

详细设计说明书

项目名称： 路况可视化 APP

作 者： 陈晓东 栾向阳 霍丽荣 李东宇

指导老师： 刘业

完成日期： 2015 年 11 月 30 日

1 引言	4
1.1 编写目的	4
1.2 背景	4
1.3 定义	4
1.4 参考资料	4
2 程序系统的结构	5
3 客户端设计	6
3.1 程序描述	6
3.2 功能	6
3.3 性能	7
3.4 输入项	7
3.5 输出项	8
3.6 流程逻辑	9
3.7 接口	10
3.8 存储分配	10
3.9 注释设计	10
3.10 限制条件	10
3.11 测试计划	10
3.12 尚未解决的问题	11
4 道路颜色标识模块	11
4.1 程序描述	11
4.2 功能	11
4.3 性能	11
4.4 输入项	11
4.5 输出项	11
4.6 算法	12
4.7 流程逻辑	13
4.8 接口	14
4.9 存储分配	14
4.10 注释设计	14
4.11 限制条件	14
4.12 测试计划	14
4.13 尚未解决的问题	14
5 服务器端设计	14
5.1 程序描述	14
5.2 功能	15
5.3 性能	16
5.4 输入项	17
5.5 输出项	20
5.6 流程逻辑	23

5.7 算法.....	25
5.7.1 基准时间计算.....	25
5.7.2 实时拥挤程度的计算.....	26
5.8 接口.....	27
5.9 存储分配.....	27
5.10 注释设计.....	27
5.11 限制条件.....	27
5.12 测试计划.....	28
5.13 尚未解决的问题.....	28
6 通信模块.....	29
6.1 程序描述.....	29
6.2 功能.....	29
6.3 性能.....	29
6.4 输入项	29
6.5 输出项	29
6.6 算法.....	29
6.7 流程逻辑.....	29
6.8 接口.....	30
6.9 存储分配.....	30
6.10 注释设计.....	30
6.11 限制条件.....	30
6.12 测试计划.....	30
6.13 尚未解决的问题.....	30

详细设计说明书

1 引言

1.1 编写目的

编写详细设计说明书是软件开发过程必不可少的部分，就是对概要设计的一个细化，就是详细设计每个模块实现算法，所需的局部结构。其目的是为了使开发人员在完成概要设计说明书的基础上完成概要设计规定的各项模块的具体实现的设计工作。以及确定怎样具体的实现系统及结构化程序设计技术及使用相应的详细设计工具，预期的读者为需要交付的用户，可以清楚的看到本系统每部分的设计及使用方法。

1.2 背景

说明：

1. 软件系统的名称：路况可视化 APP。
2. 此项目的任务提出者：中科大软件学院。
3. 开发者：陈晓东、栾向阳、霍丽荣、李东宇。
4. 用户：本系统是中科大软件学院的工程实践项目。期望可以被所有需要获取实时路况信息的相关人员使用。
5. 运行平台：Android 客户端，即安装了路况可视化 APP 的 Android 机器；后台服务器选择阿里云服务器，后台服务器需有数据库，客户端的数量没有限制。

1.3 定义

C/S 模式：C/S（Client/Server，客户机/服务器）模式又称 C/S 结构，是 20 世纪 80 年代末逐步成长起来的一种模式，是软件系统体系结构的一种。

MySQL：MySQL 是一个关系数据库管理系统。

1.4 参考资料

《软件工程导论（第四版）》 张海藩 清华大学出版社。
GB8567-88 计算机软件产品开发文件编制指南 中国标准出版社
GB9385-88 计算机软件需求说明编制指南 中国标准出版社
GB/T11457-1995 软件工程术语 中国标准出版社

2 程序系统的结构

本系统整体架构上基于 C/S 模式，系统分为前台客户端和后台服务器。前台客户端作为路况可视化终端，调用百度地图 API，将服务器数据分析出的路况状态信息显示在 UI 界面上，并提供周边路况上报功能、用户功能等相关功能。后台服务器则负责数据获取、数据分析及维护数据库，并根据客户端发来的请求，响应返回的数据结果。前后台采用 Socket 进行通信。

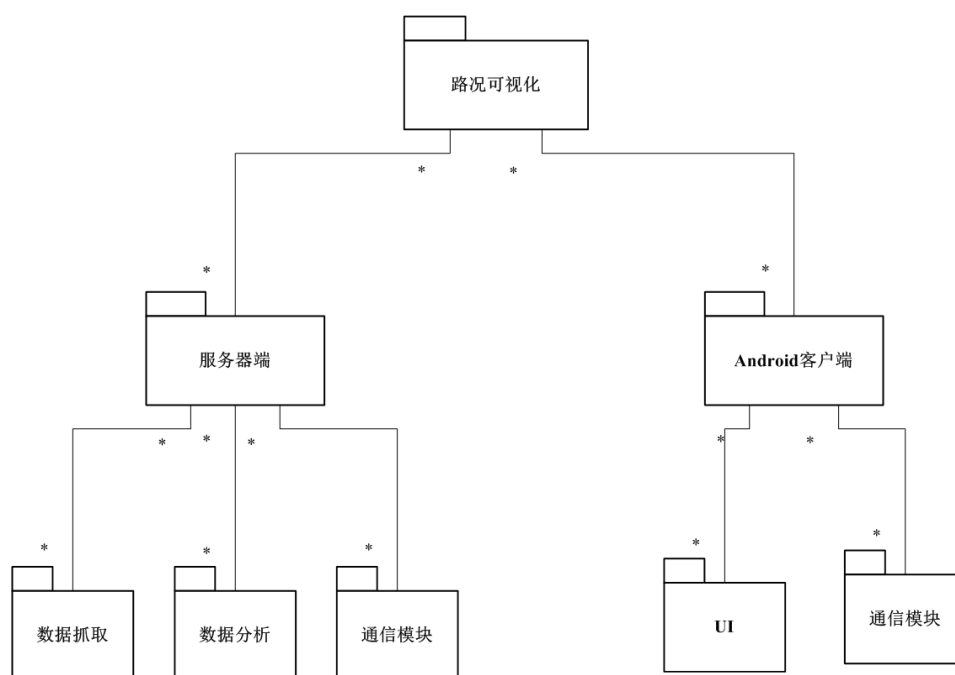


图 2.1 系统组件图

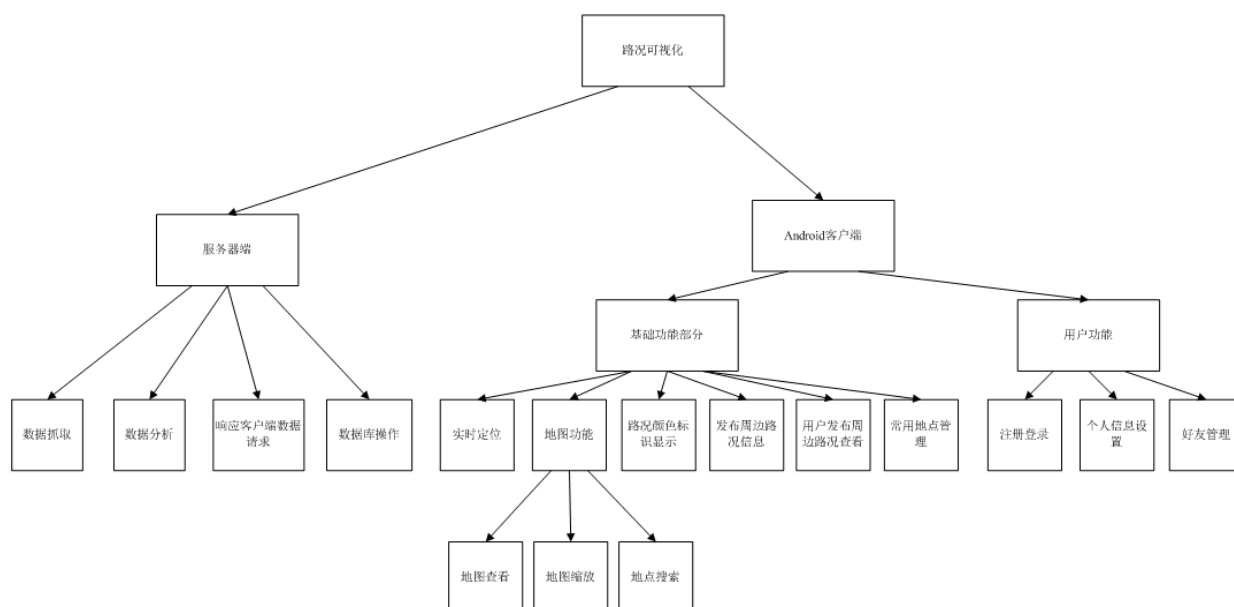


图 2.2 系统功能图

3 客户端设计

3.1 程序描述

该程序为路况可视化前台 Android 客户端。客户端使用百度地图 API 进行基础地图显示，用户从 UI 界面上进行操作，调用通信模块与服务器端进行数据交互，实现定位路况可视化功能、基础地图功能、用户管理功能等一系列功能。其中难度较大的是“道路颜色标识”表示路况的部分，将在下一章进行重点陈述。

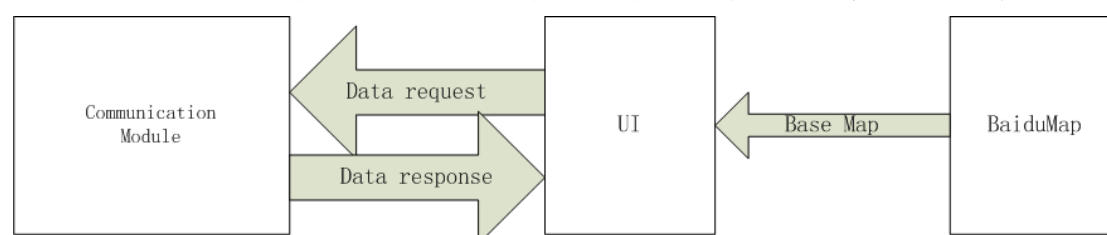


图 3.1 模块示意图

3.2 功能

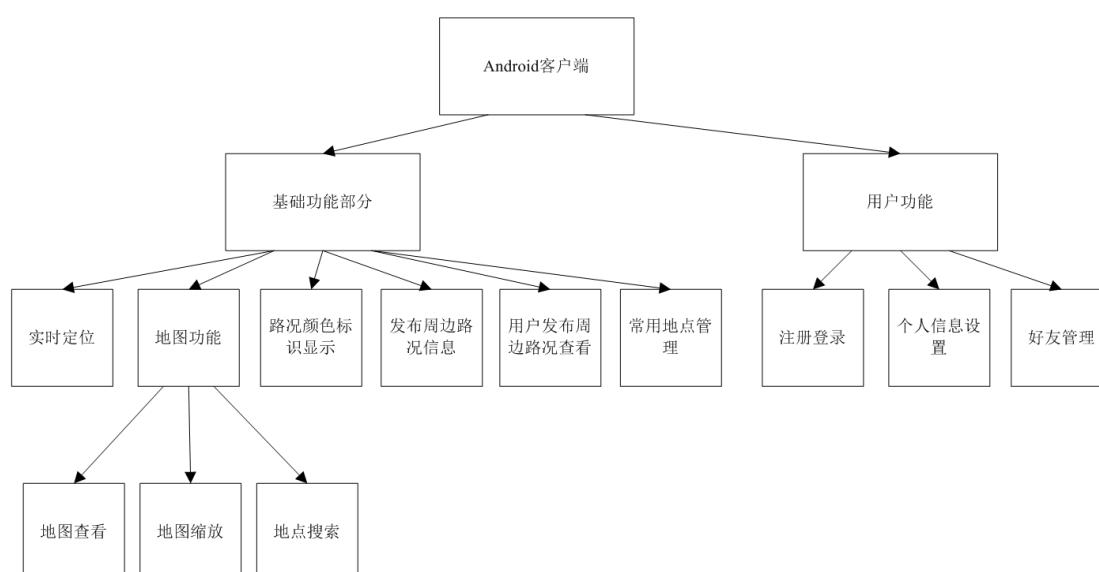


图 3.2 客户端功能图

1. 基础功能：

- 1) 实时定位：系统使用 GPS 进行自动实时定位；
- 2) 地图功能：可以进行查看地图、缩放地图、特定地点搜索等操作；
- 3) 路况信息标识：显示屏幕显示的地图区域内的道路拥堵情况，以颜色作为标识，绿色代表畅通，黄色代表轻微拥堵，橙色代表一般拥堵，红色代表严重拥堵；
- 4) 发布周边路况信息：注册用户可以通过图文信息发布到讨论区模块的

形式，上报定位点周边一定范围内的实时路况信息；

5) 查看周边路况信息：注册用户可以查看以定位点为中心的一定范围，其他注册用户在 1 小时内发布的实时路况信息；

6) 常用地点管理：注册用户可选择存储“家”、“公司”或其他常用地点；

2. 用户功能：

1) 注册登录：用户可通过手机号注册的方式成为注册用户，注册成功后以手机号和密码进行登录；

2) 个人信息设置：注册用户可以对自己的个人信息，如用户名、头像等进行修改；

3) 好友管理：注册用户可以以搜索手机号的方式寻找好友，对好友信息进行备注。

3.3 性能

使用性：要求各个功能实现简单，操作方便，系统运行正确；整个系统恢复性能好，以确保软件可使用性好。

实时性：用户操作界面时有较快的响应。

3.4 输入项

1.注册

表 3.1 注册输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
手机号	PhoneNum	char[]	字符	有效手机号	手工输入	键盘	用户
用户名	UserName	char[]	字符	任意 6-20 位字符	手工输入	键盘	用户
密码	Password	char[]	字符	任意 6-20 位字符	手工输入	键盘	用户
确认密码	PasswordCheck	Char[]	字符	任意 6-20 位字符	手工输入	键盘	用户

2.登录

表 3.2 登录输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
手机号	PhoneNum	char[]	字符	有效手机号	手工输入	键盘	用户
用户名	UserName	char[]	字符	任意 6-20 位字符	手工输入	键盘	用户
自动登录	AutoLogin	bool	单选框	勾选/不勾选	手工输入	点击	用户

3.发布路况信息

表 3.3 发布路况信息输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
路况类型	type	char[]	单选框	6 个选项	手工输入	点击	用户
路况等级	level	char[]	单选框	2-3 个选项	手工输入	点击	用户
详情描述	detail	char[]	字符	任意	手工输入	键盘	用户

3.5 输出项

前台 Android 客户端将数据协议包交给通信模块，由通信模块与服务器进行 Socket 连接后，发送给服务器端，进行相应的处理，并返回数据。

1.注册 register

phoneNum //手机号
 userName //用户名
 userPass //密码

2.登录 login

phoneNum //手机号
 userPass //密码

3. 发布路况信息

latitude //经度
 longitude //纬度
 posName //位置名称
 dateTime //时间
 type //路况类型
 level //路况等级
 detail //详情描述
 phoneNum //发布者手机号

3.6 流程逻辑

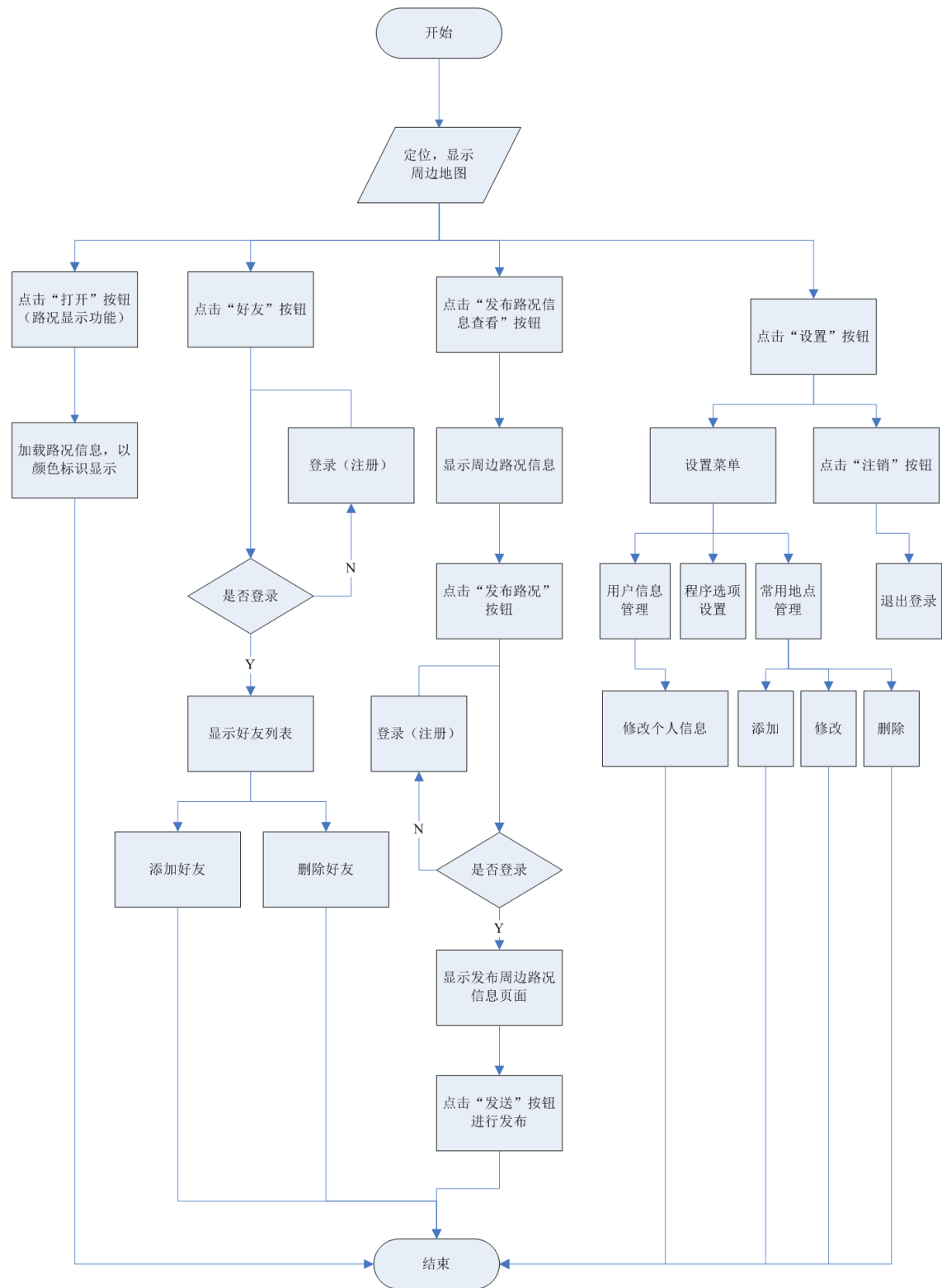


图 3.3 客户端流程图

3.7 接口

本程序所隶属的上一层模块：本程序为第一层模块

隶属于本程序的下一层模块、子程序：通信模块、百度地图 API 提供的接口模块

参数赋值和调用方式：直接调用

与本程序相直接关联的数据结构：顺序结构

3.8 存储分配

标识符	类型
phoneNum	varchar [20]
userName	varchar [40]
userPass	varchar [20]
latitude	double
longitude	double
posName	varchar[20]
dateTime	datetime
type	varchar[20]
level	varchar[20]
detail	varchar[200]

3.9 注释设计

1. 加在模块首部的注释：/*模块功能说明*/
2. 对各变量的功能、范围、缺省条件等所加的注释：//变量内容、需要说明的部分

3.10 限制条件

点击“打开”按钮获取路况状态信息，以颜色标识在地图上的过程完成速度受网速的影响。

3.11 测试计划

输入数据：多用户模拟正常使用路况可视化 APP

预期结果：各个功能都能正常处理以及响应

进度安排：已经完成

人员职责：李东宇负责

3.12 尚未解决的问题

路况状态以颜色标识的技术实现使用了百度地图 API 对于路径的规划接口，该接口每次调用只能产生一个实例，也就是说每次调用只能在图中画出一条路径状态表示线，会造成加载过程中的时间消耗，用户体验较为不好，但目前没有有效的解决方法。

4 道路颜色标识模块

4.1 程序描述

该模块用于将从服务器端取得到路况信息，分析处理出每个路段，调用百度地图 API 进行路径搜索，将返回的路径结果以颜色标识显示在基础地图上。

4.2 功能

在地图上的相应路段上，为畅通、轻微拥堵、中度拥堵、严重拥堵的路况情况画上绿、黄、橙、红色的颜色标识线。

4.3 性能

实时性：所有路段线的加载时间应在用户接受的合理范围之内。

4.4 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
用户位置	currentPt	LatLng	数组	经纬度点对	系统输入	接口	百度地图

4.5 输出项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
经度	latitude	double	数字	0-180W 0-180E	系统输入	接口	百度地图
纬度	longitude	double	数字	0-90N 0-90S	系统输入	接口	百度地图

4.6 算法

因为百度地图路径规划 API 的限制，一次只能触发一个路径搜索事件，而道路颜色标识作为路况可视化 APP 的主要功能，用户对其功能的要求程度和加载效率有着较高的要求，需要尽可能的在不考虑网络条件的情况下减少加载的时间成本。因此算法选择使用多线程进行前期的准备和信息处理，选择互斥锁来保证每次只有一个路径搜索事件可以获得在地图上进行颜色标识路径的权利，即在获得已知的起终点对应的路径之后，尝试获得锁，如果成功获得，则在地图上添加相应颜色的线段，如果获取失败，则等待目前获得锁的其他路径搜索事件完成处理，释放锁之后，再次尝试获得锁，直到获得锁并完成地图颜色线段更新为止。具体的算法流程在 4.7 的流程逻辑图中进行了详细的步骤说明。

4.7 流程逻辑

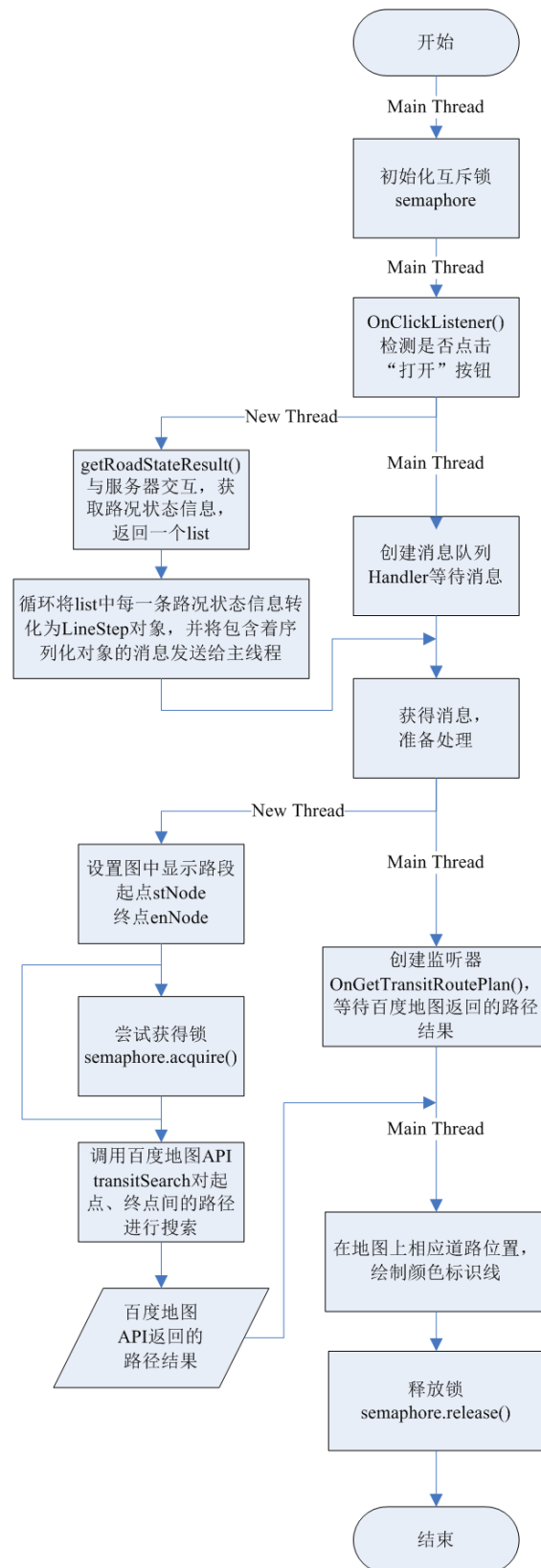


图 4.1 道路颜色标识算法流程图

4.8 接口

本程序所隶属的上一层模块：客户端 UI 界面模块

隶属于本程序的下一层模块、子程序：通信模块、百度地图路径规划 API

参数赋值和调用方式：直接调用

与本程序相直接关联的数据结构：顺序结构

4.9 存储分配

无

4.10 注释设计

1. 加在模块首部的注释：/*模块功能说明*/
2. 对各变量的功能、范围、缺省条件等所加的注释：//变量内容、需要说明的部分

4.11 限制条件

以颜色标识在地图上的过程完成速度受网速的影响。

4.12 测试计划

输入数据：多用户模拟正常使用路况可视化 APP

预期结果：各个功能都能正常处理以及响应

进度安排：已经完成

人员职责：李东宇负责

4.13 尚未解决的问题

路况状态以颜色标识的技术实现使用了百度地图 API 对于路径的规划接口，该接口每次调用只能产生一个实例，也就是说每次调用只能在图中画出一条路径状态表示线，会造成加载过程中的时间消耗，用户体验较为不好，但目前没有有效的解决方法。

5 服务器端设计

5.1 程序描述

该程序为路况可视化后台服务器端。服务器端分为四个子系统：公交信息抓

取系统、公交数据分析及路况计算系统、客户端服务系统、公交站点地理坐标获取系统。公交信息抓取模块用于从苏州市公交信息网站获取实时公交数据；公交数据分析及路况计算模块将抓取的原始数据进行处理，根据道路的公交通过信息计算实时路况；客户端服务模块负责响应客户端发起的各项请求，回复路况信息并维护用户相关的各项数据；公交站点地理坐标获取模块是服务器端的百度地图接口调用部分，负责完成公交站名的名称到地理坐标的转换工作。

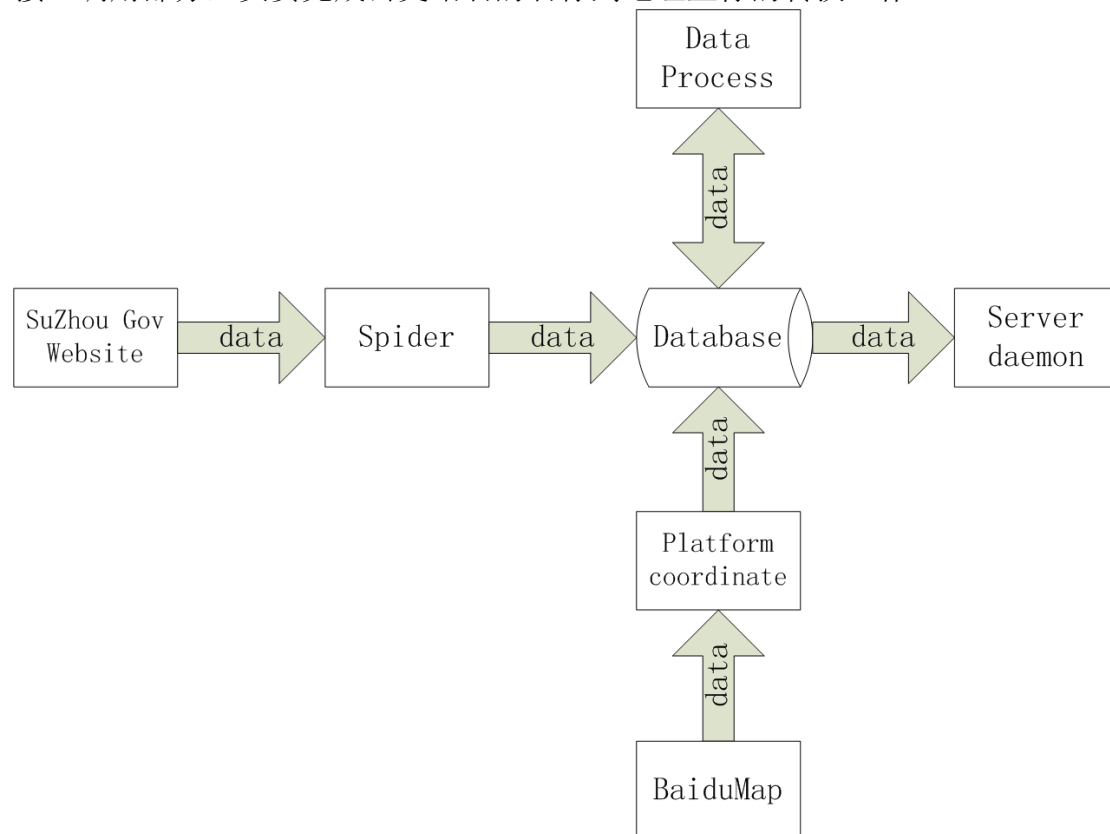


图 3.1 模块示意图

5.2 功能

1. 公交信息抓取系统

基础功能：

1) 公交线路获取：从配置文件获取待抓取的公交线路信息，所有抓取的公交线路覆盖的道路将是本软件实时路况的覆盖区域；

2) 公交停靠数据抓取：系统接受配置文件中定义的抓取间隔时间，周期性地获取各线路中的公交车到站时间信息；

3) 公交停靠数据预处理：系统为每条线路维护抓取到的到站时间表，每次抓取会进行重复信息检查，将有效数据存入数据库作为路况计算的原始数据。

2. 公交数据分析及路况计算系统

基础功能：

算法是基于历史数据价值的获取，从爬虫得到的历史数据中得到相邻两站

最短的运行时间的平均值，作为交通路网拥挤程度的评价参考。数据样本是苏州公交运行 10 天的历史数据。先获取每天出勤的所有公交车的车牌号，对每一辆车的运行情况逐一分析。

3. 客户端服务系统

基础功能：

1) 连接监听：服务开启后，监听 50001 端口，接收客户端发起的各项请求并完成处理响应；

2) 用户注册：新用户通过客户端 APP 发起注册请求，服务器接受请求完成注册；

3) 用户登录：用户在客户端输入账号密码信息，点击登录，客户端将登录信息发送给服务器，服务器收到登录请求后进行用户名及密码的校验，并将登录结果返回给客户端；

4) 用户信息查询：客户端发起用户信息查询请求，服务器收到请求后从数据库查询用户信息，并将查询结果返回给客户端；

5) 路况查询：客户端将用户屏幕显示的坐标范围发送给服务器，服务器查询指定范围内的公交路况信息，将数据封装到响应数据中返回给客户端；

6) 用户分享信息上传：服务器接受用户上传的指定位置的突发交通状况信息，此信息对于一定地理位置内的其他用户可见；

7) 用户分享信息获取：服务器接受客户端发送的用户分享信息查询请求，将用户所在地理位置范围内的所有其他用户分享信息提取并封装到响应数据中发送给客户端；

4. 公交站点地理坐标获取系统

1) 公交站点名称获取：系统从路况信息数据库获取所有公交站点名称；

2) 公交站点名称到地理坐标转换：系统调用百度地图的 API，完成公交站名到地理坐标的转换，为每个公交站点映射一个地理坐标；

3) 公交站点地理坐标数据库更新：系统维护一个公交站点地理信息数据库，对路况信息数据库中存在但地理坐标数据库中不存在的站点，完成站名到地理坐标的更新。

5.3 性能

并发性：

1) 公交信息抓取系统

采用并行抓取模式，待抓取线路及抓取结果使用 FIFO 消息队列进行存取，并行抓取线程数可在配置文件中指定；

2) 公交数据分析及路况计算系统

采用并行计算模式，将不同车辆的不同车站间的时间以多线程方式进行处理；

3) 客户端服务系统

使用 Python 的 threading 模块，为每一个来自客户端的请求创建一个线程处理，支持高并发，性能上限取决于服务器性能及 MySQL 数据库承载能力；

4) 公交站点地理坐标获取系统

此任务非时间敏感任务，无并发运行需求，采取串行查询方式。

实时性：

1) 公交信息抓取系统

本系统的数据来源于政府公交信息网站，由于此网站提供实时公交信息数据，故实时性仅取决于配置文件中定义的抓取时间间隔。即抓取的频率越高，实时性越好。

2) 公交数据分析及路况计算系统

对公交运行数据库中每一次数据的更新，都对应一次实时路况的计算及更新操作，故不存在数据的延迟问题。

3) 客户端服务系统

针对每一个请求创建一个线程进行处理，数据库为运行于服务器本地的 MySQL，查询及更新操作耗时相对于网络延迟可忽略不计，故实时性仅受限于客户端与服务器之间的网络连接状况。

4) 公交站点地理坐标获取系统

此任务非时间敏感任务，当系统添加了新的公交线路后运行一次即可；或使用 Linux 的 crontab 指定周期运行。

5.4 输入项

1. 公交信息抓取系统

表 5.1 公交信息抓取系统输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
数据库用户名	db_username	char[]	字符	有效用户名	配置文件	键盘	管理员
数据库密码	db_password	char[]	字符	有效密码	配置文件	键盘	管理员
数据库地址	db_address	char[]	字符	IP 地址或域名	配置文件	键盘	管理员
数据库名	db_name	char[]	字符	有效数据库名	配置文件	键盘	管理员
抓取线程数	work_thread_num	char[]	字符	非负整数	配置文件	键盘	管理员
抓取间隔	fetch_interval	char[]	字符	非负整数	配置文件	键盘	管理员
公交线路标识	[]	char[]	字符	线路首字母缩写	配置文件	键盘	管理员
公交线路名	name	char[]	字符	Unicode 字符	配置文件	键盘	管理员
cid	cid	char[]	字符	线路 cid	配置文件	键盘	管理员
guid	guid	char[]	字符	线路 guid	配置文件	键盘	管理员

2. 公交数据分析及路况计算系统

表 5.2 公交数据分析及路况计算系统输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
线路名称	line_name	varchar (50)	字符	Unicode 字符	数据库 获取	MySQL Connector	公交信息 抓取系统
站点名称	platform_name	varchar (20)	字符	Unicode 字符	数据库 获取	MySQL Connector	公交信息 抓取系统
站点 ID	platform_id	char(3)	字符	三位大写 字母标识	数据库 获取	MySQL Connector	公交信息 抓取系统
公交车车牌	bus_id	varchar (10)	字符	Unicode 字符	数据库 获取	MySQL Connector	公交信息 抓取系统
到站时间	arrive_time	datetime	日期时间	有效日期 时间	数据库 获取	MySQL Connector	公交信息 抓取系统

3. 客户端服务系统

表 5.3.1 客户端服务系统 login 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
手机号码	phoneNum	char[]	json 字符	有效手机号码	网络	Socket	客户端
密码	userPwd	char[]	json 字符	有效密码	网络	Socket	客户端

表 5.3.2 客户端服务系统 register 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
手机号码	phoneNum	char[]	json 字符	有效手机号码	网络	Socket	客户端
密码	userPwd	char[]	json 字符	有效密码	网络	Socket	客户端
用户名	userName	char[]	json 字符	Unicode 字符	网络	Socket	客户端

表 5.3.3 客户端服务系统 getUserInfo 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
----	----	----	----	------	------	------	--------

手机号码	phoneNum	char[]	json 字符	有效手机号码	网络	Socket	客户端
------	----------	--------	---------	--------	----	--------	-----

表 5.3.4 客户端服务系统 getRoadState 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
最小纬度	latitudeMin	char[]	json 字符	有效纬度	网络	Socket	客户端
最大纬度	latitudeMax	char[]	json 字符	有效纬度	网络	Socket	客户端
最小经度	longitudeMin	char[]	json 字符	有效经度	网络	Socket	客户端
最大经度	longitudeMax	char[]	json 字符	有效经度	网络	Socket	客户端

表 5.3.5 客户端服务系统 uploadTrafficInfo 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
手机号码	phoneNum	char[]	json 字符	有效手机号码	网络	Socket	客户端
纬度	latitude	char[]	json 字符	有效纬度	网络	Socket	客户端
经度	longitude	char[]	json 字符	有效经度	网络	Socket	客户端
上传时间	dateTime	char[]	json 字符	有效时间类型	网络	Socket	客户端
信息类型	type	char[]	json 字符	Unicode 字符	网络	Socket	客户端
信息等级	level	char[]	json 字符	Unicode 字符	网络	Socket	客户端
详情	detail	char[]	json 字符	Unicode 字符	网络	Socket	客户端

表 5.3.6 客户端服务系统 getTrafficInfo 输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
最小纬度	latitudeMin	char[]	json 字符	有效纬度	网络	Socket	客户端
最大纬度	latitudeMax	char[]	json 字符	有效纬度	网络	Socket	客户端
最小经度	longitudeM	char[]	json	有效经度	网络	Socket	客户端

	in		字符				
最大经度	longitudeMax	char[]	json 字符	有效经度	网络	Socket	客户端

4. 公交站点地理坐标获取系统

表 5.4 公交站点地理坐标获取系统输入项

名称	标识	类型	格式	有效范围	输入方式	输入媒体	输入数据来源
站点名称	platform_name	varchar (20)	字符	Unicode 字符	数据库获取	MySQL Connector	公交信息抓取系统

5.5 输出项

1. 公交信息抓取系统

表 5.5 公交信息抓取系统输出项

名称	标识	类型	格式	有效范围	输出方式	输出媒体	输出目标
线路名称	line_name	varchar (50)	字符	Unicode 字符	数据库写入	MySQL Connector	MySQL 数据库
站点名称	platform_name	varchar (20)	字符	Unicode 字符	数据库写入	MySQL Connector	MySQL 数据库
站点 ID	platform_id	char(3)	字符	三位大写字母标识	数据库写入	MySQL Connector	MySQL 数据库
公交车车牌	bus_id	varchar (10)	字符	Unicode 字符	数据库写入	MySQL Connector	MySQL 数据库
到站时间	arrive_time	datetime	日期时间	有效日期时间	数据库写入	MySQL Connector	MySQL 数据库

2. 公交数据分析及路况计算系统

表 5.6 公交数据分析及路况计算系统输出项

名称	标识	类型	格式	有效范围	输出方式	输出媒体	输出目标
起始站	platform_start	varchar (50)	字符	Unicode 字符	数据库写入	MySQL Connector	MySQL 数据库

							库
终到站	platform_end	varchar (20)	字符	Unicode 字符	数 据 库 写 入	MySQL Connector	MySQL L 数据 库
状态	state	char(3)	字符	以数字标 识的交通 状况	数 据 库 写 入	MySQL Connector	MySQL L 数据 库

3. 客户端服务系统

表 5.7.1 客户端服务系统 login 输出项

名称	标识	类型	格式	有效范围	输 出 方 式	输 出 媒 体	输出目标
登 录 结 果	loginResult	char[]	json 字 符	登录结果 状态码	网络	Socket	客户端
用户名	userName	char[]	json 字 符	有效用户 名	网络	Socket	客户端

表 5.7.2 客户端服务系统 register 输出项

名称	标识	类型	格式	有效范围	输 出 方 式	输 出 媒 体	输出目标
注 册 结 果	registerResult	char[]	json 字符	注册结果 状态码	网络	Socket	客户端

表 5.7.3 客户端服务系统 getUserInfo 输出项

名称	标识	类型	格式	有效范围	输 出 方 式	输 出 媒 体	输出目标
手 机 号 码	phoneNum	char[]	json 字 符	有效手机 号码	网络	Socket	客户端
用户名	userName	char[]	json 字 符	有效用户 名	网络	Socket	客户端

表 5.7.4 客户端服务系统 getRoadState 输出项

名称	标识	类型	格式	有效范围	输 出 方 式	输 出 媒 体	输出目标
拥堵程度 1	1	char[]	json 字 符 串 数 组	Unicode 字符	网络	Socket	客户端
拥堵程度 2	2	char[]	json	Unicode	网络	Socket	客户端

2			字符串数组	字符			
拥堵程度 3	3	char[]	json 字符串数组	Unicode 字符	网络	Socket	客户端
拥堵程度 4	4	char[]	json 字符串数组	Unicode 字符	网络	Socket	客户端

表 5.7.5 客户端服务系统 uploadTrafficInfo 输出项

名称	标识	类型	格式	有效范围	输出方式	输出媒体	输出目标
上传结果	uploadResult	char[]	json 字符	上传结果 状态码	网络	Socket	客户端

表 5.7.6 客户端服务系统 getTrafficInfo 输出项

名称	标识	类型	格式	有效范围	输出方式	输出媒体	输出目标
信息列表	retTrafficInfo	char[]	json 字符串数组	Unicode 字符	网络	Socket	客户端
发布者手机号码	phoneNum	char[]	json 字符	有效手机号码	网络	Socket	客户端
纬度	latitude	char[]	json 字符	有效纬度	网络	Socket	客户端
经度	longitude	char[]	json 字符	有效经度	网络	Socket	客户端
上传时间	dateTime	char[]	json 字符	有效时间 类型	网络	Socket	客户端
信息类型	type	char[]	json 字符	Unicode 字符	网络	Socket	客户端
信息等级	level	char[]	json 字符	Unicode 字符	网络	Socket	客户端
详情	detail	char[]	json 字符	Unicode 字符	网络	Socket	客户端

4. 公交站点地理坐标获取系统

表 5.7.4. 公交站点地理坐标获取系统输出项

名称	标识	类型	格式	有效范围	输出方式	输出媒体	输出目标
站点名称	platform_name	varchar (20)	字符	Unicode 字符	数据库写入	MySQL Connector	MySQL 数据库
纬度	latitude	double	浮点数	有效浮点数	数据库写入	MySQL Connector	MySQL 数据库
经度	longitude	double	浮点数	有效浮点数	数据库写入	MySQL Connector	MySQL 数据库

5.6 流程逻辑

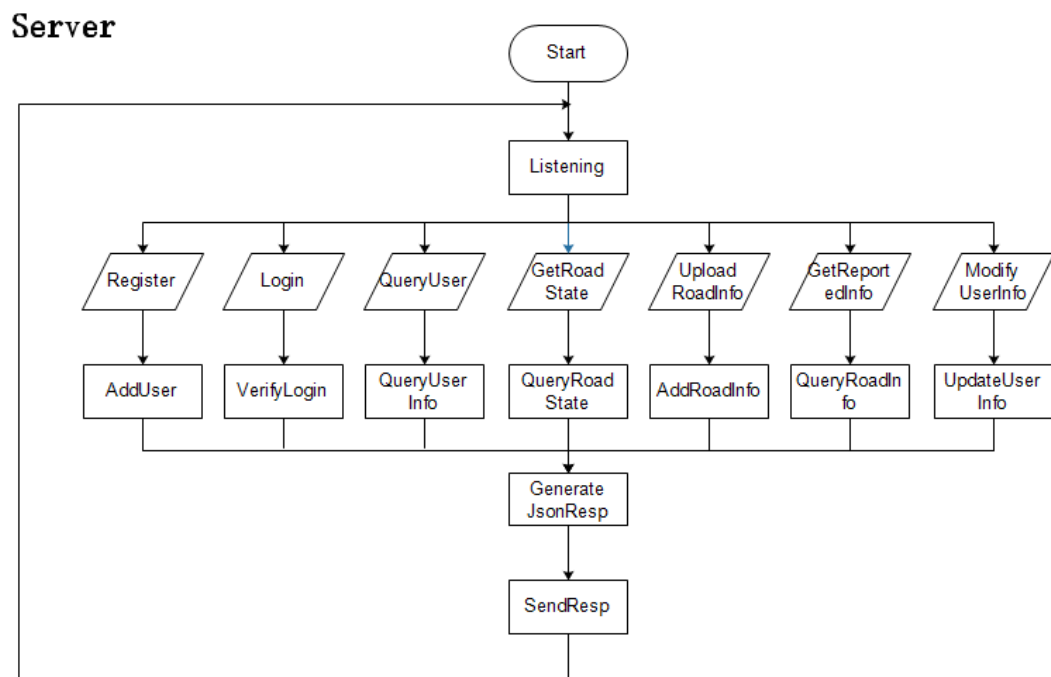


图 5.6.1 服务器端 server 流程图

Spider

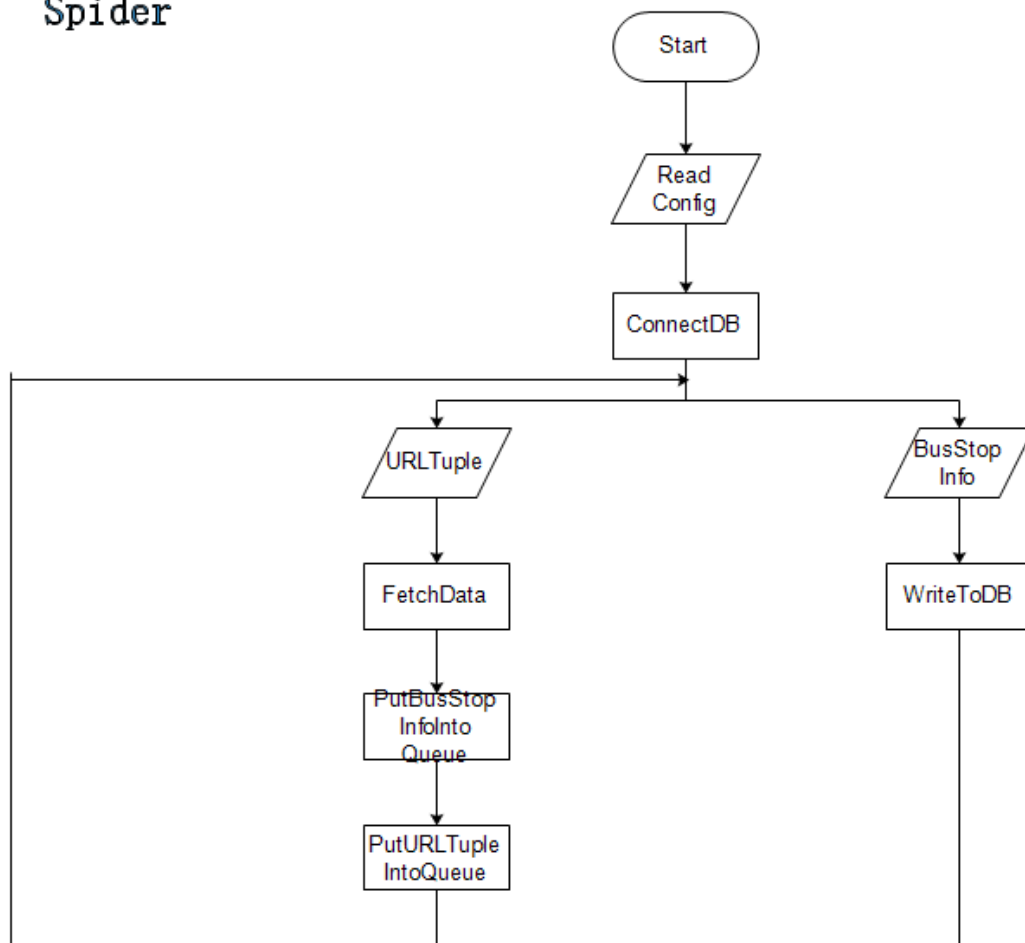


图 5.6.2 服务器端 spider 流程图

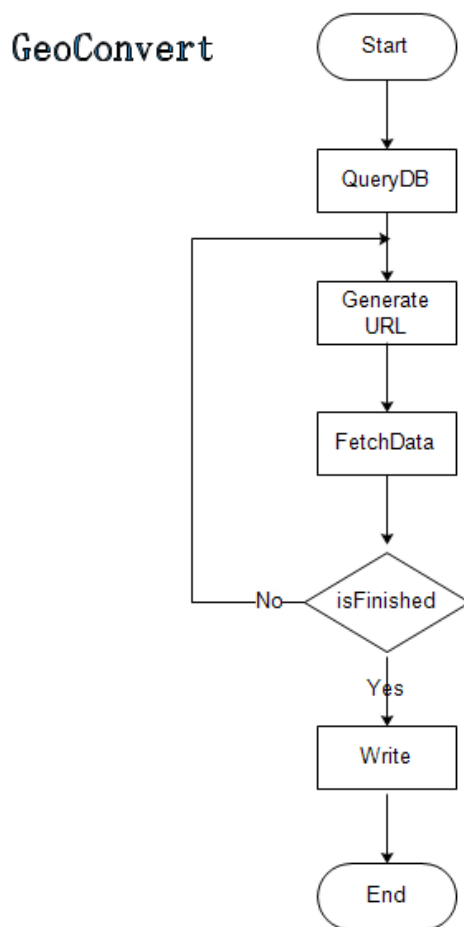


图 5.6.3 服务器端 geo_convert 流程图

5.7 算法

5.7.1 基准时间计算

算法是基于历史数据价值的获取，从爬虫得到的历史数据中得到相邻两站最短的运行时间的平均值，作为交通路网拥挤程度的评价参考。数据样本是苏州公交运行 10 天的历史数据。先获取每天出勤的所有公交车的车牌号，对每一辆车的运行情况逐一分析。

对于同一辆车，根据到站时间将到站信息排序。然后，依次计算（后一项到站时间-前一项到站时间），得到该公交车运行线路上相邻两站的运行时间。求得相邻两站的运行时间之后，与保存公交线路中相邻两站的字典中对应的值进行比较。字典的格式为：{（起始站 1，目的站 1）：到站时间，（起始站 2，目的站 2）：到站时间，…。}。当运行时间小于字典中对应的键-值对时，则对该键-值对重新赋值。通过这样的方法就可以获得当天每相邻两站最小的运行时间。

获得 10 天的数据之后，求得平均数。以该平均数作为拥挤程度的参考指标。

5.7.2 实时拥挤程度的计算

本软件的一大特点就是实时性，所以需要对实时获得公交车到站信息进行处理，来获得最新的拥挤程度。本部分因为是对数据库进行直接操作，使用了 SQL 进行代码编写。

(1) INSERT 触发

每当 BusLineDatabase 中新插入一条记录，算法就需要更新相应的数据。定义一个触发器 t1，t1 在 insert 时触发，获得新插入的记录的属性值。同时调用存储过程 p1。

(2) 存储过程 p1

存储过程 p1 主要实现两个功能 (1) 在表 result_1 中 insert 新的到站记录 (2) 调用存储过程 p2。

数据库维护一张 result_2 的表保存中间过程。这张表保存所有公交车最新的两条到站记录。格式如下，属性分别为：车牌号，站名，到站时间。

列名	数据类型	允许空
bus_id	vachar(10)	否
platform_name	varchar(20)	否
arrive_time	datetime	否

为了方便计算最新的需要调整的相邻两站的到站时间，所以 result_1 表只保存同一车牌号的时间最新的两条到站数据。如果 result_1 的表中没有新获得的公交车的信息，则 insert 新的记录；如果当前新的公交车记录中的车牌号所对应的记录数量小于 2，则直接 insert；否则先删除同一车牌对应记录中到站时间最大的记录，再将新的到站记录 insert。

同时，调用存储过程 p2。

(3) 存储过程 p2

存储过程 p2 功能是重新计算调整新 insert 的数据所对应的相邻两站的到站时间。比如两条数据同属同一辆公交车，data1（起始站 1，目的站 1）:到站时间，data2（起始站 2，目的站 2）:到站时间 data1 中的到站时间 < data2 的到站时间。当新插入一条新纪录时，data1 数据被删除，data2 变成 data1，新插入的数据则变成 data2。P2 则负责计算 data2 的到站时间减去 data1 的到站时间，获得新的相邻两站的运行时间。更新 result_2 表。

result_2 表格式如下，起始站，目的站，两站之间运行时间：

列名	数据类型	允许空
plat1	varchar(20)	是
plat2	varchar(20)	是
time	datetime	是

(4) 拥挤程度更新

将 result_2 中更新的数据与从历史记录中获得的历史基准数据进行对比，根据两者的比值。将拥挤程度赋值为 1~4，值越大，说明该路段越拥挤。路况拥挤程度保存在 RoadState 中，表格式如下，起始站，目的站，两站之间道路的拥挤

程度：

列名	数据类型	允许空
platform_start	varchar(20)	是
platform_end	varchar(20)	是
state	int(11)	是

5.8 接口

本程序所隶属的上一层模块：本程序为第一层模块

隶属于本程序的下一层模块、子程序：通信模块、百度地图 API 提供的接口模块

参数赋值和调用方式：直接调用

与本程序相直接关联的数据结构：顺序结构

5.9 存储分配

标识符	类型
phoneNum	varchar [20]
userName	varchar [40]
userPass	varchar [20]
latitude	double
longitude	double
posName	varchar[20]
dateTime	datetime
type	varchar[20]
level	varchar[20]
detail	varchar[200]

5.10 注释设计

2. 加在模块首部的注释：/*模块功能说明*/
3. 对各变量的功能、范围、缺省条件等所加的注释：//变量内容、需要说明的部分

5.11 限制条件

数据抓取部分受苏州市公交信息网站限制，抓取频率不能过高，导致部分数据时间粒度过大；

百度地图地理坐标转换 API 未涵盖全部苏州市公交站点，少部分站点坐标无法通过地理坐标转换系统完成坐标转换。

5.12 测试计划

输入数据：客户端测试用例

预期结果：各个功能都能正常处理以及响应

进度安排：已经完成

人员职责：李东宇负责

5.13 尚未解决的问题

服务器上未对客户端的登录会话跟踪连接，故存在一定的数据访问权限限制问题。后续可在服务器端为每个登录用户维持一个 `cookie` 以此鉴别登录用户身份并强制超时非活动用户下线。

6 通信模块

6.1 程序描述

该模块用于 Android 客户端与服务器间的消息通信。通信选择利用 socket 网络编程来建立连接，实现数据交互的目的。在传输数据过程中，选择 json 作为数据格式，先将要传输的数据按照事先定义好的数据格式进行封装，然后将其发送。同样的，接收端接收到的是一串字节流，还原为相应格式的数据包后才能读取其中的信息，并对其进行 json 解析。

6.2 功能

实现 Android 客户端与服务器的通信，根据客户端发出的请求，服务器端做出相应的响应，返回结果或需要的数据，供客户端显示。

6.3 性能

准确性：支持多用户的并发操作。

实时性：由于传输数据小，基本不会出现延迟情况。

6.4 输入项

详细情况请参见《路况可视化 APP-接口设计说明书》，其中对所有请求和响应做出了明确的说明。

6.5 输出项

同输入项 6.4。

6.6 算法

无。

6.7 流程逻辑

客户端将数据封装成规定的协议包发送至服务器。服务器解析协议包提前命令以及相关的数据进行操作。并反馈回客户端。

6.8 接口

本程序所隶属的上一层模块：客户端 UI 界面模块

隶属于本程序的下一层模块：TCP/IP 协议的网络传输

参数赋值和调用方式：直接调用

与本程序相直接关联的数据结构：顺序结构

6.9 存储分配

无

6.10 注释设计

1. 加在模块首部的注释：/*通信模块*/
2. 对各变量的功能、范围、缺省条件等所加的注释：

6.11 限制条件

必须在网络配置和连接和正常的情况下使用，网络必须允许发送和接收服务器端的 50001 端口的数据请求和响应。

6.12 测试计划

输入数据：大规模的注册操作已经登录考勤操作。

预期结果：正确实现注册登录考勤功能。

进度安排：延期完成。

人员职责：李东宇负责

6.13 尚未解决的问题

未经过大规模并发连接服务器的测试。