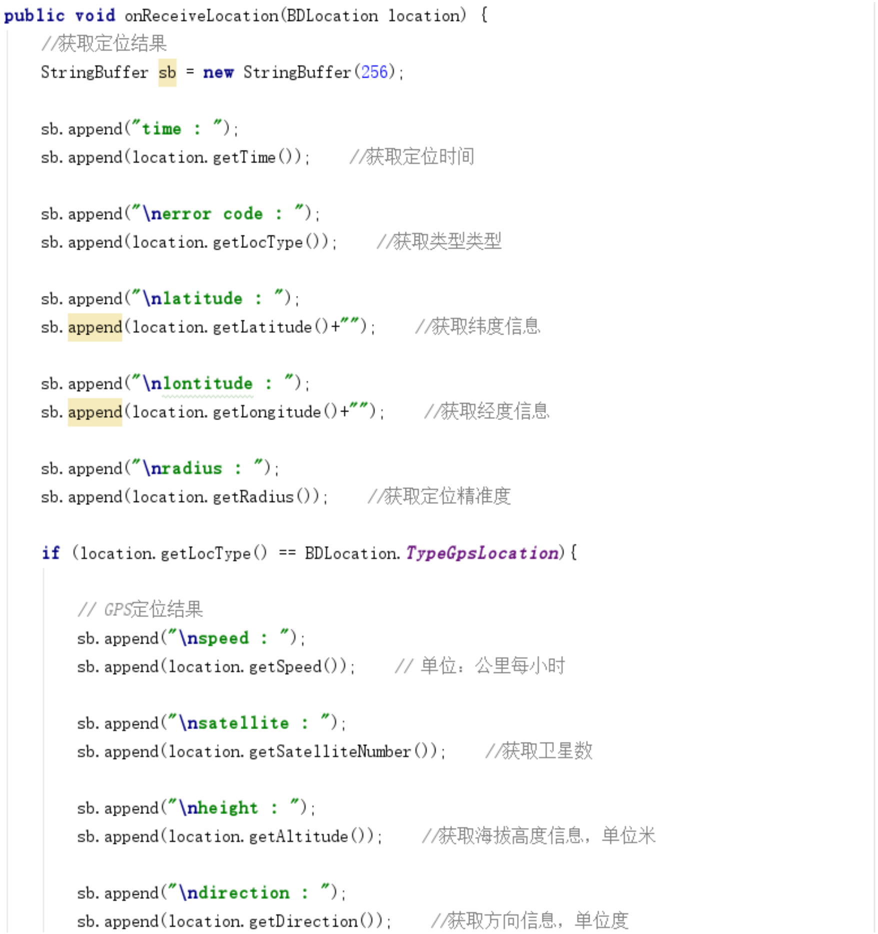
充值也是类似实现：



```
BmobUser user=BmobUser.getCurrentUser();
BmobQuery<_User> bmobQuery = new BmobQuery<~>();
String s1=user.getObjectId();
bmobQuery.getObject(s1, (object, e) -> {
        if(e==null){
            String s = String.valueOf(object.getMoney());
            int moneytmp=Integer.parseInt(s);
            moneytmp=moneytmp+Integer.parseInt(PayNum.getText().toString());
            _User p2=new _User();
            p2.setMoney(moneytmp);
            p2.update(BmobUser.getCurrentUser().getObjectId(), (e) -> {
                    if(e==null){
                        Toast.makeText(Recharge.this,"充值成功",Toast.LENGTH_SHORT).show();
                        Loginpass.setText("");
                        PayNum.setText("");
                        Paypass.setText("");
                    }
                    else{
                        Toast.makeText(Recharge.this,"充值失败"+e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
            });
        }else{
            Toast.makeText(Recharge.this,"充值余额查询失败"+e.getMessage(), Toast.LENGTH_SHORT).show();
        }
});
```

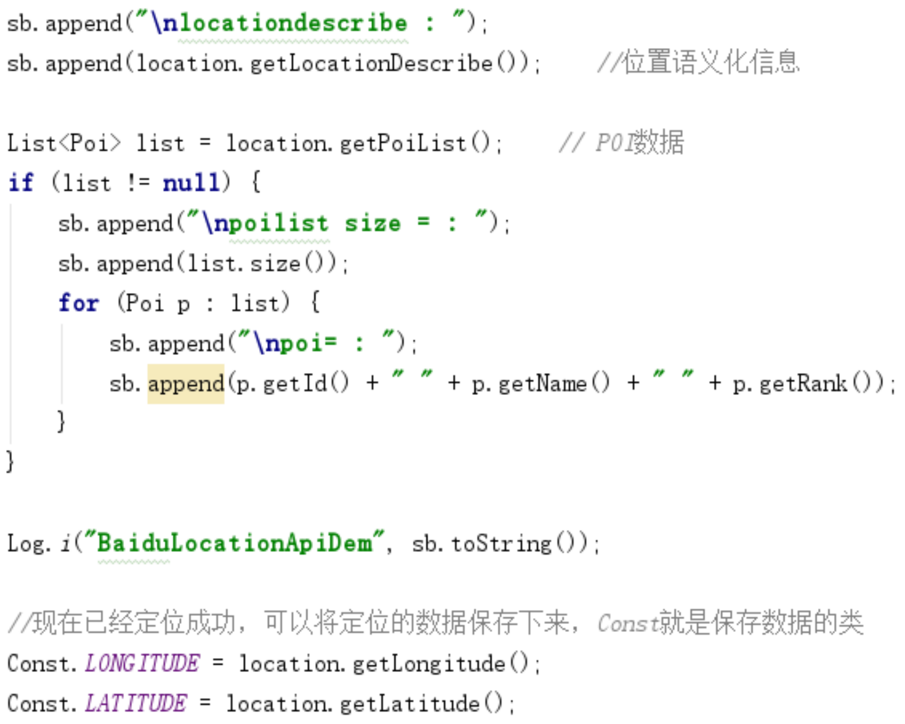
3.11 后台 GPS 监听函数

获取返回的 json 数据，进行分解获取其中的经纬度，时间等信息，供给给导航模块使用：



```
public void onReceiveLocation(BDLocation location) {
    //获取定位结果
    StringBuffer sb = new StringBuffer(256);

    sb.append("time : ");
    sb.append(location.getTime());    //获取定位时间

    sb.append("\nerror code : ");
    sb.append(location.getLocType());    //获取类型类型

    sb.append("\nlatitude : ");
    sb.append(location.getLatitude()+"");    //获取纬度信息

    sb.append("\nlontitude : ");
    sb.append(location.getLongitude()+"");    //获取经度信息

    sb.append("\nradius : ");
    sb.append(location.getRadius());    //获取定位精准度

    if (location.getLocType() == BDLocation.TypeGpsLocation){

        // GPS定位结果
        sb.append("\nspeed : ");
        sb.append(location.getSpeed());    // 单位：公里每小时

        sb.append("\nsatellite : ");
        sb.append(location.getSatelliteNumber());    //获取卫星数

        sb.append("\nheight : ");
        sb.append(location.getAltitude());    //获取海拔高度信息，单位米

        sb.append("\ndirection : ");
        sb.append(location.getDirection());    //获取方向信息，单位度
```

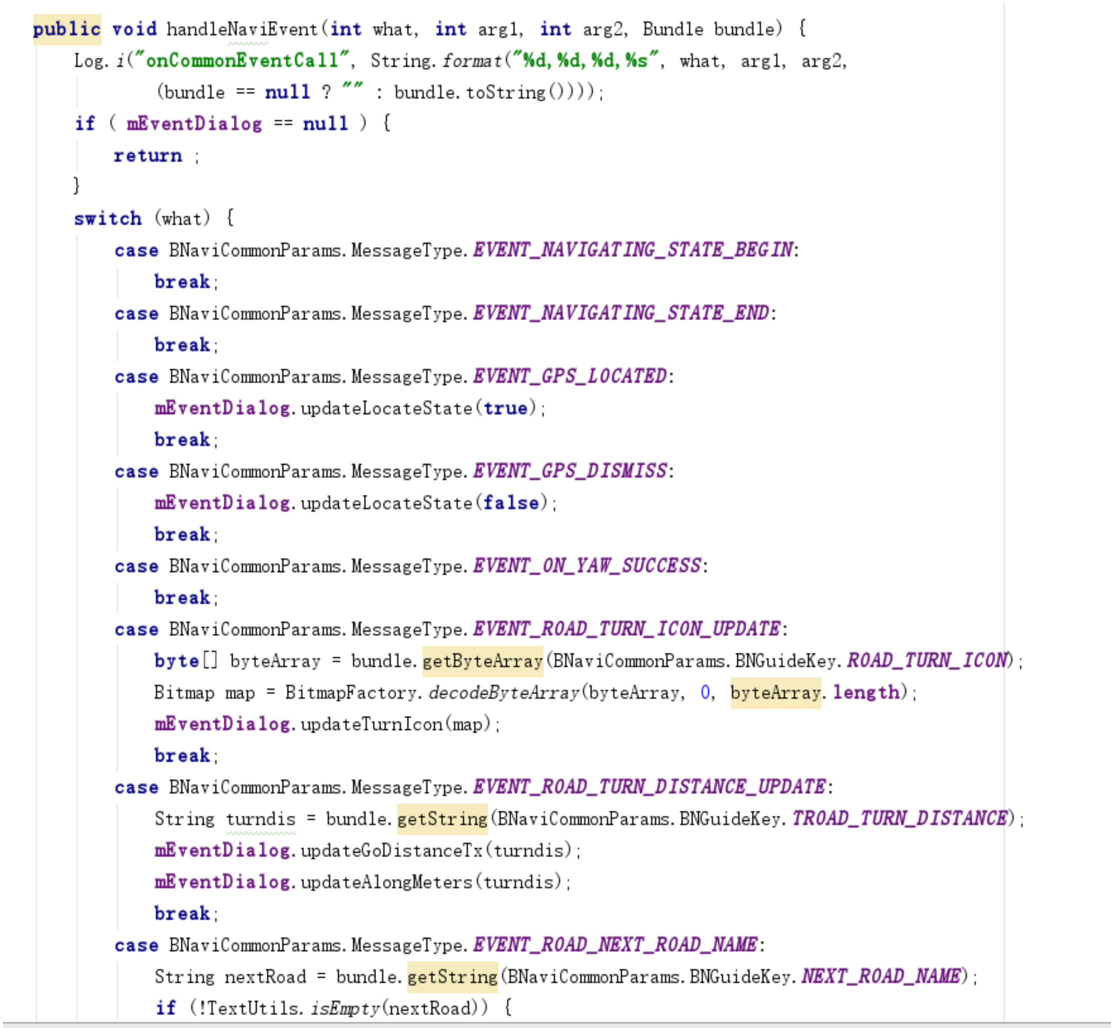
获取 POI 搜索信息以及位置语义化信息，供给给前台界面展示：



```java
sb.append("\nlocationdescribe : ");
sb.append(location.getLocationDescribe());    //位置语义化信息

List<Poi> list = location.getPoiList();    // POI数据
if (list != null) {
    sb.append("\npoilist size = : ");
    sb.append(list.size());
    for (Poi p : list) {
        sb.append("\npoi= : ");
        sb.append(p.getId() + " " + p.getName() + " " + p.getRank());
    }
}

Log.i("BaiduLocationApiDem", sb.toString());

//现在已经定位成功，可以将定位的数据保存下来，Const就是保存数据的类
Const.LONGITUDE = location.getLongitude();
Const.LATITUDE = location.getLatitude();
```

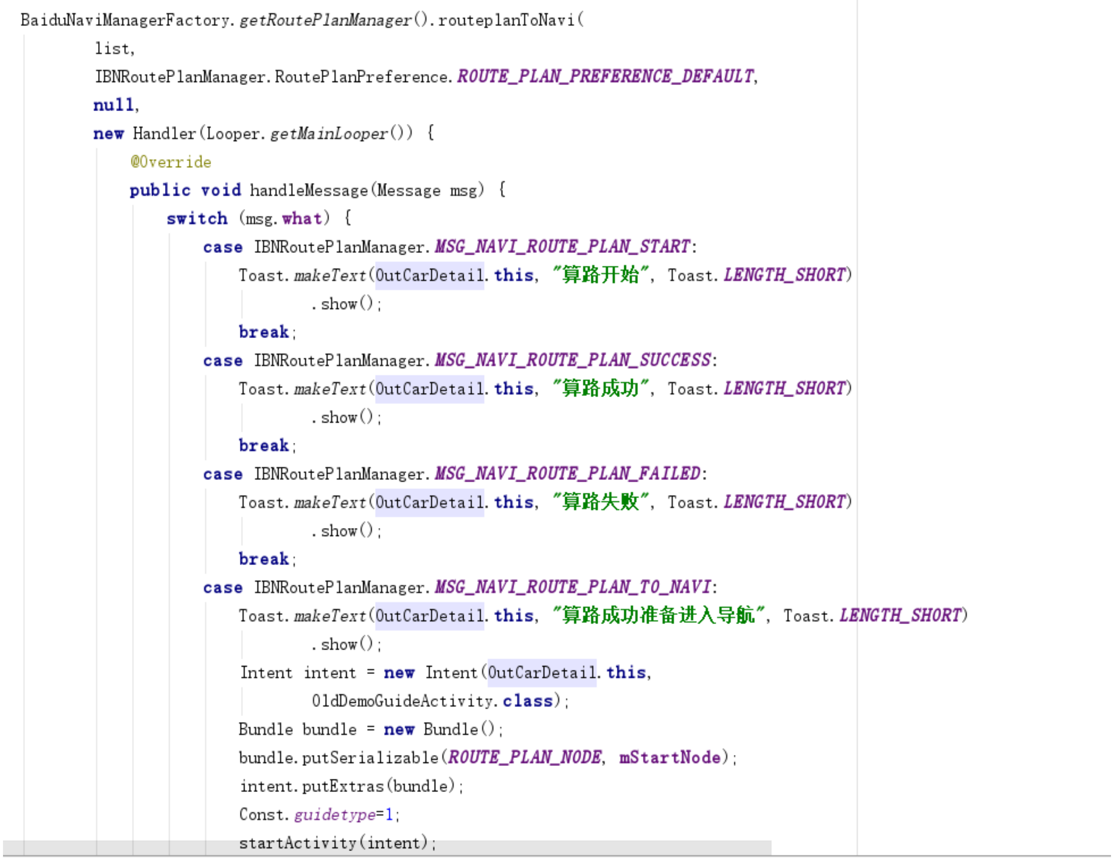
3.12 EventHandler 类及其成员函数

用于处理导航界面的点击、长按等事件触发对应的提示框及功能选择：



```java
public void handleNaviEvent(int what, int arg1, int arg2, Bundle bundle) {
    Log.i("onCommonEventCall", String.format("%d,%d,%d,%s", what, arg1, arg2,
            (bundle == null ? "" : bundle.toString())));
    if ( mEventDialog == null ) {
        return ;
    }
    switch (what) {
        case BNaviCommonParams.MessageType.EVENT_NAVIGATING_STATE_BEGIN:
            break;
        case BNaviCommonParams.MessageType.EVENT_NAVIGATING_STATE_END:
            break;
        case BNaviCommonParams.MessageType.EVENT_GPS_LOCATED:
            mEventDialog.updateLocateState(true);
            break;
        case BNaviCommonParams.MessageType.EVENT_GPS_DISMISS:
            mEventDialog.updateLocateState(false);
            break;
        case BNaviCommonParams.MessageType.EVENT_ON_YAW_SUCCESS:
            break;
        case BNaviCommonParams.MessageType.EVENT_ROAD_TURN_ICON_UPDATE:
            byte[] byteArray = bundle.getByteArray(BNaviCommonParams.BNGuideKey.ROAD_TURN_ICON);
            Bitmap map = BitmapFactory.decodeByteArray(byteArray, 0, byteArray.length);
            mEventDialog.updateTurnIcon(map);
            break;
        case BNaviCommonParams.MessageType.EVENT_ROAD_TURN_DISTANCE_UPDATE:
            String turndis = bundle.getString(BNaviCommonParams.BNGuideKey.TROAD_TURN_DISTANCE);
            mEventDialog.updateGoDistanceTx(turndis);
            mEventDialog.updateAlongMeters(turndis);
            break;
        case BNaviCommonParams.MessageType.EVENT_ROAD_NEXT_ROAD_NAME:
            String nextRoad = bundle.getString(BNaviCommonParams.BNGuideKey.NEXT_ROAD_NAME);
            if (!TextUtils.isEmpty(nextRoad)) {
```

3.13 调用百度地图引擎进行当前地点与目标位置的算路并获取返回的 Intent



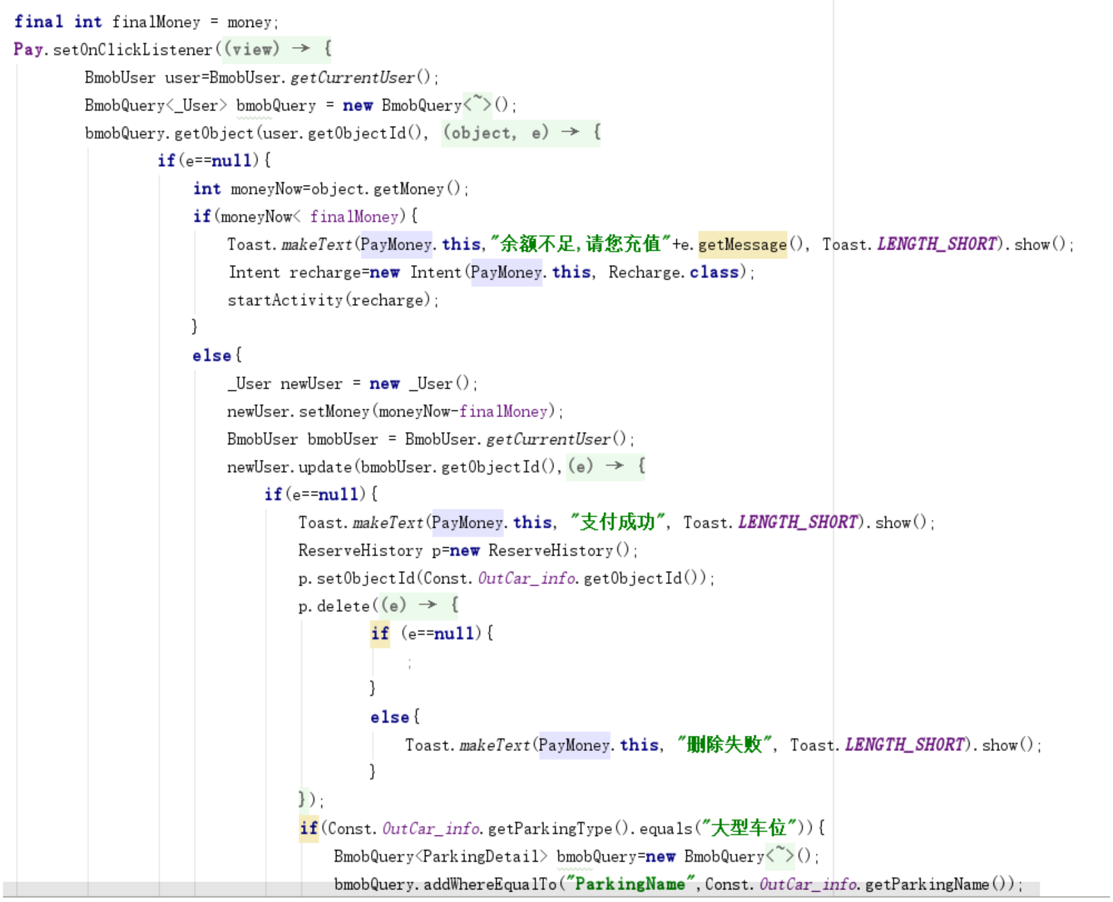
```
BaiduNaviManagerFactory.getRoutePlanManager().routeplanToNavi(
        list,
        IBNRoutePlanManager.RoutePlanPreference.ROUTE_PLAN_PREFERENCE_DEFAULT,
        null,
        new Handler(Looper.getMainLooper()) {
            @Override
            public void handleMessage(Message msg) {
                switch (msg.what) {
                    case IBNRoutePlanManager.MSG_NAVI_ROUTE_PLAN_START:
                        Toast.makeText(OutCarDetail.this, "算路开始", Toast.LENGTH_SHORT)
                                .show();
                        break;
                    case IBNRoutePlanManager.MSG_NAVI_ROUTE_PLAN_SUCCESS:
                        Toast.makeText(OutCarDetail.this, "算路成功", Toast.LENGTH_SHORT)
                                .show();
                        break;
                    case IBNRoutePlanManager.MSG_NAVI_ROUTE_PLAN_FAILED:
                        Toast.makeText(OutCarDetail.this, "算路失败", Toast.LENGTH_SHORT)
                                .show();
                        break;
                    case IBNRoutePlanManager.MSG_NAVI_ROUTE_PLAN_TO_NAVI:
                        Toast.makeText(OutCarDetail.this, "算路成功准备进入导航", Toast.LENGTH_SHORT)
                                .show();
                        Intent intent = new Intent(OutCarDetail.this,
                                OldDemoGuideActivity.class);
                        Bundle bundle = new Bundle();
                        bundle.putSerializable(ROUTE_PLAN_NODE, mStartNode);
                        intent.putExtras(bundle);
                        Const.guidetype=1;
                        startActivity(intent);
```

3.14 取车功能实现

1. 首先初始化 BmobQuery 对象

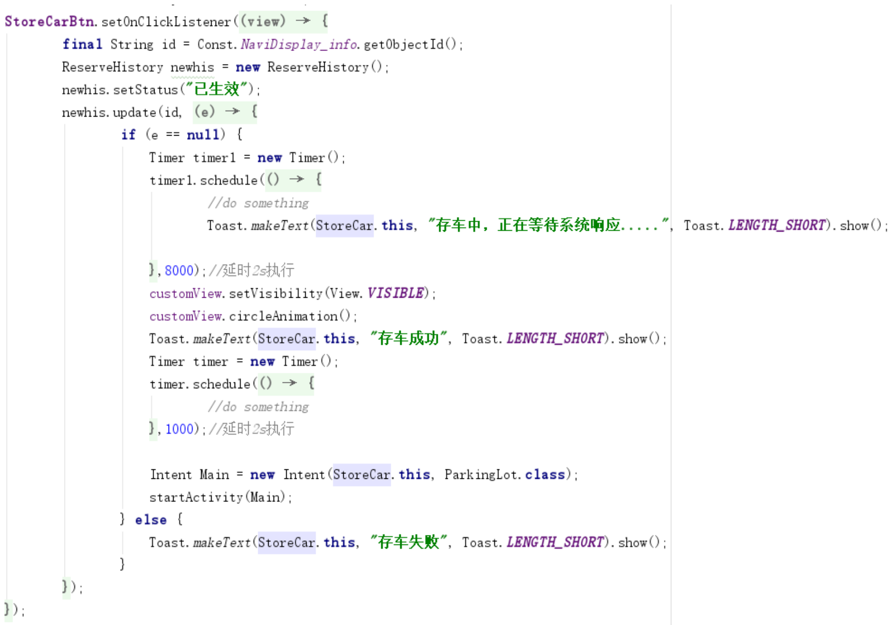
2. 根据已有记录的 ObjectId 进行查询当前账户的余额

3. 根据当前计算的消费额与余额进行比较，如果消费额大于余额则跳转到充值界面并提示余额不足，如果小于等于，则余额减去消费额并同时更新云端数据库当前账户余额以及删除当前用户的停车记录加入到历史记录一栏



```
final int finalMoney = money;
Pay.setOnClickListener((view) → {
        BmobUser user=BmobUser.getCurrentUser();
        BmobQuery<_User> bmobQuery = new BmobQuery<~>();
        bmobQuery.getObject(user.getObjectId(), (object, e) → {
                if(e==null){
                    int moneyNow=object.getMoney();
                    if(moneyNow< finalMoney){
                        Toast.makeText(PayMoney.this,"余额不足,请您充值"+e.getMessage(), Toast.LENGTH_SHORT).show();
                        Intent recharge=new Intent(PayMoney.this, Recharge.class);
                        startActivity(recharge);
                    }
                    else{
                        _User newUser = new _User();
                        newUser.setMoney(moneyNow-finalMoney);
                        BmobUser bmobUser = BmobUser.getCurrentUser();
                        newUser.update(bmobUser.getObjectId(),(e) → {
                            if(e==null){
                                Toast.makeText(PayMoney.this, "支付成功", Toast.LENGTH_SHORT).show();
                                ReserveHistory p=new ReserveHistory();
                                p.setObjectId(Const.OutCar_info.getObjectId());
                                p.delete((e) → {
                                        if (e==null){
                                            ;
                                        }
                                        else{
                                            Toast.makeText(PayMoney.this, "删除失败", Toast.LENGTH_SHORT).show();
                                        }
                                });
                                if(Const.OutCar_info.getParkingType().equals("大型车位")){
                                    BmobQuery<ParkingDetail> bmobQuery=new BmobQuery<~>();
                                    bmobQuery.addWhereEqualTo("ParkingName",Const.OutCar_info.getParkingName());
```

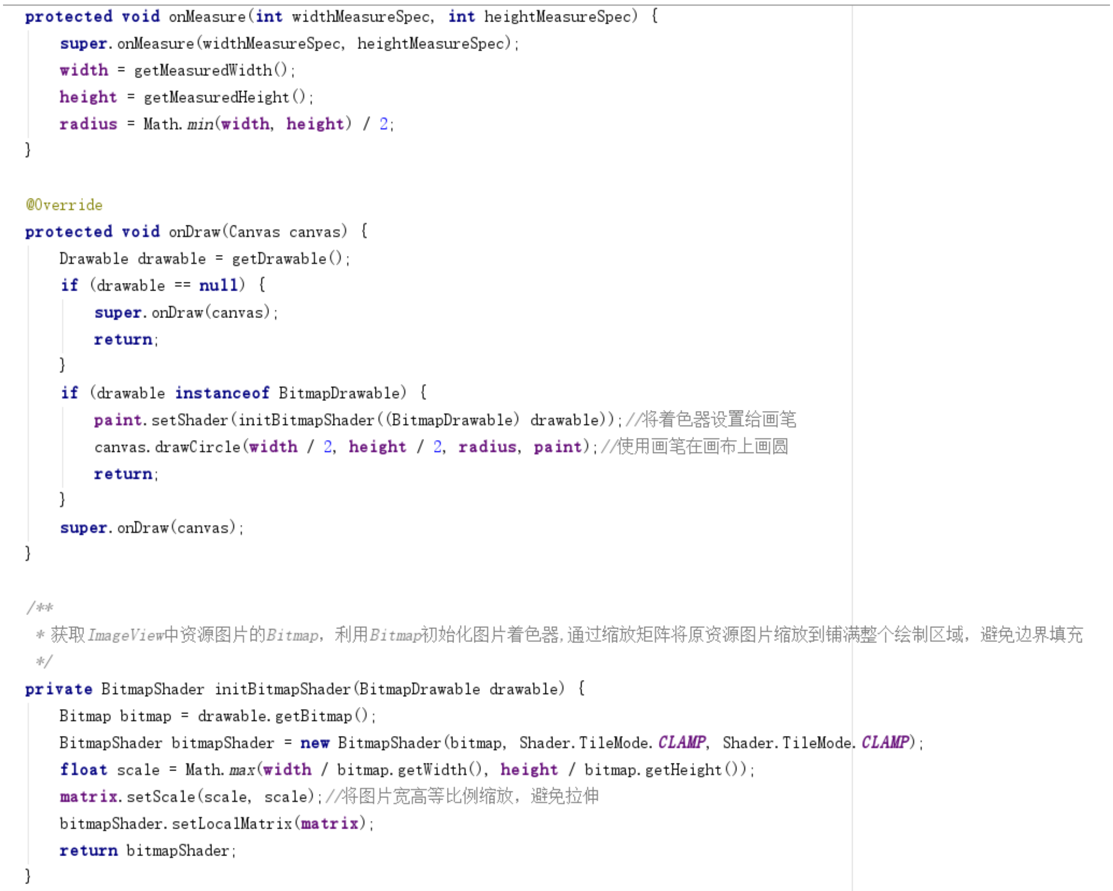
通过对云端数据库进行增删操作实现,点击对应按钮将预约信息写入数据库/从数据库删除:



```
StoreCarBtn.setOnClickListener((view) → {
        final String id = Const.NaviDisplay_info.getObjectId();
        ReserveHistory newhis = new ReserveHistory();
        newhis.setStatus("已生效");
        newhis.update(id, (e) → {
                if (e == null) {
                    Timer timer1 = new Timer();
                    timer1.schedule(() → {
                            //do something
                            Toast.makeText(StoreCar.this, "存车中, 正在等待系统响应.....", Toast.LENGTH_SHORT).show();

                    },8000);//延时2s执行
                    customView.setVisibility(View.VISIBLE);
                    customView.circleAnimation();
                    Toast.makeText(StoreCar.this, "存车成功", Toast.LENGTH_SHORT).show();
                    Timer timer = new Timer();
                    timer.schedule(() → {
                            //do something
                    },1000);//延时2s执行

                    Intent Main = new Intent(StoreCar.this, ParkingLot.class);
                    startActivity(Main);
                } else {
                    Toast.makeText(StoreCar.this, "存车失败", Toast.LENGTH_SHORT).show();
                }
        });
});
```

3.15 圆角图标实现

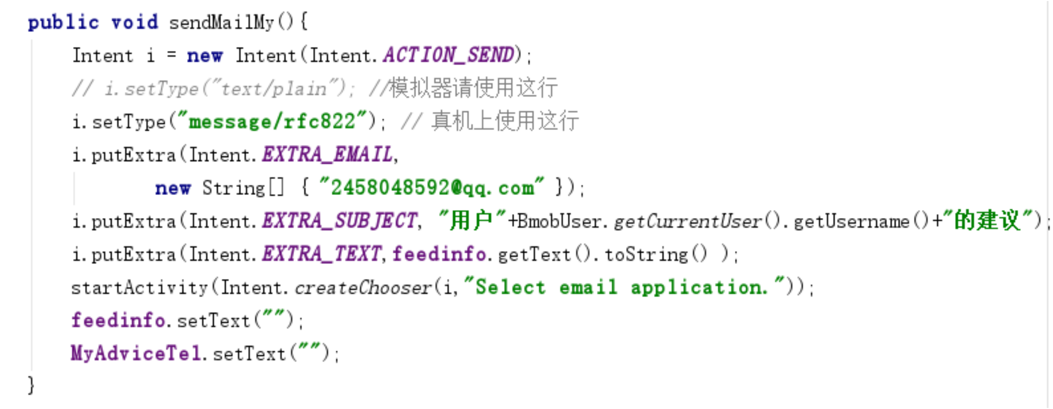
首先建立一个 ImageCircleView 类继承于 imageview 类,使用着色器设置画笔并用画笔在画布上画出圆圈:



```
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);
    width = getMeasuredWidth();
    height = getMeasuredHeight();
    radius = Math.min(width, height) / 2;
}

@Override
protected void onDraw(Canvas canvas) {
    Drawable drawable = getDrawable();
    if (drawable == null) {
        super.onDraw(canvas);
        return;
    }
    if (drawable instanceof BitmapDrawable) {
        paint.setShader(initBitmapShader((BitmapDrawable) drawable));//将着色器设置给画笔
        canvas.drawCircle(width / 2, height / 2, radius, paint);//使用画笔在画布上画圆
        return;
    }
    super.onDraw(canvas);
}

/**
 * 获取ImageView中资源图片的Bitmap，利用Bitmap初始化图片着色器,通过缩放矩阵将原资源图片缩放到铺满整个绘制区域，避免边界填充
 */
private BitmapShader initBitmapShader(BitmapDrawable drawable) {
    Bitmap bitmap = drawable.getBitmap();
    BitmapShader bitmapShader = new BitmapShader(bitmap, Shader.TileMode.CLAMP, Shader.TileMode.CLAMP);
    float scale = Math.max(width / bitmap.getWidth(), height / bitmap.getHeight());
    matrix.setScale(scale, scale);//将图片宽高等比例缩放，避免拉伸
    bitmapShader.setLocalMatrix(matrix);
    return bitmapShader;
}
```

3.16 发送邮件，用户反馈实现

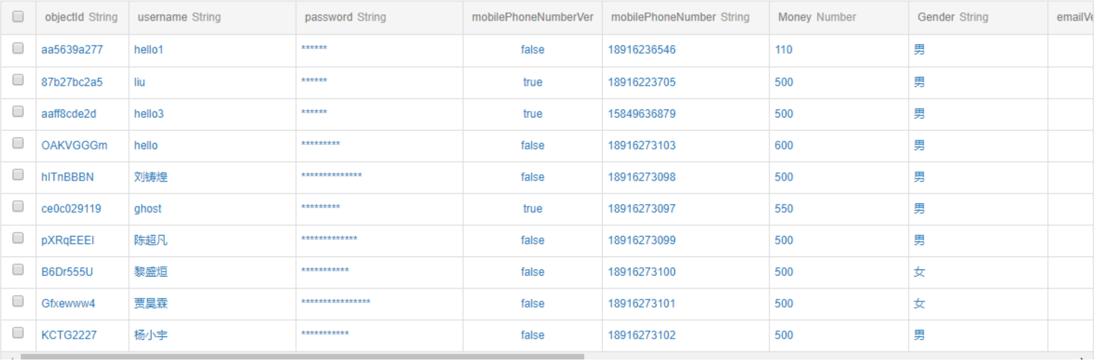
获取 Plain Text 中编辑的文字调用手机自带的邮件应用，新建 Send Intent 并设置其源mail，目标 mail（2458048592@qq.com），实现用户反馈获取



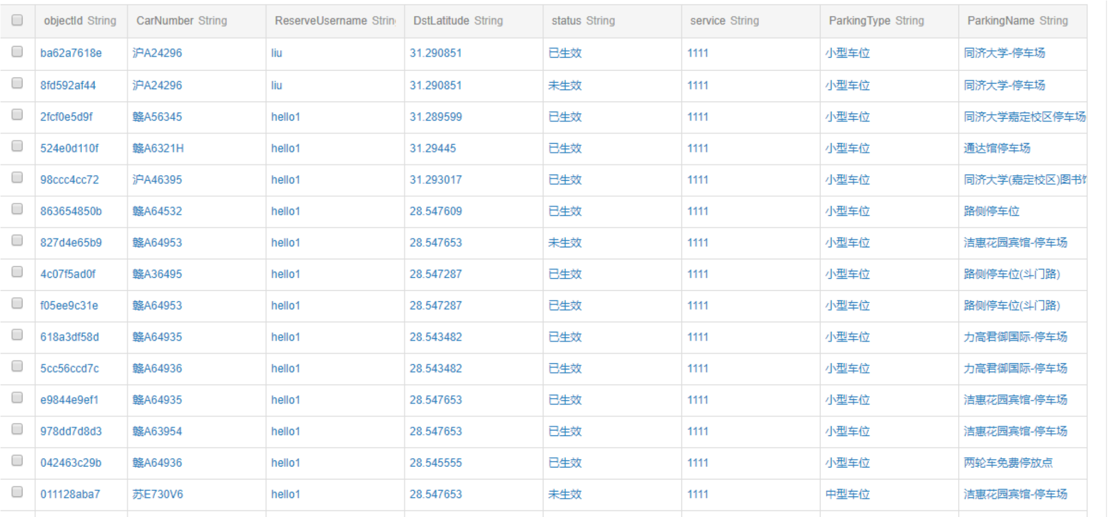
```
public void sendMailMy(){
    Intent i = new Intent(Intent.ACTION_SEND);
    // i.setType("text/plain"); //模拟器请使用这行
    i.setType("message/rfc822"); // 真机上使用这行
    i.putExtra(Intent.EXTRA_EMAIL,
            new String[] { "2458048592@qq.com" });
    i.putExtra(Intent.EXTRA_SUBJECT, "用户"+BmobUser.getCurrentUser().getUsername()+"的建议");
    i.putExtra(Intent.EXTRA_TEXT,feedinfo.getText().toString() );
    startActivity(Intent.createChooser(i,"Select email application."));
    feedinfo.setText("");
    MyAdviceTel.setText("");
}
```

3.11 云端数据库表内容

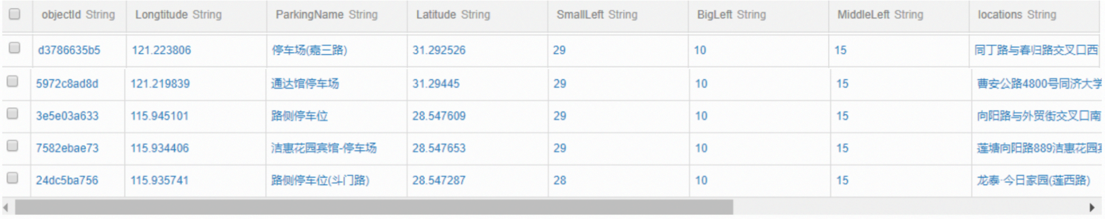
Bmob 云端数据库包含三张数据表，分别为 User、ReserveHistory、ParkingDetail 分别为用户信息表、预约信息表、停车场信息表，如下所示：



☐	objectId String	username String	password String	mobilePhoneNumberVer	mobilePhoneNumber String	Money Number	Gender String	emailVe
☐	aa5639a277	hello1	******	false	18916236546	110	男	
☐	87b27bc2a5	liu	******	true	18916223705	500	男	
☐	aaff8cde2d	hello3	******	true	15849636879	500	男	
☐	OAKVGGGm	hello	*********	false	18916273103	600	男	
☐	hITnBBBN	刘铸煌	**************	false	18916273098	500	男	
☐	ce0c029119	ghost	*********	true	18916273097	550	男	
☐	pXRqEEEI	陈超凡	*************	false	18916273099	500	男	
☐	B6Dr555U	黎盛烜	***********	false	18916273100	500	女	
☐	Gfxewww4	贾昊霖	****************	false	18916273101	500	女	
☐	KCTG2227	杨小宇	***********	false	18916273102	500	男	



☐	objectId String	CarNumber String	ReserveUsername Strin	DstLatitude String	status String	service String	ParkingType String	ParkingName String
☐	ba62a7618e	沪A24296	liu	31.290851	已生效	1111	小型车位	同济大学-停车场
☐	8fd592af44	沪A24296	liu	31.290851	未生效	1111	小型车位	同济大学-停车场
☐	2fcf0e5d9f	赣A56345	hello1	31.289599	已生效	1111	小型车位	同济大学嘉定校区停车场
☐	524e0d110f	赣A6321H	hello1	31.29445	已生效	1111	小型车位	通达馆停车场
☐	98ccc4cc72	沪A46395	hello1	31.293017	已生效	1111	小型车位	同济大学(嘉定校区)图书
☐	863654850b	赣A64532	hello1	28.547609	已生效	1111	小型车位	路侧停车位
☐	827d4e65b9	赣A64953	hello1	28.547653	未生效	1111	小型车位	洁惠花园宾馆-停车场
☐	4c07f5ad0f	赣A36495	hello1	28.547287	已生效	1111	小型车位	路侧停车位(斗门路)
☐	f05ee9c31e	赣A64953	hello1	28.547287	已生效	1111	小型车位	路侧停车位(斗门路)
☐	618a3df58d	赣A64935	hello1	28.543482	已生效	1111	小型车位	力高君御国际-停车场
☐	5cc56ccd7c	赣A64936	hello1	28.543482	已生效	1111	小型车位	力高君御国际-停车场
☐	e9844e9ef1	赣A64935	hello1	28.547653	已生效	1111	小型车位	洁惠花园宾馆-停车场
☐	978dd7d8d3	赣A63954	hello1	28.547653	已生效	1111	小型车位	洁惠花园宾馆-停车场
☐	042463c29b	赣A64936	hello1	28.545555	已生效	1111	小型车位	两轮车免费停放点
☐	011128aba7	苏E730V6	hello1	28.547653	未生效	1111	中型车位	洁惠花园宾馆-停车场



☐	objectId String	Longtitude String	ParkingName String	Latitude String	SmallLeft String	BigLeft String	MiddleLeft String	locations String
☐	d3786635b5	121.223806	停车场(嘉三路)	31.292526	29	10	15	同丁路与春归路交叉口西
☐	5972c8ad8d	121.219839	通达馆停车场	31.29445	29	10	15	曹安公路4800号同济大学
☐	3e5e03a633	115.945101	路侧停车位	28.547609	29	10	15	向阳路与外贸街交叉口南
☐	7582ebae73	115.934406	洁惠花园宾馆-停车场	28.547653	29	10	15	莲塘向阳路889洁惠花园
☐	24dc5ba756	115.935741	路侧停车位(斗门路)	28.547287	28	10	15	龙泰·今日家园(莲西路)