

# 室内交互式引导 APP 概要设计说明书

编 撰 人：孙晟元、杨豪

审 核 人：杨秋辉

批 准 人：杨秋辉

最终版本：005

批准日期：2015-5-25

## 版本历史

[illegible]

---

# 目 录

<b>1. 引言</b>	<b>1</b>
1.1. 编写目的	1
1.2. 背景	1
1.3. 术语	1
1.4. 参考资料	3
<b>2. 总体设计</b>	<b>4</b>
2.1. 运行环境	5
2.1.1. 运行环境网络结构说明	5
2.1.2. 运行环境说明细则	5
2.1.3. 支持软件	5
2.2. 系统的关键技术	5
<b>3. 构件设计</b>	<b>7</b>
3.1. 构件列表	7
3.2. 构建集成说明	7
3.3. 构建 1—机场服务	7
3.4. 构建 2—航班信息	12
3.5. 构建 3—室内导航	15
<b>4. 接口设计</b>	<b>24</b>
4.1. 外部接口	25
4.2. 内部接口	28
<b>5. 系统出错处理</b>	<b>39</b>
6.1. 出错信息	39
6.2. 补救措施	39

# 1. 引言

## 1.1. 编写目的

本文档将在大赛所要求的需求分析的基础上，对项目室内交互式引导 APP 做概要设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。以便于评审专家能够更好，更快地对项目有一个整体、全面的逻辑构架。对项目的功能点、创新点、实现算法、结构模块等有一个清晰准确的认识。

在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。

## 1.2. 背景

1. 开发的软件系统的名称：室内交互式引导 APP--POCKET AIRPORT

2. 列出此项目的任务提出者：第五届“中国软件杯”组委会

开发者：孙晟元、杨豪

3. 软件系统应用范围：Android 手机用户

## 1.3. 术语

ChatEntity--聊天实体表

UserInfo--用户信息表

name--姓名

sex--性别

Location--位置

LocationRequestEntity--位置请求表

LocationResponseEntity--位置请求的响应表

password--密码

content--聊天

ConnectedApp--连接系统服务器，调用数据库

RegisterActivity--记录实体信息

ChatMsgReceiver--聊天记录接收

ChatServiceData--聊天数据调用和存储

FriendListInfo--好友列表信息

LocationRequestReceiver--位置信息请求和响应

MyLocation--个人位置信息定位

ChatEmoticonUtil--在线好友匹配查找

ReadyMadeDialog--讯息接收状态

LocationAcceptActivity--位置请求接收状态

SearchFriendByNameActivity--通过姓名查找好友

SearchFriendByOnlineActivity--通过在线状态查找好友

FriendSearchResultActivity--好友查找结果反馈

FriendListGroupItem--好友管理

DbSaveOldMsg--信息记录状态

LocationInfo--位置信息

ChatInfo--聊天记录

UserInfo--<object>--用户信息

FriendInfo--好友信息

FriendList--好友列表

FriendSearchResult--好友查询

FightDetailActivity--记录并存储航班详细信息

SearchFightResultActivity--航班信息匹配查询并返回结果

CarInfo--机场停车信息查询与更新

SearchresultActivity--航班信息查询并规划实时线路室内导航

planTakeoffTime--计划起飞时间

realTakeoffTime--实际起飞时间

planArrivalTime--计划到达时间

realArrivalTime--实际到达时间

takeoffcity--出发地

arrivalcity--目的地

enter--登机口

MapActivity--地图显示

ShopDetailActivity--商店等详细信息查询

SearchresultActivity--输入信息的匹配查询功能

ArrayList<LocationRegion>--起终点

ArrayList<LocationRegion>--中转点

ImageView lockcenter--定位点

lrpointto--LocationRegion--触发点

zoomin--ImageView--放大点

zoomout ImageView--缩小点

List<PathRoutingManager.SwitchFloorInfo> routeinfo--路径导航

lockcenter--中心定位

distanceView--距离计算

List<PathRoutingManager.SwitchFloorInfo> routeinfo--导航路径

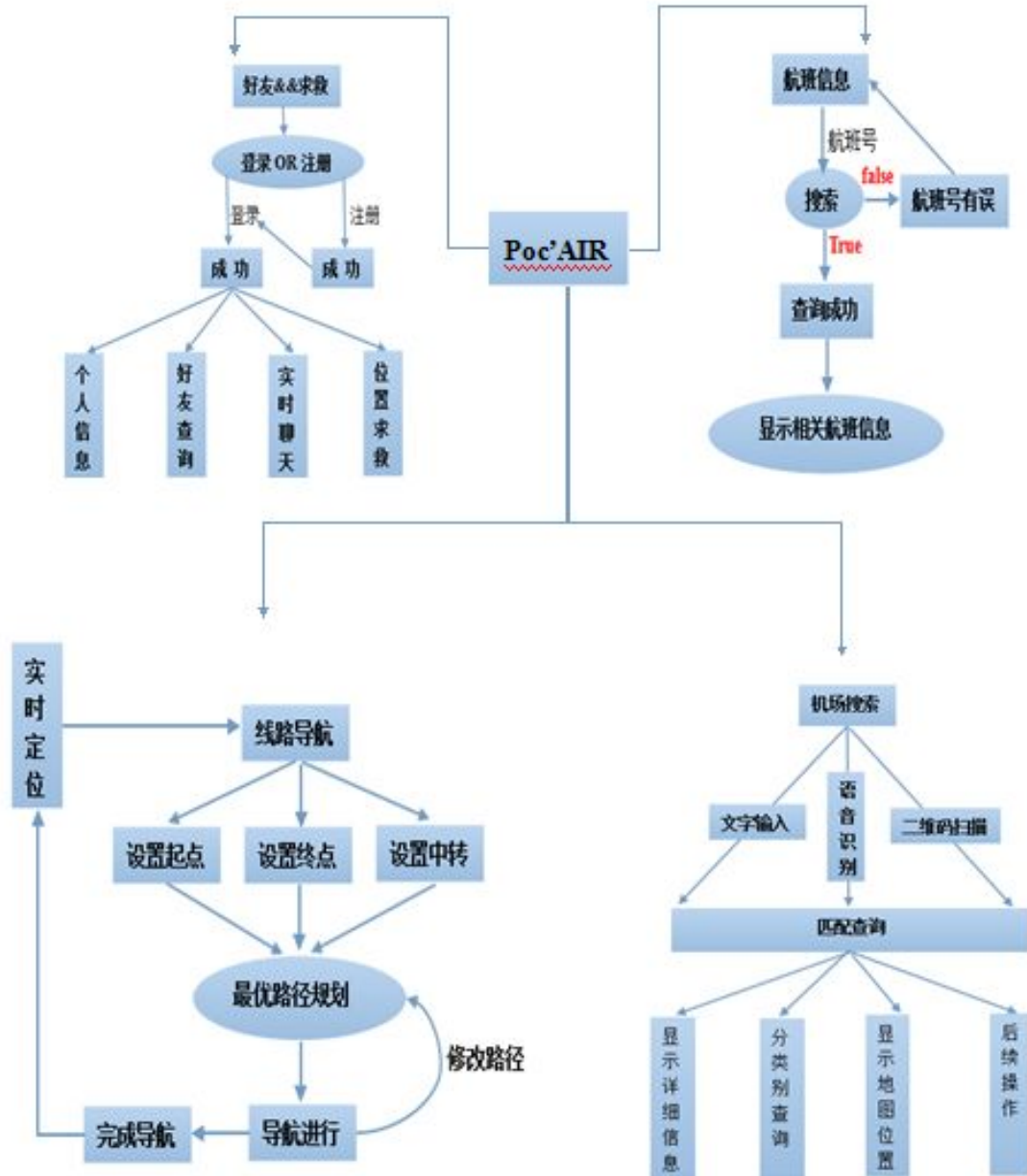
distance--路径距离

List<PathRoutingManager--路线查询

## 1.4. 参考资料

1. 《Android 设计与开发》--机械工业出版社
2. 《疯狂 Android 讲义》--机械工业出版社
3. 《Android 系统级深入开发》--机械工业出版社

## 2. 总体设计—系统功能结构



## 2.1. 运行环境

### 2.1.1. 硬件环境

1. 本软件运行对服务器、客户端的硬件要求：

服务器：一台完备的计算机系统

客户端：两台 Android4.3 及以上版本的智能手机、手机可用内存不小于 10M

2. 本软件运行所使用的外围设备：

无线 WiFi 网络支持

### 2.1.2. 软件环境

本软件运行所使用的计算机软件及版本：

- 1) 操作系统：Win7 及以上
- 2) 数据库系统：MYSQL
- 3) 开发平台及工具：装载有完备 SDK 的 Eclipse 开发环境
- 4) 通信协议：HTTP 协议

## 2.2. 系统的关键技术

1. 实时定位：通过无线 WiFi 对用户进行实时定位，然后进行后续导航和线路规划服务。
2. 最优线路规划：通过设置起始点、中转点，自动规划出最优导航路径。
3. 线路纠偏：在规划路径当中实现实时纠偏功能，当用户偏离规划路线时，系统自动实时规划出合适的路线保证最优化。
4. 紧急求救：通过紧急求救功能可以向自己的好友发送消息和自己的定位消息，可以起到安全保护作用和处置各种突发情况。
5. 语音识别：通过语音识别，作为输入信息进行各种查询。
6. 二维码扫描：通过对航班、商店、厕所灯建筑的二维码进行扫描，从而达到输入信息的作用，进而进行信息查询。



7. 机场地图路径设计：JOSM。
8. 社交娱乐：可以对使用同一软件的人通过添加好友，进行聊天、位置分享、紧急求救等。
9. 搜索查询：可以进行楼层、地点名、类别名等进行快速、准确的查询并显示在机场地图上。

### 3. 构件设计

#### 3.1. 构件列表

构件列表				
序号	构件分类	构件名称	依赖关系说明	复用情况
1	功能构件	机场服务	无	无
2	功能构件	航班信息	无	无
3	功能构件	室内导航	无	无

#### 3.2. 构件集成说明

集成顺序：

机场服务—>航班信息—>室内导航

集成的策略：

自底向上的集成策略；

集成就绪准则：

- （1）各个模块能够实现且模块测试通过。
- （2）各个接口完成且通过运行。
- （3）完成统一规则。

集成结束准则：

- （1）程序可以完整无误运行。
- （2）模块功能的最优化。
- （3）各模块的最低耦合性。
- （4）多次运行，系统服务器无崩溃等问题。

#### 3.3. 构件 1——机场服务

该构件属于三大功能构件之一，名称是“机场服务”，主要功能和设计目标是：用户在机场时，软件的该功能构件可以为用户提供人性化的相关配套机场服务，如附近出租车信息、接机服务等等。该构件的核心功能在于“好友&&求救”

模块，实现了该软件的社交功能，可以为使用该软件的用户提供一个社交网络平台，可以相互加好友、实时聊天等，最重要的是能够向好友紧急求救，通过发送自己的位置信息来应对某些突发情况以求能够快速解救。

### 3.3.1. 功能模块列表

模块列表					
构件名称	模块编号	模块名称	对应需求功能编号	所对应需求功能	实现优先级
机场服务	1-1	好友&&求救	自己添加功能	自己添加功能	低
	1-2	其他功能	自己添加功能	自己添加功能	低

### 3.3.2. 功能模块 1—好友&&求救

#### 【模块编号】

1-1

#### 【功能描述】

“好友&&求救”模块，实现了该软件的社交功能，可以为使用该软件的用户提供一个社交网络平台，可以相互加好友、实时聊天等，最重要的是能够向好友紧急求救，通过发送自己的位置信息来应对某些突发情况以求能够快速解救。

#### 【前置条件】

- (1) 用户个人信息数据库。
- (2) 位置定位信息准确性。
- (3) 网络覆盖。

#### 【操作者】

用户

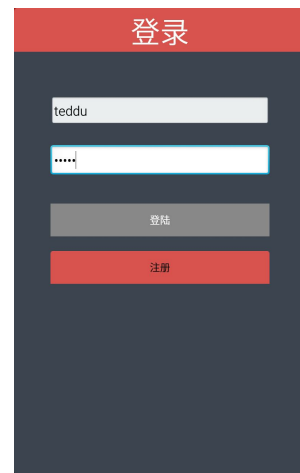
## 【与本模块相关的代码表和表】

名称	中文注释	类型		作用
		代码表	表	
ChatEntity	聊天实体表		√	用户数据表
LocationRequestEntity	位置请求表		√	提供用户的位置请求数据表
LocationResponseEntity	位置请求的响应表		√	对位置请求所响应的数据表
UserInfo	用户信息表		√	用户基本信息

## 【界面】



1-1.机场服务模块主界面



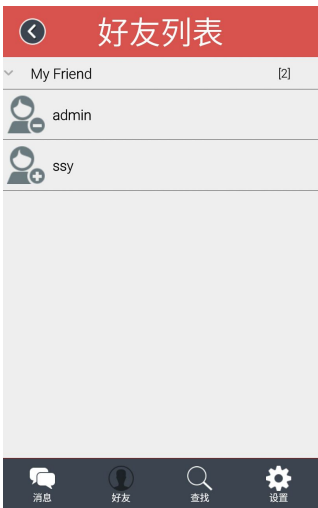
1-2.好友&amp;&amp;求救登录界面



1-3.用户注册界面



1-4.好友聊天及位置请求



1-5.好友列表界面



1-6.好友查找界面



1-7.个人信息设置界面

**【输入】**

用户名: name--String

密 码: password--String

性 别: sex--String

生 日: mBirthYear--int

mBirthDay--int

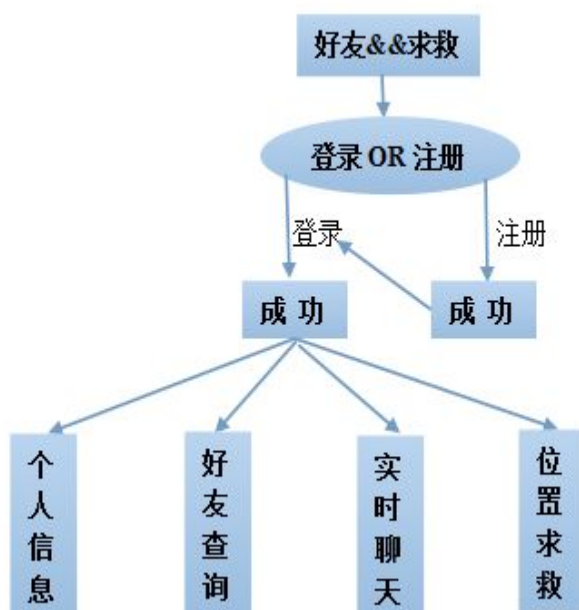
mBirthDay--int

聊 天: content--String

位 置: Location--String

**【处理流程及说明】**

流程图:



流程说明:

**【算法】**

- 1.ConnectedApp--连接系统服务器，调用数据库
- 2.RegisterActivity--记录实体信息
- 3.ChatMsgReceiver--聊天记录接收
- 4.ChatServiceData--聊天数据调用和存储

- 5.FriendListInfo--好友列表信息
- 6.LocationRequestReceiver--位置信息请求和响应
- 7.MyLocation--个人位置信息定位
- 8.ChatEmoticonUtil--在线好友匹配查找
- 9.ReadyMadeDialog--讯息接收状态
- 10.LocationAcceptActivity--位置请求接收状态
- 11.SearchFriendByNameActivity--通过姓名查找好友
- 12. SearchFriendByOnlineActivity--通过在线状态查找好友
- 13.FriendSearchResultActivity--好友查找结果反馈
- 14.FriendListGroupItem--好友管理
- 15.DbSaveOldMsg--信息记录状态

#### 【输出】

- 位置信息: LocationInfo--String
- 聊天记录: ChatInfo--String
- 用户信息: UserInfo--<object>
- 好友信息: FriendInfo--String<object>
- 好友列表: FriendList--<object>
- 好友查询: FriendSearchResult--<object>

### 3.4. 构件 2--航班信息

该构件主要实现的功能是：用户到达机场之后，可以通过输入航班次号对自己乘坐的航班进行相关查询，如登机口、值机信息、起飞时间、到达时间等，达到人性化需求，使乘客更能从此款软件中体验方便、快捷的使用体验。

### 3.4.1. 功能模块列表

模块列表					
构件 名称	模块 编号	模块 名称	对应需求 功能编号	所对应 需求功能	实现 优先级
航班信息	2-1	航班查询	/	人性化需求	中

### 3.4.2. 功能模块 1

#### 【模块编号】

2-1

#### 【功能描述】

用户到达机场之后，可以通过输入航班次号对自己乘坐的航班进行相关查询，如登机口、值机信息、起飞时间、到达时间等，达到人性化需求，使乘客更能从此款软件中体验方便、快捷的使用体验。

#### 【前置条件】

- (1) 航班信息数据库
- (2) 无线网络覆盖

#### 【操作者】

用户

#### 【与本模块相关的代码表和表】

名称	中文注释	类型		作用
		代码表	表	
FightInfo	航班信息表		√	航班信息表



## 【界面】



2-1.航班搜索主界面



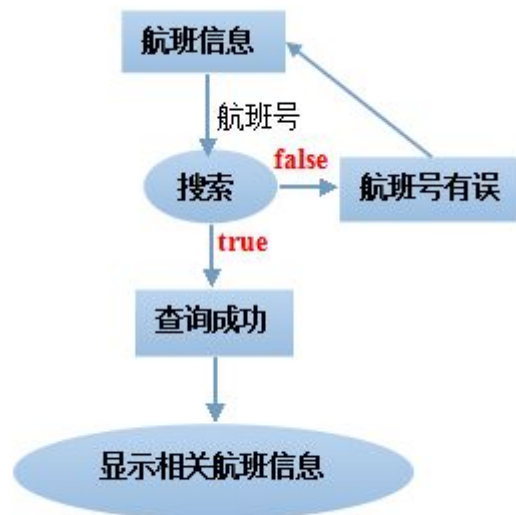
2-2.航班详情主界面

## 【输入】

航班号: flightnumber--String

## 【处理流程及说明】

流程图:



流程说明：

**【算法】**

- 1.FightDetailActivity--记录并存储航班详细信息
- 2.SearchFightResultActivity--航班信息匹配查询并返回结果
- 3.CarInfo--机场停车信息查询与更新
- 4.SearchresultActivity--航班信息查询并规划实时线路室内导航

**【输出】**

计划起飞时间：planTakeoffTime--String

实际起飞时间：realTakeoffTime--String

计划到达时间：planArrivalTime--String

实际到达时间：realArrivalTime--String

出 发 地：takeoffcity--String

目 的 地：arrivalcity--String

登 机 口：enter--String

### 3.5. 构件 3—室内导航

该构件是项目的最核心构件，也集中了比赛的大部分需求，该构件主要实现的核心功能如下：

1. 实时定位：通过无线 WiFi 对用户进行实时定位，然后进行后续导航和线路规划服务。
2. 最优线路规划：通过设置起始点、终点和中转点等，自动规划出最优导航路径。
3. 线路纠偏：在规划路径当中实现实时纠偏功能，当用户偏离规划路线时，系统自动实时规划出合适的路线保证最优化。
4. 语音识别：在机场设施搜索的时候，通过语音识别，作为输入信息进行各种查询。
5. 二维码扫描：通过对航班、商店、厕所灯建筑的唯一二维码进行扫描，从而达到输入信息的作用，进而进行信息查询。
6. 搜索查询：可以进行楼层、地点名、类别名等进行快速、准确的查询并显

示在机场地图上。

### 3.5.1. 功能模块列表

模块列表					
构件名称	模块编号	模块名称	对应需求功能编号	所对应需求功能	实现优先级
机场搜索	3-1	实时定位	\	实时定位	高
	3-2	搜索查询	\	线路导航	高
	3-3	线路导航	\	搜索查询	高

### 3.5.2. 功能模块 1

#### 【模块编号】

3-1

#### 【功能描述】

用户在指定的机场活动时，通过对自己所在位置进行实时定位，完成后续的线路导航以及其他相关服务。

#### 【前置条件】

- (1) 在指定机场活动
- (2) 无线网络覆盖
- (3) 服务器连接运行正常

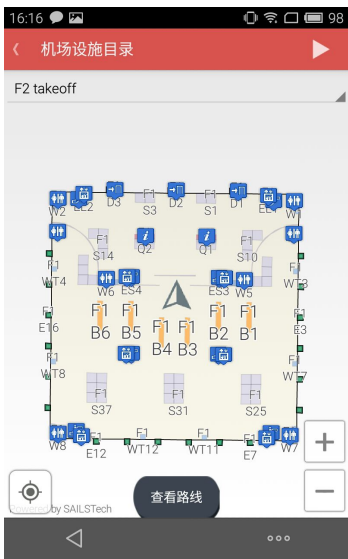
#### 【操作者】

用户

#### 【与本模块相关的代码表和表】

名称	中文注释	类型		作用
		代码表	表	
无	无	\	\	\

【界面】



3-1 实时定位界面

【输入】

触发定位按钮

【处理流程及说明】

流程图：

无

流程说明：

【算法】

通过调用 SAILS\_SDK\_V1.73\_Publisher.jar 实现该定位功能。

【输出】

在地图上显示当前位置并标识出来。

3.5.3. 功能模块 2

【模块编号】

3-2

【功能描述】

该功能模块是比赛项目的最核心要求之一，主要有以下功能：

- (1) 输入查询：通过输入文字信息或者通过语音信息，作为输入项，触发搜索按钮，实现快速查找的功能，同时在地图上显示出查询结果，以

便进行后期操作。

- (2) 直接查询：该软件已经实现分类别、分楼层搜索查询，不需要输入信息，直接触发相应的搜索事件进行分楼层、分类别查询。
- (3) 扫描查询：通过扫描商店、安检口、地标等建筑的二维码，可以查询相关建筑的详细信息，包括：显示地图位置、实时距离、商店信息等。

【前置条件】

- (1) 在指定机场活动
- (2) 无线网络覆盖
- (3) 服务器连接运行正常

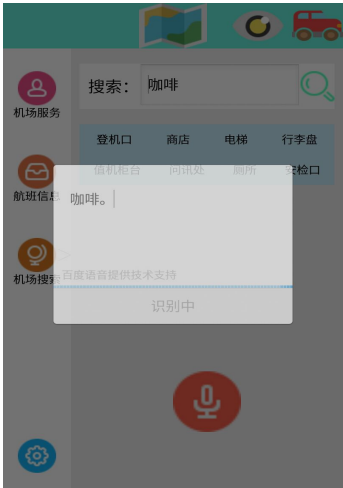
【操作者】

用户

【与本模块相关的代码表和表】

名称	中文注释	类型		作用
		代码表	表	
SearchEntity	搜索实体表	√		实体数据库表
ShopInfo	商店信息表	√		商店信息表

【界面】



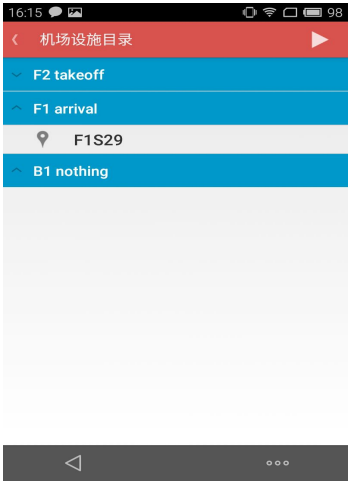
3-2 语音输入界面



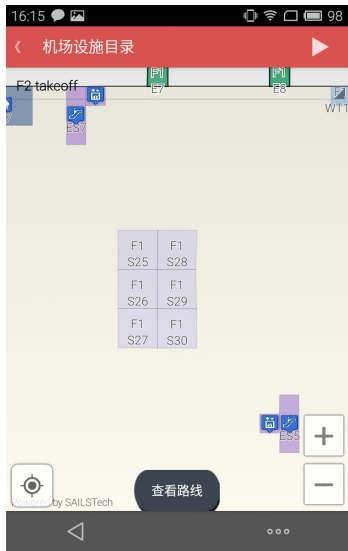
3-3 二维码扫描输入界面



3-4 搜索结果列表 1



3-5 搜索结果列表 2



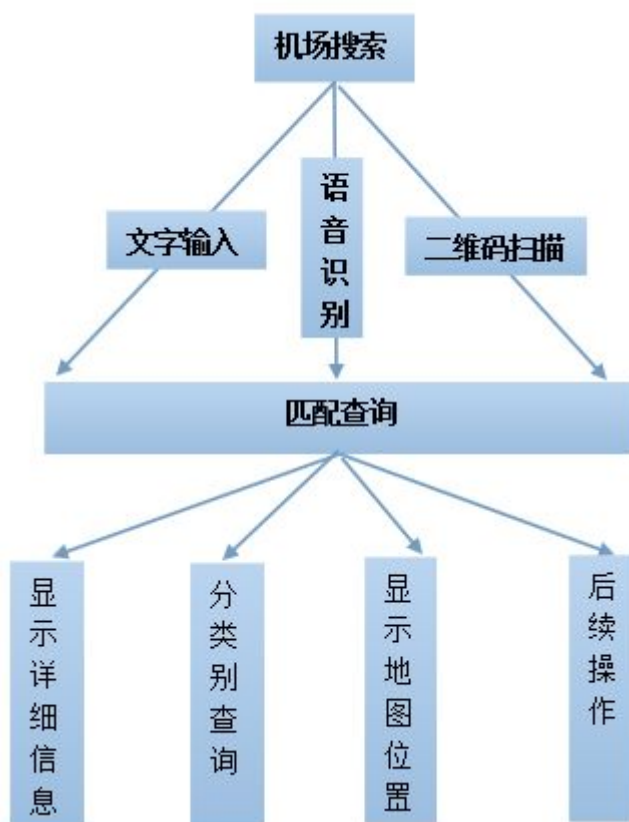
3-6 搜索结果地图显示

【输入】

- (1) 输入查询：通过输入文字信息或者通过语音信息，作为输入项。
- (2) 直接查询：该软件已经实现分类别、分楼层搜索查询，不需要输入信息，直接触发相应的搜索事件进行分楼层、分类别查询。
- (3) 扫描查询：扫描商店、安检口、地标等建筑的二维码。

**【处理流程及说明】**

流程图：



流程说明：

**【算法】**

- 1.MapActivity--地图显示
- 2.ShopDetailActivity--商店等详细信息查询
- 3.SearchresultActivity--输入信息的匹配查询功能

**【输出】**

- (1) 相关建筑的详细信息，包括：显示地图位置、实时距离、商店信息等。
- (2) 在地图上显示出查询结果。

### 3.5.4. 功能模块 3

**【模块编号】**

3-3

**【功能描述】**

该功能模块是比赛项目的最核心要求之一，主要有以下功能：

- （1）进入室内地图导航功能，通过自我实时定位默认作为起始点，点击图中的建筑物，设置中转点和终点，系统自动规划导航路线，并显示出该路径的路程。
- （2）在导航过程中，如果用户出现了路径偏差，系统会自动以终点为中心，实时调整导航路线。
- （3）在即将到达中转点或终点时，自动感应方向，实行精准导航。

**【前置条件】**

- （1）在指定机场活动
- （2）无线网络覆盖
- （3）服务器连接运行正常
- （4）能够正常定位

**【操作者】**

用户和服务器共同完成

**【与本模块相关的代码表和表】**

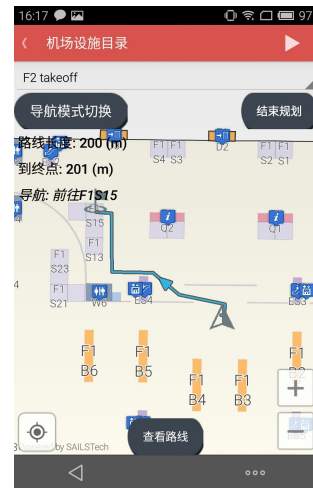
名称	中文注释	类型		作用
		代码表	表	
TabMsgItemEntity	导航路径数据实体	√		导航路径数据实体
LocationRequestEntity	位置请求数据库		√	位置请求数据库
LocationResponseEntity	位置请求相应数据库		√	位置请求相应数据库



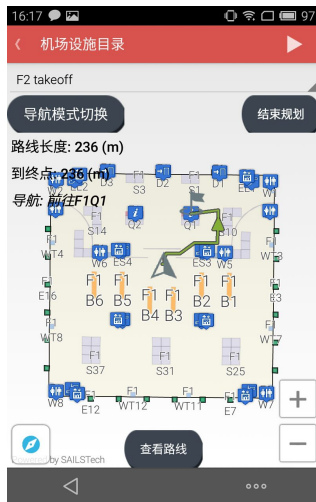
## 【界面】



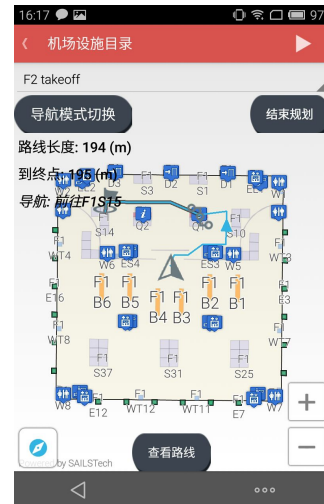
3-7 设置起点、终点、中转点等



3-8 自动规划路线导航



3-9 设置中转点



3-10 查看路线

## 【输入】

起终点: ArrayList<LocationRegion>

中转点: ArrayList<LocationRegion>

定位点: ImageView lockcenter

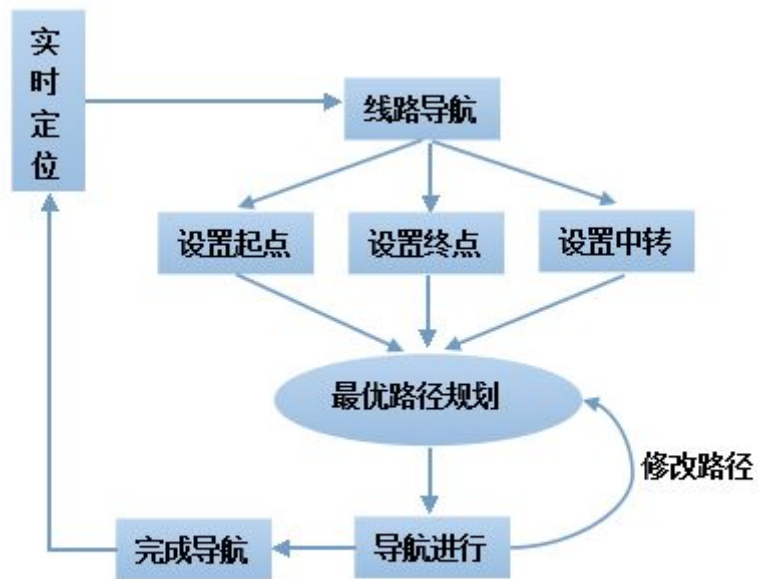
触发点: Irpointto--LocationRegion

放大点: zoomin--ImageView

缩小点: zoomout ImageView

**【处理流程及说明】**

流程图：



流程说明：

**【算法】**

1. List<PathRoutingManager.SwitchFloorInfo> routeinfoList--路径导航
2. lockcenter--中心定位
3. distanceView--距离计算

**【输出】**

导航路径：List<PathRoutingManager.SwitchFloorInfo> routeinfoList

路径距离：distance--int

## 4. 接口设计

构件接口列表				
构件名称	接口编号	接口名称	接口类型	说明
机场服务	01-01	UserInfo	内部	用户数据信息接口
	01-02	SAILS_SDK_V1.73_Published	外部	实现位置定位
航班信息	02-01	FightInfo	内部	航班数据信息接口
室内导航	03-01	ShopInfo	内部	商店数据信息接口
	03-02	zxing	外部	二维码扫描
	03-03	VoiceRecognition	外部	语音识别
	03-04	galaxy	外部	百度语音识别 1
	03-05	Baidu-SpeechRecognitionUI-S DK	外部	百度语音识别 2
	03-06	commons-math	外部	数学计算--距离计算
公共接口	04-01	Config	内部	统一编码类型
	04-02	Constants	内部	全局变量
	04-03	DptoPx	内部	获取屏幕像素
	04-04	ClientListenThread	内部	服务器数据返回
	04-05	ClientSendThread	内部	服务器发送请求
	04-06	NetConnect	内部	网络连接
	04-07	NetStateReceiver	内部	接收网络状态，检测能不能 连接到服务器
	04-08	NetworkService	内部	发消息之前协议规定

## 4.1. 外部接口

### 4.1.1. 接口 1

接口编号	01-02
接口名称	SAILS_SDK_V1.73_Published
接口说明	实现位置定位
数据来源	网络获得
调用者	LocationAcceptActivity
输入	/
输出	/
调用方式说明	Import/extend

### 4.1.2. 接口 2

接口编号	03-02
接口名称	zxing
接口说明	二维码扫描
数据来源	接口获得
调用者	SearchActivity
输入	/
输出	/
调用方式说明	import/extend

### 4.1.3 接口 3

接口编号	03-03
接口名称	VoiceRecognition
接口说明	语音识别
数据来源	语音输入
调用者	SearchActivity
输入	语音
输出	语音识别结果
调用方式说明	import/extend

### 4.1.4 接口 4

接口编号	03-04
接口名称	galaxy
接口说明	百度语音识别 1
数据来源	语音输入
调用者	SearchActivity
输入	语音
输出	语音识别结果
调用方式说明	import/extend

### 4.1.5 接口 5

接口编号	03-05
接口名称	Baidu-SpeechRecognitionUI-SDK
接口说明	百度语音识别 2
数据来源	语音输入
调用者	SearchActivity
输入	语音
输出	语音识别结果
调用方式说明	import/extend

### 4.1.3. 接口 6

接口编号	03-06
接口名称	commons-math
接口说明	数学计算--距离计算
数据来源	位置坐标转换
调用者	MapActivity
输入	起点、终点位置
输出	线路路径
调用方式说明	import/extend

## 4.2. 内部接口

### 4.2.1. 接口 1

#### (1) 接口属性设计

接口编号	01-01
接口名称	UserInfo
接口说明	用户数据信息接口
数据来源	数据库表存储
调用者	MainHelpActivity
输入	界面输入
输出	用户信息生成
处理流程	界面输入->用户信息生成->数据存储->数据调用

#### (2) 类设计

类名称	UserInfo
分类	/
描述	用户数据信息
使用到的其他类	无
属性及方法描述	get()/set()
使用/交互	import
其他	/

## 4.2.2 接口 2

### (1) 接口属性设计

接口编号	02-01
接口名称	FightInfo
接口说明	航班数据信息
数据来源	数据库表存储
调用者	SearchActivity
输入	FightNumber
输出	FightInfo
处理流程	输入航班号->匹配查询->输出航班信息

### (2) 类设计

类名称	FightInfo
分类	/
描述	航班数据信息
使用到的其他类	无
属性及方法描述	get()/set()
使用/交互	import
其他	/



### 4.2.3 接口 3

#### (1) 接口属性设计

接口编号	03-01
接口名称	ShopInfo
接口说明	商店数据信息接口
数据来源	数据库表存储
调用者	SearchActivity
输入	ShopID
输出	ShopInfo
处理流程	二维码扫描->系统调用->返回商店详细信息

#### (2) 类设计

类名称	ShopInfo
分类	/
描述	商店数据信息
使用到的其他类	无
属性及方法描述	get()/set()
使用/交互	import
其他	/

## 4.2.4 接口 4

### (1) 接口属性设计

接口编号	04-01
接口名称	Config
接口说明	统一编码类型
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	Config
分类	/
描述	统一编码类型
使用到的其他类	无
属性及方法描述	void setCurrentPropIndex
使用/交互	import
其他	/

## 4.2.5 接口 5

### (1) 接口属性设计

接口编号	04-02
接口名称	Constants
接口说明	全局变量
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	Constants
分类	/
描述	全局变量
使用到的其他类	无
属性及方法描述	static final
使用/交互	import
其他	/

## 4.2.6 接口 6

### (1) 接口属性设计

接口编号	04-03
接口名称	DptoPx
接口说明	获取屏幕像素
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	DptoPx
分类	/
描述	获取屏幕像素
使用到的其他类	无
属性及方法描述	static int dip2px
使用/交互	import
其他	/

## 4.2.7 接口 7

### (1) 接口属性设计

接口编号	04-04
接口名称	ClientListenThread
接口说明	服务器数据返回
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	ClientListenThread
分类	/
描述	服务器数据返回
使用到的其他类	无
属性及方法描述	ClientListenThread(Context par, Socket s)、void run()
使用/交互	import
其他	/

## 4.2.8 接口 8

### (1) 接口属性设计

接口编号	04-05
接口名称	ClientSendThread
接口说明	服务器发送请求
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	ClientSendThread
分类	/
描述	服务器发送请求
使用到的其他类	无
属性及方法描述	synchronized void start(Socket socket, String str0)
使用/交互	import
其他	/

## 4.2.9 接口 9

### (1) 接口属性设计

接口编号	04-06
接口名称	NetConnect
接口说明	网络连接
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	NetConnect
分类	/
描述	网络连接
使用到的其他类	Thread
属性及方法描述	void run()
使用/交互	import
其他	/

## 4.2.10 接口 10

### (1) 接口属性设计

接口编号	04-07
接口名称	NetStateReceiver
接口说明	接收网络状态，检测能不能连接到服务器
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	NetStateReceiver
分类	/
描述	接收网络状态，检测能不能连接到服务器
使用到的其他类	无
属性及方法描述	void onReceive(Context context, Intent intent)
使用/交互	import
其他	/



## 4.2.11 接口 11

### (1) 接口属性设计

接口编号	04-08
接口名称	NetworkService
接口说明	发消息之前协议规定
数据来源	/
调用者	系统公共接口
输入	/
输出	/
处理流程	/

### (2) 类设计

类名称	NetworkService
分类	/
描述	发消息之前协议规定
使用到的其他类	无
属性及方法描述	void setupConnection()
使用/交互	import
其他	/

## 5. 系统出错处理

### 5.1. 出错信息

错误分类	子项及其编码	错误名称	错误代码	备注
数据库错误	连接	连接超时	100001001	
		连接断开	100001002	
	数据库本身错误代码	数据库本身错误代码	100002+数据库错误代码	
服务器连接错误	连接	连接超时	101001001	
		连接断开	101001002	
	其它 TCP 连接错误(server 自身错误代码)		101002+ server 错误代码	
配置信息错误	未配置输入参数		102001	
	未配置输出参数		102002	

### 5.2. 补救措施

- 1.后备技术：当原始系统数据万一丢失时启用的副本的建立和启动的技术。  
例如，周期性地把磁盘信息记录到磁带上。
- 2.降效技术：使用另一个效率稍低的系统或方法来求得所需结果的某些部分。  
例如，一个自动系统的降效技术可以是手工操作和数据的人工记录。
- 3.恢复及再启动技术：说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行。