

文件编号: [有一说一 确实]-SWC2020-[20200133]

受控状态: 受控 非受控

保密级别: 公司级 部门级 项目级 普通级

采纳标准: CMMI DEV V1.2



基于人流量监测的室内导航与监管平台

PublicOverwatch

项目开发文档

Version 10.0

2020.6.1

Written by 有一说一 确实



All Rights Reserved

目录

1	项目概述	1
1.1	项目背景	1
1.2	项目定位	1
1.2.1	应用场景	1
1.2.2	目标人群	1
1.3	项目方案	2
1.4	项目目标	2
1.5	项目价值	2
2	开发计划	3
2.1	最终呈现形式	3
2.2	主要功能描述	3
2.3	运行环境	3
2.4	验收标准	4
2.5	关键问题	6
2.6	进度安排	6
2.7	开发预算	7
3	可行性分析	7
3.1	技术可行性分析	7
3.1.1	功能简述	7
3.1.2	物联网端可行性分析	8
3.1.3	快应用客户端可行性分析	8
3.1.4	人工智能算法可行性分析	9
3.1.5	服务器端可行性分析	11
3.2	资源可行性分析	11
3.2.1	数据集可行性分析	11
3.2.2	模型训练可行性分析	13
3.2.3	服务器资源可行性分析	13
3.3	市场可行性分析	14
3.3.1	寻路	14
3.3.2	高德导航室内地图	15
4	需求分析	16
4.1	数据需求	16
4.1.1	静态数据	16
4.1.2	动态数据	16
4.1.3	数据词典	17
4.1.4	数据采集	18
4.2	功能需求	18
4.2.1	功能模块	18
4.3	性能需求	22
4.3.1	时间特性	22

4.3.2	适应性.....	23
4.4	界面需求.....	23
4.5	接口需求.....	30
4.5.1	硬件接口.....	30
4.5.2	软件接口.....	30
4.6	其他需求.....	34
5	概要设计.....	35
5.1	处理流程.....	35
5.2	总体结构设计.....	37
5.3	功能设计.....	37
5.4	用户界面设计.....	38
5.4.1	用户端.....	38
5.4.1	管理端.....	48
5.5	数据结构设计.....	49
5.6	接口设计.....	50
5.7	错误/异常处理设计.....	55
5.7.1	错误/异常输出信息.....	55
5.7.2	错误/异常处理对策.....	55
5.8	系统配置策略.....	55
5.9	系统部署方案.....	56
5.10	其他相关技术与方案.....	56
6	数据库设计.....	56
7	详细设计.....	58
7.1	室内定位功能模块.....	58
7.1.1	功能描述.....	58
7.1.2	性能描述.....	58
7.1.3	输入.....	58
7.1.4	输出.....	59
7.1.5	程序逻辑.....	59
7.1.6	限制条件.....	59
7.2	人数预测功能模块.....	60
7.2.1	功能描述.....	60
7.2.2	性能描述.....	60
7.2.3	输入.....	60
7.2.4	输出.....	60
7.2.5	程序逻辑.....	61
7.2.6	限制条件.....	61
7.3	最佳逃生路线规划功能模块.....	61
7.3.1	功能描述.....	61
7.3.2	性能描述.....	62
7.3.3	输入.....	62
7.3.4	输出.....	62
7.3.5	程序逻辑.....	62
7.3.6	限制条件.....	62

7.4	事故/意外情况上报功能模块.....	63
7.4.1	功能描述.....	63
7.4.2	性能描述.....	63
7.4.3	输入.....	63
7.4.4	输出.....	63
7.4.5	程序逻辑.....	63
7.4.6	限制条件.....	63
7.5	快应用—Web 通信解析功能模块.....	64
7.5.1	功能描述.....	64
7.5.2	性能描述.....	64
7.5.3	输入.....	64
7.5.4	输出.....	64
7.6	人群计数算法功能.....	64
7.6.1	功能描述.....	64
7.6.2	性能描述.....	64
7.6.3	输入.....	65
7.6.4	输出.....	65
7.7	快应用—修改节点状况功能模块.....	65
7.7.1	功能描述:	65
7.7.2	性能描述.....	65
7.7.3	输入.....	65
7.7.4	输出.....	65
7.8	快应用—节点警报的发送/取消.....	65
7.8.1	功能描述:	65
7.8.2	性能描述.....	65
7.8.3	输入.....	66
7.8.4	输出.....	66

记录更改历史

序号	更改原因	版本	作者	更改日期	备注
1	初始版本	1.0	赵彧涵	2019.11.30	
2	完成项目概述	1.1	赵彧涵	2019.12.1	
3	完成物联网部分可行性分析	1.2	曹志铭	2019.12.2	
4	完成计数算法部分可行性分析	1.3	赵彧涵	2019.12.2	
5	完成快应用管理端可行性分析及验收标准	1.4	宋晋瑜	2019.12.5	
6	基本完成需求分析	2.0	曾益	2019.12.7	
7	完成市场可行性分析	2.1	赵彧涵	2019.12.7	
8	完成快应用用户端文档	3.0	宋晋瑜	2019.12.7	
9	完成需求分析	4.0	曾益	2020.2.20	
10	完成概要设计	5.0	曾益	2020.2.25	
11	完成数据库设计与详细设计	6.0	曾益	2020.3.23	
12	完成界面设计	7.0	宋晋瑜	2020.3.29	
13	完成管理端详细设计	8.0	宋晋瑜	2020.4.1	
14	算法内容更新	9.0	曾益, 赵彧涵	2020.5.7	
15	完成最终修改	10.0	宋晋瑜	2020.6.1	

1 项目概述

1.1 项目背景

现代生活中，随着交通的发展，人们出行越来越方便，我们可以通过地图和导航的方式避开拥挤路段，到达我们想要到达的地点，然而在商场、学校等大型建筑中，我们很难做到高效的避开拥挤人流到达我们想要去的地点，通常这些拥挤的地点也伴随着如踩踏，摔倒的风险，商场和学校的管理人员也很难做到随时随地的监管每个拥挤地点的风险情况的发生，基于此我们提出了本项目。

随着人工智能技术的逐步发展与成熟，通过人工智能算法的方式快速且精确的识别图像中的人流量密度成为可能，我们结合人工智能技术和物联网技术，通过部署物联网边缘端和识别算法的方式，实现了对于建筑内各个地点的人流量自动监控，并且通过蓝牙的方式实现了建筑的用户室内定位，这样我们一方面可以给用户提供避免拥挤人群的室内导航功能，另一方面也可以给商场等大型建筑的管理人员提供全自动的人流密度监控、预测以及紧急状况的通知发布等功能，从而减轻商场等大型建筑区域的安防压力。

1.2 项目定位

1.2.1 应用场景

本项目主要应用于商场，学校等人流量较为密集的大型建筑，基于建筑内已有的监控摄像头网络，我们通过部署人工智能算法和物联网硬件的方式，构建起建筑内的人流量监控网络，从而为出行者提供避免拥挤的室内导航功能，为建筑管理人员提供提供全自动的人流密度监控、预测以及紧急状况的通知发布等功能。

出行者来到已经部署 PublicOverwatch 系统的建筑附近时，我们通过快应用的位置检测推送的方式，提示用户附近建筑的拥挤程度一览，用户可以打开快应用卡片查看建筑的拥挤程度，或是手动选择想去的地点进行室内导航。当发生紧急情况时，快应用会通过推送通知的方式及时提示用户当前建筑内发生了某种紧急情况，并给予用户的位置和出口的拥塞情况，自动规划疏散路线，除此之外，我们在建筑内部署的物联网边缘端的屏幕也会指引用户前往最合适最近的出口进行疏散。

商场等人流密集的大型建筑的管理者可以通过部署 PublicOverwatch 系统实现对于商场内人流量的检测，并且可以根据各个地点的历史分时人流数据预测不同地点的人流量压力，及时进行相关的安防调整；当出现人流量异常时，系统会自动提示管理人员人数异常，管理员可以根据实际的情况及时发布紧急情况的通知，该通知会通过应用推送和物联网边缘端屏幕的方式及时通知建筑内的人群紧急状况，并自动给出合理的疏散路线，实现全自动监控和疏散，从而减轻安防的压力。

1.2.2 目标人群

因为本项目的功能包括两个方面：一方面本项目的可以给用户提供避免拥挤人群的

室内导航功能,另一方面也可以给商场等大型建筑的管理人员提供全自动的人流密度监控、预测以及紧急状况的通知发布等功能,因此本项目的目标人群主要由两部分构成

- 建筑内想要避开人流密集区域到达指定位置的出行者
- 商场等人流密集的大型建筑的管理者

1.3 项目方案

本项结合人工智能技术和物联网技术,实现了对于建筑内各个地点的人流量自动监控以及用户室内导航,从而减轻商场等人流密集的大型建筑区域的安防压力,我们通过以下几个关键部分的方案来实现该项目的功能:

- 通过部署人工智能算法实现监控区域的人流量自动识别
- 基于建筑内已有的监控网络,部署蓝牙等物联网模块,实现用户的室内定位和电子屏幕引流功能
- 基于快应用技术,为用户提供上手即用的室内拥塞避免的室内导航服务
- 基于快应用等 web 技术,为管理人员提供建筑内的人流监测、预测以及紧急情况的及时发布功能
- 基于快应用推送和电子屏幕,为室内人员提供紧急情况自动引导疏散功能

1.4 项目目标

该项目的目标包括以下几个方面

- 依靠建筑内的监控网络,部署人工智能算法,实现对建筑内各个地点人流量的有效监测,并记录各地点的分时人流量数据
- 通过建筑内部署的物联网边缘端系统,实现室内定位和导航功能,并且结合部署的电子屏幕,实现紧急情况的人员引流和疏散功能
- 通过高效的路径规划算法,为用户提供快速规划避免拥挤人群的路径导航以及紧急情况的疏散路线规划的功能
- 通过快应用的方式,为用户提供免安装的客户端,并通过较为美观的界面为用户提供高效的室内导航,拥挤程度一览以及紧急情况通知及疏散功能
- 通过快应用平台,为建筑管理者提供包含人流监控,预测,监控视频观看和以及紧急情况发布等功能的客户端

1.5 项目价值

- 通过人工智能算法和物联网系统实现对于建筑内各个地点的人流量检测,为建筑管理者提供了高效的结合安防检测、应急处理已经人流疏散于一体的平台
- 通过历史人流数据为管理人员提供了安防工作开展的信息依据,让安防工作可以更加侧重人流可能较多区域,减轻了安防压力,同时也为商场等商业建筑提供用户行为分析的支持
- 通过方便快捷的快应用客户端,为用户提供上手即用的室内导航功能,让用户免

于路径规划和拥挤人群的烦恼

- 系统提供紧急情况的发布通知功能,通过快应用和物联网电子屏幕的方式覆盖建筑内所有人员,提供了高效的室内疏散引流的功能
- 人流量检测是基于图像的人工智能算法,无需部署额外的硬件既可实现人数识别,基于建筑内的监控平台既可完成快速部署
- 物联网平台采用了 arduino 单片机和相关的传感器,功耗低,价格低,方便部署和维护

2 开发计划

2.1 最终呈现形式

- 用户端的快应用平台
- 管理端的快应用平台
- 运行人流量计算算法的计算平台
- 运行在服务器上的服务端程序
- 接口的所有 Web Api 及其文档
- 边缘端以搭载 Android 系统的终端机为用户提供路线指引的方式呈现。摄像头搭载 hc05 蓝牙模块和 uno 开发板为用户端提供定位支持。

2.2 主要功能描述

项目主要实现了以下功能:

- 传统的监控安防系统只能通过视频监控,视频中出现的状况的分析依赖于监管人员的分析,因此存在有监控空隙,我们通过算法实现对于人流的监控、分析和预警,从而更好的帮助管理人员应对突发情况的发生
- 通过快应用免安装客户端,为用户提供了为用户提供上手即用的室内拥挤避免的室内导航服务
- 我们通过快应用推送以及物联网的电子屏幕,实现了紧急情况的通知对于建筑内人员的有效覆盖,并且全自动的路径规划和引导有效避免了疏散时发生拥堵现象的可能
- 通过历史人流数据为管理人员提供了安防工作开展的信息依据,让安防工作可以更加侧重人流可能较多区域,减轻了安防压力,同时对于商场等建筑,也提供了根据人流数据布置商业区域的参考,从而产生更多的商业价值
- 人流量检测是基于图像的人工智能算法,无需部署额外的硬件既可实现人数识别,基于建筑内的监控平台既可完成快速部署,相较于当下的一些安防系统成本显著降低
- 物联网平台采用了 arduino 单片机和相关的传感器,功耗低,价格低,方便部署和维护

2.3 运行环境

- 人流量计算平台环境

python 2.7 torch 1.0.1 tensorboardX tensorboard
TensorFlow easydict pandas numpy numba SciPy
OpenCV x264lib matplotlib requests PIL ffmpeg

- 物联网边缘端环境：
Android 6.0+
Arduino General Uno
- 快应用端环境
Window 快应用开发工具
Window Chrome
Android 快应用预览版
Android 快应用调试器
- 后端平台：
Java 8u231
Springboot v2.2.0
Nginx 1.16.1

2.4 验收标准

- 边缘端：
位于边缘端的方向指引功能可以正常使用，在 3 秒内完成收到后端返回的方向数据。
位于边缘端的蓝牙辅助定位功能可以正常使用，在保证正常供电时提供有效的蓝牙信号。
- 前端-管理端：
位于管理端的登录功能可以正常使用，在 1s 内保证登录结果。
位于管理端的点击 tab-bar 显示不同的 tab-content 功能可以正常使用，在 0.5s 内显示功能。
位于管理端的根据当前楼层显示实时人数、人数阈值和摄像头位置功能可以正常使用，在 3s 内显示楼层、摄像头位置和实时人数。
位于管理端的点击地图按钮实现当前楼层地图功能可以正常使用，在 0.5s 内显示当前楼层地图。
位于管理端的点击摄像头显示实时监控视频、实时人数、修改阈值和发送警报功能可以正常使用，在 1s 内显示功能界面
位于管理端的点击查看实时监控视频功能可以正常使用，在 3s 内显示实时监控视频
位于管理端的修改当前摄像头阈值功能可以正常使用，在 1s 内完成修改阈值功能
位于管理端的发送警报功能可以正常使用，在 1s 内显示发送提示和在 1s 内完成发送警报功能
位于管理端的查看超阈值次数统计功能可以正常使用，在数据统计模块可以直接显示。
位于管理端的查看各地点历史人数统计的功能可以正常使用，在数据统计模块可以选择摄像头和日期进行查看，在 2s 内显示历史人数统计。
- 前端-客户端

位于客户端的快应用主应用部分可以正常启动，全过程在 0.5s 内完成。

位于客户端的人流量监控服务可以正常启用，并在 0.1s 内完成服务的启用（不包括刷新）。

位于客户端的 tab-bar 可以通过点击和滑动显示不同的 tab-content，并在 0.1s 内完成显示。

位于客户端的刷新功能可以正常使用，当用户不在 PublicOverwatch 建筑内时定位功能正常时在 0.5s 内完成数据刷新；当用户在 PublicOverwatch 建筑内且定位功能正常时在 5s 内完成数据刷新。

位于客户端的获取紧急事件功能可以正常使用，并在 0.5s 内完成数据的拿取。

位于客户端的获取当前位置功能可以正常使用，并在 0.5s 内完成数据的拿取。

位于客户端的获取当前位置人流量级别功能可以正常使用，并在 0.5s 内完成数据的拿取。

位于客户端的界面路由功能可以正常使用，并在 0.5s 内完成页面的切换。

位于客户端的启动轮询功能可以正常使用，并在 0.5s 内完成数据轮询功能的启动。

位于客户端的紧急风格界面功能可以正常使用，并在获取到紧急事件的 0.5s 内完成页面风格的切换。

位于客户端的紧急情况推送功能可以正常使用，并在获取到紧急事件的 0.5s 内完成消息推送。

位于客户端的当前紧急事件清单功能可以正常使用，并在 0.5s 内获取到清单的详细信息。位于客户端的人流量图功能可以正常使用，并在 0.5s 内完成人流量图全部信息的显示。

位于客户端的人流量图楼层切换功能可以正常使用，并在 0.5s 内完成楼层的切换和重绘。

位于客户端的室内导航功能可以正常使用，并在 0.5s 内给出导航的路径或者无法导航的说明。

位于客户端的室内导航路径展示功能可以正常使用，并当有路径时在 0.5s 内给出导航的多层路径。

位于客户端的应急路线展示功能可以正常使用，并当有路径时在以 0.5s 内给出导航的多层路径。

位于客户端中快应用卡片的数据刷新功能可以正常使用，并在 5s 内完成数据的更新。

位于客户端中快应用卡片的查看紧急事件清单功能可以正常使用，并在 0.5s 内完成紧急事件清单的展示。

位于客户端中快应用卡片的查看详细功能可以正常使用，并在 0.5s 内完成。

位于客户端中快应用卡片的应急路线功能可以正常使用，并在 0.5s 内给出基于当前位置的应急路线。

- 人流量计算平台

人流量识别的帧数满足视频传输的要求

在无外界干扰画面的情况下，精度误差控制在 8 以内

在保证视频清晰程度的情况下，将视频传输码率控制在 1000kbps 内

- 后端

后端所有的接口响应时间<50ms

后端所有接口的报错类型都已经在文档中进行标注

后端所有的值实时更新，数据保持一致性和实时性

后端 api 接口文档能够实时更新并提供线上测试功能

2.5 关键问题

- 边缘端的关键问题：

用户端的快应用可以接收到多个摄像头的蓝牙信号，区分哪一个是距离用户最近的较为困难。

 - 对策：使用官方接口提供的 RSSI 字段描述的信号强度进行区分，信号最强的摄像头近似用户的位置。
- 人流量计算端的关键问题：需要权衡算法的性能开销和精度之间的关系
 - 对策：我们训练并测试使用了多个模型如 CSRNet, MCNN, CNTL 等网络，对比精度与性能开销，最终选择了基于 ResNet50 自行优化相关网络
- 前端-管理端关键问题：

管理端如何提示非当前楼层的摄像头中出现了人数超出阈值两倍的情况。

 - 对策：使用一个变量在每次轮询的时候自增，count 到一定值时，用数组来存放人数超出阈值两倍的摄像头信息，然后全部抛出发送警告。自增变量回到一个初始值接着循环，来实现定时检测人数超出阈值两倍的较为突发的情况。
- 前端-客户端关键问题：

客户端的数据获取中基于蓝牙的设备通信耗时过长，且时长频繁波动不确定

 - 当前使用的蓝牙硬件设备为 hc05，其信号比较弱，导致每次蓝牙需要搜索很久才能搜到设备。之后打算使用蓝牙 beacon 作为蓝牙设备，beacon 可以轻易的放置并且能够将连续的信号发送到传输范围内的任何移动设备上，借助 beacon 可以更好的实现楼宇建筑内的位置服务。

因为服务卡片需要跟官方协作，当用户进入 PublicOverwatch 建筑时，暂时只能使用后台轮询实现服务的启动并进行消息推送实现告知。

 - 对策：与快应用厂商协作，使用服务卡片来实现位置驱动的服务卡片，当用户进入 PublicOverwatch 建筑时，服务卡片的推送会提醒用户已经进入 PublicOverwatch 建筑，并启动快应用实现更自然的服务启动。
- 后端关键问题：

在原型设计中，在规划路径这种需要访问大量数据的过程中，持续的依赖数据库连接进行数据获取将会极大的增加接口响应时间，达到 1500ms 以上。

 - 对策：在后续优化中，把需要连接数据库获取的数据全部移到服务器来进行处理，降低在路径计算中所需的数据库连接次数。经过数据处理源的优化，路径计算接口的响应时间达到 40ms 以下。

2.6 进度安排

- 2019.10.20
 - 决定项目和具体细节，决定基本的清单，项目开始，文档编写第一版
- 2019.10.20~2018.10.27
 - 查阅资料，选择可行性较好的实现方案，设计数据字典，找寻合适的数据集，

前端开始学习快应用文档

- 2019.10.28~2018.11.5
 - 前后端初步搭建,完成数据库部分的初步设计,深度学习网络部分搭建基本完成,开始开发快应用,并进行物联网边缘端开发
- 2019.11.6~2019.11.19
 - 完善前后端的搭建,编写所有的 Api 接口,编写路径算法和相关接口,完成深度学习模型的训练,完成平台主要功能的开发。完成快应用功能和物联网边缘端开发
- 2019.11.20~2018.11.26
 - 项目测试和除错阶段,编写和完善文档
- 2019.11.26~2018.11.29
 - 完成剩余的文档,拍摄项目展示视频

2.7 开发预算

- 阿里云服务器的费用
- 边缘段硬件费用
- 深度学习训练费用

3 可行性分析

3.1 技术可行性分析

3.1.1 功能简述

本项结合人工智能技术和物联网技术,实现了对于建筑内各个地点的人流量自动监控以及用户室内导航,从而减轻商场等人流密集的大型建筑区域的安防压力,项目主要实现了以下功能:

- 传统的监控安防系统只能通过视频监控,视频中出现的状况的分析依赖于监管人员的分析,因此存在有监控空隙,我们通过算法实现对于人流的监控、分析和预警,从而更好的帮助管理人员应对突发情况的发生
- 通过快应用免安装客户端,为用户提供了为用户提供上手即用的室内拥挤避免的室内导航服务
- 我们通过快应用推送以及物联网的电子屏幕,实现了紧急情况的通知对于建筑内人员的有效覆盖,并且全自动的路径规划和引导有效避免了疏散时发生拥堵现象的可能
- 通过历史人流数据为管理人员提供了安防工作开展的信息依据,让安防工作可以更加侧重人流可能较多区域,减轻了安防压力,同时对于商场等建筑,也提供了根据人流数据布置商业区域的参考,从而产生更多的商业价值
- 人流量检测是基于图像的人工智能算法,无需部署额外的硬件既可实现人数识别,基于建筑内的监控平台既可完成快速部署,相较于当下的一些安防系统成本显著降低
- 物联网平台采用了 arduino 单片机和相关的传感器,功耗低,价格低,方便部署和维护

3.1.2 物联网端可行性分析

3.1.2.1 开源硬件支持

本项目使用 Arduino 作为硬件支持。Arduino 是一款便捷灵活、方便上手的开源硬件产品，具有丰富的接口，有数字 I/O 口，模拟 I/O 口，同时支持 SPI,IIC,UART 串口通信。能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他装置来反馈、影响环境。它没有复杂的单片机底层代码，没有难懂的汇编，只是简单而实用的函数。而且具有简便的编程环境 IDE，极大的自由度，可拓展性能非常高。标准化的接口模式为项目硬件端的实现提供了便利条件。

3.1.2.2 快应用蓝牙接口支持

在蓝牙辅助定位模块中，我们使用快应用提供的蓝牙接口与硬件部分的蓝牙模块进行通信。快应用对于蓝牙的支持十分完整，从建立蓝牙发现，建立 BLE 连接到关闭蓝牙发现，各个接口都十分稳定。为手机端配合硬件联动提供了基础保障。

3.1.3 快应用客户端可行性分析

3.1.3.1 快应用管理端可行性分析

- 快应用程序本身是比赛的基本技术要求。快应用的入口场景丰富，使用方便且无需下载安装。快应用技术规范层面做了统一，能在 10 大主流手机厂商的手机设备上运行，满足大多数安保人员可以使用的前提。
- 快应用拥有丰富且稳定的接口以及组件，完全满足开发者的需求。
- 快应用提供的 video 组件支持 http 协议且支持主流的音视频编码格式，可以实现在手机上实时查看摄像头的监控视频的功能，解决了大型建筑内安保人员随时查看监控视频的需求。
- 安保人员根据情况动态地调整摄像头阈值或者是针对该摄像头区域发出警报。
- 快应用可以通过 web 组件引入外部 web 实现对数据进行表格统计可视化，弥补了快应用暂时没有表格组件的缺陷。
- 管理端可以提供动态的人数预测、人数超阈值统计以及历史监控数据分析等给安保人员做参考，大大减轻了安保人员的安防压力。

3.1.3.2 快应用用户端可行性分析

- 快应用作为一种新的应用形态，能够让用户无需下载安装即可流畅的体验应用内容，对于用户而言，即点即用无疑是一种更方便的使用方式。
- 快应用提供了丰富的组件和前端化的应用开发支持，足以满足开发需求。
- 快应用的蓝牙接口提供了与物联网边缘端设备交互的能力
- 快应用的 fetch 接口提供了前后端数据交互能力，满足数据交互的需求。
- 快应用的支持厂商几乎覆盖了国内所有的 Android 厂商，国内用户基数非常大，适

合开发基于安卓的、本地化的应用。

- Public Overwatch 人流量监控服务作为一种空档期长的公告场所服务并不需要、也不应该、更不会被长期的安装在用户手机上,而以快应用这种即点即用的应用形态,加上与应用厂商的定制化 Android 系统深度整合带来的负一屏应用卡片、大量入口场景等特性,再加上其位置触发、事件驱动等服务卡片形式,可以给 Android 用户带来全新的更沉浸的应用体验。使用快应用开发 Public Overwatch, 可以使它不再仅仅是一个 APP, 而更像是一种原生自带服务而存在。

3.1.4 人工智能算法可行性分析

随着人工智能算法的快速发展,人群计数(Crowd Counting)作为计算机视觉的一个重要方向的技术也在不断更新,为了尽可能的提高人数预测的精确度,我们浏览了业界主流的神经网络与相关的数据集,我们从中筛选出了一些精度较高,性能较好的网络作为我们的候选网络,包括: AlexNet, MCNN, CSNet, SANet, ResNet series, CMTL, VGG 等,然后我们选择了分辨率较高 shanghaiTech_part_B(简称 SHTB)人群数据集作为我们的测试数据集。



图 SHTB 数据集图片与标注

随后我们对这些网络进行训练并测试实际的检测结果,根据训练得出的实际结果作为我们测试的根据,这里列举部分网络的训练截图:



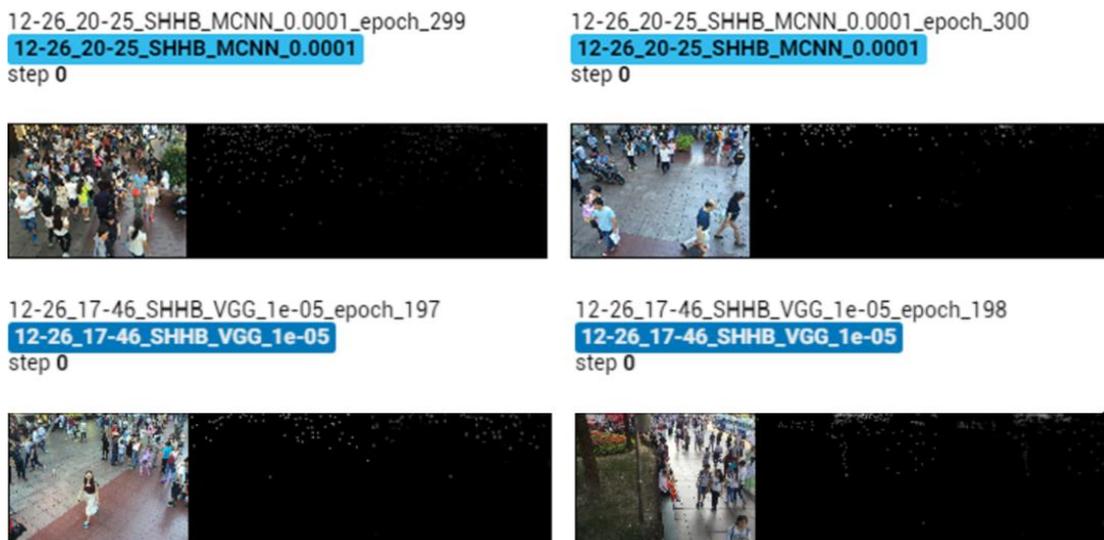


图2 从上到下分别是 CSRNet, Res50, MCNN, VGG 的 TensorBoard 训练 Img 截图 我们使用 SHTB 的测试集进行测试, 得到各个模型的 MSE 与 MAE 信息如下:

方法	MAE	MSE
ResNet-50	7.4	12.0
MCNN	26.4	41.3
CMTL	16.0	31.1
CSRNet	10.6	16.0
SANet	8.4	13.6
AlexNet	21.5	38.1
VGG-16	10.3	16.5

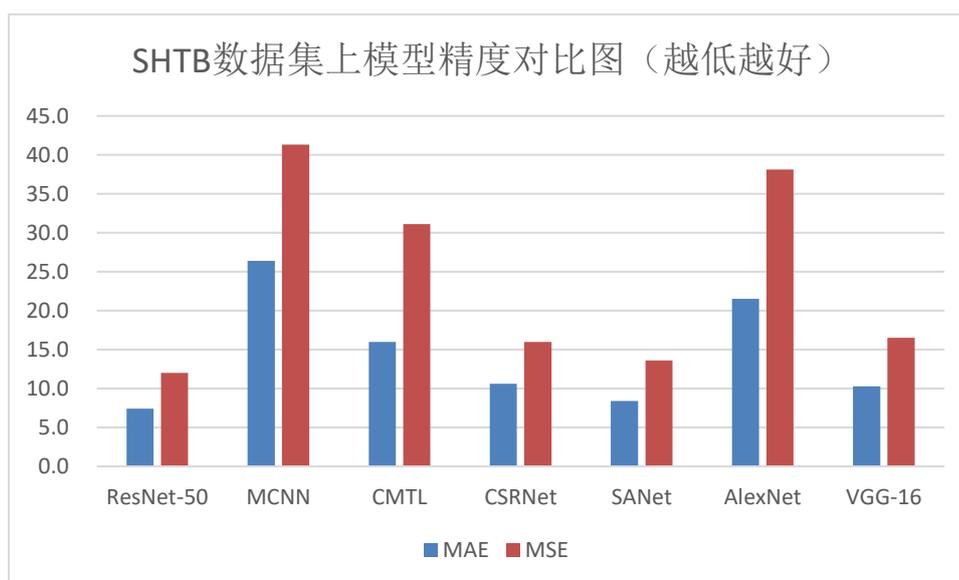


图 模型精度对比图

从测试结果分析, ResNet-50 凭借其强大的学习能力取得了非常好的精度, 是我们测试网络中最好的, 因此我们决定尝试使用更深层数的 ResNet101, 测试是否能够获得更好的精度, 以下分别为训练时的 Tensorboard 的 MAE, MSE, loss 的截图:

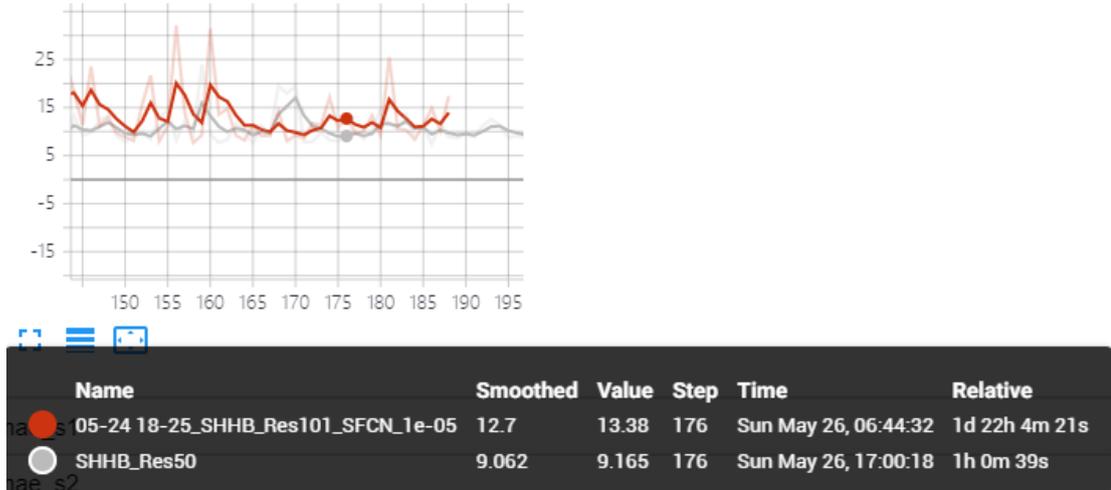


图 Tensorboard 的 MAE, MSE, loss

从测试结果分析, ResNet101 的确获得了相对 ResNet50 略微更好一些的精度 (MAE 7.2 MSE 11.6), 然而 ResNet101 运行与训练时的 GPU 负载相对 ResNet50 有非常多的提高, 因此综合考虑精度和性能, 我们在众多的神经网络中选择了 ResNet50。

3.1.5 服务器端可行性分析

本平台后端主要用于提供路径计算平台和数据中间枢纽。在业界领域有生态丰富的 Java 可以协助完成这个任务。前端开发者使用 http 请求访问 rest api 来提交和获得 json 序列化后的信息。同时使用 MySQL 数据库可以存储在程序运行期间收到和产生的数据, 做到数据与程序分离, 保证数据的安全性。在作为系统的中间件的运行模式中, 选择使用 SSM (Spring-SpringMVC-Mybatis) 来构建整个系统。这样的开发模式在 Java 生态中十分成熟, 同时也是中小服务程序的首选方案, 所以在技术上后端的技术路线具有极高的可行性。

在传输视频方面, 我们采用了更方便易用的 nginx 的视频流插件来实现。Nginx-http-flv-module 是一款更新的 nginx 视频流直播插件, 使用这一模块可以便捷的开启一个视频中间服务。

1. 使用 Springboot 搭建主体程序
2. 使用 MySQL 作为数据存储
3. 使用 Swagger 生成 api 文档
4. 使用 NGINX 和 nginx-http-flv-module 来传输视频流

3.2 资源可行性分析

3.2.1 数据集可行性分析

在我们尝试使用 SHTB 数据集上训练得到的 ResNet50 模型时, 遇到了复杂场景下无法很好的识别图像的问题, 也就是说模型针对我们的泛化能力并不强, 因此我们首先需要提高模型的泛化能力。

在提高模型的泛化能力上, 我们使用了最新的 CVPR2019 中一篇顶会论文: Learning from Synthetic Data for Crowd Counting in the Wild 中提出的方法, 使用论文中方法生成的多场景, 高分辨率的超大人群众数据集 GCC 数据集对 ResNet50 进行预训练, 因为 GCC 中包括了夜晚,

街道等众多的场景数据，因此我们使用 GCC 数据集对我们的模型进行预训练，从而极大的提高我们模型的泛化能力。

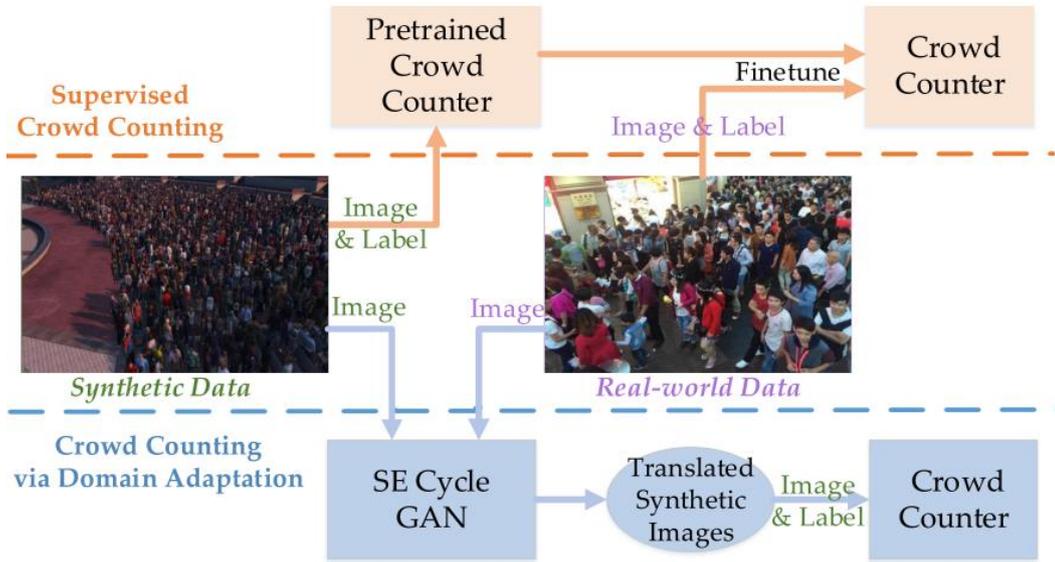


图 论文中提出的训练方法

Dataset	Number of Images	Average Resolution	Count Statistics			
			Total	Min	Ave	Max
UCSD [8]	2,000	158 × 238	49,885	11	25	46
Mall [9]	2,000	480 × 640	62,325	13	31	53
UCF_CC_50 [14]	50	2101 × 2888	63,974	94	1,279	4,543
WorldExpo'10[41]	3,980	576 × 720	199,923	1	50	253
SHT A [43]	482	589 × 868	241,677	33	501	3,139
SHT B [43]	716	768 × 1024	88,488	9	123	578
UCF-QNRF [15]	1,525	2013 × 2902	1,251,642	49	815	12,865
GCC	15,212	1080 × 1920	7,625,843	0	501	3,995

图 6 论文中提出的 GCC 数据集与其他人群数据集的对比



图 GCC 数据集中的众多场景

为了适配我们的监控建筑内公共地点的使用场景，我们采用了 World Expo 10 数据集对 GCC 预训练的 ResNet50 模型进行迁移学习，从而使得模型更好的适配我们的场景，此外为了适应高分辨率的输入图像，对于 ResNet 为了保证密度图的大小不至于过小（不小于原图尺寸的 1/8），我们修改了 res.layer3 中第一层 stride 的大小（将原本的 2 改为 1），以此当做 encoder。以下为训练部分的部分截图：

05-27_16-50_WE_Res50_1e-05_epoch_200

05-27_16-50_WE_Res50_1e-05_latest

step 0 Tue May 28 2019 18:37:54 GMT+0800 (中国标准时间)

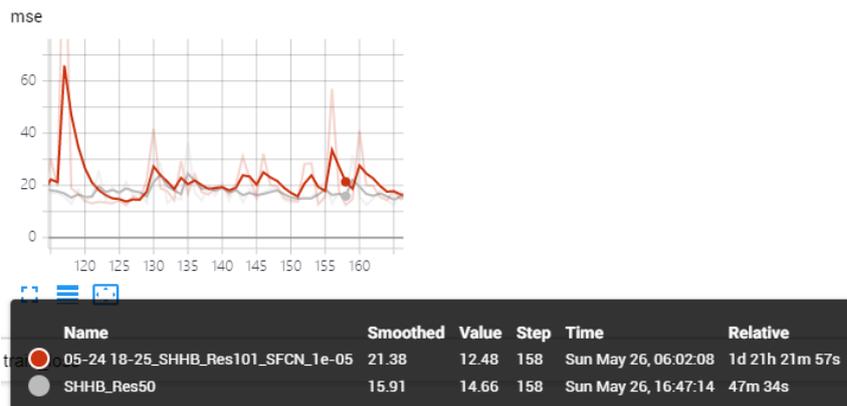


图 训练截图

最终通过合理的数据集选择，我们可以实现较好的模型精度。

3.2.2 模型训练可行性分析

在模型选择阶段，我们对各个模型在分辨率较高 shanghaiTech_part_B（简称 SHTB）人群数据集上的性能进行了对比，在训练阶段，在我们的单卡 1080ti 平台，即使是训练计算量最大的 ResNet101，训练到 MAE 显著收敛也只需要大约两天，如图



因此模型训练的是具有较好的可行性的。

3.2.3 服务器资源可行性分析

现在的云计算市场可以提供的通用计算服务器，可以提供不同配置、灵活升降的配置方案。普通的单核、2G 内存和 40G 云存储空间的配置服务器可满足服务端程序的所有运算和存储要求。

3.3 市场可行性分析

我们通过分析对比市面上几款拥有与本项目类似功能的竞争品，进而分析本项目的市场可行性

3.3.1 寻路



寻路 软件界面

介绍：寻路（Path Guide），提供低成本、即插即用的室内导航服务。

特点：完全不需要室内地图信息，也不需要室内预装任何硬件设备，在智能手机上下载应用后即可立即使用，完全实现了低成本、即插即用和易扩展的室内导航推广需求。

原理：通过收集智能手机在运动过程中的磁场、加速度传感器、陀螺仪和气压计等数据的变化来记录用户的相对运动轨迹（包括步数信息、转弯和上下楼等），进而生成一个参考路线供用户分享。后者只需要沿着“领路人”的路线指引即可找到对应的位置。在路线创建的过程中，“寻路”也支持添加个性化文字、语音及照片等功能，既为分享路线增添了乐趣也能帮助后续用户更容易地找到终点。



寻路 软件界面

对比分析：寻路不需要室内预装任何东西，另外它丰富使用了智能手机上的各种传感器来记录用户的相对运动轨迹是一个很新颖的方式。但是，它需要一个“领路人”作为一切导航的前提。所以当不存在“领路人”的情况下，用户会出现比较尴尬的情况。若是真的对一个大型建筑内进行导航，则是需要“领路人”对于大型建筑内的每个地点都留下一条路线才行。本项目对于寻路的优势在于提供了管理端方便安保人员以及提供了对于突发紧急情况的应对措施。

3.3.2 高德导航室内地图

介绍：高德导航室内地图。

特点：渲染了室内地图，可通过选择楼层来显示不同的地图、提供室内 POI 搜索，以及室内的路线规划。

对比分析：高德室内地图，涵盖全国 21 个省市，超 1000 家大型建筑，包括交通枢纽，机场，火车站，大型体育馆，大型购物中心等，满足用户的室内地图需求。与本项目对比，高德导航的室内地图明显更偏向于商业化一些，它提供了室内的 POI 搜索以及同一楼层的商品浏览，对于用户来说可以很好地满足自身的需求。但是对于大型建筑内的安保人员来说，使用可能并不是那么友好，安保人员还是需要传统的视频监控方式进行安保活动。本项目则既可以在兼顾到用户室内地点导航的情况下，提供给了建筑内安保人员需要的安保帮助，同时对于紧急情况，本项目也能及时的提醒用户并给到用户最佳的撤离方案。



高德室内导航 地图界面

4 需求分析

4.1 数据需求

4.1.1 静态数据

服务器的 ip 地址
 数据库服务器的配置信息
 建筑建模的基础点坐标信息

4.1.2 动态数据

账号信息和保安人员的联系信息
 所有监控节点的状态信息
 所欲监控节点发生的事件信息

4.1.3 数据词典

表名	列名	类型	说明
accident	id	int(10)	事件 id
	description	varchar(255)	事件说明
camera	id	int(10)	相机 id
	position	varchar(255)	相机的位置描述
	threshold	int(5)	路径节点的负载
	longitude	decimal(10,6)	纬度
	latitude	decimal(10,6)	经度
	times	int(11)	过载次数
	param	int(11)	参数
	lastshoturl	varchar(255)	上次记录人数的快照
	nums	int(10)	现在人数, 最新人数纪录
	isexit	tinyint(1)	是否是个出口
	floor	int(10)	层数
isfree	tinyint(1)	是否空闲	
camera_accident	cameravid	int(10)	相机 id
	accidentid	int(10)	事件 id
camera_history	id	int(11)	记录 id
	camid	int(10)	相机 id
	nums	int(11)	记录对应的人数
	time	timestamp(6)	记录产生时时间
	shoturl	varchar(255)	快照链接
guard	id	int(20)	保卫人员 id
	name	varchar(10)	保安人员名字
	phone	varchar(15)	手机号
	longitude	decimal(10,6)	纬度
	latitude	decimal(10,6)	经度
neighbors	sourceid	int(10)	相邻关系的一方 id
	neighborid	int(10)	相邻关系的一方 id
user	account	varchar(20)	用户账号
	password	varchar(20)	用户密码
	phone	varchar(15)	用户手机号
video	id	int(11)	视频素材的 id
	name	varchar(70)	视频素材的名字
	time	timestamp(6)	视频素材的记录时间
	length	int(11)	视频素材的长度
	position	varchar(40)	视频素材的位置
	file_position	varchar(100)	视频素材的文件路径
	camera_id	int(11)	对应的相机 id

4.1.4 数据采集

保卫人员、可达关系和相机信息：管理员的 web 端使用接口进行导入、修改和删除，其中 id 等信息时系统自动生成的。

图像信息：通过建筑内的监控摄像头网络获取

人数记录信息：摄像头捕获的数据经过计算平台处理后自动上传人数记录

文件记录相关信息：在文件上传时，系统自动生成事件和 id 等信息。

4.2 功能需求

4.2.1 功能模块

表 核心功能模块描述

功能模块	功能	功能描述	优先级
路径规划模块	最佳逃生路径	客户端调用接口时输入目前所在地的位置标识，可以得出最短的逃生路径	高
	室内导航最佳路径	客户端调用接口时输入当前所在地的位置标识，同时还有目的地的位置标识，就可以得到在室内的最短路径	高
	路口指引	当前节点的最佳逃生路径上的下一个节点	中
管理人员模块	修改节点状态	管理人员可以在某个出口或者通道发生事故时，在系统中记录事故，同时这一操作会影响路径规划	高
	查看人流情况	管理人员可以查看当前建筑内，所有摄像机的画面中的人流情况，为应急疏散做准备	高
	解除紧急事件	当一个通道节点的紧急事件结束之后，管理员删除事件	低
	配置监控节点	对所有的节都进行配置，修改负载限制和位置描述以及其他信息	高
路径可达关系模块	可达关系控制	当增加了一个节点时，需要添加可达信息或者减少一个可达关系	高
	查询相邻节点	可以在接口中查询本地点的相邻节点	高
人数	更新人数	边缘端提供实时的人数解析值更新到系统当中	高

记录接口	查询人数	客户端查询对象节点的现在人数记录	高
	历史记录	在每次更新人数时,节点都会将对应的记录记入到历史记录中	高

添加紧急事件用例规约

用例名称	添加一种紧急事件
功能简述	在系统中添加一种新的紧急事件类型
用例编号	001
执行者	系统管理员
前置条件	系统正常运行
后置条件	系统中增加一种新的紧急事件类型
涉众利益	系统管理员
基本路径	a.系统管理员登录管理端界面 b.系统管理员进入事件管理界面 c.系统管理员选择添加一种新的事件 d.系统管理员输入事件的描述 e.完成添加
扩展路径	1.管理员取消添加
字段列表	事件
设计规则	
未解决的问题	
备注	

为节点添加紧急事件用例规约

用例名称	为节点添加一个紧急事件
功能简述	在系统中为节点添加一种新的紧急事件类型
用例编号	002
执行者	系统管理员
前置条件	系统正常运行 节点处发生事件被管理员发现
后置条件	节点处增加一种新的紧急事件类型 所有的路径规划都会规避该节点
涉众利益	系统管理员 系统使用者
基本路径	a.系统管理员登录管理端界面

	b.系统管理员进入事件管理界面 c.系统管理员选择添加一种新的事件到指定节点 d.系统管理员输入相机的标识符，同时指定一种已存在的事件类型 e.完成添加
扩展路径	1.管理员取消添加 2.管理员取消一个事件
字段列表	事件 相机-事件的相机 id 和事件 id
设计规则	
未解决的问题	
备注	

获取最短逃生路径用例规约

用例名称	最佳逃生路径
功能简述	在当前节点位置获得最佳的逃生路径
用例编号	003
执行者	客户端用户
前置条件	系统正常运行 用户所在的节点确实存在一条可以安全离开的路径
后置条件	用户获得一条逃生的最短路径
涉众利益	客户端使用者
基本路径	1.用户进入客户端主界面 2.系统检测到当前的地址完成定位 3.客户端更新界面最佳逃生路径
扩展路径	1.用户手动切换路线
字段列表	相机表人数字段和可达关系表
设计规则	
未解决的问题	
备注	

室内导航用例规约

用例名称	室内导航
功能简述	在系统中为节点添加一种新的紧急事件类型
用例编号	004
执行者	客户端用户
前置条件	系统正常运行

	用户输入的目的地存在 确实存在一条可行路径
后置条件	用户获得一条通往目的地的最短路径
涉众利益	客户端用户 系统使用者
基本路径	1.用户进入客户端主界面 2.系统检测到当前的地址完成定位 3.客户输入希望到达的目的地 4.客户端联系服务器获取最短路径 5.在地图上绘制最短路径
扩展路径	1.用户更换目的地 2.用户输入错误的目的地 3.系统提示用户目的地不在系统中，进行错误引导
字段列表	事件 相机-事件的相机 id 和事件 id
设计规则	
未解决的问题	
备注	

更新人数用例规约

用例名称	用户查看当前发生中的紧急事件
功能简述	用户在客户端可以查看当前发生中的紧急事件
用例编号	005
执行者	客户端用户
前置条件	系统正常运行 系统中存在事故事件记录
后置条件	用户获得现在系统中的紧急事件列表
涉众利益	客户端用户 系统使用者
基本路径	1.用户进入客户端主界面 2.系统检测到当前的地址完成定位 3.检测到紧急事件的存在，警告用户 4.用户选择查看紧急事件列表 5.给用户展示紧急事件列表
扩展路径	
字段列表	事件 相机-事件的相机 id 和事件描述

设计规则	
未解决的问题	
备注	

查看节点人数用例规约

用例名称	用户查看当前所在位置节点的人数
功能简述	用户在客户端可以查看节点周围的人数
用例编号	006
执行者	客户端用户
前置条件	系统正常运行 系统能够收到边缘端传来的实时人数
后置条件	用户获得现在节点的实时人数
涉众利益	客户端用户 系统使用者
基本路径	1.用户进入客户端主界面 2.系统检测到当前的地址完成定位 3.在主界面展示当前节点的人流量级别
扩展路径	
字段列表	相机 相机记录表的当前实时人数字段
设计规则	
未解决的问题	
备注	

4.3 性能需求

4.3.1 时间特性

- 响应时间：
 - 后端系统的正常访问响应时间不应超过 0.5 秒
- 更新处理时间：
 - 所有数据保证实时更新，边缘端的上传数据和客户端获取到的数据保持一致
- 数据转换与传输时间
 - 视频流的传输延迟不超过 20s
 - 图片快照文件的上传时间不超过 1s

4.3.2 适应性

人流量计数算法可以在支持 Pytorch-GPU 的主流计算平台上,均以较快的速度完成精确的人流量计数

后端平台运行于任何可以运行 Java runtime environment 8u231 以上版本的平台
快应用客户端可以流畅运行在支持快应用的主流国产手机上

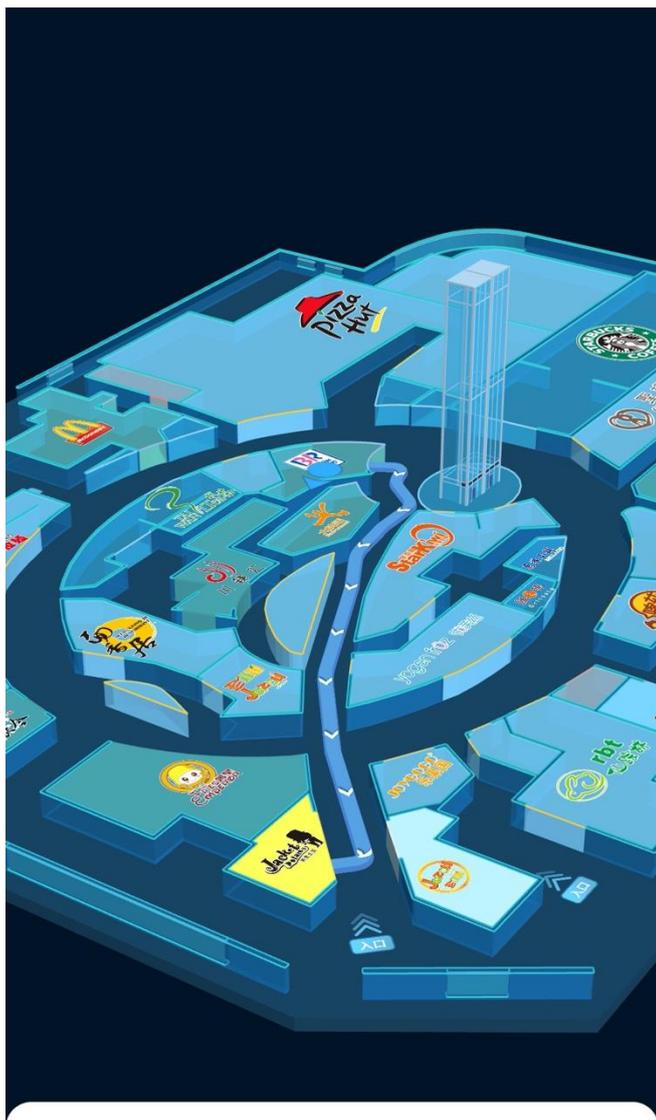
4.4 界面需求

快应用客户端对应的功能需求:

最佳逃生路径



室内导航最佳路径



夹克土豆



距您47.6m | 商户1FA座15号

查看店铺



搜索



收藏



分享

路线

导航

查看人流情况

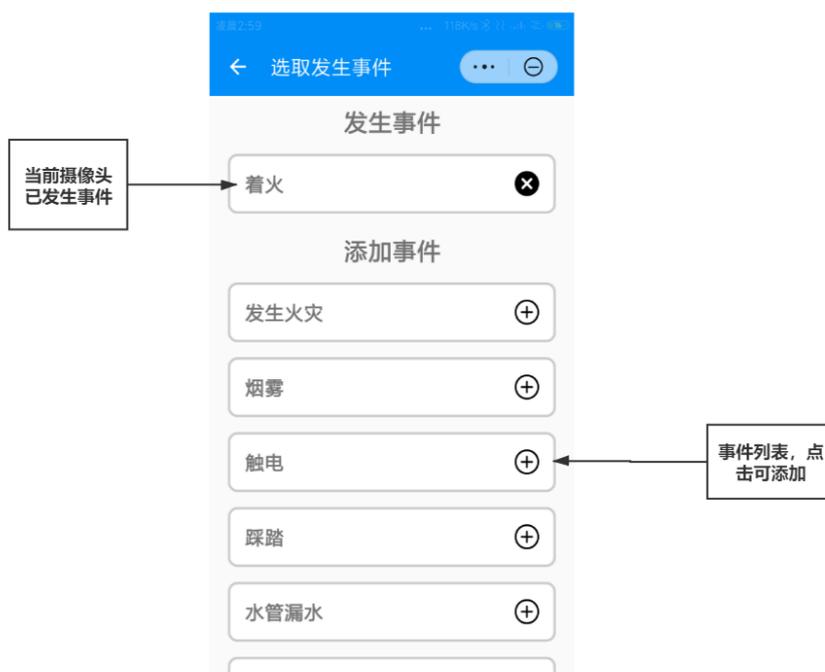
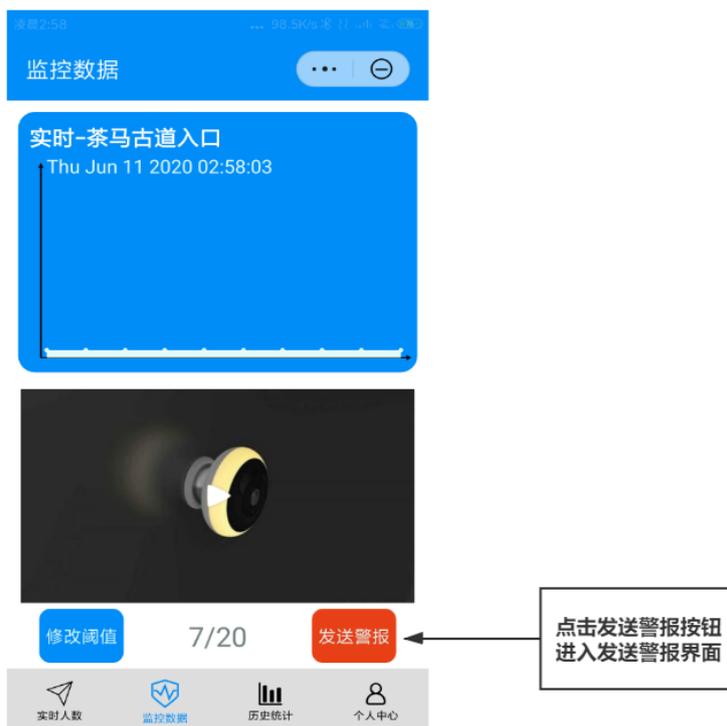


🚩 紧急事件



快应用管理端对应的功能需求:

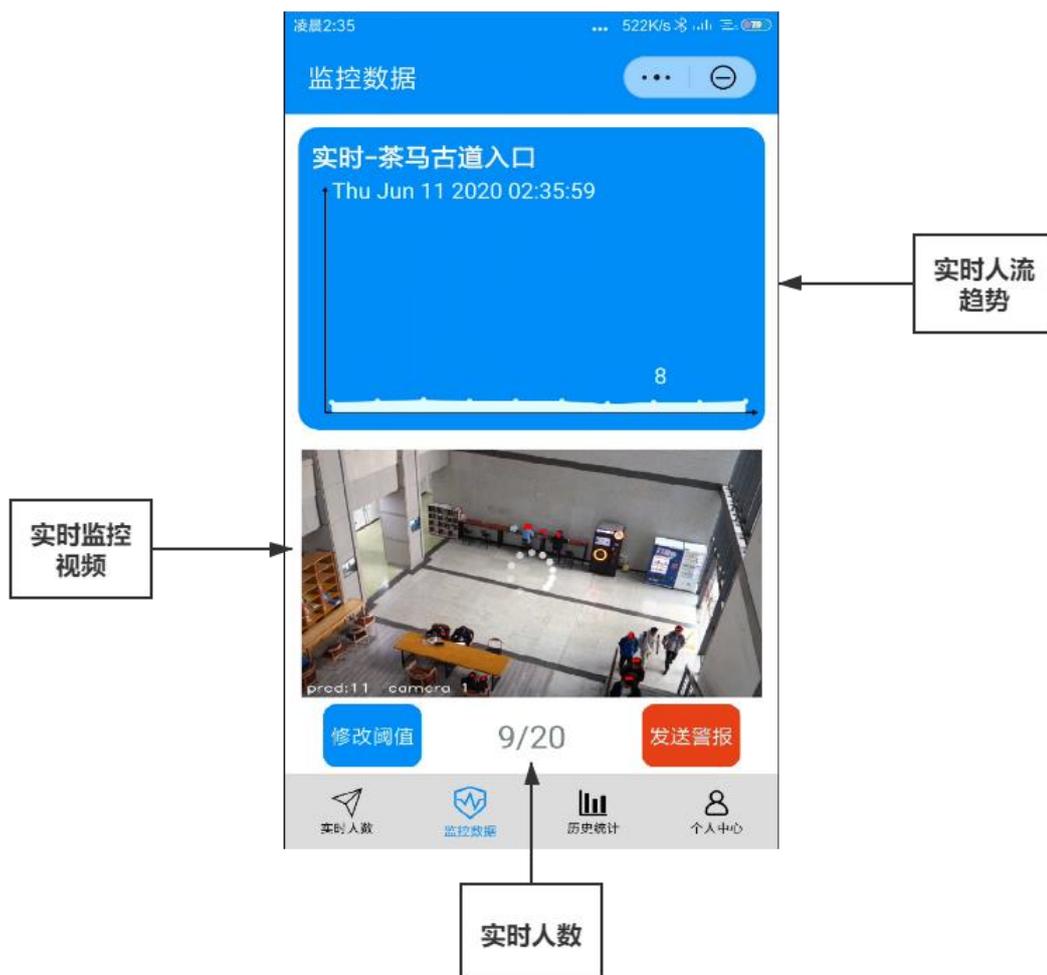
- 为节点添加一个紧急事件



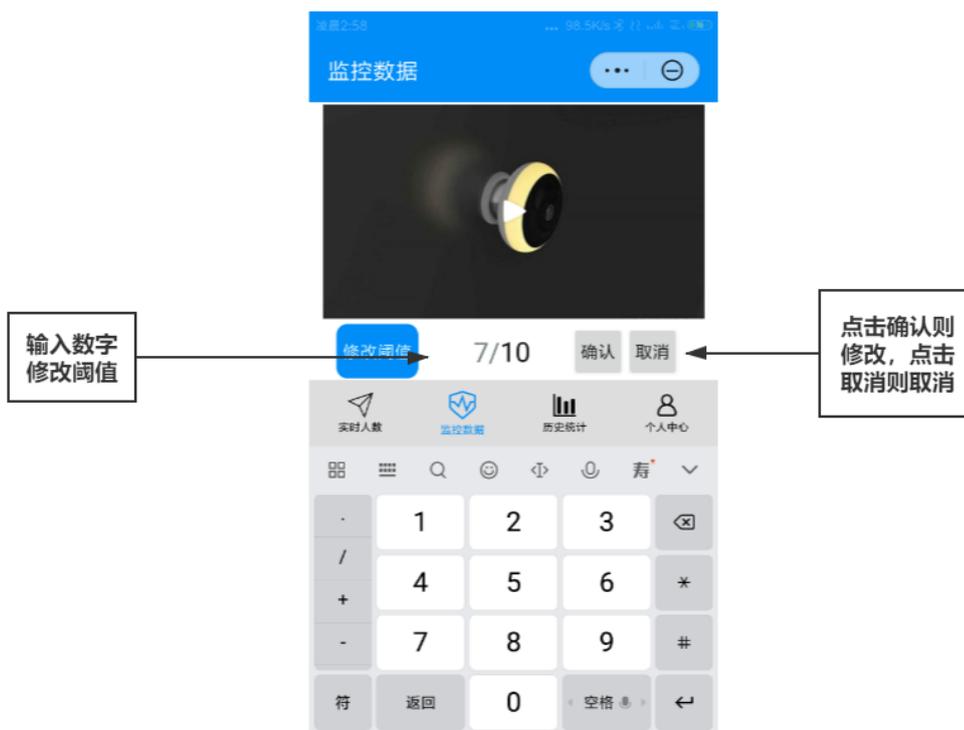
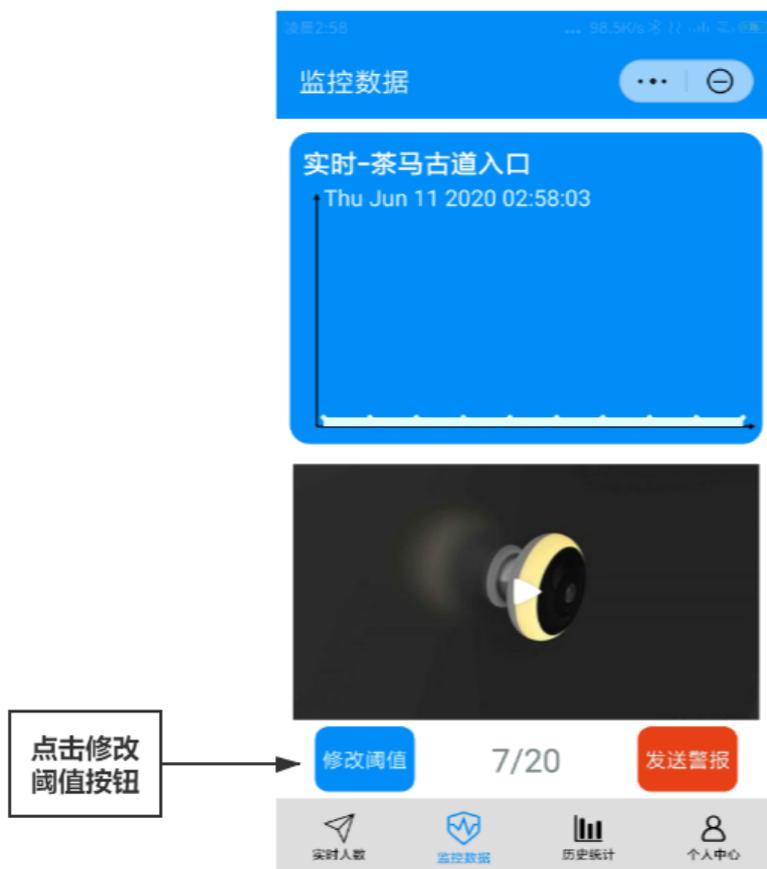
- 为摄像头添加一个新事件



● 查看人流情况



- 修改阈值



4.5 接口需求

4.5.1 硬件接口

- 需要通过光纤或是 usb 接口连接的摄像头
- 需要支持 pci-e 3.0 接口的 GPU

4.5.2 软件接口

接口名	接口描述	接口参数	接口返回实例
/addAccidentType	增加一种出口的事件	description -string	{ "description": "string", "id": 0 }
/deleteAccident2Camera	为摄像头去掉指定事件	accid - integer(\$int32) cameraid - integer(\$int32)	"string"
/getBasePoints	建筑基准点, 依次是左下, 右下, 左上, 右上	/	[[0]]
/getCurrentAccidents	获取现在整个建筑中发生 的所有 事故, 如果没有事故就为空	/	[{ "accidentid": 0, "cameraid": 0, "description": "string", "position": "string" }]
/getPredictedValue	获取预测人数	cameraid - integer(\$int32)	0
/raiseAccident2Camera	为摄像头增加一个事件	accidentid - integer(\$int32) cameraid - integer(\$int32)	"string"

/getAllPath	获得所有的路径	cameraid integer(\$int32)	- [{ "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }]
/getBestPath	获取最佳逃生路径	cameraid integer(\$int32)	- { "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }
/getBestPathWithDest	室内目的地导航	destid integer(\$int32) cameraid integer(\$int32)	- { "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }
/getCurrentNumById	获取相机的当前人数	cameraid integer(\$int32)	- 0

/getHistorys	获得指定相机的所有历史数据	cameraid integer(\$int32)	- [{ "camid": 0, "nums": 0, "shoturl": "string", "timestamp": { "date": 0, "day": 0, "hours": 0, "minutes": 0, "month": 0, "nanos": 0, "seconds": 0, "time": 0, "timezoneOffset": 0, "year": 0 } }]]
/setCurrentNums	设置当前人数, 在历史数据中, 快照一栏为 null	camid integer(\$int32) newNums integer(\$int32)	- "string"
/addCamera	添加摄像头, 返回新摄像头的详情	floor integer(\$int32) isexit -boolean latitude number(\$double) longitude number(\$double) positionDescription -String threshold integer(\$int32)	- { "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }
/addNeighbor	为指定相机增加一个邻里关系	cameraid1 integer(\$int32) cameraid2	- "string"

		integer(\$int32)	
/getNeighbor	查询节点的邻里关系	cameraid - integer(\$int32)	[0]
/getRtmpPlay	获取 RTMP 播放地址和推流地址	cameraid - integer(\$int32)	"string"
/getFlvPlay	获取 Flv 播放地址和推流地址	cameraid - integer(\$int32)	"string"
/login	登录, 身份验证接口	account - string password - string	"string"
/sign	注册新用户, 管理员数据导入接口	account - string password - string	"string"
/getNearestFreeExit	通过 id 返回最佳逃生路线上的下一个节点	cameraid - integer(\$int32)	{ "nodeName": "string", "nodeid": 0 }
/getAllCamera	获取所有的摄像头信息和属性值	/	[{ "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }]
/getPicNow	获得相机当前快照图片路径	cameraid - integer(\$int32)	"string"

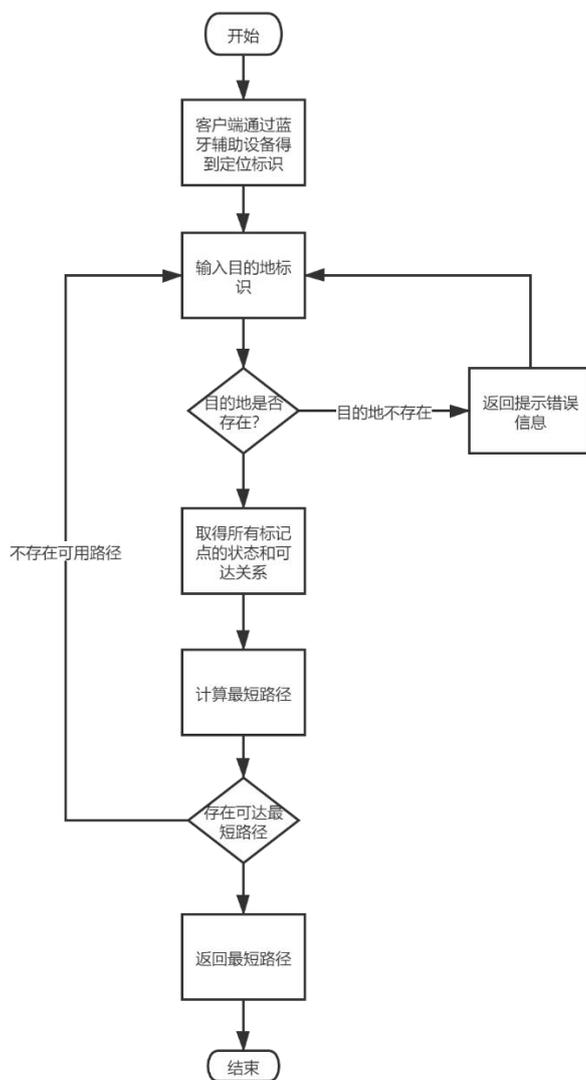
/searchByName	根据名字搜索, 给快应用的搜索功能	positionName string	- [{ "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }]
---------------	-------------------	------------------------	---

4.6 其他需求

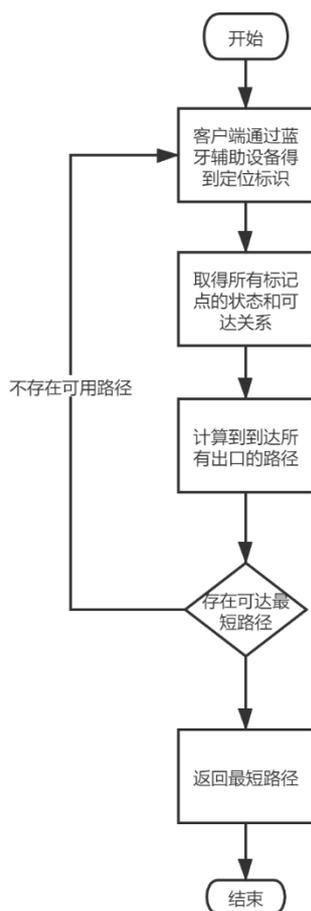
- 系统的后端代码应该有较好的可移植性, 以便于进行服务的迁移
- 系统将可以使用短信验证码登陆或更改密码
- 后端接口应当限制不同用户对于文件的访问权限
- 前端接口应当方便于各种类型的模型调用

5 概要设计

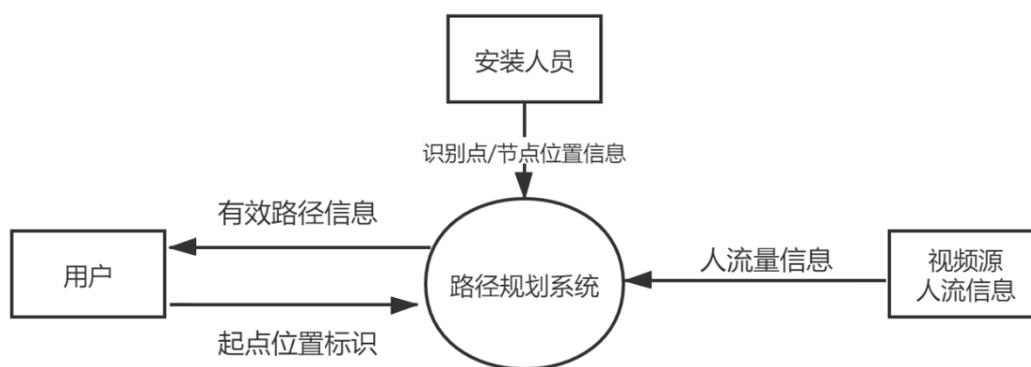
5.1 处理流程



(导航路径请求流程图)

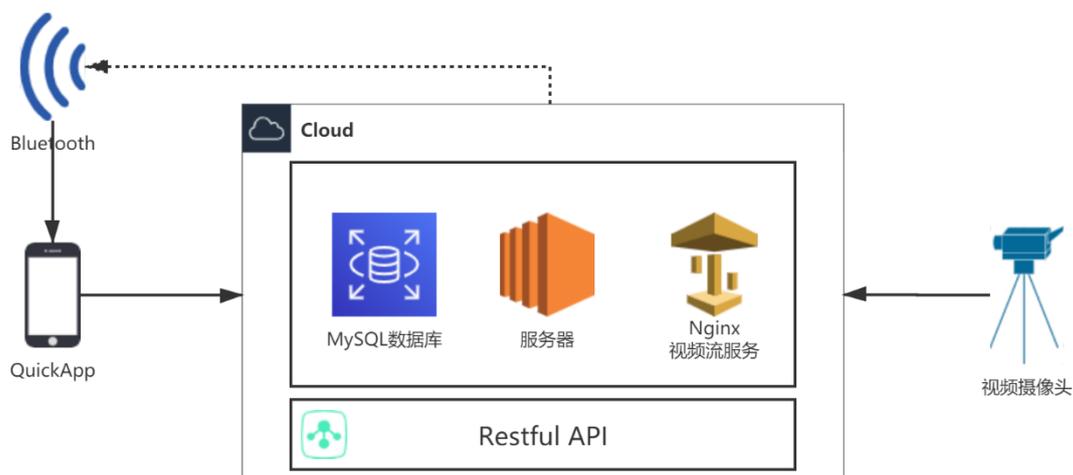


(逃生路径规划流程图)



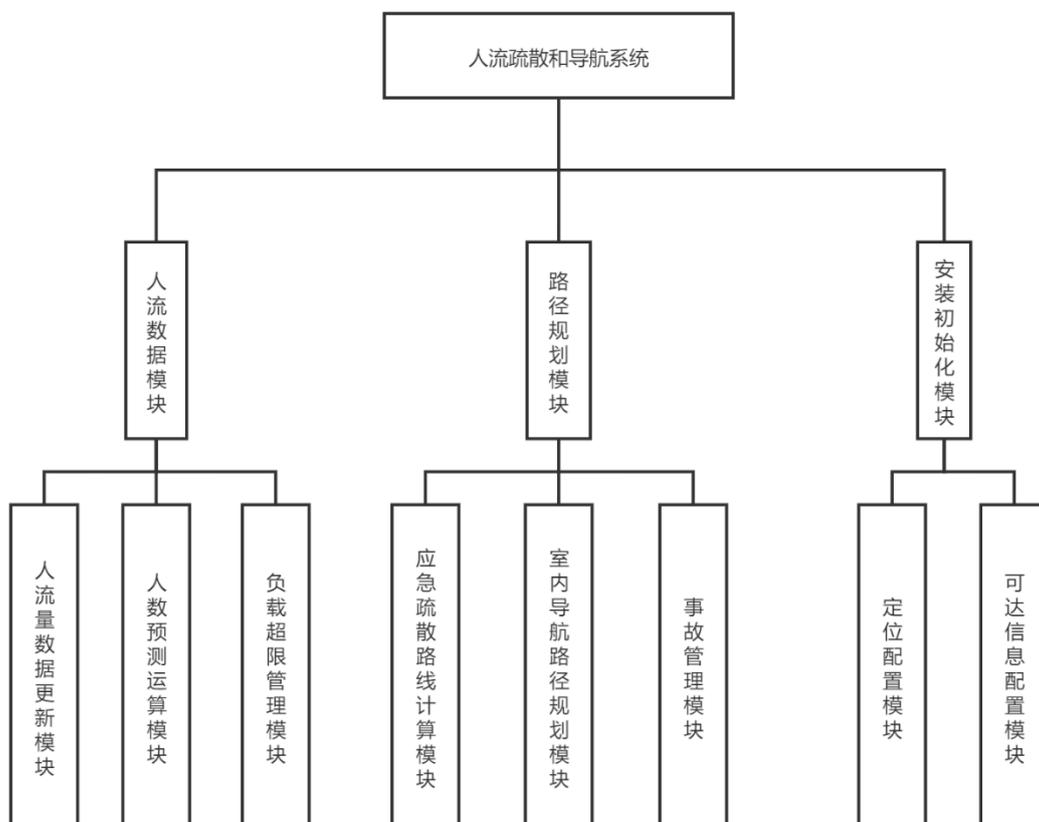
(系统数据流图)

5.2 总体结构设计



(系统结构设计图)

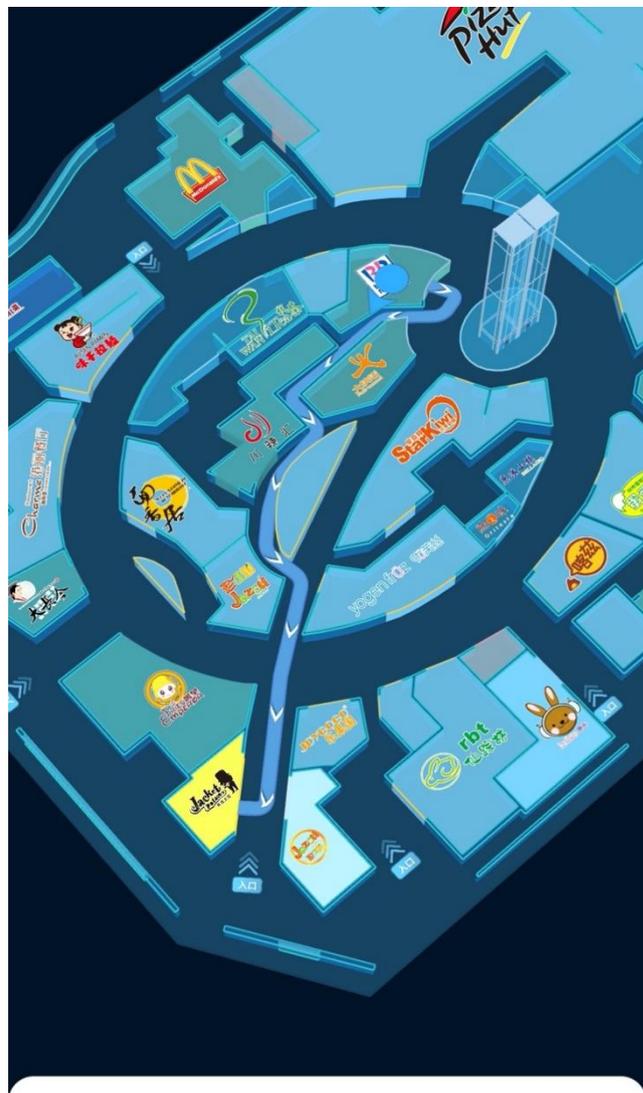
5.3 功能设计



5.4 用户界面设计

5.4.1 用户端

路径规划



夹克土豆



距您47.6m | 商户1FA座15号

查看店铺



搜索



收藏

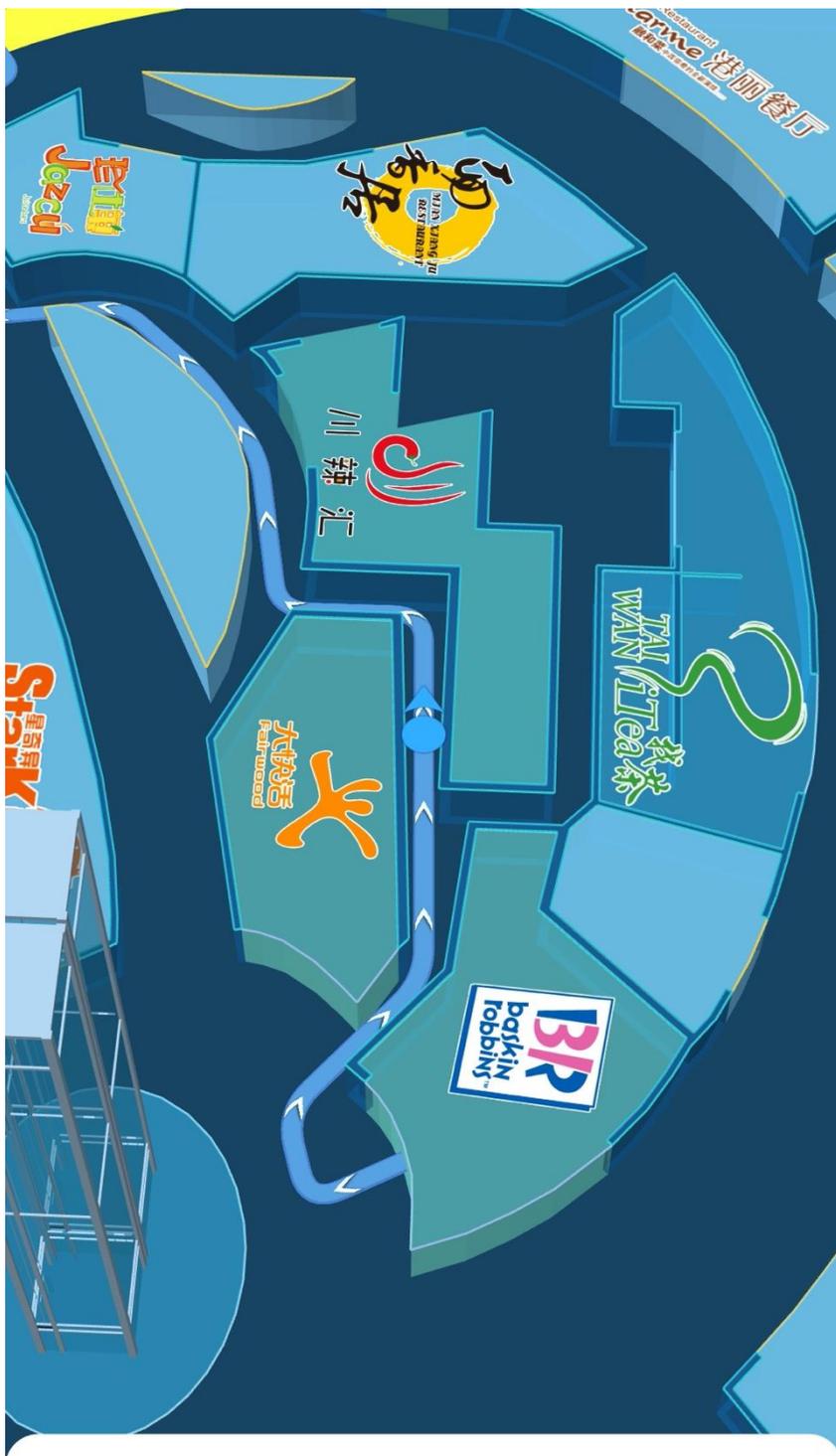


分享

路线

导航

导航



正在前往 薯皇

还剩 48.0米



直行4米 左转

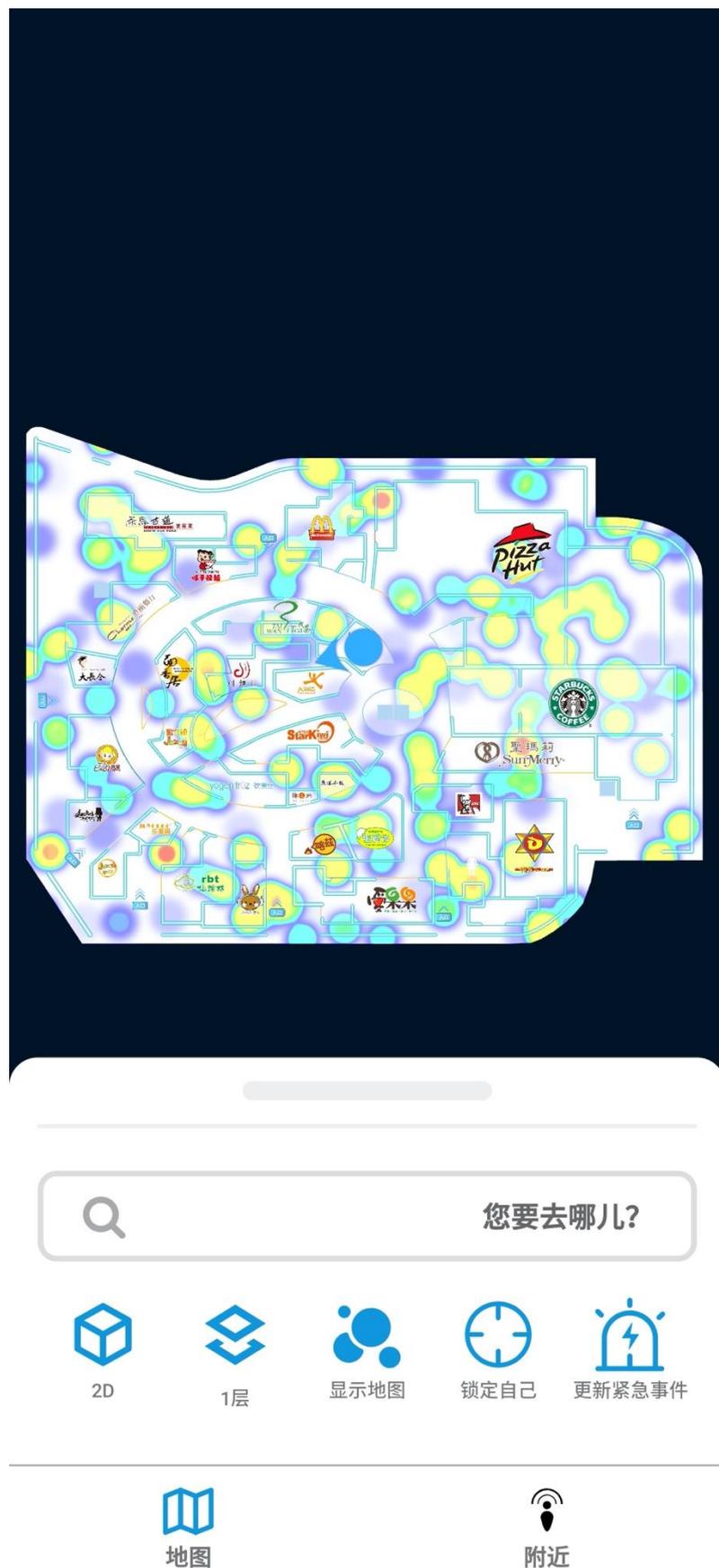
直行5米 右转

0分36秒

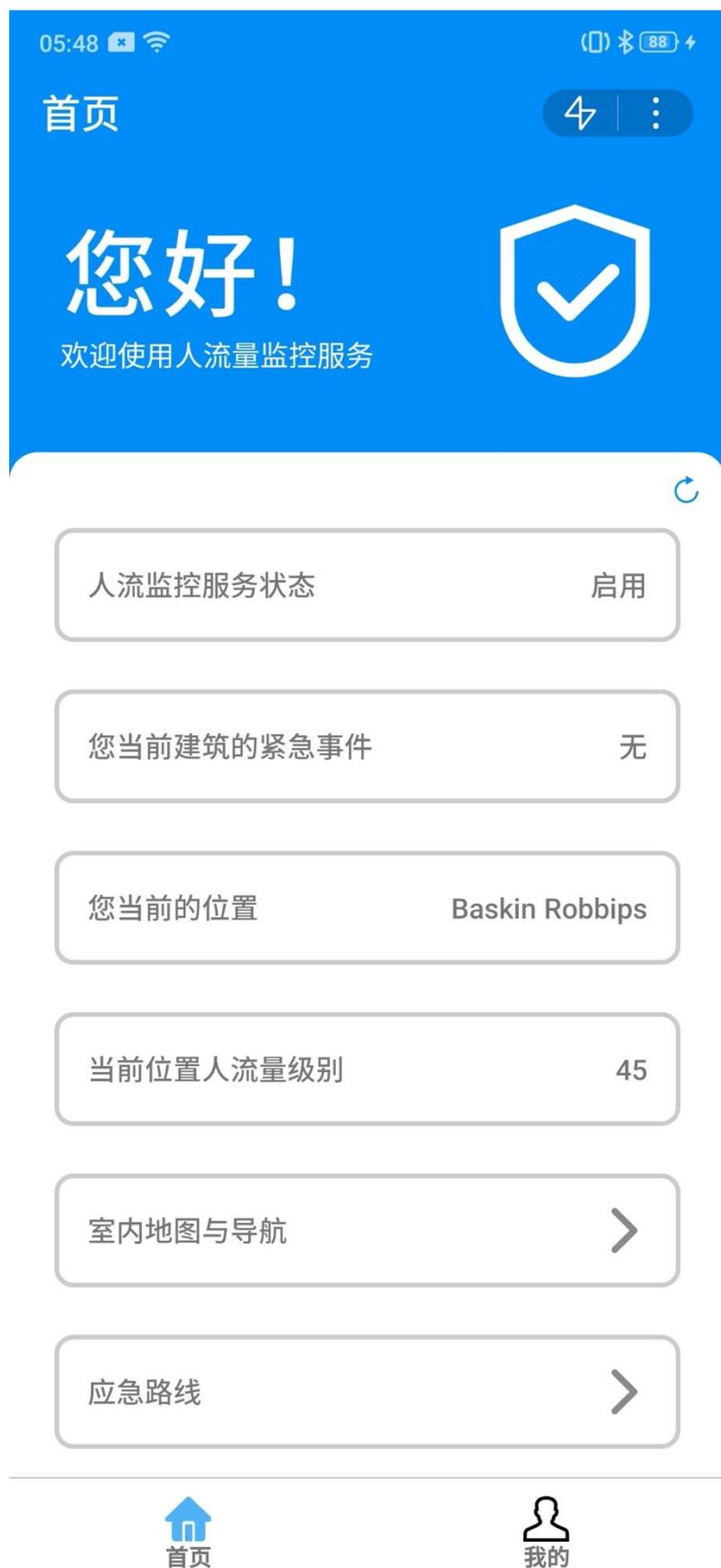
重置导航

取消导航

人流量热力图

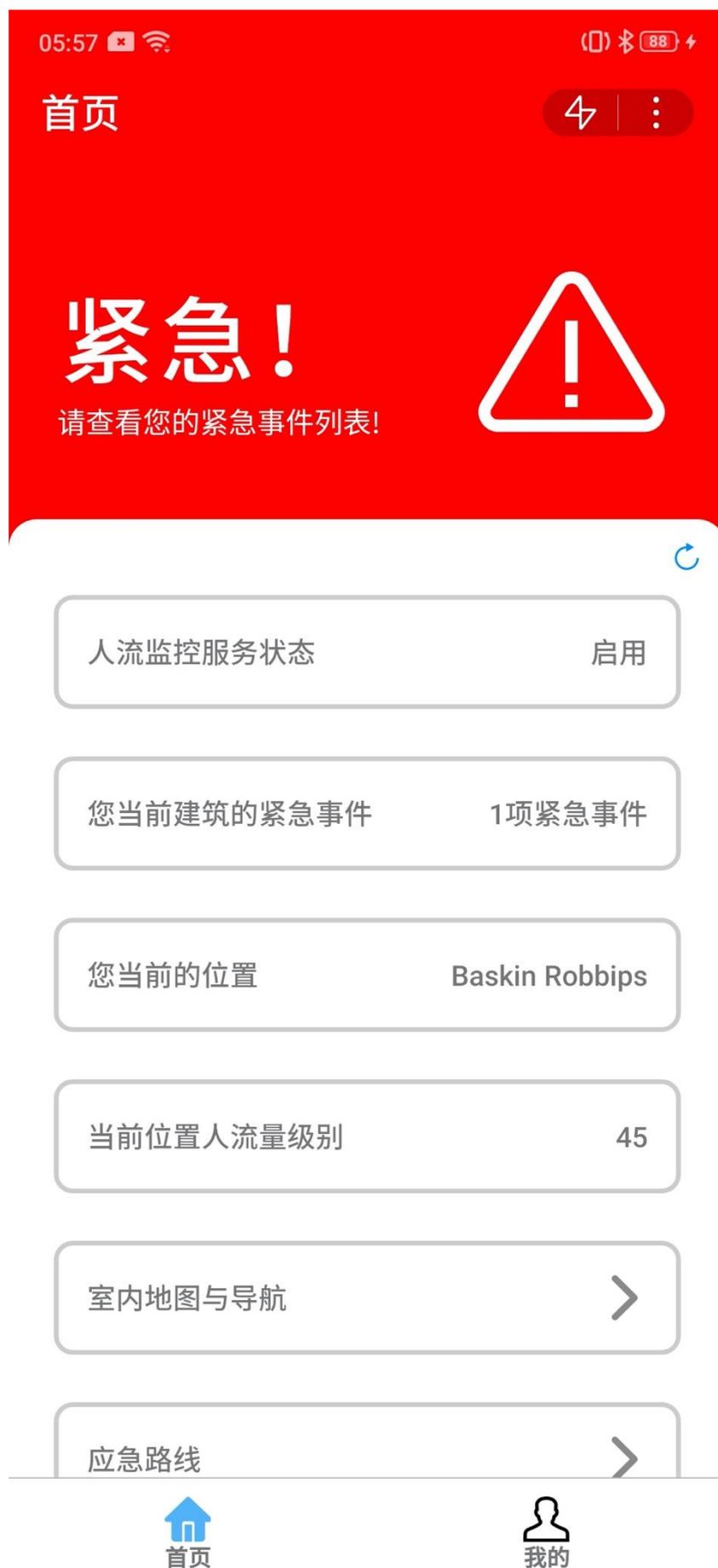


人流量级别



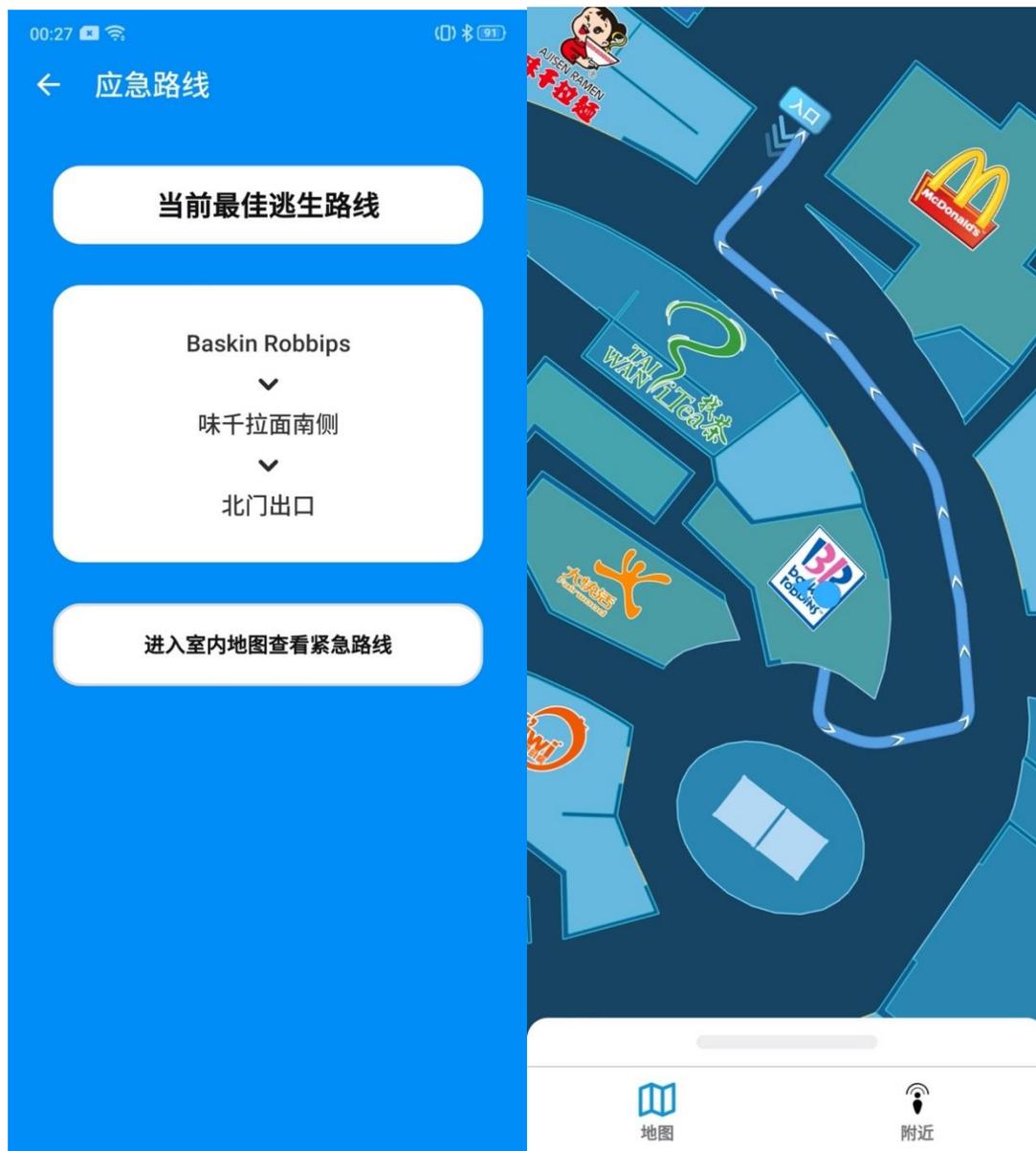
紧急事件列表



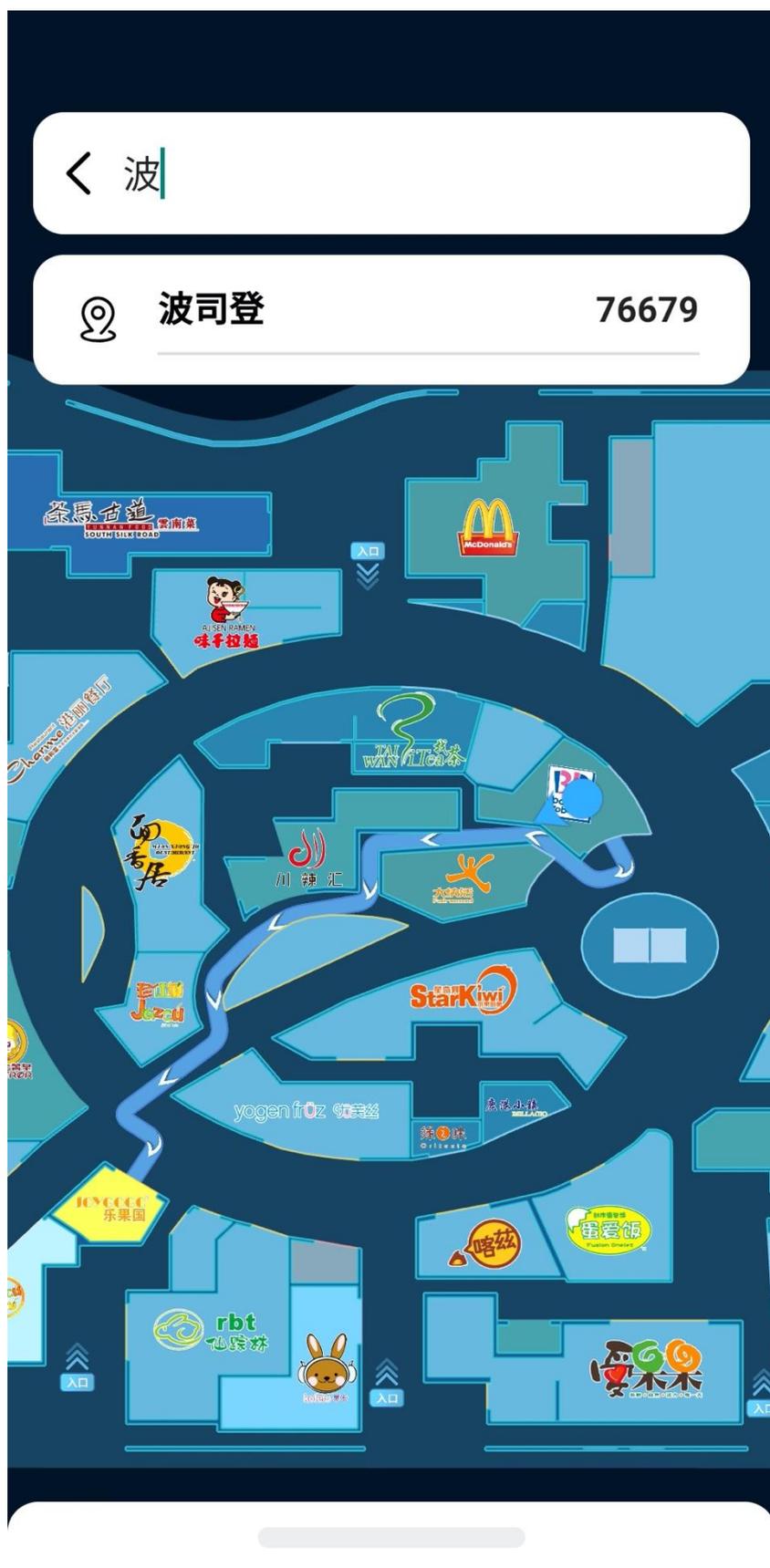




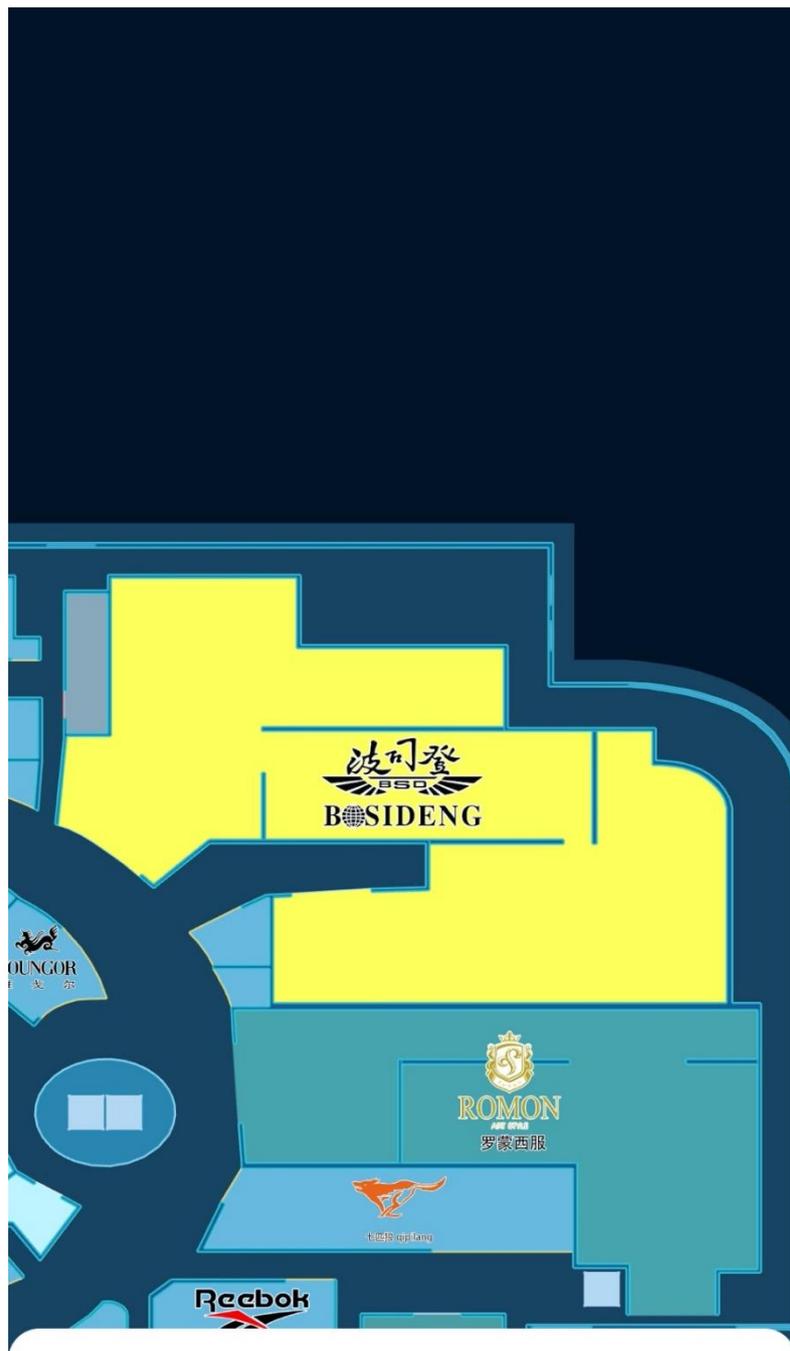
应急路线（最佳逃生路线）



搜索商店



展示商店信息



波司登



距您30.3m | 商户2FA座1号

[查看店铺](#)



搜索



收藏

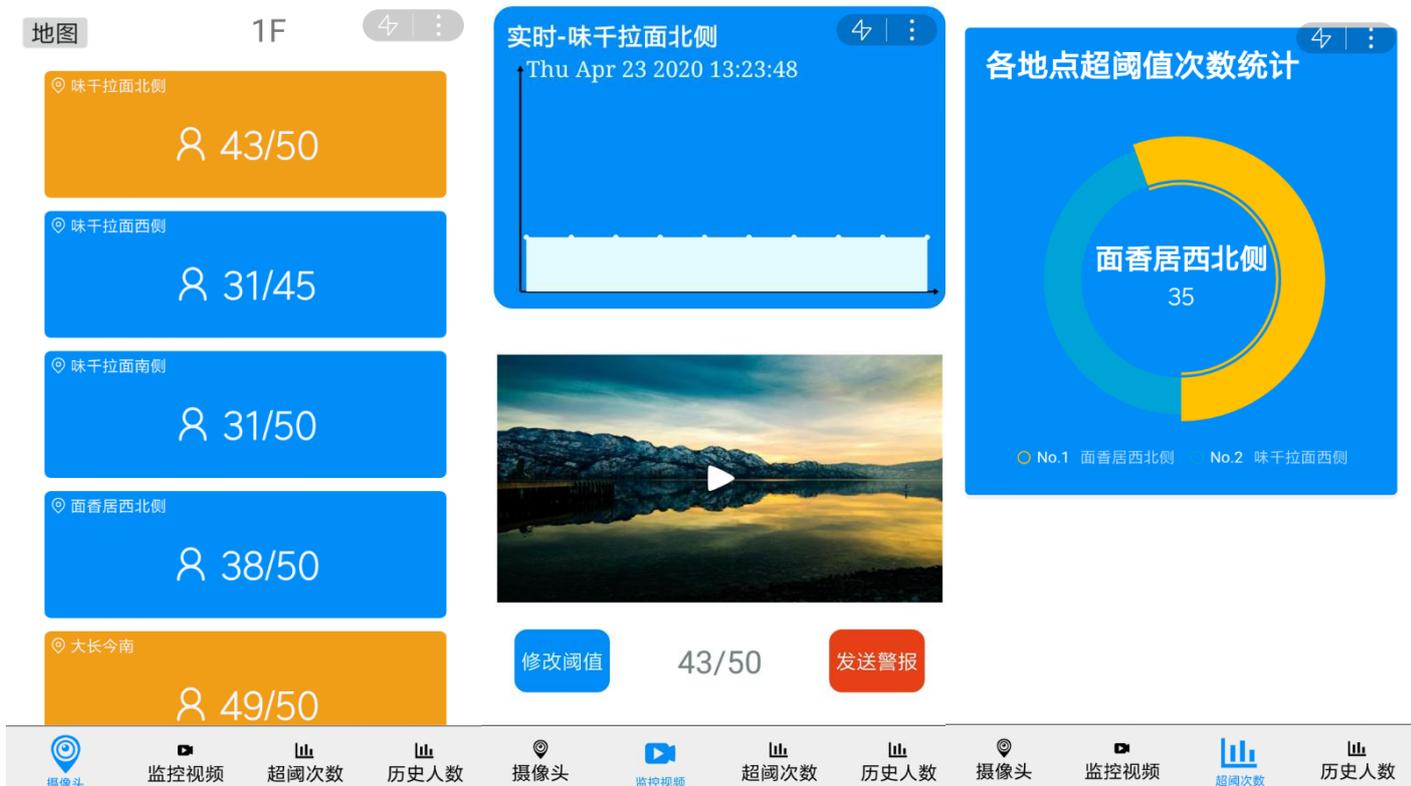


分享

路线

导航

5.4.1 管理端



上图从左到右分别摄像头列表功能、监控视屏功能和超阈值次数统计功能



上图从左到右分别为摄像头历史人数统计功能、修改阈值功能和发送警报功能

5.5 数据结构设计

实体名	属性名	类型	属性说明
Accident2Camera	accidentid	String	事件 id
	description	String	事件说明
	camid	String	相机的对应 id
	longitude	double	纬度
	latitude	double	经度
camera	id	String	相机 id
	position	String	相机的位置描述
	threshold	int	路径节点的负载
	longitude	double	纬度
	latitude	double	经度
	times	int	过载次数
	param	int	参数
	lastshoturl	String	上次记录人数的快照
	nums	int	现在人数, 最新人数纪录
	isexit	boolean	是否是个出口
	floor	int	层数
	isfree	boolean	是否空闲
	accidentid	int	事件 id
camera_history	id	int	记录 id
	camid	int	相机 id
	nums	int	记录对应的人数
	time	timestamp	记录产生时时间
	shoturl	String	快照链接
guard	id	String	保卫人员 id
	name	String	保安人员名字
	phone	String	手机号
	longitude	double	纬度
	latitude	double	经度
neighbors	sourceid	int	相邻关系的一方 id
	sname	String	相邻关系的一方名字
	Neighborname	String	相邻关系的一方名字
	neighborid	int	相邻关系的一方 id
user	account	String	用户账号
	password	String	用户密码
	phone	String	用户手机号
video	id	int	视频素材的 id
	name	String	视频素材的名字

	time	long	视频素材的记录时间
	length	int	视频素材的长度
	position	String	视频素材的位置
	file_position	String	视频素材的文件路径
	camera_id	int	对应的相机 id
PathWrapper	fromNode	int	起始点的 id
	ToNode	Int	目的地点的 id
	Pathdata	PathNode[]	内部的路径
PathNode	nodeid		节点的 id
	Longitude	节点的纬度	节点的纬度
	Latitude		节点的经度
	Floor		节点的楼层

5.6 接口设计

接口名	接口描述	接口参数	接口返回实例
/addAccidentType	增加一种出口的事件	description - string	{ "description": "string", "id": 0 }
/deleteAccident2Camera	为摄像头去掉指定事件	accid - integer(\$int32) cameraid - integer(\$int32)	"string"
/getBasePoints	建筑基准点，依次是左下，右下，左上，右上	/	[[0]]
/getCurrentAccidents	获取现在整个建筑中发生的所有事故，如果没有事故就为空	/	[{ "accidentid": 0, "cameraid": 0, "description": "string", "position": "string" }]
/getPredictedValue	获取预测人数	cameraid - integer(\$int32)	0
/raiseAccident2Camera	为摄像头增加一个事件	accidentid - integer(\$int32)	"string"

ra		cameraid integer(\$int32)	-	
/getAllPath	获得所有的路径	cameraid integer(\$int32)	-	[{ "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }]
/getBestPath	获取最佳逃生路径	cameraid integer(\$int32)	-	{ "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }
/getBestPathWithDest	室内目的地导航	destid integer(\$int32) cameraid integer(\$int32)	- -	{ "fromNode": 0, "path": [{ "nodeName": "string", "nodeid": 0 }], "pathLength": 0, "toNode": 0 }
/getCurrentNumById	获取相机的当前人数	cameraid integer(\$int32)	-	0

/getHistorys	获得指定相机的所有历史数据	cameraid - integer(\$int32)	[{ "camid": 0, "nums": 0, "shoturl": "string", "timestamp": { "date": 0, "day": 0, "hours": 0, "minutes": 0, "month": 0, "nanos": 0, "seconds": 0, "time": 0, "timezoneOffset": 0, "year": 0 } }]
/setCurrentNums	设置当前人数,在历史数据中,快照一栏为 null	camid - integer(\$int32) newNums - integer(\$int32)	"string"
/addCamera	添加摄像头,返回新摄像头的详情	floor - integer(\$int32) isexit -boolean latitude - number(\$double) longitude - number(\$double) positionDescription -String threshold - integer(\$int32)	{ "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }
/addNeighbor	为指定相机增加一个邻里关系	cameraid1 - integer(\$int32) cameraid2 - integer(\$int32)	"string"
/getNeighbor	查询节点的邻里关系	cameraid - integer(\$int32)	[0]
/getRtmpPlay	获取 RTMP 播放地址和	cameraid -	"string"

	推流地址	integer(\$int32)	
/getFlvPlay	获取 Flv 播放地址和推流地址	cameraid - integer(\$int32)	"string"
/login	登录, 身份验证接口	account - string password - string	"string"
/sign	注册新用户, 管理员数据导入接口	account - string password - string	"string"
/getNearestFreeExit	通过 id 返回最佳逃生路线上的下一个节点	cameraid - integer(\$int32)	{ "nodeName": "string", "nodeid": 0 }
/getAllCamera	获取所有的摄像头信息和属性值	/	[{ "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }]
/getPicNow	获得相机当前快照图片路径	cameraid - integer(\$int32)	"string"
/searchByName	根据名字搜索, 给快应用的搜索功能	positionName - string	[{ "floor": 0, "id": 0, "isexit": true, "latitude": 0, "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }]

<p>/navigateWithAStar</p>	<p>返回使用 Astar 算法给出的最短路径</p>	<p>cameraid * integer(\$int32)</p> <p>destCameraId * integer(\$int32)</p>	<pre>{ "fromNode": 0, "path": [{ "floor": 0, "id": 0, "isexit": true, "isfree": true, "latitude": 0, "logicalPosition": "string", "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }], "pathLength": 0, "toNode": 0 }</pre>
<p>/getBestPathWithDes t</p>	<p>迪杰斯特拉版本的求最短路径</p>	<p>cameraid * integer(\$int32)</p> <p>destCameraId * integer(\$int32)</p>	<pre>{ "fromNode": 0, "path": [{ "floor": 0, "id": 0, "isexit": true, "isfree": true, "latitude": 0, "logicalPosition": "string", "longitude": 0, "nums": 0, "param": 0, "position": "string", "threshold": 0, "times": 0 }], "pathLength": 0,</pre>

			<pre>"toNode": 0 }</pre>
--	--	--	--------------------------

5.7 错误/异常处理设计

5.7.1 错误/异常输出信息

序号	异常情况	异常输出信息
1	用户输入的目标地址不存在	客户端返回 404，在前端界面显示
2	用户所在位置没有合适的逃生路径	客户端输出“路线查找失败”
3	用户现在输入的导航目的地不可达	客户端输出“路线查找失败”

5.7.2 错误/异常处理对策

序号	错误/异常	处理对策
1	用户输入的目的地不存在	后端程序进行判断之后返回错误代码，前端向用户发起提醒消息，告知目的地不存在
2	用户定位失败	前端客户端程序告知用户定位失败，要求打开相应的功能开关：GPS 定位或者蓝牙模块，以链接我们的定位设备。
3	用户当前请求逃生路径时，没有可用的逃生路径	后端程序返回代表没有可用路径的错误代码，前端程序提醒用户现在没有可用的逃生路径，提醒用户现在没有可用的路径，原地等待救援。
4	在路径上存在节点处于意外情况当中	前端向后端拉取异常信息，同时在前端页面进行相应展示。

5.8 系统配置策略

边缘端设备	辅助定位设备	蓝牙 4.0 信号收发器
	电子展示板	安卓平板等电子面板
	摄像头设备	网络摄像头
后端设备	WEB 服务器	操作系统：Linux 发行版 - Ubuntu

		处理器：高性能处理器 内存：2G 大小及以上高性能内存
	人流量识别服务器	运行环境要求：Python 解释器 GPU：高性能图像处理器 CPU：高性能处理器 内存：32G 及以上内存
快应用程序运行设备	终端快应用程序	运行所有支持快应用平台的终端设备上。

5.9 系统部署方案

边缘端设备	蓝牙设备	作为辅助定位设备，安装在建筑内所有需要作为路径标记点的位置
	摄像头设备	作为整个系统最重要的数据来源，需要安装在左右需要采集人流量信息的位置：例如商户入口、楼梯间、通道转角
	外置显示终端设备	安装在需要指示牌的区域，使用安卓平板的屏幕来进行提示。终端设备的软件就运行在平板设备上
服务端程序	WEB 服务后端程序	位于大楼 IT 中心机房的服务器上运行
	摄像头人流量识别程序	位于大楼 IT 中心机房的服务器上运行
客户端程序	客户端快应用程序	运行在每个系统使用者的终端设备上

5.10 其他相关技术与方案

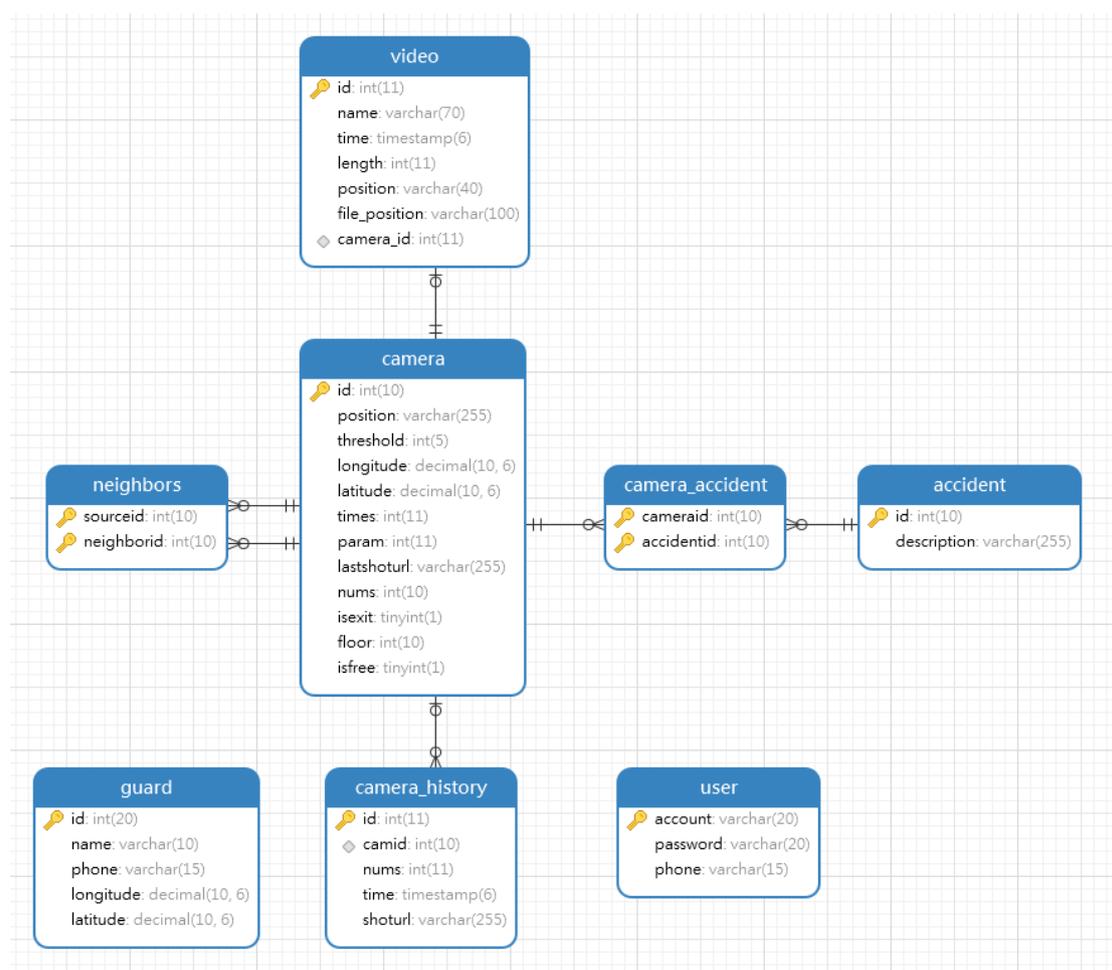
6 数据库设计

6.1 数据字典设计

表名	列名	类型	说明
accident	id	int(10)	事件 id
	description	varchar(255)	事件说明

camera	id	int(10)	相机 id
	position	varchar(255)	相机的位置描述
	threshold	int(5)	路径节点的负载
	longitude	decimal(10,6)	纬度
	latitude	decimal(10,6)	经度
	times	int(11)	过载次数
	param	int(11)	参数
	lastshoturl	varchar(255)	上次记录人数的快照
	nums	int(10)	现在人数, 最新人数纪录
	isexit	tinyint(1)	是否是个出口
	floor	int(10)	层数
	isfree	tinyint(1)	是否空闲
camera_accident	cameravid	int(10)	相机 id
	accidentid	int(10)	事件 id
camera_history	id	int(11)	记录 id
	camid	int(10)	相机 id
	nums	int(11)	记录对应的人数
	time	timestamp(6)	记录产生时时间
	shoturl	varchar(255)	快照链接
guard	id	int(20)	保卫人员 id
	name	varchar(10)	保安人员名字
	phone	varchar(15)	手机号
	longitude	decimal(10,6)	纬度
	latitude	decimal(10,6)	经度
neighbors	sourceid	int(10)	相邻关系的一方 id
	neighborid	int(10)	相邻关系的一方 id
user	account	varchar(20)	用户账号
	password	varchar(20)	用户密码
	phone	varchar(15)	用户手机号
video	id	int(11)	视频素材的 id
	name	varchar(70)	视频素材的名字
	time	timestamp(6)	视频素材的记录时间
	length	int(11)	视频素材的长度
	position	varchar(40)	视频素材的位置
	file_position	varchar(100)	视频素材的文件路径
	camera_id	int(11)	对应的相机 id

ER 图



7 详细设计

7.1 室内定位功能模块

7.1.1 功能描述

为终端的定位提供蓝牙辅助。使用 js 接口的方式为快应用端提供准确的室内定位数据。

7.1.2 性能描述

在蓝牙设备发现后，能够立刻计算出结果传入回调。

7.1.3 输入

无

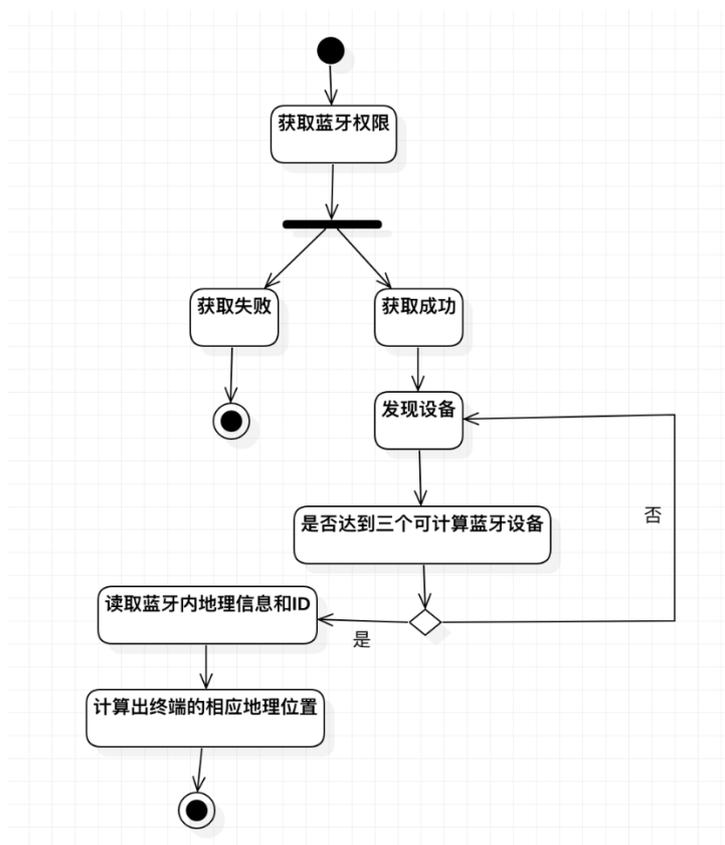
7.1.4 输出

下面是室内定位功能的一个典型的输出样例：

Data Structure:

```
{
  "id": "摄像机 id-{Number}",
  "point": {
    "根据摄像机的物理位置计算的经度-{Number}",
    "根据摄像机的物理位置计算的纬度-{Number}"
  }
}
```

7.1.5 程序逻辑



7.1.6 限制条件

室内定位要求设备正常授权蓝牙权限，并且设备的蓝牙功能正常

7.2 人数预测功能模块

7.2.1 功能描述

用于进行每个观测点在 24 制时间中任意时刻的人流量预测

7.2.2 性能描述

50ms 内完成预测模型的生成同时返回预测结果

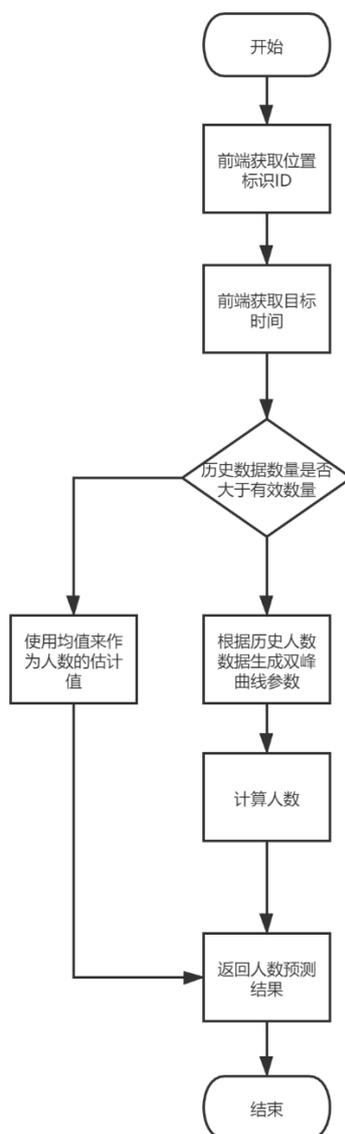
7.2.3 输入

1. 观测点的标识 id
2. 需要预测时间（一天中的分钟数）

7.2.4 输出

人流量预测数据

7.2.5 程序逻辑



7.2.6 限制条件

输入的相机标识符存在于系统当中，已经有了至少一条人流量历史数据

7.3 最佳逃生路线规划功能模块

7.3.1 功能描述

用户所在建筑中发生了意外情况需要进行疏散，需要对当前用户位置进行应急路线规划。

7.3.2 性能描述

系统可以在 50ms 内完成路径查找，同时返回有效的指向建筑出口逃生路径

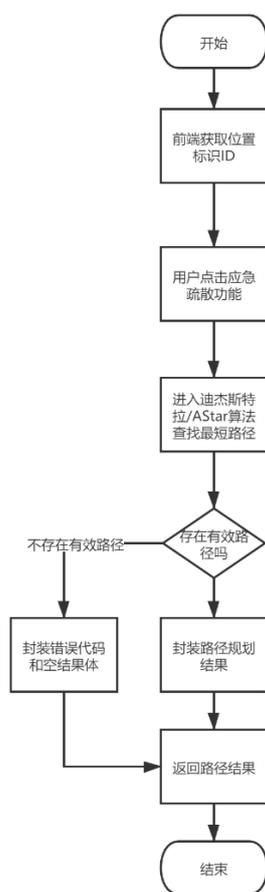
7.3.3 输入

输入为用户的当前位置标识 ID

7.3.4 输出

返回一条有效的指向 EXIT 建筑出口的有效逃生路径

7.3.5 程序逻辑



7.3.6 限制条件

用户当前已经被定位到了有效的节点上。

7.4 事故/意外情况上报功能模块

7.4.1 功能描述

当我们的智能设备或者安保人员发现了建筑内的意外事件之后，需要将事故上报到系统当中，以提醒中控管理人员，同时路径规划系统也会进行危险避免。

7.4.2 性能描述

接口在 50ms 内完成数据注入，同时完成结果返回。

7.4.3 输入

输入的是建筑的位置标识，可以通过快应用设备进行自动配对和读取，同时需要明确故障类型。

7.4.4 输出

返回事故上报到结果

7.4.5 程序逻辑



7.4.6 限制条件

上报端系统已经完成了定位的整个过程。

7.5 快应用—Web 通信解析功能模块

7.5.1 功能描述

采用 Web 开发的室内地图来实现可视化，基于 `system.message` 消息通道，自定义了一套通信协议。快应用侧的消息报文带有 `Hap` 前缀，地图侧的消息报文带有 `Web` 前缀，解析前会验证前缀是否正确，接着在双端都实现了 `runProtocol` 方法，实现核心的消息解析，最终完成快应用和 Web 侧地图的互操作和通信。

7.5.2 性能描述

7.5.3 输入

协议内所规定的对象或字符串，其中，对象形如

```
{
  type: "loc",
  detail: {
  }
}
```

不同的 `type` 对应不同的 `detail` 结构，

而字符串包括但不限于 `2D`、`3D`、`F1`、`F2`、`F3`、`Ready`、`applyHeatMap`、`removeHeatMap`。

7.5.4 输出

执行对应信号的操作，如 Web 侧收到 `2D` 命令将视图显示模式切换为 `2D`，`3D` 亦然，`F1`、`F2`、`F3` 则切换楼层等等。

7.6 人群计数算法功能

7.6.1 功能描述

采用基于密度图的深度学习算法，对于摄像头传输的视频帧进行人群计数，使用基于 `ResNet50` 深度改进的网络作为计数的 `encoder`，使用 `GCC` 数据集进行预训练给了网络很好的鲁棒性，在 `SHTB` 和 `WorldExpo` 数据集上获得了 `SOTA` 级别的精度。

7.6.2 性能描述

人群计数算法在 `SHTB` 和 `WorldExpo` 数据集上获得了 `SOTA` 级别的精度，在配置有 `Nvidia GTX 1080ti` 的人工智能计算端上运行算法处理视频的帧数可以达到 `22.1fps`。

7.6.3 输入

从边缘端摄像头获取的视频流作为输入

7.6.4 输出

向后端持续发送 int 型的估计人数作为输出，此外以 RTMP 流的方式向后端传输实时的监控录像

7.7 快应用—修改节点状况功能模块

7.7.1 功能描述：

管理人员可以根据实时情况与实际信息，修改节点的阈值。

7.7.2 性能描述

管理人员在修改节点阈值，在 1s 内成功实现，保证警报的及时发送。

7.7.3 输入

- 修改摄像头节点阈值：

管理人员通过点击修改阈值，输入要修改的阈值数，点击确认后发送带有修改的阈值数以及当前修改的摄像头 id 发送给服务器修改。

7.7.4 输出

修改摄像头节点阈值：

当前界面上摄像头的阈值数被修改。

7.8 快应用—节点警报的发送/取消

7.8.1 功能描述：

管理人员可以根据实时情况与实际信息，对突发事件发送警报。

7.8.2 性能描述

管理人员对突发事件的摄像头发送警报，在 1s 内成功实现，保证警报的及时发送。

事件发生结束后，管理人员对该摄像头事件进行取消，在 1s 内实现。

7.8.3 输入

- 发送摄像头的警报事件：
管理人员通过点击发送警报按钮，选择当前摄像头内发生的事件点击
- 取消摄像头警报事件
管理人员点击相应摄像头当前发生事件中要取消的事件

7.8.4 输出

- 发送摄像头的警报事件：
添加事件至已发生的事件中
- 取消摄像头警报事件
点击的取消事件在当前已发生事件列表中消除。