

智能停车场管理系统

软件说明文档



内容模块总览：

一. 项目总览

- [一]项目名称
- [二]项目背景
- [三]项目功能
- [四]项目意义

二. 项目开发过程说明

- [一]整体架构描述
- [二]功能程序流程图设计
- [三]界面模块设计
- [四]关键功能对应代码说明
- [五]软件测试过程及结果

三. 其他

- [一]术语表
- [二]文档修改历史
- [三]参考文献

一. 项目总览

[一]项目名称：智能停车场管理系统

[二]项目背景：

当今时代发达城市土地紧缺问题尤为明显,而越来越多的车辆也使得对于停车位的需求也越来越大,所以合理利用有限的土地建造更多的停车位成了关键性问题,智能停车场管理系统设计可以合理的缓解这个问题,利用地上立体停车场大楼使得有限的土地可以容纳更多的车辆而智能停车场引导及取车系统可以免去车主找车所花费的时间,利用手机 app,车主可以即时查看车辆情况或者通过 app 选择相应的车辆服务以实现车辆服务一体化,省去车主来往于汽车维修点的时间,智能系统可以对车辆中涉及到的危险问题对车主进行示警,以达到安全行车的目的。由此我们想着手做智能停车场管理系统。

[三]项目功能：

由于项目只涉及一种用户,所以项目对应的功能韦恩图如下:



[四]项目意义：

[1]重要性：

可以通过与实际的智能停车场的合作,对周边车位进行智能推荐从而达到便利停车的目的,并且通过智能停车场管理系统的预约服务对车辆服务资源进行整合,并对车辆状况进行评估从而为用户提供车辆一体化且更为安全的服务,同时提供停车场导航功能为用户提供便捷的车辆管理服务。

[2]必要性:

随着现代社会自动化程度的不断提高,效率低下的现有管理系统将逐渐不适用,要实现高效管理,朝这方面进行研究开发是必要的。

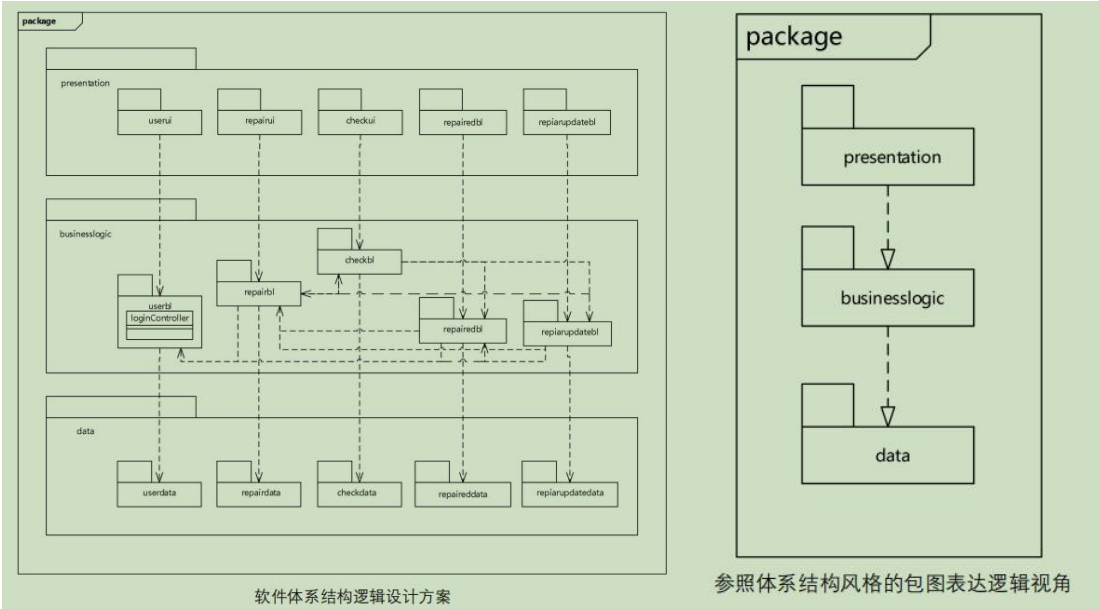
二. 项目开发过程说明

[一]整体架构描述

根据体系结构的设计,系统分为用户界面层,业务逻辑层,数据层。每一层之间为了增加灵活性,我们会添加接口.业务逻辑层和数据层之间我们添加了 dataservice.parkinglotdataservice.ParkinglotDataService 接口。为了隔离数据层职责和业务逻辑层 职责,我们增加了 UserController, 这样 UserController 会将用户管理信息进行委托给 ParkinglotDataService 对象。ParkingDataList 封装了关于 ParkingDataLineItem 的数据集合的数据结构 的秘密。Usermemberinfo、Parkinglotinfo、Reserveinfo 都是 根据 依赖 倒置 的原则, 为了消除循环依赖而产生的接口。

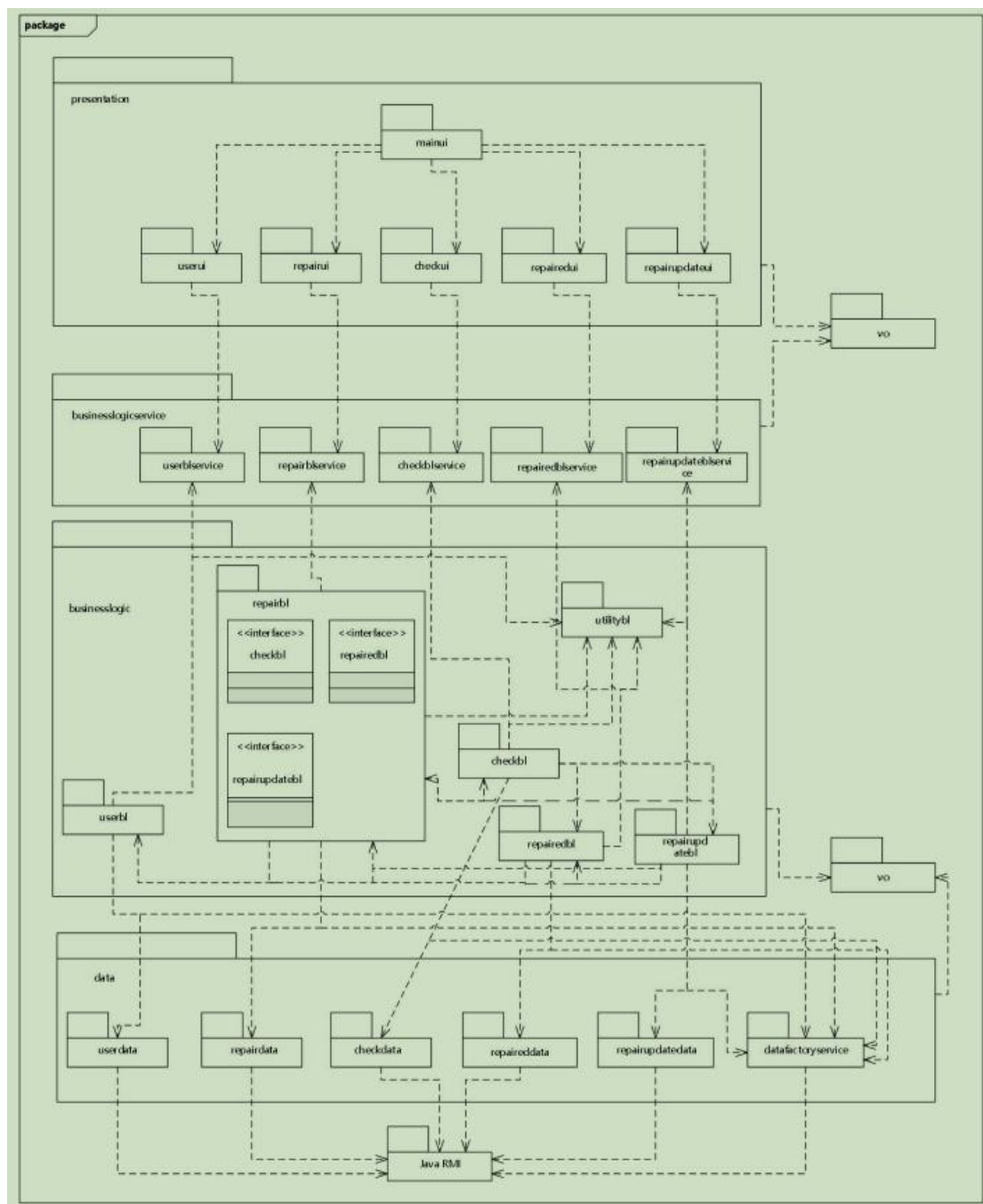
1.1 逻辑视角

开发过程中该软件将系统分为 3 层: 展示层 (presentation) 业务逻辑层 (businesslogic) 数据层 (data)。其中, 展示层包含 GUI 页面的实现, 业务逻辑层包含业务逻辑处理的实现, 数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下:



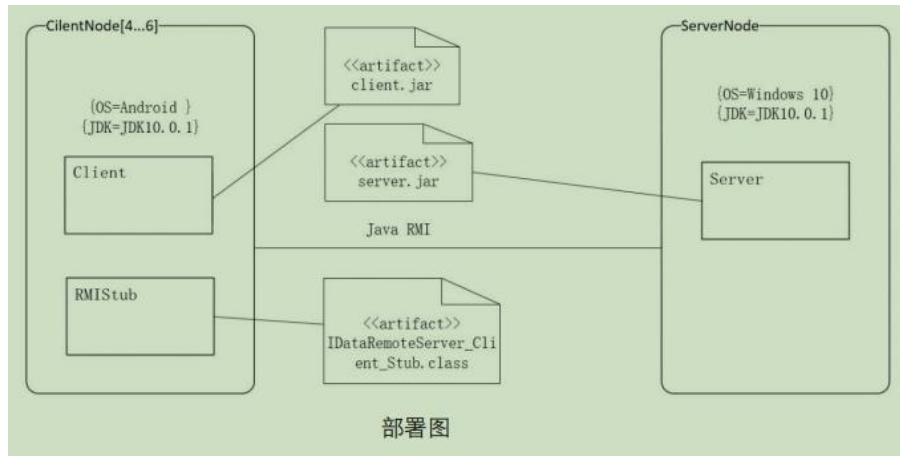
1.2 组合视角

智能停车场管理系统客户端开发包图如下:



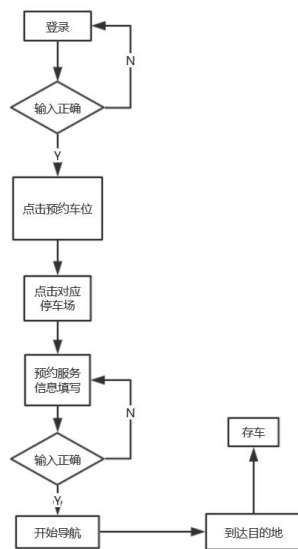
1.3 物理部署

智能停车场管理系统中客户端构件是放在客户端机器上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署 RMISTub 部件，由于 Java RMI 构件属于 JDK10,0.1 的一部分。所以，在系统 JDK 环境已经设置好了的情况下，不需要再独立部署。部署图如下图所示。

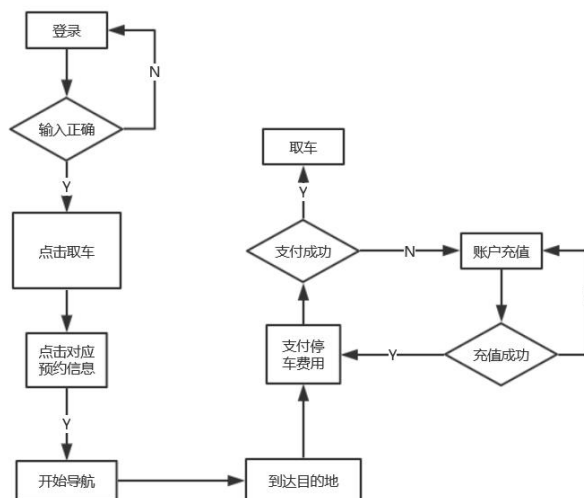


[二]功能程序流程图设计

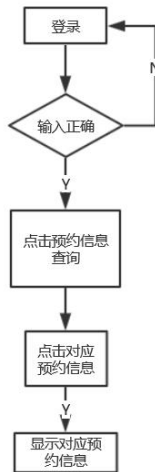
2.1 存车功能流程图如下:



2.2 取车功能流程图如下:



2.3 预约记录查询功能流程图如下:



[三]界面模块设计

3.1 界面跳转设计

3.1.1 首先点击软件进入登录界面，点击注册按钮跳转注册活动界面，填写相关信息点击注册跳转管理系统主界面，在登录界面输入对应信息点击登录跳转系统主界面

3.1.2 主界面点击预约车位按钮跳转周边停车场展示选择界面，点击预约信息查询按钮跳转到车位预约信息查询界面，点击取车按钮跳转到已生效预约信息展示界面

3.1.3 主界面点击侧滑栏，在侧滑栏中点击个人信息按钮跳转个人信息查询界面，点击余额查询按钮跳转余额展示及充值说明界面，点击反馈建议按钮跳转反馈信息填写界面，点击注销账户跳转登录界面，点击修改密码按钮跳转修改密码界面，点击关于我们按钮跳转项目信息展示界面，在个人信息展示界面中点击我的位置跳转个人实时定位展示界面，点击退出登录跳转登录界面

3.1.4 周边停车场展示选择界面点击对应停车场跳转该停车场信息展示界面，在停车场信息展示界面点击预约跳转停车场车辆服务预约界面，在该界面中点击导航跳转停车场导航界面，到达停车场后跳转存车界面，点击存车跳转系统主界面

3.1.5 已生效预约信息展示界面点击对应预约信息，跳转该停车场预约信息展示界面，点击导航按钮跳转停车场导航界面，到达停车场后自动跳转到取车界面，点击支付按钮，如果支付成功则跳转系统主界面，支付失败(余额不足)跳转充值界面

3.2 界面风格：

界面采用的风格为简约风格，以单一色调为主，背景设置为动画背景。界面直观对用户透明，用户接触该软件之后对界面上的功能一目了然，无需培训便可使用该软件。界面设计采用以线性布局 LinearLayout 为主，相对布局 RelativeLayout 为辅的设计方法。字体为灰底黑字，部分为灰底红字，字体大小为宋体，大小为 14sp，部分显示带边框，14 对齐方式为居中对齐。

3.3 关键界面实现解释

3.3.1 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.ParkLongitude = loc.longitude;
        Const.ParkLatitude = loc.latitude;
        Const.ParkName = name;
        Intent intent = new Intent(P0IDisplay.this, ParkDetailDisplay.class);
        Bundle bundle = new Bundle();
        bundle.putString("Name", name);
        bundle.putString("Address", address);
        bundle.putString("Distance", distance);
        intent.putExtras(bundle);
        startActivity(intent);
    }
});
```

3.3.2 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：


```

<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>

<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>

```

3.3.3 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：

```

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>

```

3.3.4 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>

```

3.4 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

[四]关键功能对应代码说明

4.1 周边停车场展示功能

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")
    .setType(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;
    }
}

```

4.2 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歧义，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(
        01dDemoGuideActivity.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 IBNRRouteGuideManager.OnNavigationListener mOnNavigationListener =
    new IBNRRouteGuideManager.OnNavigationListener() {

        @Override
        public void onNaviGuideEnd() {
            // 退出导航
            finish();
            if (Const.guidetype==0) {
                Intent Parking=new Intent(01dDemoGuideActivity.this, StoreCar.class);
                startActivity(Parking);
            }
            else if (Const.guidetype==1) {
                Intent Parking = new Intent(01dDemoGuideActivity.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(01dDemoGuideActivity.this, StoreCar.class);
                    startActivity(Parking);
                }
                else if (Const.guidetype==1) {
                    Intent Parking = new Intent(01dDemoGuideActivity.this, PayMoney.class);

```

4.3 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(getApplicationContext(), "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(getApplicationContext(), "创建数据失败: " + e.getMessage(), Toast.LENGTH_SHORT).show();
    }
});
} else {
    Toast.makeText(getApplicationContext(), "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();
        }
    });
});
```

4.4 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 MyLocationListener implements BDLatLngListener {
    @Override
    public void onReceiveLocation(BDLatLng location){
        if (location == null){
            sendBroadcast("定位失败!");
            return;
        }
        sendBroadcast(location.getAddrStr());
    }
    public void onReceivePoi(BDLatLng 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);
}

```

4.5 预约信息查询/个人信息查询/修改密码/存取车功能

预约信息查询/个人信息查询/修改密码/存取车功能均为使用 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());
    }
});

```

存取车功能则是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();
        }
    });
});

```


而修改用户信息则是通过 update(objectId,updateListener)方法来更新

4.6 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	mobileNumberVer	mobilePhoneNumber String	Money Number	Gender String	emailV
	aa5639a277	hello1	*****	false	18916236546	110	男	
	87b27bc2a5	liu	*****	true	18916223705	500	男	
	aa98cde2d	hello3	*****	true	15849636879	500	男	
	OAKVGGGm	hello	*****	false	18916273103	600	男	
	hiTnBBN	刘博煜	*****	false	18916273098	500	男	
	ce0c029119	ghost	*****	true	18916273097	550	男	
	pXRqEEEl	陈超凡	*****	false	18916273099	500	男	
	B6Dr555U	黎盛垣	*****	false	18916273100	500	女	
	Gfxewww4	黄昊霖	*****	false	18916273101	500	女	
	KCTG2227	杨小李	*****	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
	2fc0e5d9f	沪A56345	hello1	31.289599	已生效	1111	小型车位	同济大学嘉定校区停车场
	524e0d110f	沪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)图书馆
	863654850b	沪A64532	hello1	28.547609	已生效	1111	小型车位	路侧停车位
	827d4e65b9	沪A64953	hello1	28.547653	未生效	1111	小型车位	清惠花园宾馆-停车场
	4c07f5ad0f	沪A36495	hello1	28.547287	已生效	1111	小型车位	路侧停车位(斗门路)
	f05ee9c31e	沪A64953	hello1	28.547287	已生效	1111	小型车位	路侧停车位(斗门路)
	618a3df58d	沪A64935	hello1	28.543482	已生效	1111	小型车位	力高君御国际-停车场
	5cc56ccd7c	沪A64936	hello1	28.543482	已生效	1111	小型车位	力高君御国际-停车场
	e9844e9ef1	沪A64935	hello1	28.547653	已生效	1111	小型车位	清惠花园宾馆-停车场
	978dd7d8d3	沪A63954	hello1	28.547653	已生效	1111	小型车位	清惠花园宾馆-停车场
	042463c29b	沪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	清惠花园宾馆-停车场

	objectId String	Longitude String	ParkingName String	Latitude String	SmallLeft String	BigLeft String	MiddleLeft String	locations String
	e0bbf3c02c	121.218345	同济大学(嘉定校区)开物馆-	31.29479	28	10	15	上海市嘉定区曹安公路48
	d3786635b5	121.223806	停车场(南三路)	31.292526	29	10	15	同丁路与泰归路交叉口西
	8c89b8bdb0	121.219048	同济大学(嘉定校区)图书馆-	31.293017	28	10	15	曹安公路4800号
	3645491bb8	121.222051	同济大学嘉定校区学生事务	31.292277	29	10	15	曹安公路4800号同济大学
	f5dc9a3607	121.512346	同济大学-停车场	31.290851	29	10	15	国康路11同济大学
	622136d559	121.218486	同济大学嘉定校区停车场(清	31.289599	29	10	15	上海市嘉定区曹安公路48
	5972c8ad8d	121.219839	通达馆停车场	31.29445	29	10	15	曹安公路4800号同济大学
	3e5e03ae33	115.945101	路侧停车位	28.547609	29	10	15	向阳路与外环线交叉口南
	7582ebae73	115.934406	清惠花园宾馆-停车场	28.547653	29	10	15	莲塘向阳路889清惠花园
	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)

[五]软件测试过程及结果

5.1 总测试结果概述

测试活动总结：智能停车场管理系统的测试从 2019-2-16 开始持续到 2019-2-24，共持续 8 天，按时完成测试计划，所有测试结果与测试任务均完成，软件各项功能需求以及非功能需求均得到调试满足。

产品质量评估：产品总体质量到达预期要求，可以支持普通用户登录等一系列功能达到测试要求，虽然测试时存在少数情况与时间段出现无法处理的情况，但经过调试后，大体使用预期目标达到，服务器端数据的增删改减信息功能达标。该智能停车场管理系统项目，经过项

目开发人员、测试人员的协力合作，达到交付标准，满足用户的需求。

5.2 度量数据总结

普通用户人员度量数据:

用例数量:5

平均每个用例的场景数:13

平均用例行数:220

功能需求数量:35

输入:21

输出:25

查询:6

逻辑文件:13

对外接口:0

测试编号	度量描述与度量
User.Login.input	系统允许用户在登录任务(输出: 预约车位主界面)中进行键盘输入和操作
User.Login.Cancel	当用户发出取消请求时(登录), 系统不做任何逻辑处理
User.Invalid	当输入的信息不合法时, 系统提示错误, 要求重新输入或取消修改(输出)
User.Modify	系统应该允许用户修改输入信息
User.Modify.Null	当用户不改变输入信息时, 系统不做任何逻辑处理
User.Modify.End	当用户修改完毕并确认后(修改), 系统更新数据, 提示修改成功(输出)
User.Register.input	系统允许用户在注册任务(输出: 系统登录主界面)中进行键盘输入和操作
User.Register.Cancel	当用户发出取消请求时(注册), 系统不做任何逻辑处理
User.Register.Invalid	当输入的信息不合法时, 系统提示错误, 要求重新输入或取消修改(输出)
User.Search.input	系统允许用户在查询车位任务(输出:附近车位显示界面)中进行键盘输入和操作
User.Search.Cancel	当用户发出取消请求时(查询), 系统不做任何逻辑处理, 返回上级界面
User.Guide.input	系统允许用户在查询车位任务(输出:附近车位显示界面)中进行键盘输入和操作
User.Guide.Cancel	当用户发出取消请求时(查询), 系统不做任何逻辑处理, 返回上级界面
User.Info.input	系统允许用户在查询个人信息任务(输出:个人信息界面)中进行键盘输入和操作
User.Info.Cancel	当用户发出取消请求时(查询信息), 系统不做任何逻辑处理, 返回上级界面
User.Reserve.input	系统允许用户在预约车位任务(输出:目标车位导航界面)中进行键盘输入和操作

User.Reserve.Cancel	当用户发出取消请求时(预约), 系统不做任何逻辑处理, 返回上级界面
User.Logout	系统应该允许用户退出登录
User.Recharge	系统应该允许用户为账户充值
User.Recharge.input	系统允许用户在充值任务中进行键盘输入和操作
User.Recharge.Cancel	当用户发出取消请求时(充值), 系统不做任何逻辑处理, 返回上级界面
User.Pay	系统应该允许用户在取车时支付费用
User.Feedback.input	系统允许用户在反馈建议任务中进行键盘输入和操作
User.Feedback.Cancel	当用户发出取消请求时(反馈建议), 系统不做任何逻辑处理, 返回上级界面
User.Changepwd.input	系统允许用户在修改密码任务中进行键盘输入和操作
User.Changepwd.Cancel	当用户发出取消请求时(修改密码), 系统不做任何逻辑处理, 返回上级界面

智能停车场管理系统属于中等系统, 所以其功能点测度总数为 $18 \times 4 + 15 \times 5 + 5 \times 4 + 9 \times 10 + 0 \times 4 = 257$, 系统的复杂度因子为 36, 功能点数 $FP = 257 \times (0.65 + 0.01 \times 36) = 259.57$

5.3 测试结果决策

成员认为作出软件‘通过’决定的原因有以下几点:

1. 系统产生的部分错误在日常软件使用过程中遇到的概率较低, 对于整体结构运行的影响不大, 可以通过运行测试
2. 软件所达到的预期效果目标良好, 可以实现三类人群的各类功能需求, 可移植性良好, 可供后期开发人员改良
3. 本项目在软件测试过程中没有出现致命错误, 且软件运行时产生的效果与界面优化已做到可以普通人也可快速掌握使用的程度, 所以可以通过

三. 其他

[一]术语表

词汇名称	词汇含义	备注
DPM	智能停车场管理系统	Diligent Parkinglot manager
Bmob	云端数据库	在线云端数据库
SD	System Design	系统设计
SMR	Senior Management Review	高级管理评审
SPI	Software Process Improvement	软件过程改进
UCD	User Case Diagram	用例图
UT	Unit Testing	单元测试
UAT	User Acceptance Test	用户验收测试

[二]文档修改历史

修改人员	日期	变更原因	版本号
------	----	------	-----

涂远鹏	2018-5-7	添加总体设计概要、项目名称、项目内容	V1.1.1
涂远鹏	2018-7-9	绘制软件功能韦恩图	V1.1.2
涂远鹏	2018-9-16	添加项目意义并开始编写功能程序流程图，对软件功能流程进行优化	V1.1.4
涂远鹏	2018-11-25	设计软件界面，并添加软件测试分析，修改界面跳转流程	V1.1.5
涂远鹏	2018-12-17	对测试分析表添加表项，对软件进行功能、结构及跳转测试，并将测试结果添加到对应表项	V2.1.1
涂远鹏	2018-1-27	添加关键功能代码展示及解释模块并对软件界面进行MVP+MVVC优化	V2.1.2
涂远鹏	2019-2-25	添加术语表、参考文献、修改历史表并对文档整体进行结构性修改	V3.1.1

[三]参考文献

- 1) 郭霖. 第一行代码[M]. 人民邮电出版社:郭霖, 2014. 69-307
- 2) IEEE[1471-2000]标准
- 3) 骆斌.软件工程与计算（卷二）——软件开发的技术基础[M].机械工业出版社:骆斌, 2016.82-118, 403-410
- 4) 郭志宏. Android 应用开发详解[M].电子工业出版社. 2010.
- 5) 杨丰盛.Android 应用开发揭秘[M]. 机械工业出版社. 2010.
- 6)Frank Ableson.Introduction to Android development[J].developerWorks,2009,10(7).
- 7)余志龙, 陈昱勋, 郑名杰, 陈小凤, 郭秩均. Google Android SDK 开发范例 大全[M].人民邮电出版社. 2009.
- 8)李宁. Android/OPhone 开发完全讲义[M]. 中国水利水电出版社. 2010.
- 9)Ed Burnett, Hello, Android: Introducing Google ' s Mobile Development Platform[J]. PRAGMATIC BOOKSHELF,2010.7:10-11.