[项目LOGO]

[商超导购]

**[基于Wi-Fi指纹的超市定位导航系统]**

软件设计说明书

**Version [1.7]**

[2022.7.14]

**Written by [Team Name]**

[Team LOGO]

**All Rights Reserved**

目录

[1 摘要 1](#_Toc108715496)

[2 引言 1](#_Toc108715497)

[2.1 项目研究背景及意义（应用场景、目标人群） 1](#_Toc108715498)

[2.2 相关产品及对比 1](#_Toc108715499)

[2.3 项目价值及创新 1](#_Toc108715500)

[3 开发计划 2](#_Toc108715501)

[3.1 主要功能描述 2](#_Toc108715502)

[3.2 关键难点问题 2](#_Toc108715503)

[3.3 进度安排 3](#_Toc108715504)

[4 可行性分析 3](#_Toc108715505)

[4.1 市场可行性分析 3](#_Toc108715506)

[4.2 技术可行性分析 3](#_Toc108715507)

[5 需求分析 4](#_Toc108715508)

[5.1 数据采集及描述 4](#_Toc108715509)

[5.2 功能性需求 5](#_Toc108715510)

[5.2.1 功能模块结构图 5](#_Toc108715511)

[5.2.2 核心功能模块描述 5](#_Toc108715512)

[5.3 非功能性需求（精度、性能、易用性、可扩展性、可维护性等） 5](#_Toc108715513)

[5.4 用例图 6](#_Toc108715514)

[6 概要设计 6](#_Toc108715515)

[6.1 系统架构设计 6](#_Toc108715516)

[6.2 系统总体功能结构设计 7](#_Toc108715517)

[6.3 关键技术设计 8](#_Toc108715518)

[6.3.1 前端部分 8](#_Toc108715519)

[6.3.2 后端部分 8](#_Toc108715520)

[6.3.3 算法部分 9](#_Toc108715521)

[6.4 数据库设计 9](#_Toc108715522)

[6.5 错误/异常处理设计 10](#_Toc108715523)

[6.5.1 个人信息部分 10](#_Toc108715524)

[6.5.2 接口实现部分 10](#_Toc108715525)

[6.5.3 前后端数据传输部分 10](#_Toc108715526)

[6.5.4 界面死循环部分 11](#_Toc108715527)

[6.5.5 Wi-Fi信号扫描部分 11](#_Toc108715528)

[6.5.6 室内定位部分 11](#_Toc108715529)

[7 详细设计与实现 11](#_Toc108715530)

[7.1 总流程图 11](#_Toc108715531)

[7.2 用户登录注册子系统的设计与实现 12](#_Toc108715532)

[7.2.1 类图 12](#_Toc108715533)

[7.2.2 流程图： 12](#_Toc108715534)

[7.2.3 用户界面展示 13](#_Toc108715535)

[7.3 定位与导航子系统设计与实现 13](#_Toc108715536)

[7.3.1 类图 13](#_Toc108715537)

[7.3.2 流程图 13](#_Toc108715538)

[7.3.3 用户界面展示 14](#_Toc108715539)

[7.4 附近聊天室子系统设计与实现 14](#_Toc108715540)

[7.4.1 类图 14](#_Toc108715541)

[7.4.2 流程图 14](#_Toc108715542)

[7.4.3 用户界面展示 14](#_Toc108715543)

[7.5 核心算法（伪代码） 14](#_Toc108715544)

[7.5.1 AP信息记录 14](#_Toc108715545)

[7.5.2 坐标调整 15](#_Toc108715546)

[7.5.3 相似度计算 15](#_Toc108715547)

任务记录

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **完成任务** | **占比（总和100%）** | **备 注** |
| 陈杰 20301033 | 服务器及数据库搭建 | 18.5% |  |
| 陈增20301034 | 负责安卓开发及测试 | 18% |  |
| 韩熔 20301036 | 负责安卓开发 | 17.166666666% |  |
| 韦昌靖20301025 | 负责文档撰写及算法设计 | 16% |  |
| 江桀 20301042 | 服务器及数据库搭建 | 15.833333333% |  |
| 周炳旭20301032 | 负责文档撰写及算法设计 | 15% |  |

# 摘要

室内环境下的定位一直是一个很多问题未被解决的领域。由于信号的严重衰减和多径效应，通用的室外定位设施（比如GPS）并不能在建筑物内有效地工作。定位准确性也是一个问题，GPS也许可以指出移动设备在哪一个建筑物，但是室内场景下，人们希望得到更精确的室内位置，这需要更精密的地图信息和更高的定位精度。GPS难以解决室内环境下的一些定位问题，大部分室内环境下都存在Wi-Fi，因此利用Wi-Fi进行定位无需额外部署硬件设备，是一个非常节省成本的方法。然而Wi-Fi并不是专门为定位而设计的，传统的基于时间和角度的定位方法并不适用于Wi-Fi。近十年来，在室内Wi-Fi场景下的定位中，位置指纹法被广泛研究和采用。因此，我们根据Wi-Fi指纹这一信息，设计研究这一超市定位系统。

# 引言

## 项目研究背景及意义（应用场景、目标人群）

Wi-Fi广泛使用在家庭、旅馆、咖啡馆、机场、商场等各类大型或小型建筑物内，这样使得Wi-Fi成为定位领域中一个最引人注目的无线技术。通常，一个Wi-Fi系统由一些固定的接入点组成，它们部署在在室内一些便于安装的位置，系统或网络管理员通常知道这些AP的位置。能连接Wi-Fi的移动设备（比如笔记本电脑、移动电话）相互之间可以直接或间接地通信，因此可以考虑在通信功能外同时实现定位功能。但是Wi-Fi信号并不是为定位而设计的，通常是单天线、带宽小，室内复杂的信号传播环境使得传统的基于到达时间/到达时间差的测距方法难以实现，基于到达信号角度的方法也同样难以实现，如果在Wi-Fi网络中安装能定向的天线又需要额外的花费。因此，近年来大家详细研究的主要是位置指纹方法，通过使用Wi-Fi指纹，我们能够容易增加Wi-Fi的一项功能，能够使在家庭、旅馆、咖啡馆、机场、商场等各类大型或小型建筑物内的人们明确自己的位置，而不是毫无方向。

## 相关产品及对比

相关产品：维小帮3D实景导航

该产品提供了三个万达广场，一个工厂的地图导航服务，能为地图提供3D视角，能够拖动地图调整角度观察，寻找所要前往的店铺，店铺标注明显，可以通过选择起始位置和终点进行导航，通过明显的蓝色导航先提供路线规划。该产品能够进行实景模拟导航，用户可以通过该功能查看实景，熟悉路线。

## 项目价值及创新

我们可以在室内搭建一套完整的基础设施用来定位，但是这样需要很大的代价，包括定位信号占用的频谱资源、用于感知定位信号的嵌入在移动设备中的额外硬件、安装在固定位置的用来发送定位信号的锚节点。因此，大家倾向于使用那些已有的被广泛部署的无线设备去实现室内定位。因此，通过Wi-Fi作为媒介进行定位服务，不乏是一种合适的解决方案。当前，Wi-Fi信号无处不在，通过Wi-Fi进行定位能够更加方便的定位用户在超市中的的位置。我们还能通过Wi-Fi定位为用户提供室内导航，方便用户前往所要挑选商品的地方。

# 开发计划

## 主要功能描述

基本功能：

1.用户登录、注册（手机验证），APP功能主页

2.定位显示

3.移动路径显示

扩展功能

1.超市导航

2.超市采购

关键难点问题

在服务器方面：在此项目中，服务器端包括数据库的部分。主要承担着数据处理等相关功能。主要难点在于：如何存储与提取用户信息，实现增删查改等功能；如何完成相关接口并与前端进行连接；如何将服务器部署到阿里云服务器中。

在安卓开发方面：在此项目中，安卓部分承担着用户交互界面，地图动态显示等功能。

在算法方面：此部分需要实现基于Wi-Fi的室内定位，目标跟踪，附近设施推荐等算法。算法部分的难点在于：如何尽可能提高定位的精度；如何实现移动过程中的实时更新；如何搜索用户体验良好的最短路径；

## 进度安排



最终，项目小组成员基本按照以上计划完成了各项任务。

# 可行性分析

## 市场可行性分析

移动通信服务人均80%的时间在室内完成，移动电话使用和数据连接有80%在室内场景下使用。随着室内活动的丰富，室内定位将为零售、制造、医疗、急救、物流、机器人等行业提供物联网能力和位置信息数据，成为物联网时代的重要基础。

室内场景越来越庞大复杂，大型商超、综合性医院、机场、停车场等场所对于定位和导航的需求也逐渐增多。大型商场希望能够借助室内定位技术为前来购物的消费者提供实时导引服务，同时基于位置提供对应的营销服务。医院希望对医疗设备进行实时定位，便于需要时快速调用;希望能对特殊病患进行定位监护，防止其发生意外。高危化工厂需要对人员进行定位管理，防止发生安全事故等。可以说，室内定位在零售、餐饮、物流、制造、化工、电力、医疗等行业均展现出了广阔的市场前景。

相较于GPS、北斗在室外定位的地位，室内定位目前还没有大规模地普及，是一个很有发展前景的行业。简单来说，现在人们在室外找路，一定会想到用高德、百度等等地图软件，但在室内可真的没有一款软件让人们想起来使用，还是得看商场、交通枢纽里的电子屏地图或者询问路人。

在当前Wi-Fi随处可见的情况下，并且大多数导航并不具备室内导航方案，只具备室外导航，无法做到精确室内导航，因此我们制作这款为用户在超市提供导航的服务软件，定位用户当前位置并对想要去的位置进行导航，具有较高的应用前景，有望推广至各大商超。

## 技术可行性分析

1） 前端部分

前端部分本项目使用Java进行Android开发。其优势在于庞大而完善的生态系统。Java拥有大量的第三方类库，能安全高效跨平台执行。Java编译并运行程序时，必须检查所有可能会遇到的问题以消除错误。它提供了自动垃圾回收功能，防止了程序员在管理内存时犯错误。Java具有良好的可移植性，独立于平台的功能使其易于移植到互联网上的不同计算机上。Java具有高性能的特点。与其他解释和执行语言不同，Java字节码的设计使其可以轻松地将其直接转换为与特定CPU相对应的机器代码，从而获得更高的性能。

基于以上的优势和特点，使用Java进行Android具有实际操作性和可行性。

2） 后端部分

后端部分分为数据库和服务器。数据库中我们使用MySQL，数据库存储用户信息与场景交互信息，使用DMS管理系统进行MySQL数据库实例的管理。MySQL是⼀种关系型数据库管理系统，其将数据保存在不同的表中，增加了速度并提高了灵活性。它具有体积小、速度快、总体拥有成本低和开放源码等特点，与本项目的目标特点相符。

服务器部分选用Java+Spring+MySQL的方案。Spring具有低侵入设计，代码污染极低，独立于各种应用服务器，基于Spring架构的应用可以实现Write Once，Run Anywhere的承诺，开发者可以自由选择Spring框架的部分或者全部。扩展性能强，有利于本应用的研发。

3） 算法部分

算法部分分为核心功能算法和其他功能算法。核心功能算法选用了基于Wi-Fi信号强度的位置指纹法，指纹法是以多个AP的信号强度作为基础，通过数据收集建立一个位置指纹的样本网格。用户方则使用KNN算法的思想，通过用户当前所在位置的多个AP信号强度与样本网格的对比，来获得当前的定位和周边设施的推荐信息。此室内定位算法在市场中有较广泛的应用，因此具有相当的可行性和可实现性。

而其他功能算法包括导航中的路径搜索算法，周边推荐中的推荐算法。路径搜索算法选用Floyd算法完成，来找到符合用户目标的最短路径。

# 需求分析

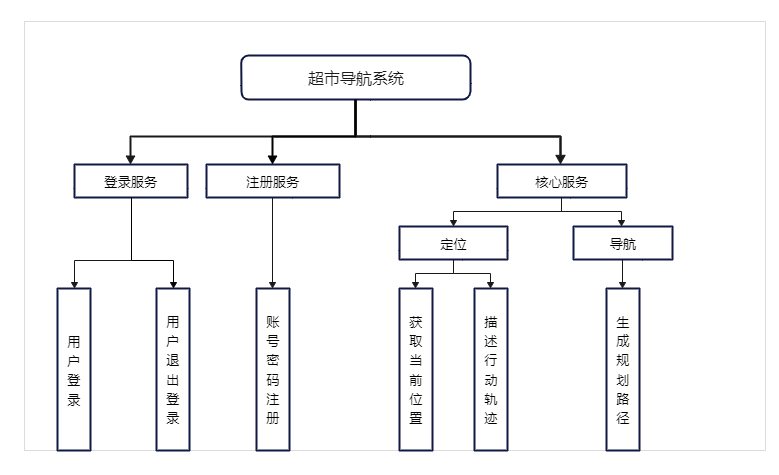
## 数据采集及描述

由于线上课程，小组人员无法见面，因此选择我们成员的家里模拟超市Wi-Fi信号，采集不同地方的Wi-Fi指纹作为超市中的不同场所，来进行分析导航。

具体的收集方案为：我们实现了一个用于收集数据的APP，此APP能够在地图上定位坐标并扫描此坐标周围的Wi-Fi信号强度。我们通过扫描定位点周围Wi-Fi信号强度信息形成json数组传送给服务器，存储在数据库中。在采集时，我们采用多次采集取平均值的方式来确定每个定位点的最终Wi-Fi信号强度信息。在采集完所有数据点后，服务器端将直接生成指纹样本网格。

## 功能性需求

### 功能模块结构图



### 核心功能模块描述

1) 登录注册：用户通过用户名，密码登录和注册，并可以通过邮箱的验证码进行验证。

2) 定位：通过本软件能够定位出使用者当前位置，如果使用者进行移动，本软件还会生成相关轨迹

3） 导航：通过选择目的地，软件获取当前位置，进行路径规划，为用户寻找最佳路径。

## 非功能性需求（精度、性能、易用性、可扩展性、可维护性等）

1）响应时间

系统响应时间主要分两个类别进行考虑：日常交互类和日常查询类。日常交互类主要包括用户个人信息的修改等，这类业务具有较高的响应需求。而日常查询类则包括超市专栏查询，设施评价，实时定位等，此类业务受到查询的复杂程度，数据量大小等因素的影响，需要根据具体情况而定。

本项目为Wi-Fi应用项目，需要用户在使用过程中全程连接Wi-Fi，因此此APP延时时长与当前网络强度、云服务器与传输数据量有关。在网络强度基本符合要求的情况下，在在线用户量较小时能够达到对用户任意需求的实时响应。当在线数量较多时，查询详细周边设施详细信息和附近的人功能受到影响较大，而其他功能基本无影响。

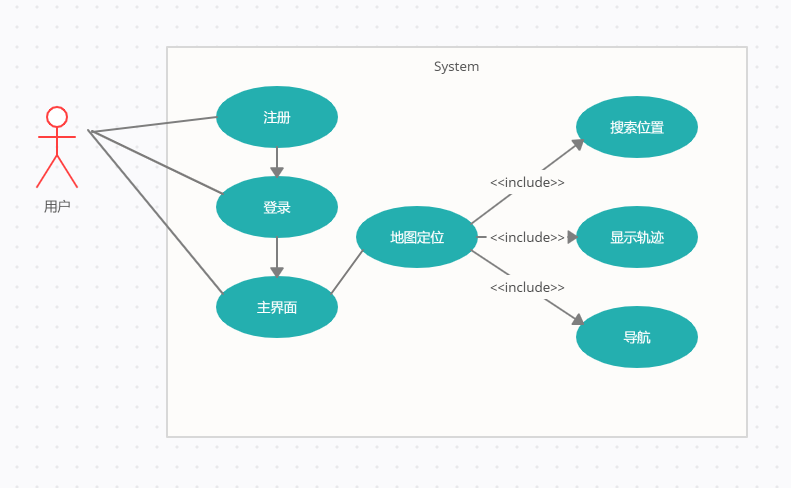
2） 用户数

本项目将考虑总用户数、峰值在线用户数、峰值并发用户数、平均在线用户数等。根据项目的定位，此项目的总用户数与超市容量与大小相关，而峰值在线用户数与人们的时间安排有关。比如，周内超市人数比较少，周末超市人数或增多，使用量加大。

3） 可扩展性

本项目在提高可扩展性方面做了许多工作，做了一些设想，比如用户靠近超市中的标点，会弹出该标点的信息，为用户查看。也有在标点处建立聊天室，当用户来到标点的时候便可以加入改聊天室进行讨论。

## 用例图



# 概要设计

## 系统架构设计

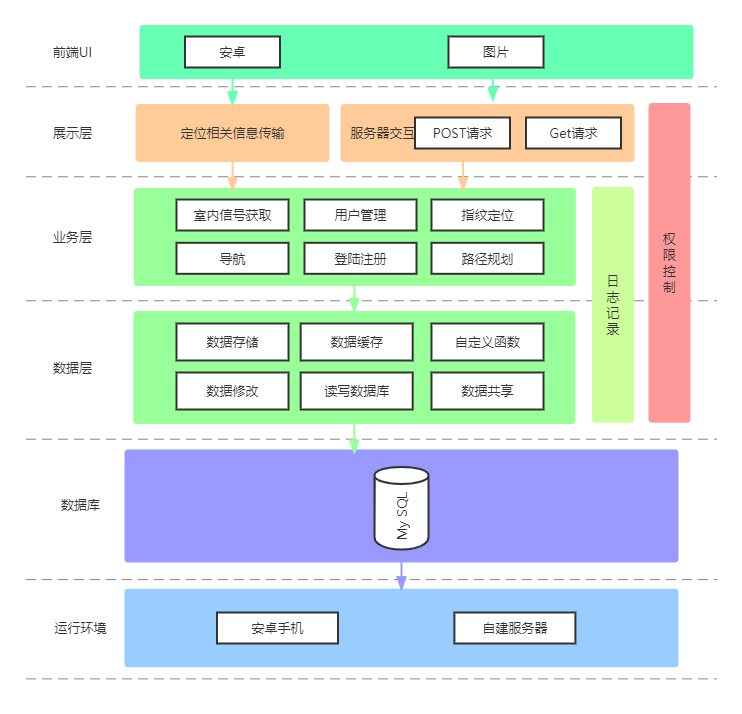
本项目使用三层架构将系统分为三层，展示层，业务逻辑层，以及数据层，并添加前端UI，数据库和运行环境部分增加图表可读性与丰富性。前端部分主要使用安卓进行开发，并使用自定义图片来改善用户体验。

展示层和UI构成架构的最上层，主要包括实现用户界面功能，将用户的需求传达和反馈，并保证用户体验，以及传输定位相关信息与服务器的两种请求交互。

业务逻辑层是对具体问题进行逻辑判断与执行操作，位于表示层与数据层中间位置，同时也是表示层与数据层的桥梁。用于实现三层间数据连接和指令传达，通过接收数据进行逻辑处理，并将处理结果反馈到表示层 UI 中，来实现软件功能。

数据访问层是数据库的主要操作系统，实现数据的增删查改等操作，并将结果返回给业务逻辑层。本项目使用MYSQL数据库，服务器选择在自建服务器部署，客户端允许用户在安卓手机中运行。

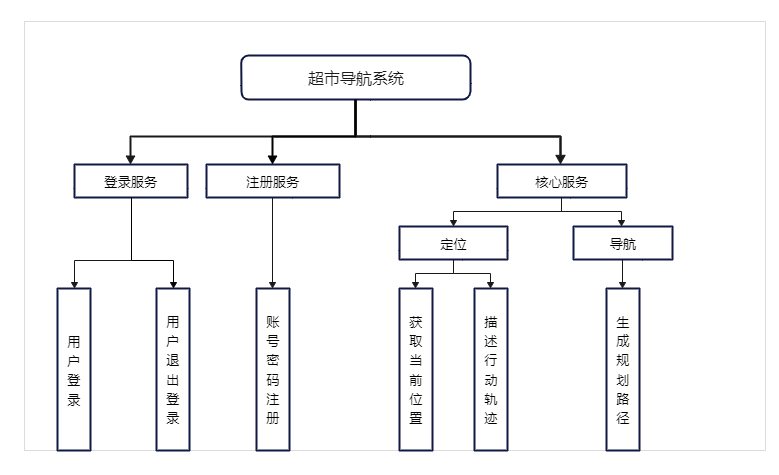
以下为本项目的系统架构图：



## 系统总体功能结构设计

本项目与定位相关的核心业务共包含三个部分：位置界面，个人界面，定位导航界面。室内位置定位主要包括当前WIFI信号强度的获取，定位相关辅助传感器信号获取，以及定位相关算法。目标搜索导航主要包括对导航目标的模糊搜索，路径查找算法，和当前移动的位置记录。

以下为本项目主要功能模块图：



## 关键技术设计

### 前端部分

基于Android的室内定位系统是基于移动端的应用，下面将按照功能模块的划分对使用流程进行说明，具体分为一下模块：数据采集模块、定位模块、轨迹跟踪、查找最短路径模块。

（1）数据采集模块

在使用本模块时首先进入主页面，之后点击信息采集按钮，在信息采集页面填写采集当前采集信号的位置信息，最后点击采集按钮完成WiFi信息采集。在该界面中我们将看到每个参考AP点的信号强度，方便用户感知WiFi信号强度的变化。

（2）定位模块

在使用本模块时首先进入主页面，点击定位按钮，在定位页面中，我们将看到自定义地图，之后点击定位按钮，当前的位置信息将显示在界面上。

（3）轨迹跟踪

本模块和定位模块一起集成在定位页面中。

（4）查找最短路径模块

在使用模块中首先进入主页面，点击路径规划按钮，在路径规划页面（图5.4）中，我们首先点击定位按钮，获得当前位置，之后输入想要到达的位置，点击导航按钮，完成查找路径功能。

### 后端部分

#### 数据库选择方案-MYSQL

本项目使用RDS云数据库存储用户信息与场景交互信息。RDS云数据库数据由本地已有数据导入云端。并使用DMS管理系统进行MySQL云数据库实例的管理。

MySQL是⼀种关系型数据库管理系统，其将数据保存在不同的表中，增加了速度并提高了灵活性。它所使用的SQL语言是用于访问数据库的最常用标准化语言。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这⼀特点，⼀般中下型项目的开发都选择 MySQL 作为数据库。

#### 服务器选择方案-java+Spring+MySQL

服务器部分选用Java+Spring+MySQL的方案。Spring具有低侵入设计，代码污染极低，独立于各种应用服务器，基于Spring架构的应用可以实现Write Once，Run Anywhere的承诺，开发者可以自由选择Spring框架的部分或者全部。扩展性能强，有利于本应用的研发。

### 算法部分

#### 数据采集方法设计

我们通过调用WIFI Manager来完成Wi-Fi信号强度的扫描与收集，并通过实现一个单独的数据采集APP来完成数据采集。此APP将在后续加入开发端方便地图更新。

而在数据信息方面，程序首先获取Wi-Fi强度列表，通过排序获取强度最大的前10个AP并记录，在同一数据点通过3-5次扫描记录，得到最终的AP列表并计算均值作为次数据点的“指纹”

#### 定位基础算法设计

此项目选用位置指纹法作为定位的基础算法，位置指纹实际上是把环境中的位置和“指纹”联系起来，每一个位置对应一个独特的指纹，而这个指纹包括了位置的信号强度的多维数据。我们通过收集样本指纹点来构建一个样本网格，用户通过当前的信号强度信息与样本网格中的信号强度进行对比，来获得相关信息并估计用户的位置。

基础算法主要包括坐标调整，相似度计算和定位估计3个部分。

1) 坐标调整：收集用户当前位置的信号强度信息并取最强的5个AP构成一个5维数据。在进行相似度计算前，首先进行坐标匹配工作，将此5个AP的BSSID在样本记录的AP中进行查找，若存在则构成样本的5维数据，否则填充极值100防止其对结果产生影响。

2) 相似度计算：本项目使用余弦相似度进行计算，并通过欧式距离进行再次计算，再通过实验比较两者准确性来决定使用的算法。目前我们使用余弦相似度来计算用户和样本网格的5维数据之间的相似性，并生成排序列表用于坐标估计。

3) 定位估计：我们使用KNN回归的思想，首先获取相似性排序列表中前k个数据，然后根据权值求得位置的平均值作为用户的位置估计。此外，我们尝试编写了KNN中k值的求取代码，方便进行地图更新等。

#### 定位改进算法设计

目前使用的位置指纹法由于数据采集密度较稀疏，样本点较少，且分布不够均匀，使用基础算法进行定位时会出现精度不够的情况。此外，Android规定了2分钟内的Wi-Fi扫描次数为4次，而室内定位需要显示用户的实时位置，这会导致定位点闪现的问题。而在Wi-Fi信号浮动时，扫描得出的结果也不完全相同，这可能产生定位坐标来回跳动的问题。

## 数据库设计

使用MySQL数据库进行信息存储与交互。

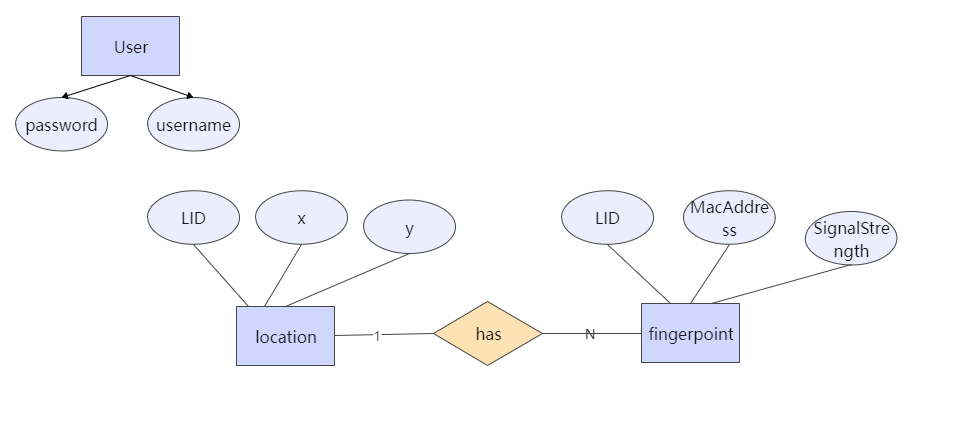
本项目采用一个数据库，数据库使用较为简单

下面来详细介绍表设计



表总共有三个，分别是user，location和fingerpoint，user中存储注册用户的用户名和密码，location中存储Wi-Fi指纹id和所在坐标，figerpoint中存储Wi-Fi指纹的id和mac地址和信号强度。

此项目的E-R图如下：



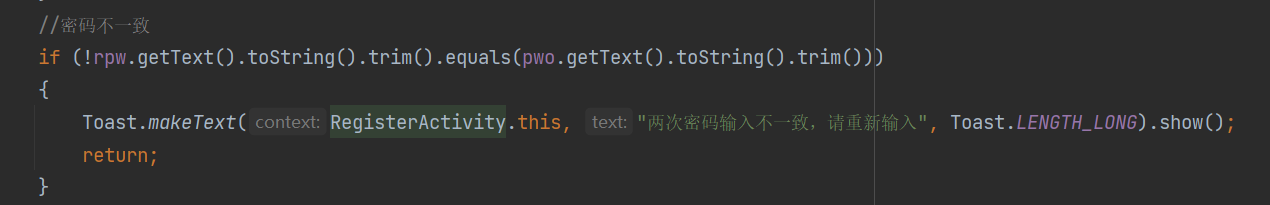
## 错误/异常处理设计

### 个人信息部分

如果重复注册会显示用户已注册



注册时两次密码不一致会报错

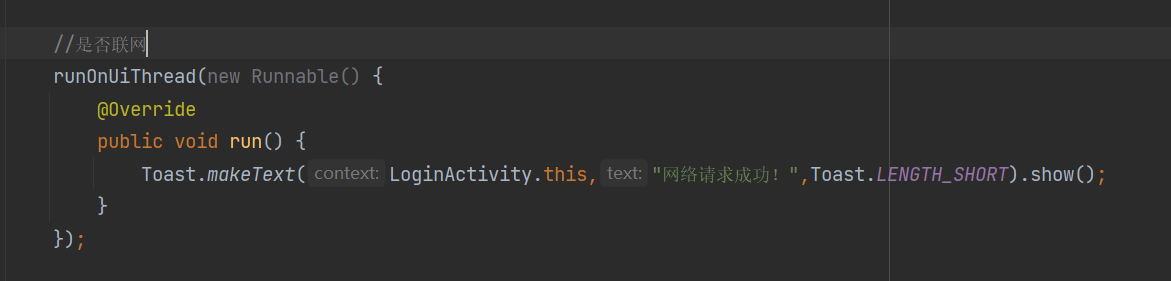


密码错误会显示登录失败



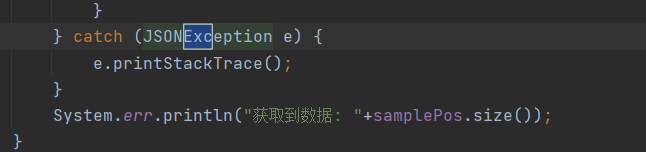
### 接口实现部分

如果前端连接到服务器，点击按钮时会显示“网络请求成功”



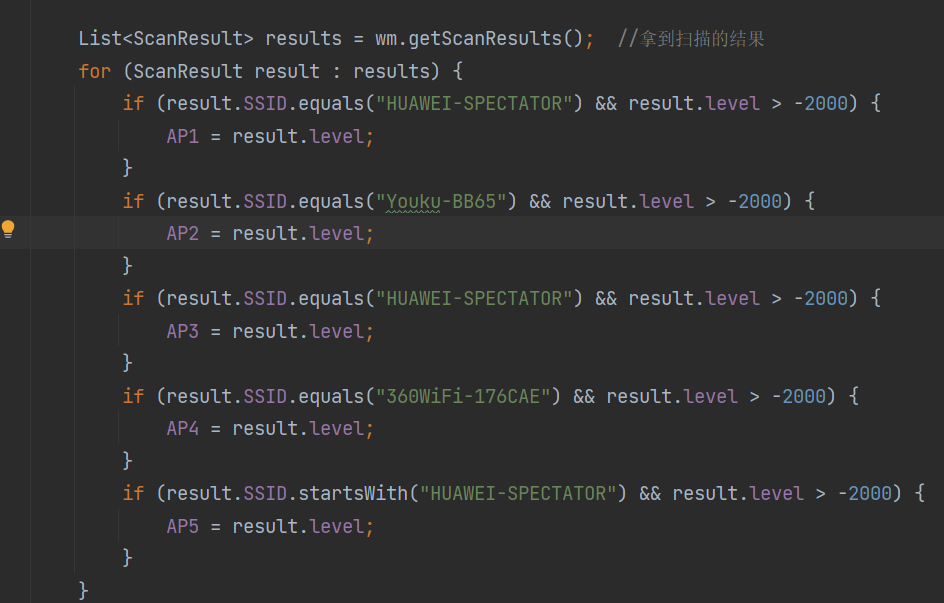
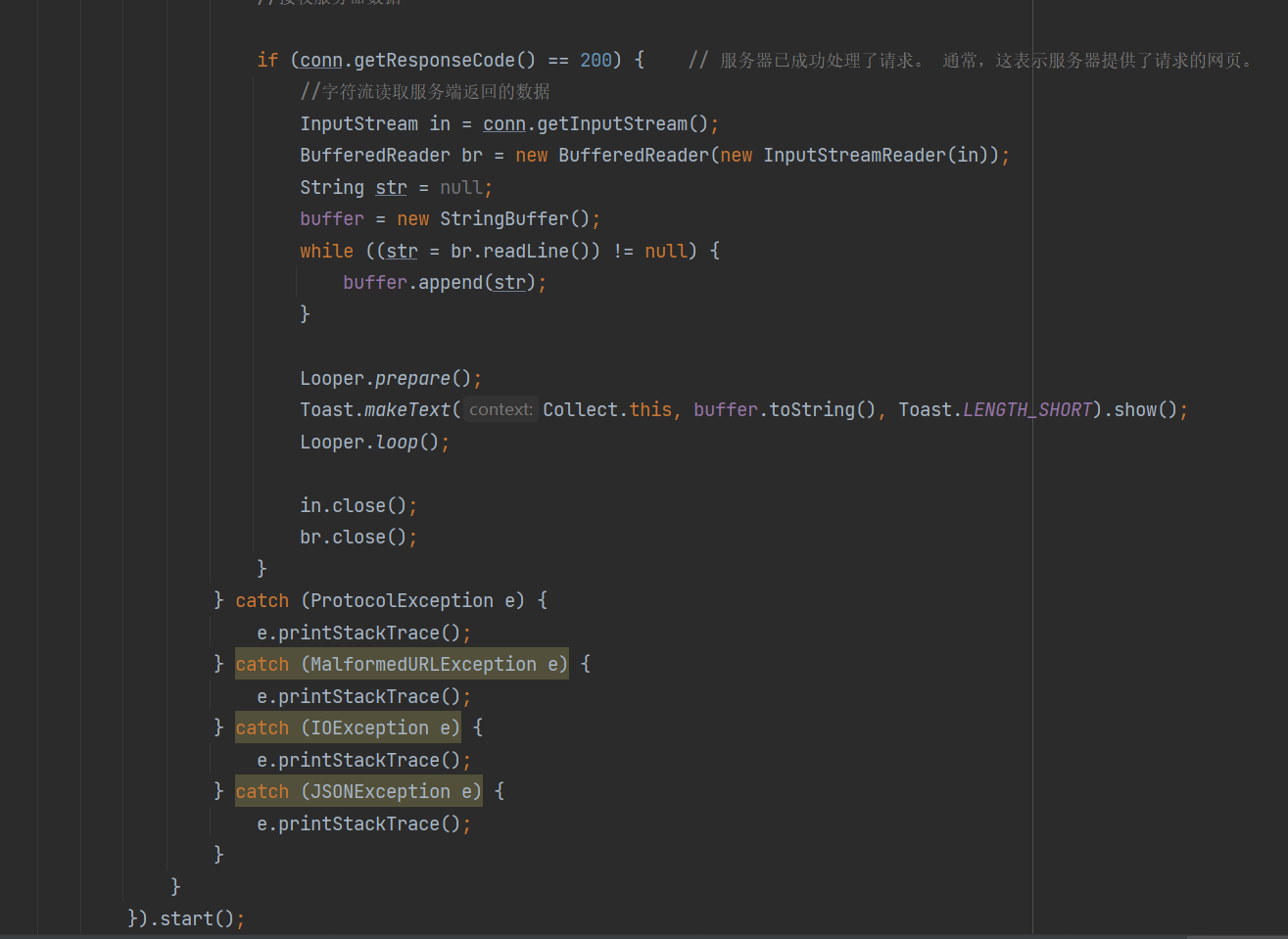
### 前后端数据传输部分

这里主要是所传输的数据是否传输成功，传输成功会提示获取到数据，传输失败会提示获取数据异常，如果是IO的异常情况，会提示关闭流异常。



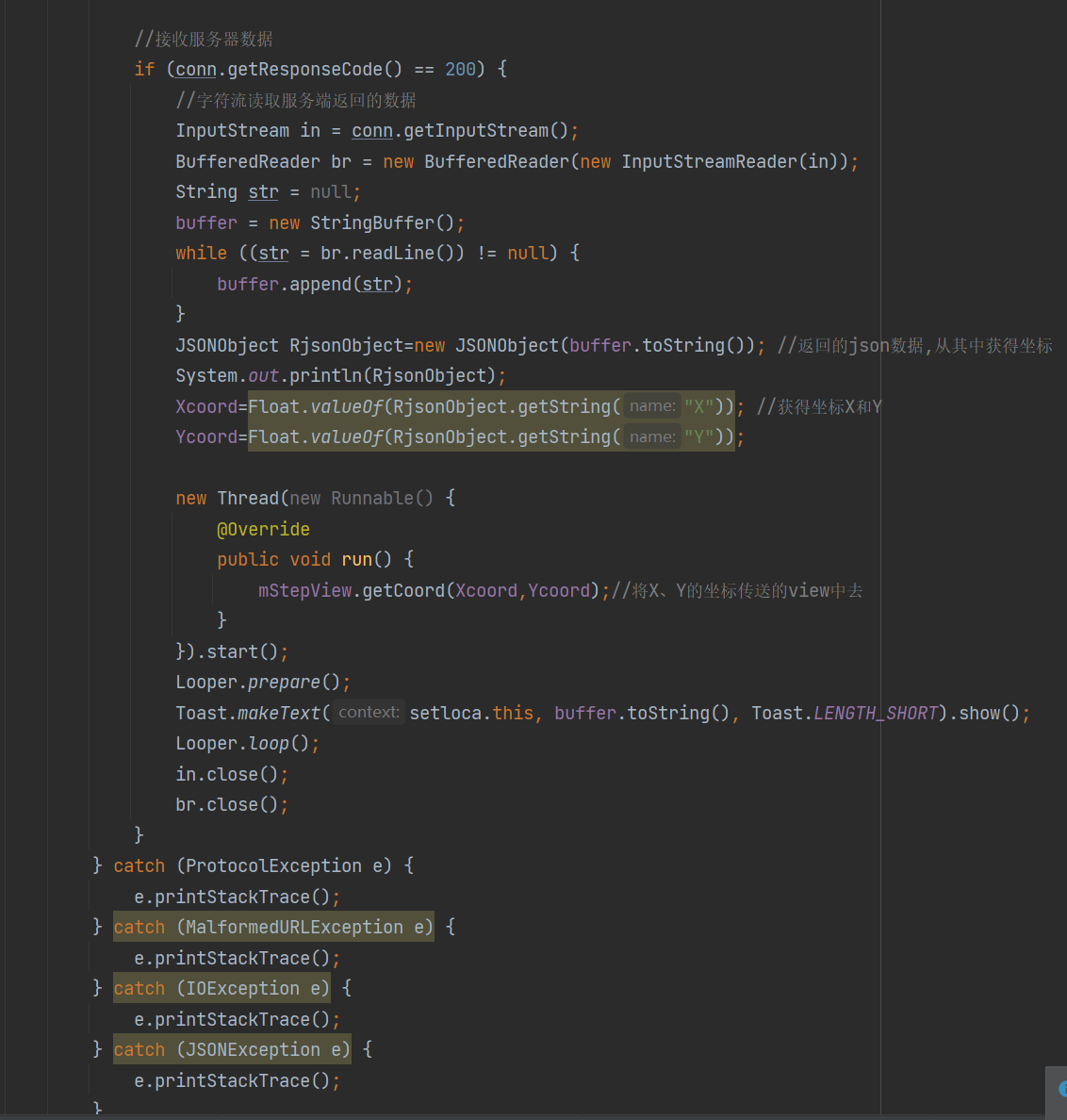
### Wi-Fi信号扫描部分

只有扫描到WIFI才可定位，否则报异常



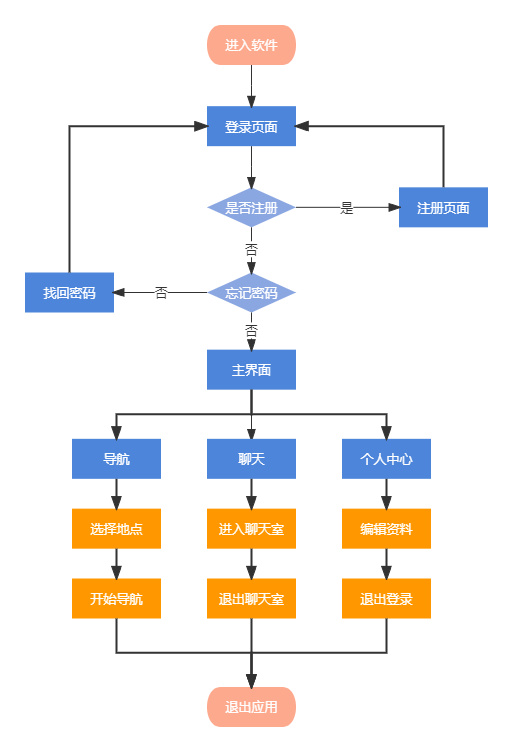
### 室内定位部分

获取到服务器返回的数据才可定位，否则报异常

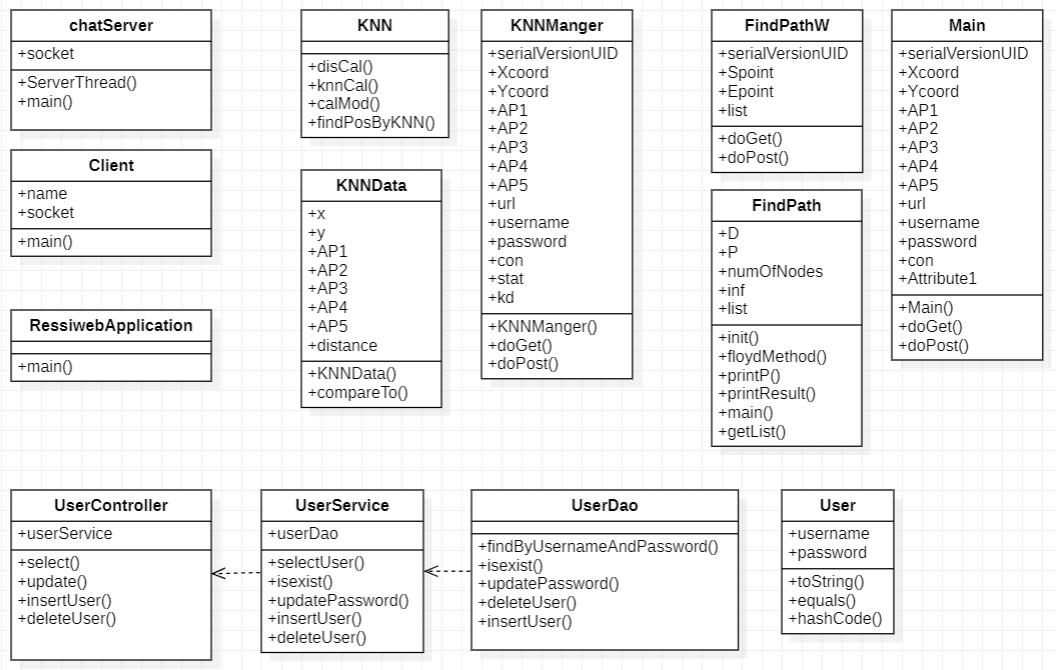


# 详细设计与实现

## 总流程图

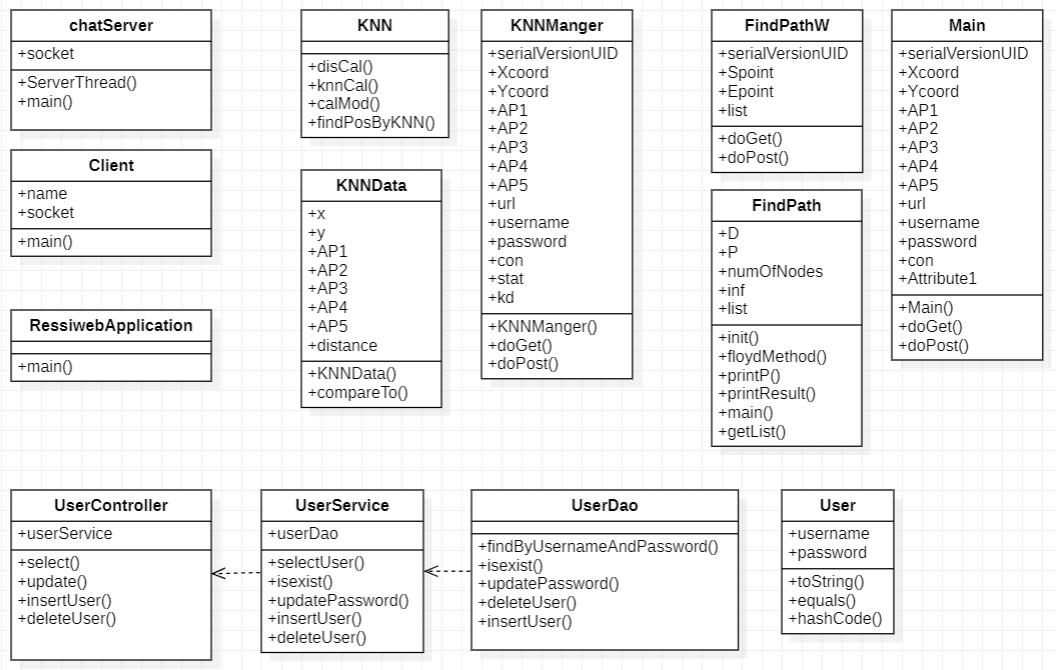


## 总类图

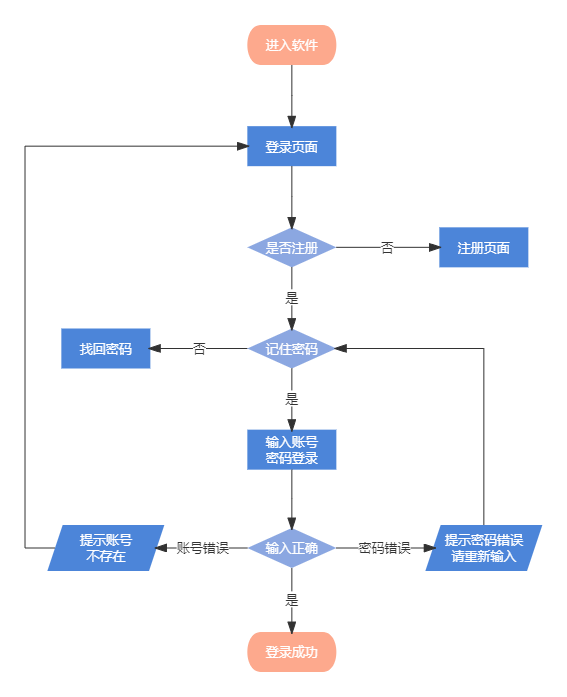


## 用户登录注册子系统的设计与实现

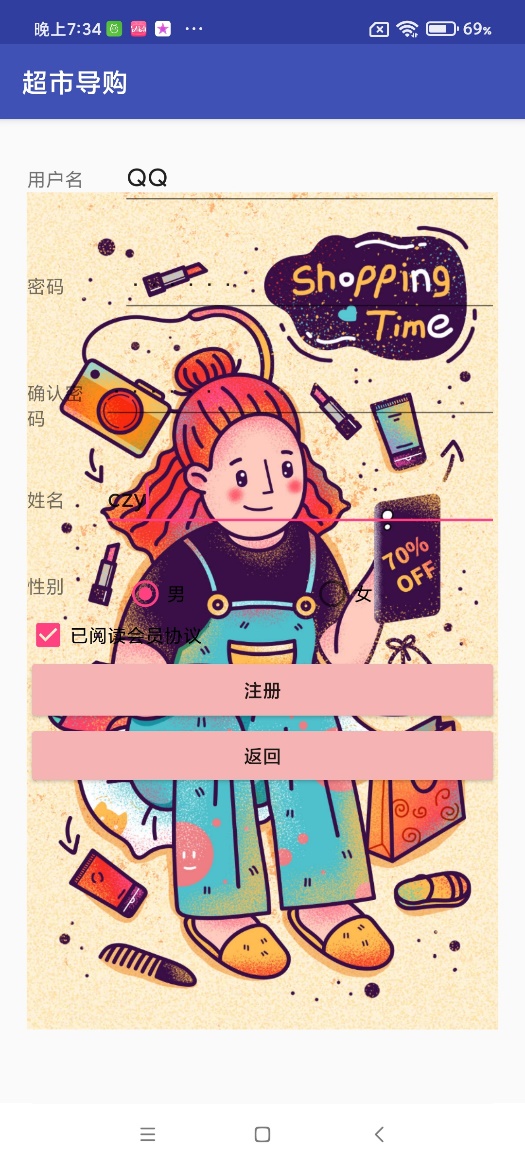
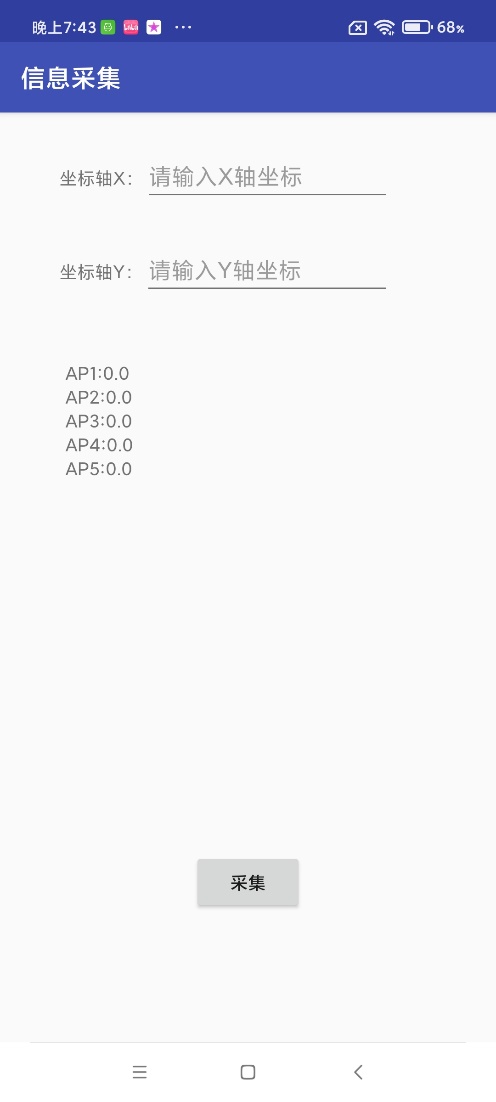
### 类图



### 流程图：

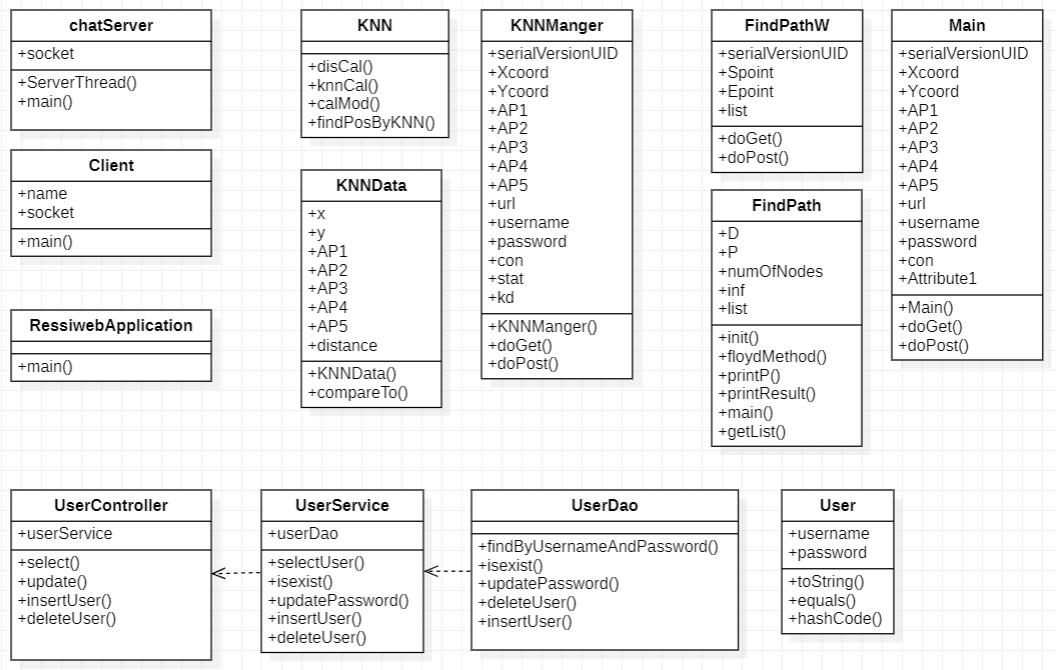


### 用户界面展示

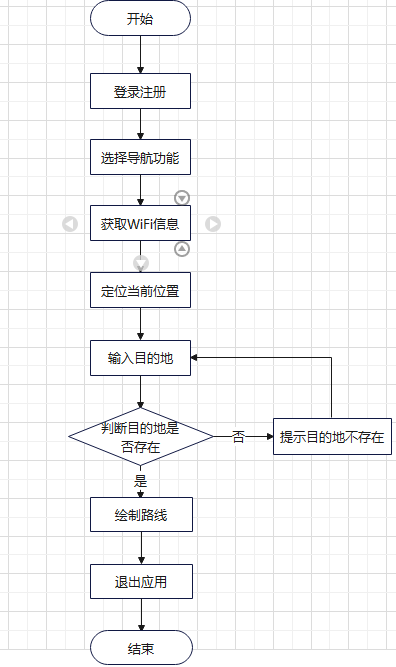
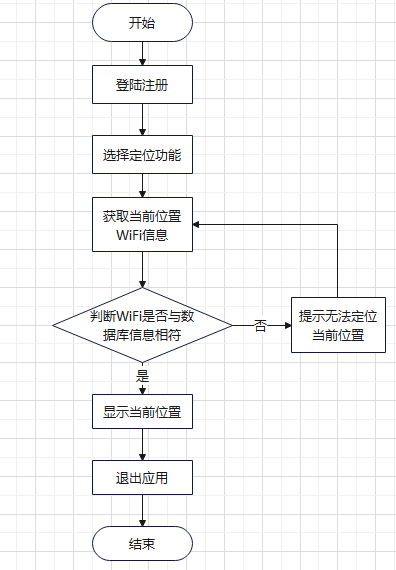


## 定位与导航子系统设计与实现

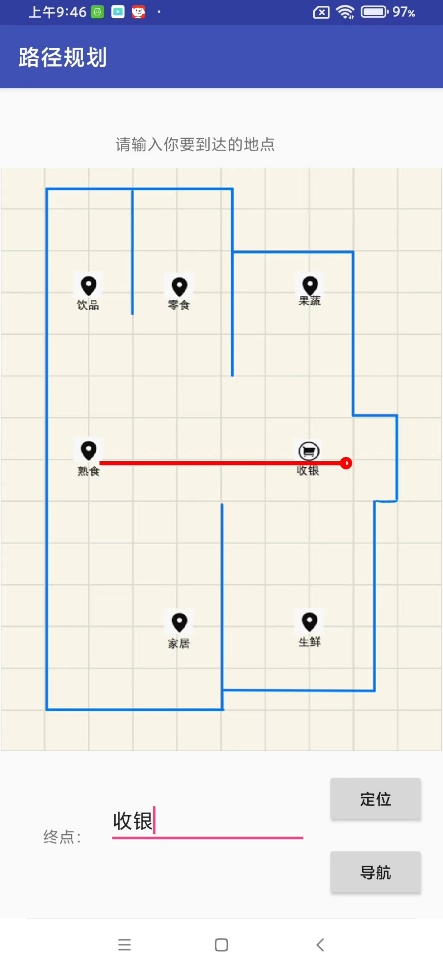
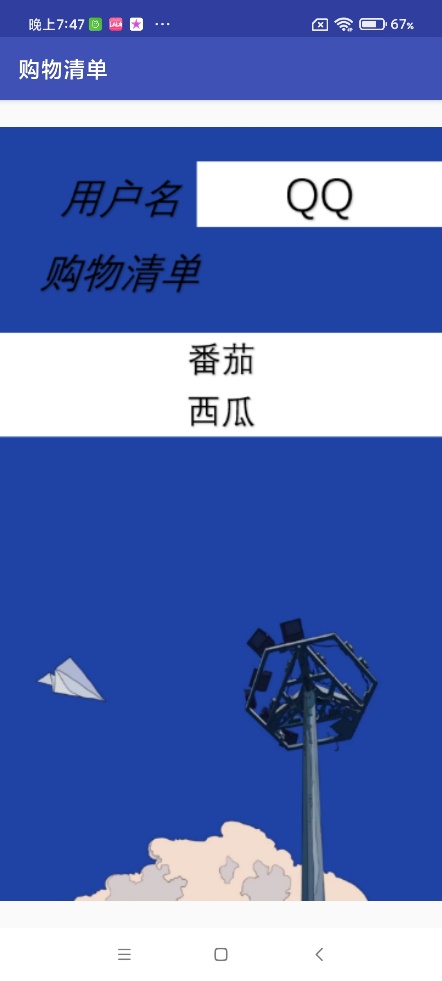
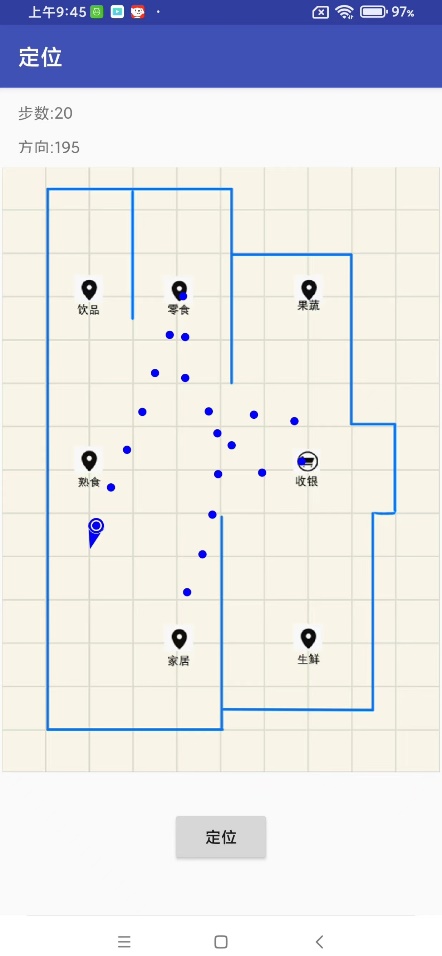
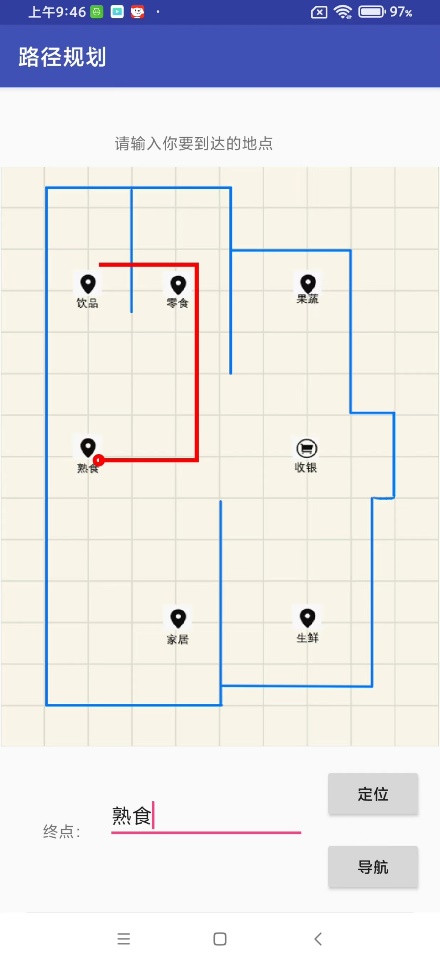
### 类图



### 流程图



### 用户界面展示



## 核心算法（伪代码）

### KNN算法

基于我们的商超导航程序，我们给定采集好的wifi指纹数据集，对新的wifi指纹输入实例，在数据集中找到与该实例最邻近的K个实例，这K个实例的多数属于某个类，就把该输入实例分类到这个类中。

算法流程：

1. 计算待测实例到数据集中每个实例的距离

2. 将所有实例按照与待测实例的距离的增序排列

3. 选取与A最近的k个数据集实例，即k个最近的邻居

4. 统计这k个邻居的类别频率

5. 找到k个邻居中频率最高的类别，作为测试实例

的类别。





### Floyd算法（导航）

通过一个图的权值矩阵求出它的任意两点间的最短路径。 从图的带权邻接矩阵A=[a(i,j)] n×n开始，迭代地进行n次更新，即由矩阵D(0)=A，按公式map[i,j]:=min{map[i,k]+map[k,j],map[i,j]}，构造出矩阵D(1)；又用同样地公式由D(1)构造出D(2)；最后又用同样的公式由D(n-1)构造出矩阵D(n)。矩阵D(n)的i行j列元素便是i号顶点到j号顶点的最短路径长度，称D(n)为图的距离矩阵，同时引入一个后继节点矩阵path来记录两点间的最短路径。

算法流程：把图用邻接矩阵G表示出来，如果从Vi到Vj有路可达，则G[i][j]=d，d表示该路的长度；否则G[i][j]=无穷大。定义一个矩阵D用来记录所插入点的信息，D[i][j]表示从Vi到Vj需要经过的点，初始化D[i][j]=j。把各个顶点插入图中，比较插点后的距离与原来的距离，G[i][j] = min( G[i][j], G[i][k]+G[k][j] )，如果G[i][j]的值变小，则D[i][j]=k。在G中包含有两点之间最短道路的信息，而在D中则包含了最短通路径的信息。

