

详细设计说明书

One 团队

目录

一. 编写目的	3
二. 背景	3
三. 系统结构	3
四. 模块设计	4
1.服务器选择模块.....	4
2.地图导航交互模块.....	5
3.目的地选择模块.....	12
五. 程序类设计	13
1.CookieHelper 类.....	13
2.Beeline 类	14
3.Tools 类	15
4.DataBase 类	17
5.DBItem、XYPos 类.....	17
六. UI 界面设计	18
1.界面设计	18
2.动画设计	21
3.UI 控件优化.....	23

一. 编写目的

本说明书用于将前期的软件需求、设计思路进一步具体化，形成较为明确的 程序需求。本文档包括详细的程序描述，读者为用户代表以及本团队的成员。

二. 背景

项目名称：室内交互式引导 APP

委托单位：第四届中国软件杯大学生软件设计大赛组委会

开发单位：One 团队

开发日期：2015 年 5 月

三. 系统结构

系统功能模块包括：服务器地址选择模块，地图导航交互模块，目的地选择模块。

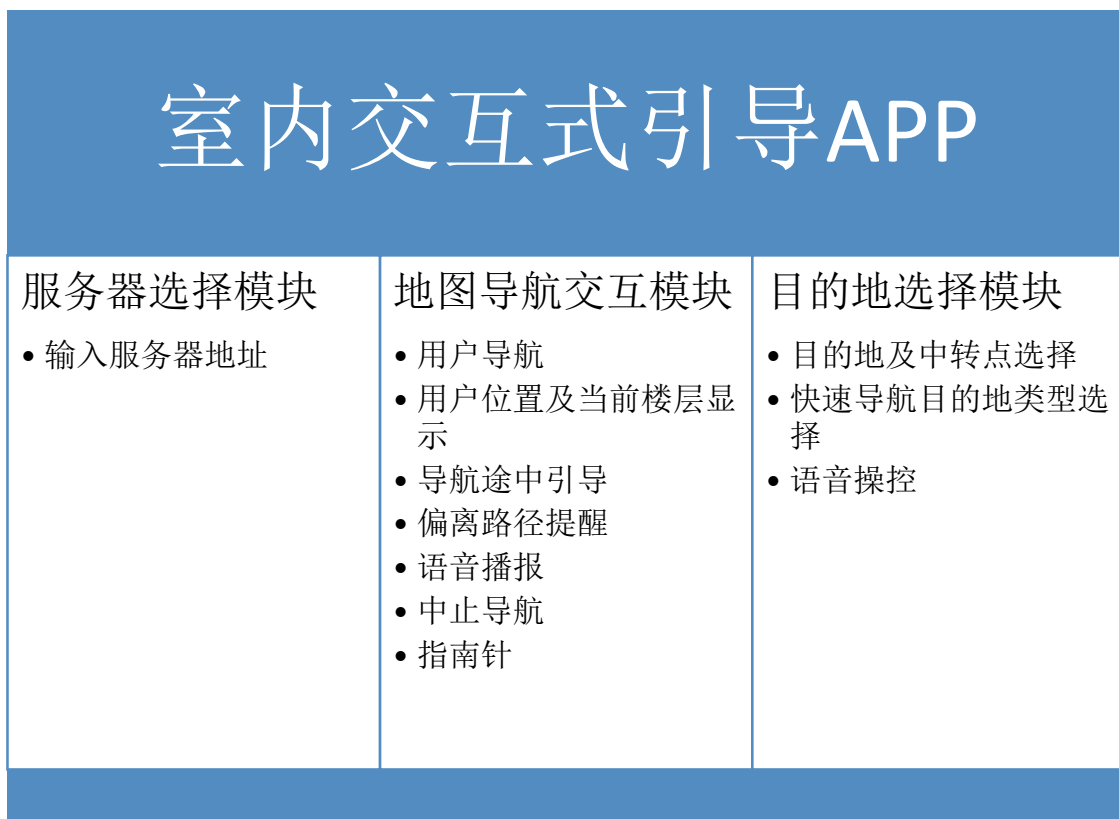


图 1 系统功能模块图

服务器选择模块包括：服务器地址选择。

地图导航交互模块包括：用户导航、用户位置及当前楼层显示、用户交互提醒、偏离路径提醒、语音播报、中止导航。

目的地选择模块包括：目的地及中转点选择、快速导航目的地类型选择、语音操控

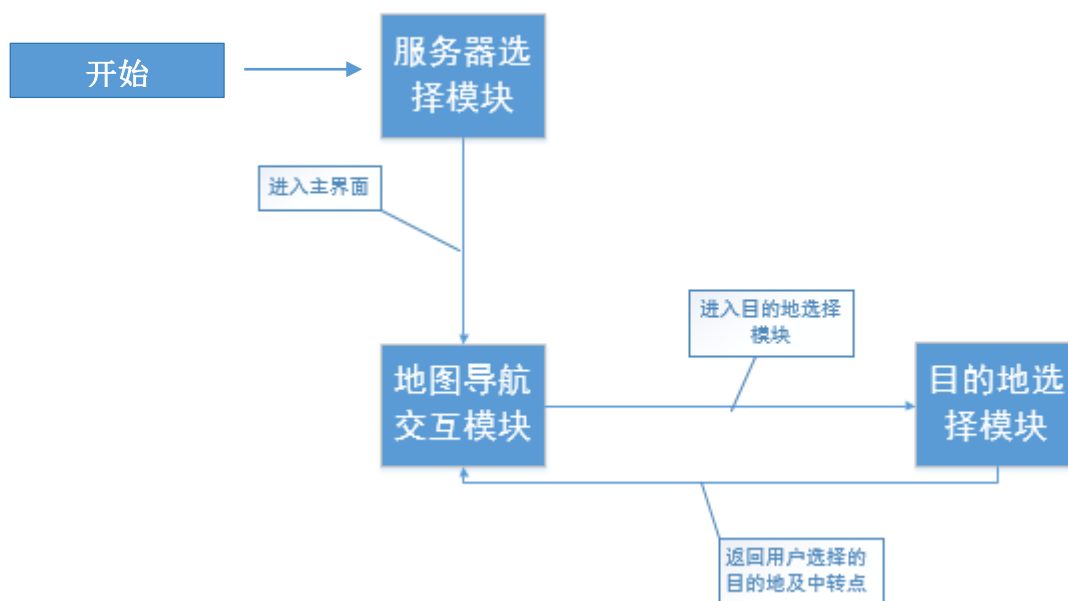


图 2 系统模块关系图

四. 模块设计

1.服务器选择模块

模块描述

该模块主要是为了方便使用不同的服务器运行定位程序的时候能够方便的修改服务器地址，不必每次修改程序源码。

模块功能

实现用户输入服务器地址，在进入主界面的时候存入本地缓存中，以供

程序调用获取用户位置信息。

模块性能

该模块过于简单，所以不涉及性能评估。

模块输入输出项

输入项：用户输入的服务器地址字符串

输出项：本地缓存中服务器地址

模块设计

用 **TextBox** 控件来接收用户输入数据，通过调用本地缓存 **API** 来讲用户输入内容存入本地缓存。

模块流程



图 3 服务器选择模块流程图

2.地图导航交互模块

模块描述

该模块为 **APP** 主要功能模块，大部分内容、功能均在此模块。其他两个模块均是为此服务的。在此模块内，用户可以看见自己当前所在的位置以及进行和导航相关的所有事情，完成一次室内导航的全部需求。

模块功能

该模块可以实现如下的功能：实现地图的显示，并且可以根据地理方向

进行相应的旋转；根据本地缓存中服务器地址，发送 **Http Get** 请求从服务器获取用户当前位置数据并显示在地图上；根据从目的地选择模块传过来的目的地或中转点进行路径规划并显示在地图上；可以根据地理北极自动旋转地图并显示指南针，保证地图的正确方向；中止当前的导航，恢复初始状态；根据用户路径途径点来让走过的路径逐步消失；偏离路径检测，当用户偏离预定路线的时候将进行震动和弹窗提醒用户，并重新规划路径；导航开始、更改楼层、到达目的地友好交互，弹窗提醒并语音播报；定位功能，可突出显示用户当前的位置。

模块性能

精度

本应用是导航应用，对于数据处理的精度要求较严格。对于位置数据，数据精度为 **0.0001**，严格匹配服务器数据显示在地图上。对于点坐标数据，数据精度为 **0.01**，偏差为 **0.005**。对于地图上点之间距离数据精度为 **0.1cm**，偏差为 **0.2cm**。

时间特性

本应用要求实时显示用户当前的位置，并对用户选择的路径进行快速响应。所以设置刷新用户位置数据的时间为 **1s**，设置用户路径规划时间为毫秒级。整体用户响应为即时响应，毫秒级延迟。

可靠性

由于本应用为室内导航，要求精确导航，需要应用的可靠性较高。所以在程序中不仅仅设置各种用户违法操作的静止提示，同时对用户数据进行严格检查限制，每次导航结束、导航终止、程序挂起、页面跳转时候对标志域字段进行严格控制初始化。

灵活性

在导航过程中可能出现目的地更改、临时想去厕所、取消导航等等多种

情况，为了应对各种用户使用情况，增加程序的灵活性，在程序中增加临时中止导航等多种情况处理，可在任意时候更改当前导航状态，保证程序灵活实用性。

模块输入输出项

输入项：服务器位置信息

输出项：地图上用户位置点以及当前楼层显示

输入项：从目的地选择输入的目的地信息

输出项：在地图上画出当前位置到目的地的路径

输入项：用户位置信息移动

输出项：逐步删除走过的路径，如到达目的地提醒，如偏离路径提醒

模块设计

地图显示：三个 **Canvas** 画布显示地图，分别控制三个地图的显隐来显示当前所在的楼层。同时通过将陀螺仪的角度数据绑定到地图的选择角度数据来实现地图根据地理方向自动旋转的功能。

用户位置获取和显示：通过 **Http Get** 请求从服务器获取位置信息，因为两次请求间隔为 1s，**Http API** 默认会调用缓存信息，为了拿到真实的位置信息，将 **Http Get** 请求的 **HttpRequestHeader** 的 **Pragma** 和 **CacheControl** 设置为 "no-cache" 来禁止读取缓存；通过控制 **Canvas** 上 **Ellipse** 控件的位置来显示用户位置，具体位置是通过服务器相对坐标映射到地图上。

导航路径显示：首先计算距离用户当前位置最近的点，然后通过用该点和用户选择好的目的地来调用 **Calculate** 函数来获取最短路径所途径的所有点信息，通过查询点坐标库来依次画到地图上，并在点之间连线画出路径，然后画出用户当前位置到第一个点的线。

用户导航：根据规划好的路径，存储当前用户所走在的两点之间的两个点，如果到达下一个途径点，则将当前两点间的线段擦除，同时更新存储的两点为路径上接下来的两个点，直到到达目的地。

偏离路径重规划:根据用户当前所走在的两点之间,根据两点画围栏(两点连线,在线上每隔 0.3 画一个半径为 0.5 的圆,封装在 `IsOffset` 函数中),判断用户当前位置是否超出围栏,如果超出,则弹窗提醒用户,并终止当前导航,重新开始规划路径。

导航途中引导:导航开始提醒用户,程序中存储着用户选择的目的地或中转点数组,每次到达一个途径点即进行判断,如果该点是目的地或中转点则弹窗延迟 5s 提醒用户到达,并震动 2s。同时对楼层更改进行判断,提醒上楼或者下楼。

导航中止:用户临时取消或更改目的地或中转点,选用此功能,会遍历 `Canvas` 控件的 `Children` 可视化树,将全部 `Line` 控件,即路径全部清除。并初始化关键点数组、路径数据、两点坐标等标志域字段,使程序恢复初始状态。

快速导航:如果选择导航到卫生间,则首先判断用户所在区域,然后取所在区域内的所有卫生间点分别为目的地调用路径规划算法,比较得到最短的一条路径显示在地图上并同正常导航一样进行导航。

用户定位:在各个楼层地图上各放有一个较大的红色圆圈隐藏显示,用户选择定位后,会立即从服务器拿回当前位置信息,并根据位置信息定位红色圆圈并显示 3s,以防止用户找不到当前所在位置而醒目提醒用户。

语音播报:全部提醒都有语音播报,方便行走的时候没注意到手机。

模块标志域字段命名约束

`curPos` 当前所在位置

`lastPoint` 当前点段起点

`nextPoint` 当前点段终点

`route` 规划好路径的所有途径点

`keyPoint` 中转点及目的地

`mapWidth` 地图宽度

`nextPointNum` 当前路径上下一个点编号

`uri` 存储服务器地址

`IsCompassOn` 手机是否支持陀螺仪标志

模块逻辑函数

函数名: **DrawLines**

函数描述: 将路径画在地图上

输入: 无

输出: 一组添加到地图上的 Line 控件

函数名: **DeleteLastLine**

函数描述: 清除最近一条线

输入: 无

输出: 删除地图上最先画的一条线

函数名: **DrawFirstLine**

函数描述: 画出当前位置到第一个关键点路径

输入: 无

输出: 在地图上增加当前位置到第一个途径点的线段

函数名: **PointContinueNext**

函数描述: 检查当前点段是否走完并当前点段继续后移

输入: 无

输出: 如果到达当前点段的终点则将点段后移

函数名: **SetCurPosition**

函数描述: 设置当前坐标的相关内容

输入: 无

输出: 从服务器获取位置信息并将地图正确设置显示

函数名: **PointSectionInit**

函数描述: 点段初始化

输入: 无

输出: 设置当前点段的起止点信息

函数名: **ReachDestination**

函数描述: 判断到达目的地或中转点提醒用户

输入: 无

输出: 到达目的地则通过弹窗和震动提醒用户

函数名: **OffsetCheck**

函数描述: 路径是否走偏并重新规划路径

输入: 无

输出: 检测是否偏离路径, 偏离则提醒用户

函数名: **StopNavigate**

函数描述: 终止导航

输入: 无

输出: 清除地图上路径所有点并初始化点段等数据

函数名: **Toast**

函数描述: 自定义 **Toast** 通知

输入: 提示的内容 (**string**)

输出: 自定义的弹窗消息提醒

函数名: **Speech**

函数描述: 语音播报

输入: 播报的内容 (**string**)

输出: 语音说出 **String** 的内容

函数名: **Cortana**

函数描述: 语音操控

输入: 用户声音输入

输出: 根据用户指令执行相应操作

函数名: ScreenSet

函数描述: 根据设备分辨率适配控件大小

输入: 设备信息

输出: 各种自适应控件大小指数

模块流程

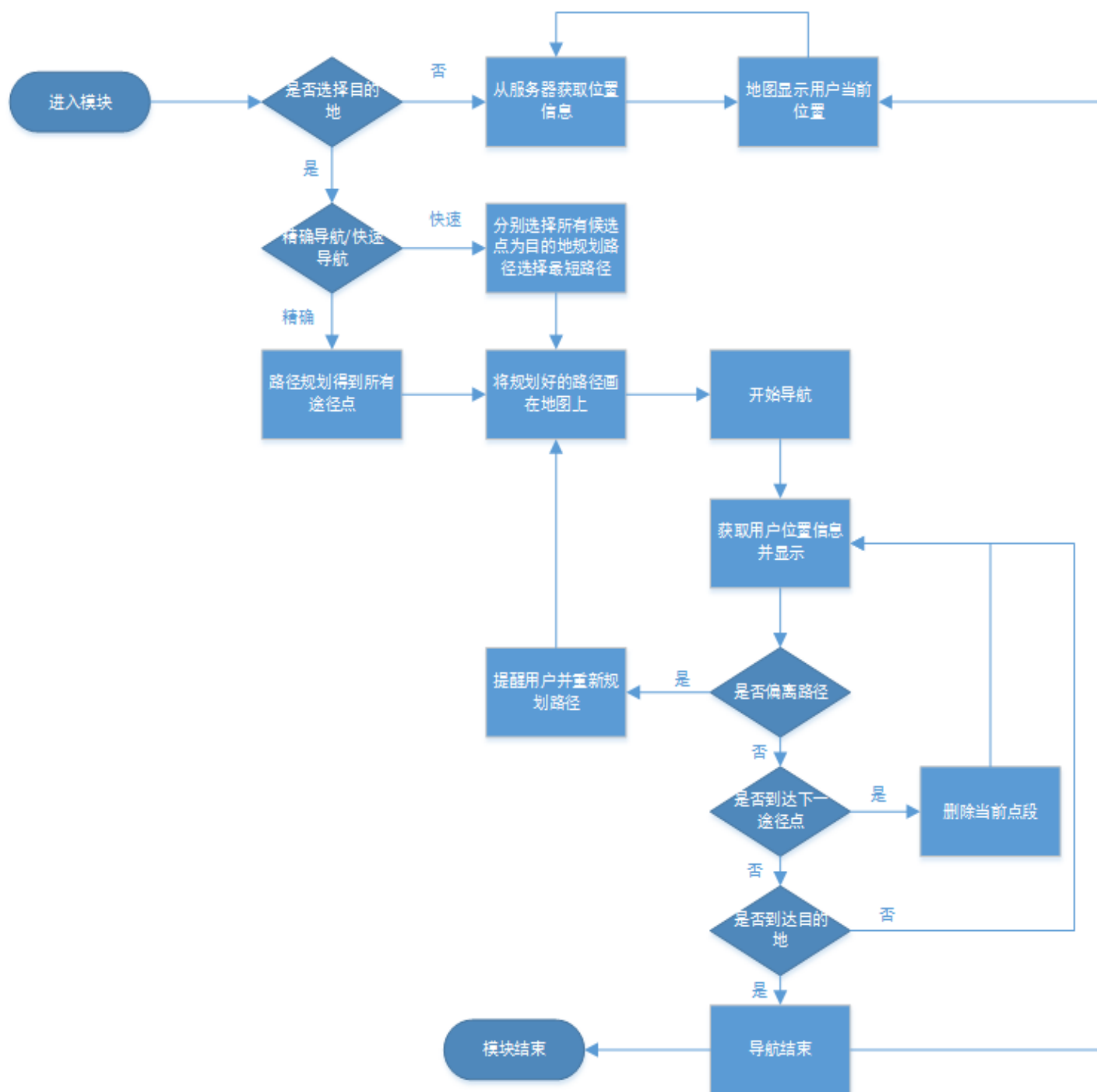


图 4 地图导航交互模块流程图

3.目的地选择模块

模块描述

该模块主要是为实现用户选择中转点及目的地功能。

模块功能

该模块主要是从数据库中取得所有的点以列表形式显示供用户选择，并得到用户的选择后返回给地图导航交互模块进行使用。

模块性能

时间特性

从进入页面到绑定所有点数据，需要毫秒级别延迟，同时在用户选择后能够及时加入已选择列表中，反应需要迅速，以免影响用户体验。

灵活性

保证目的地选择模块能够灵活的增删添改选择的目的地，保证目的地选择的灵活性。

模块输入输出项

输入项：用户选择的目的地或中转点

输出项：所有中转点或目的地的信息

模块设计

准确导航：为了方便用户选择，用五个 **Combobox** 控件来分别绑定五个

区域的点，不但可以减少每个列表的待选点数量，同时分区域选择更方便用户寻找点。用户选择后加入待选列表并显示出来，并可对其进行修改，点击确定按钮后，将待选列表通过页面传值带出去。

快速导航：为方便用户快速使用导航，精简不必要的选择时间，可以快速选择指定类型目的地，程序将自动选择离用户最近的该类型目的地进行导航。

语音操控：内置 **Cortana** 语音助手实现语音操控，用户说出关键词将自动进行该类型目的地的导航。

模块标志域命名约束

List<int> selected 已选择目的地点集合

模块流程

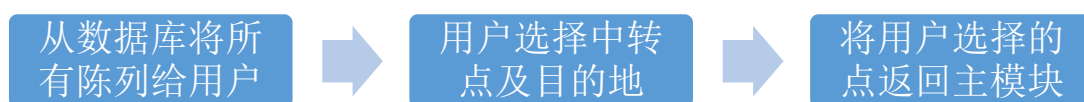


图 5 目的地选择模块流程图

五. 程序类设计

1.CookieHelper 类

该类是封装了本地缓存操作函数，含有 3 个方法。

CookieHelper
Write
Read
DeleteCookie

函数名：Write

函数描述：存入本地缓存

输入：键名、键值

输出：本地缓存存储该键值

函数名：Read

函数描述：读取本地存储

输入： 键名

输出： 键值

函数名：DeleteCookie

函数描述：删除本地缓存

输入： 键名

输出： 删除本地缓存该键值

2.Beeline 类

该类是存储三层拓扑化地图并进行最短路径规划的工具类。

Beeline
arcs D[]
Beeline SetSP Dijkstra OutPoint

字段名：arcs

描述:存储所有点的邻接矩阵

字段名：D[]

描述:保存起点到每一个点的最短距离

函数名：Beeline

函数描述:初始化邻接矩阵

输入： 每个相邻两点的距离

输出： arcs 邻接矩阵

函数名: SetSP

函数描述: 设置路径规划起点

输入: 点编号

输出: 设置类起点为指定点

函数名: Dijkstra

函数描述: 路径规划算法

输入: 无

输出: 算出起点到每一点最短路径途径的点

函数名: OutPoint

函数描述: 输出指定终点的最短路径

输入: 终点

输出: 起点到终点按照顺序的所有途径的点

3.Tools 类

该类包含了所有算法工具类，实现了大部分的底层算法功能封装。

Tools
Calculate
Distance
GetDivision
GetCurPos
IsOffset
GetLateKeyPoint
IsReachPoint

函数名: Calculate

函数描述: 进行整体路径规划

输入: 起点、中转点、目的地组成的集合

输出： 从起点到目的地的所有途径点集合

函数名： **Distance**

函数描述： 计算两点间最短距离

输入： 起点、终点组成的集合

输出： 两点间最短路径长度

函数名： **GetDivision**

函数描述： 根据 **xy** 坐标获取当前坐标所在区域

输入： 位置信息 (**x**, **y**, **f**)

输出： 区域代号

函数名： **GetCurPos**

函数描述： 通过服务器获得当前坐标信息

输入： 服务器地址

输出： 位置信息 (**x**, **y**, **f**)

函数名： **IsOffset**

函数描述： 判断当前位置是否偏离当前点段

输入： 当前点段起点、终点、当前位置 (**XYPos**)

输出： **bool** 值，是否偏离

函数名： **GetLateKeyPoint**

函数描述： 得到当前坐标最近的关键点

输入： 位置信息 (**x**, **y**, **f**)

输出： 最近点

函数名： **IsReachPoint**

函数描述： 判断是否到达指定点附近

输入： 当前位置、判断点 (**XYPos**)

输出： bool 值，两点是否距离相近

4.DataBase 类

该类为存储用户点坐标库的数据库。

DataBase
Point

字段名： Point

描述： 存储点坐标库的 List 数组。

5.DBItem、XYPos 类

这两个类为 Model 数据实体类，DBItem 为点坐标数据库实体类型，XYPos 为点位置信息数据实体类。

DBItem
Id
Name
X
Y
Region

XYPos
Floor
X
Y

六. UI 界面设计

1.界面设计



图 6 Logo



图 7 APP 启动页面

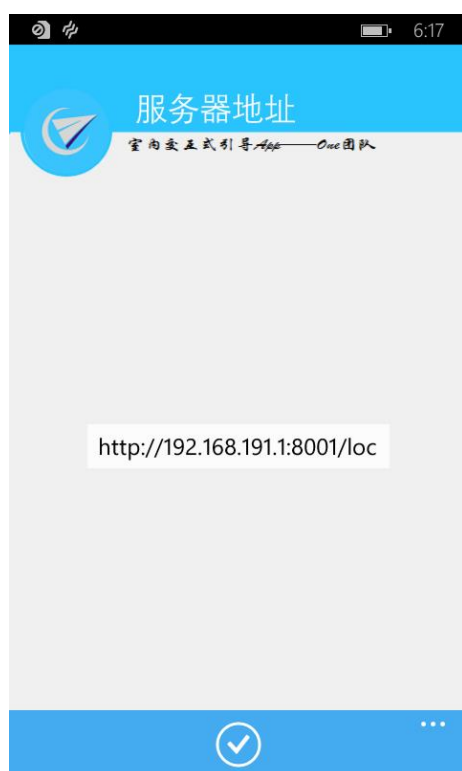


图 8 服务器地址选择页面

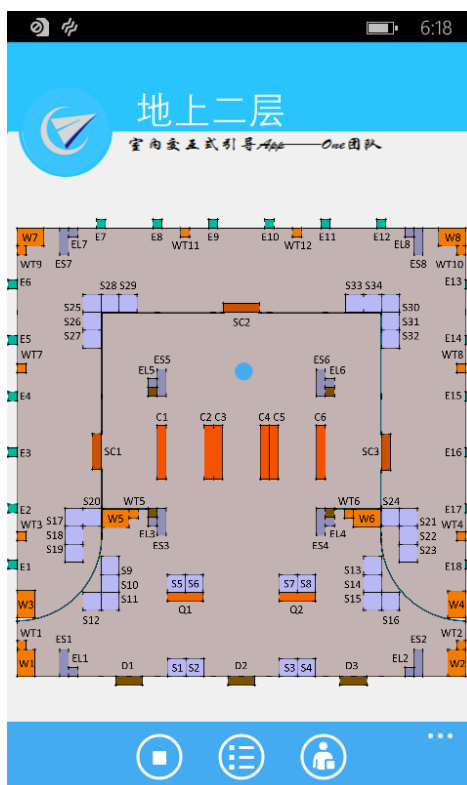


图 9 地图导航交互页面

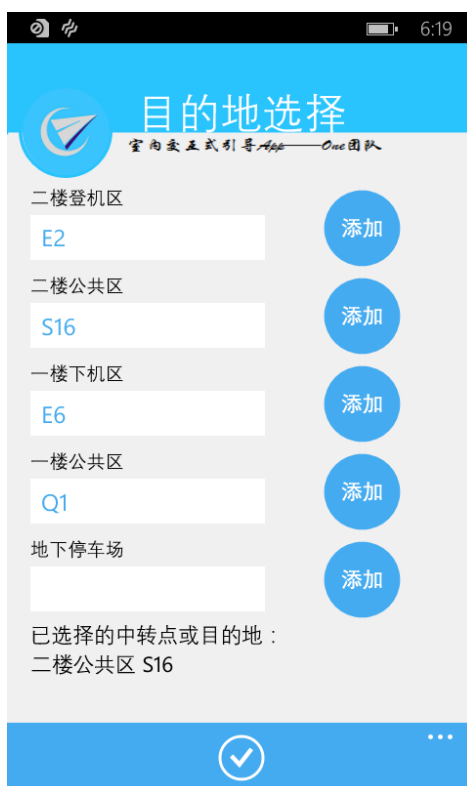


图 10 目的地选择-精确导航页面



图 11 目的地选择-快速导航页面

2.动画设计

用户当前位置点：采用呼吸效果来显示，渐变改变大小和透明度。

定位效果：较大醒目的红色圆圈渐变消失显示。

提醒通知：重写 **Toast** 通知，延迟五秒，屏幕上部逐渐消失显示，并伴有手机震动。

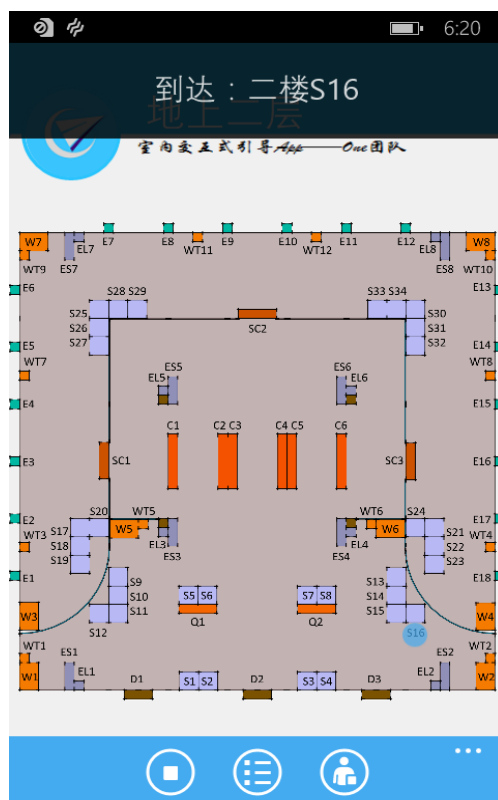


图 11 到达目的地或中转点提醒通知效果

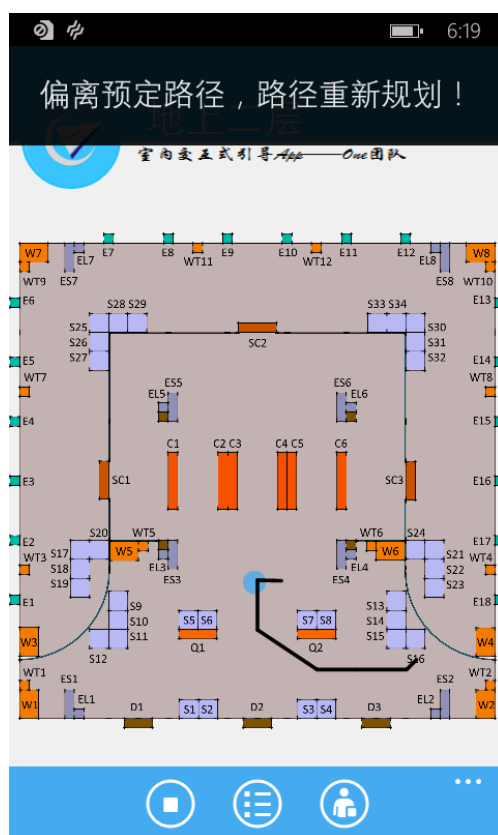


图 12 偏离预定轨迹提醒通知效果

3.UI 控件优化

双击退出键退出应用：为防止用户勿碰后退键退出应用，重写主界面后退键功能，双击才可退出。

重写后退键：为增加各页面联系，重新其余页面后退键，保证其他页面可以在 App 中正常导航，而不是直接退出应用。

重写通知界面：增加渐变效果通知界面，更适合本导航应用使用。

按钮 UI 重写：更改方形按钮为圆形无边缘深蓝色按钮，同时增加视觉状态，为其增加点击动画效果。

照相硬件按钮重写：更改其功能为调出语音识别功能，方便在用户不方便看屏幕的时候能快速调出语音操控功能开始导航。