**软件工程课程设计**

**《智能物流管理系统项目》**

**详细设计说明书**

**小组：目标是成为抽象带师**

**设计组学生：**

**阿琪 1120161708**

**周赫斌 1120161737**

**万伯阳 1120161725**

**指导教师：汤世平**

**北京理工大学**

**2019年11月**

|  |  |
| --- | --- |
| 项目名称 | 智能物流管理系统 |
| 文档 | **详细说明书** |
| 版本 | **V1.0** |
| 作者 | **周赫斌** |
| 最后更新时间 | **2019-11-14** |

目录

[1. 引言 4](#_Toc24666590)

[1.1 编写目的 4](#_Toc24666591)

[1.2 项目背景 4](#_Toc24666592)

[1.3 定义 4](#_Toc24666593)

[1.4 参考资料 4](#_Toc24666594)

[2. 总体设计 5](#_Toc24666595)

[2.1 设计概述 5](#_Toc24666596)

[2.2 软件结构 5](#_Toc24666597)

[3. 表示层服务模块设计说明 6](#_Toc24666598)

[3.1 设计概述 6](#_Toc24666599)

[3.2 功能 6](#_Toc24666600)

[3.3 性能 6](#_Toc24666601)

[3.4 接口 7](#_Toc24666602)

[4. 应用逻辑服务模块设计说明 9](#_Toc24666603)

[4.1 设计概述 9](#_Toc24666604)

[4.2 算法描述 9](#_Toc24666605)

[4.2.1 货物摆放空间最优化算法 9](#_Toc24666606)

[4.2.2 仓库内货物摆放位置调度算法 10](#_Toc24666607)

[4.2.3 仓库间货物调度算法 11](#_Toc24666608)

[4.2.4 最优路径计算算法 11](#_Toc24666609)

[4.3 功能 12](#_Toc24666610)

[4.4 性能 13](#_Toc24666611)

[4.5 接口 13](#_Toc24666612)

[5. 数据存储服务模块设计说明 16](#_Toc24666613)

[5.1 设计概述 16](#_Toc24666614)

[5.2 功能 16](#_Toc24666615)

[5.3 性能 16](#_Toc24666616)

[5.4 接口 17](#_Toc24666617)

# 引言

## 编写目的

详细设计说明书主要介绍各个模块更底层的逻辑，它在概要设计说明书的基础上描述了各个模块的具体实现。为了给开发人员的开发工作提供便利，编写本文档。

## 项目背景

随着电子商务的兴起，我国物流产业的业务量暴增，从业人员的数量也随之迅速增长。但是由于物流行业的成本高居不下，同行业之间的低价竞争，也使得物流企业的利润不升反降。同时，作为劳动密集型产业，众多物流企业规模小，信息化程度低，缺少物流标准化体系，过于激烈的竞争降低服务质量等都现如今急需解决的问题。

本系统将物流网络系统和运输系统深度结合，实现智能运输、智能识别和自动分拣的“物自（智）流”过程，提高物流企业的品质和效率。

## 定义

## 参考资料

《智能物流管理系统需求规格说明书》

《智能物流管理系统概要设计说明书》

# 总体设计

## 设计概述

本系统旨在为用户提供信息的系统化管理与物流调度的自动化设置。本系统可分为表示层服务、应用逻辑服务、数据存储服务三个模块。

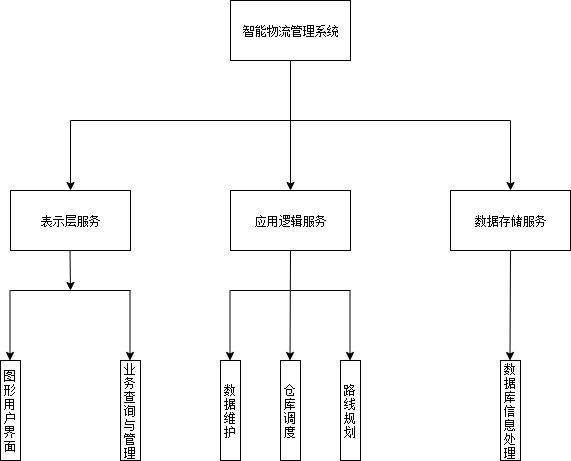
表示层服务主要为用户提供友好的图形化界面，以方便用户操作。同时，表示层服务也提供用户的业务查询、管理服务的接口。

应用逻辑服务负责处理物流管理系统各种数据的实时维护工作。它对表示层服务的业务查询、管理服务提供支持。应用逻辑服务也对物流管理系统的核心业务——仓储调度管理和物流路径规划提供支持。

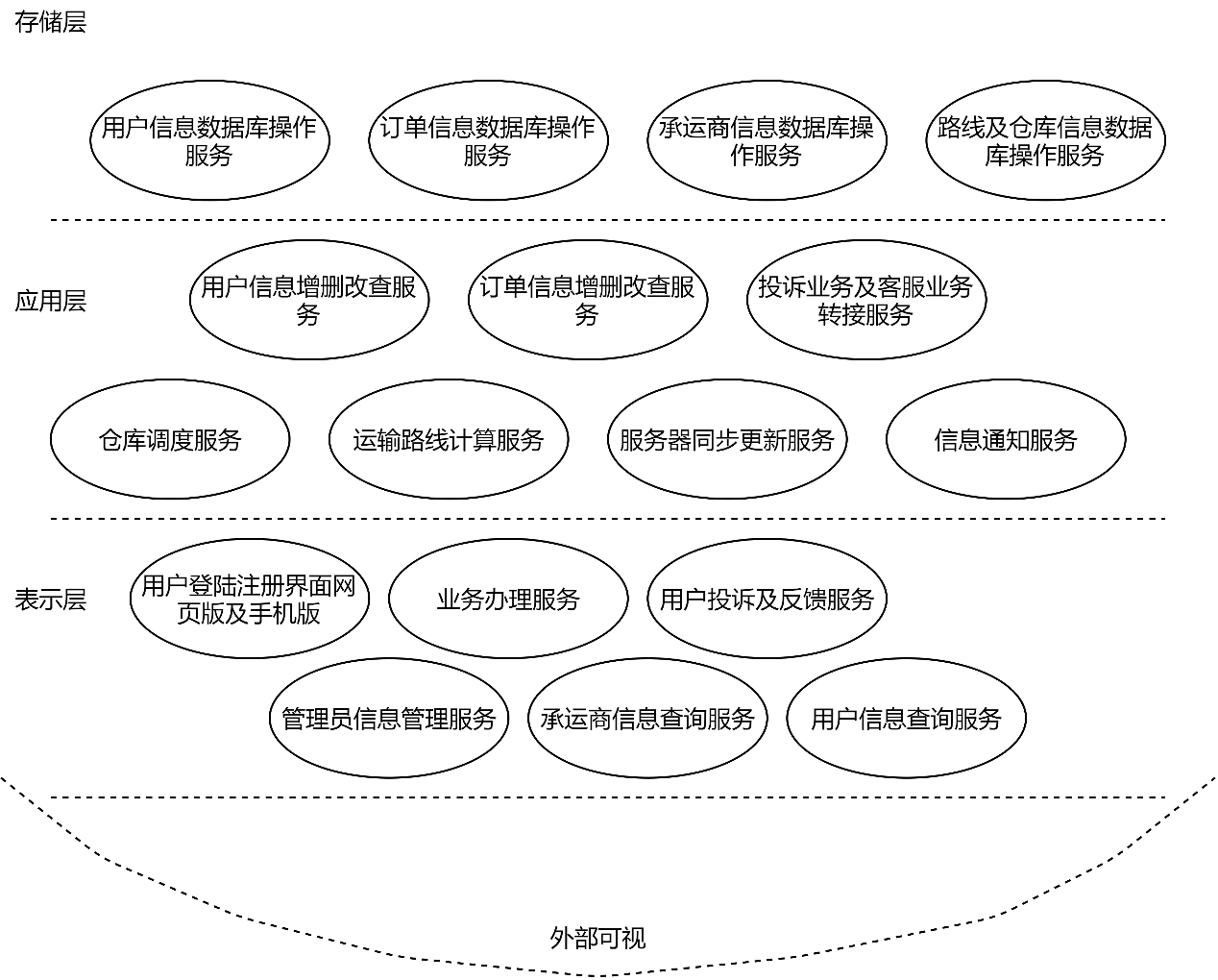
数据存储服务负责对用户信息等静态数据进行维护。

## 软件结构

本系统的总体模块结构与主要功能如下：



其中，每个模块的功能又可细分，如下：



# 表示层服务模块设计说明

## 设计概述

表示层服务主要为用户提供友好的图形化界面，以方便用户操作。同时，表示层服务也提供用户的业务查询、管理服务的接口。

表示层服务的业务查询、管理功能通过调用应用逻辑服务的功能实现。

## 功能

表示层提供用户图形界面服务、业务办理服务、用户投诉与反馈服务、管理员信息管理服务、承运商信息查询服务、用户信息查询服务。

## 性能

由于本模块对系统外部环境（如网络环境）有一定依赖，对本模块的及时反馈性能要求如下：

除去网络传输时间开销，模块处理时间（单次）应不超过3秒。

当网络信号不佳（以15秒内未收到有效反馈为界）时，应当有醒目标识提醒用户对网络环境进行检查。

## 接口

提供接口如下：

|  |  |
| --- | --- |
| 函数原型 | int register(char\* phone, char\* verCode, char\* username, char\* password) |
| 输入描述 | [username:用户名][password:密码][phone:电话号码][verCode:短信验证码] |
| 返回值描述 | 注册成功时，返回0。失败时，返回-1。 |
| 备注 | 需要通过手机短信验证，并且不存在用户名冲突等不合法情况，才可以注册。 |

|  |  |
| --- | --- |
| 函数原型 | int login(char\* username, char\* password) |
| 输入描述 | [username:用户名][password:密码] |
| 返回值描述 | 登陆成功时，返回ID，表示用户标识码。失败时，返回-1。 |
| 备注 | 无 |

|  |  |
| --- | --- |
| 函数原型 | int changeInfo(int ID, int operation, char\* text) |
| 输入描述 | [ID:用户标识码][operation:指示待修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时，返回-1。 |
| 备注 | 需要用户先登录，才能得到用户标识码。同时当用户权限不足时，也会修改失败。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askInfo(int ID, int operation) |
| 输入描述 | [ID:用户标识码][operation:指示查询字段] |
| 返回值描述 | 查询成功时，返回一个字符串。失败时，返回null。 |
| 备注 | 需要用户先登录，才能得到用户标识码。同时当用户权限不足时，也会查询失败。 |

|  |  |
| --- | --- |
| 函数原型 | int addOrder(int ID, char\* text) |
| 输入描述 | [ID:用户标识码][text:待运输货物的信息（压在一个JSON类型的字符串中）] |
| 返回值描述 | 创建订单成功时，返回一个0。失败时，返回-1。 |
| 备注 | 需要用户先登录，才能得到用户标识码。 |

|  |  |
| --- | --- |
| 函数原型 | int addComplaint(int ID, char\* text) |
| 输入描述 | [ID:用户标识码][text:投诉信息] |
| 返回值描述 | 提交投诉成功时，返回一个0。失败时，返回-1。 |
| 备注 | 需要用户先登录，才能得到用户标识码。 |

# 应用逻辑服务模块设计说明

## 设计概述

应用逻辑服务负责处理物流管理系统各种数据的实时维护工作。它对表示层服务的业务查询、管理服务提供支持。应用逻辑服务也对物流管理系统的核心业务——仓储调度管理和物流路径规划提供支持。

应用逻辑服务的业务查询、管理服务需要在接收到表示层服务的查询请求后，转发到数据存储服务中进行操作。

仓储调度管理和物流路径规划使用了货物摆放空间最优化算法、仓库内货物摆放位置调度算法、仓库间货物调度算法、最优路径计算算法。

## 算法描述

## 货物摆放空间最优化算法

因为货物形状不一，且包含特殊要求，情况过于复杂，无法用一种单一算法概括，这里分类讨论。

该算法优先考虑有破损、爆炸的风险以及有各种特殊要求的货物。对于可叠放、无特殊要求、破损风险较小的普通货物（后称普通货物）则后考虑。

首先对于有爆炸风险的货物，不应将之与其他可能会发热或运送途中易发生碰撞的零散货物摆放在一起。应安全摆放该种货物。可将货物的长宽高在货物的实际长宽高基础上适当提升，以便留出一定安全距离。同时有危险的货物附近应当做好消防措施，如准备灭火器等。

对于上方无法放置物品的货物，当它是小体积货物时放在货架上则不会有问题，当它是大体积货物时上方不会堆叠其他货物。

## 仓库内货物摆放位置调度算法

对于每个仓库，将仓库按矩形计算，保存该矩形的长宽。将该仓库对应的一整个大矩形分成若干较小的矩形，该较小的矩形应该能容纳可能入库的最大批次货物，同时在矩形之间保留能够容纳自动运货机器人或工人移动的最小道路。对于所有入库的货物，将其考虑为一个长方体（通过货物摆放空间最优化算法得出）。能够得到该长方体的长宽高。根据长宽高的属性，将货物分为小体积货物和大体积货物两类。小体积货物定义为长宽高都不超过30cm的货物，大体积货物定义为长宽高中存在一维超过30cm的货物。所有危险品都应当该增加长宽高至少达到大体积货物的标准。小体积货物和大体积货物应当分开存放。对于小体积货物，统一安排货架盛放。对于大体积货物，首先考虑对于不要求顶部向上的货物翻转长宽高使得占地面积最小。由于大体积货物一般重量不轻，所以不考虑叠放。因为对于大体积货物重新排列的成本太高，所以只考虑在空隙处放下物品。从俯视图看，每个货物是一个二维的矩形。把所有货物的长宽上取整到10cm的倍数，然后暴力枚举每个货物的四个顶点，如果以这个点为左上顶点可以放下待处理物品就放下。

## 仓库间货物调度算法

当仓库满时，应当设定所有送达该仓库的路线的权值为充分大，以停止往该仓库送货。若有发往该仓库的货物已经开始发送，在到达后没有足够的空间容纳货物，则先在仓库外等待，直到仓库有足够的容量容纳货物。当仓库的货物容量下降到80%以下，则重新设定送达该仓库的路线，以使得它能正常工作。实现如下：

在每次最新的仓库库存数据上传后，统一更新所有仓库的点的数据：

仓库权值 = （1-仓库当前大型货物仓储能力 / 仓库总大型货物仓储能力） + （1-仓库当前小型货物仓储能力 / 仓库总小型货物仓储能力）

当仓库权值大于1.95时将之设为无穷大，并停止路径设定，当新计算出来的仓库权值小于1.6且路径设定被停止时，重新启动该仓库的路径设计。

## 最优路径计算算法

每个仓库为一个点，每条路径为一条边，代表点到点之间的第条路径。每条边有权值，该权值由该路径的平均通过时间，事故发生率和运费加权求和得出。在每次路径数据更新时，对于每两个仓库之间的所有路径，找出权值最小的路径，用表示，代表直接从一个仓库到达另一个仓库的最优路径。

在得到所有后用Dijkstra算法得出仓库之间的最短路。

实现如下：

路径权值 = \*路径平均通过时间 + \*运费 + \*事故发生率

(需要由客户即物流公司自己设定)



## 功能

应用层提供用户信息增删改查服务、订单信息增删改查服务、投诉业务及客服业务转接服务、仓库调度服务、运输路线计算服务、服务器同步更新服务、信息通知服务。

## 性能

对本模块的时间性能要求如下：

对于不涉及到仓储调度管理和物流路径规划的操作（所有“增删改查”的操作），应当在3秒内返回结果。

对于涉及到仓储调度管理和物流路径规划的操作，要求在10秒内返回结果，同时对高并发操作有一定的支持。

对于本模块的空间性能要求如下：

对于不涉及到仓储调度管理和物流路径规划的操作，没有空间要求。

对于涉及到仓储调度管理和物流路径规划的操作，要求处理的物流路径级别在千万左右，使用的内存空间应当不超过128GB。同时，在操作数较多的高峰时段，允许临时使用不超过128GB的额外空间。

## 接口

|  |  |
| --- | --- |
| 函数原型 | void solveComplaint(char\* username, int ID, char\* text) |
| 输入描述 | [username:投诉人用户名][ID:处理人用户标识码][text:投诉处理记录] |
| 返回值描述 | 无 |
| 备注 | 无 |

|  |  |
| --- | --- |
| 函数原型 | int refreshRoad(int ID, int roadID, int C1, int C2, int C3) |
| 输入描述 | [ID:信息提交人用户标识码][roadID:道路编号][C1/C2/C3:新的道路加权系数] |
| 返回值描述 | 修改成功时，返回0。失败时，返回-1。 |
| 备注 | 修改道路信息的时候，需要信息提交人有足够权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeInfo(int ID, int operation, char\* text) |
| 输入描述 | [ID:用户标识码][operation:指示待修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时，返回-1。 |
| 备注 | 当用户权限不足时，会修改失败。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askInfo(int ID, int operation) |
| 输入描述 | [ID:用户标识码][operation:指示查询字段] |
| 返回值描述 | 查询成功时，返回一个字符串。失败时，返回null。 |
| 备注 | 当用户权限不足时，会查询失败。 |

|  |  |
| --- | --- |
| 函数原型 | int noticeInventory(char\* username, char\* text) |
| 输入描述 | [username:待通知用户名][text:入库通知信息] |
| 返回值描述 | 发送信息成功时，返回一个0。失败时，返回-1。 |
| 备注 | 无 |

|  |  |
| --- | --- |
| 函数原型 | void addItem(int itemID, int inventoryID) |
| 输入描述 | [itemID:入库货物编号][inventoryID:仓库编号] |
| 返回值描述 | 无 |
| 备注 | 同时调用RefreshInventory()检测仓库过满 |

|  |  |
| --- | --- |
| 函数原型 | void delItem(int itemID, int inventoryID) |
| 输入描述 | [itemID:出库货物编号][inventoryID:仓库编号] |
| 返回值描述 | 无 |
| 备注 | 同时调用RefreshInventory()检测仓库不再过满 |

|  |  |
| --- | --- |
| 函数原型 | void RefreshInventory(int inventoryID) |
| 输入描述 | [inventoryID:仓库编号] |
| 返回值描述 | 无 |
| 备注 | 刷新仓库信息，检测是否过满或不再过满 |

|  |  |
| --- | --- |
| 函数原型 | void RefreshGraph(int S) |
| 输入描述 | [S:出发点] |
| 返回值描述 | 无 |
| 备注 | 每隔一段时间刷新从S点出发到其他点最短距离 |

|  |  |
| --- | --- |
| 函数原型 | void inInventory(int itemID, int ID) |
| 输入描述 | [itemID:入库的货物编号][ID:记录人用户标识码] |
| 返回值描述 | 无 |
| 备注 | 同时会调用noticeInventory()通知客户 |

|  |  |
| --- | --- |
| 函数原型 | void outInventory(int itemID, int ID) |
| 输入描述 | [itemID:出库的货物编号][ID:记录人用户标识码] |
| 返回值描述 | 无 |
| 备注 | 无 |

# 数据存储服务模块设计说明

## 设计概述

数据存储服务负责对用户信息等静态数据进行维护。主要涉及数据库的增删改查等方面。

## 功能

存储层提供用户信息数据库操作服务、订单信息数据库操作服务、承运商信息数据库操作服务、路线及仓库信息数据库操作服务。

## 性能

对本模块的时间性能要求：模块处理时间（单次）不应超过2秒，同时对于高并发读写有一定处理能力。

对本模块的存储性能要求：应当有较大规模数据的存储能力，保存十亿级规模的数据。同时，对于断电等突发情况有一定的抗灾害能力，以及做好备份工作。

## 接口

|  |  |
| --- | --- |
| 函数原型 | int register(char\* username, char\* password, char\* phone) |
| 输入描述 | [username:注册用户名][password:注册密码][phone:注册手机号] |
| 返回值描述 | 注册成功时，返回0。失败时返回-1。 |
| 备注 | 无 |

|  |  |
| --- | --- |
| 函数原型 | int login(char\* username, char\* password) |
| 输入描述 | [username:登陆用户名][password:密码] |
| 返回值描述 | 登陆成功时返回用户标识码ID。失败时返回-1。 |
| 备注 | 无 |

|  |  |
| --- | --- |
| 函数原型 | char\* askUserInfo(char\* username, int ID, int operation) |
| 输入描述 | [username:待查询用户名][ID:查询者用户标识码][operation:标识查询字段] |
| 返回值描述 | 查询成功时，返回待查询信息。失败时返回null。 |
| 备注 | 需要查询人有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeUserInfo(int ID, int operation, char\* text) |
| 输入描述 | [ID:修改者用户标识码][operation:标识修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时返回-1。 |
| 备注 | 需要修改人有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askOrderInfo(int RFID, int ID, int operation) |
| 输入描述 | [RFID:待查询订单号][ID:查询者用户标识码][operation:标识查询字段] |
| 返回值描述 | 查询成功时，返回待查询信息。失败时返回null。 |
| 备注 | 需要查询人有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeOrderInfo(int RFID, int ID, int operation, char\* text) |
| 输入描述 | [RFID:待修改订单号][ID:修改者用户标识码][operation:标识修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时返回-1。 |
| 备注 | 需要修改人有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int addTransInfo(int trID, int ID, char\* text) |
| 输入描述 | [trID:待添加承运商编号][ID:操作者用户标识码][text:承运商信息(JSON格式)] |
| 返回值描述 | 添加成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int delTransInfo(int trID, int ID) |
| 输入描述 | [trID:待删除承运商编号][ID:操作者用户标识码] |
| 返回值描述 | 删除成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askTransInfo(int trID, int ID, int operation) |
| 输入描述 | [trID:待查询承运商编号][ID:操作者用户标识码][operation:标识查询字段] |
| 返回值描述 | 查询成功时，返回待查询字段。失败时返回null。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeTransInfo(int trID, int ID, int operation, char\* text) |
| 输入描述 | [trID:待修改承运商编号][ID:操作者用户标识码][operation:标识修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int addLine(int routeID, int ID, char\* text) |
| 输入描述 | [routeID:待添加路径编号][ID:操作者用户标识码][text:路径信息(JSON格式)] |
| 返回值描述 | 添加成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int delLine(int routeID, int ID) |
| 输入描述 | [routeID:待删除路径编号][ID:操作者用户标识码] |
| 返回值描述 | 删除成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askLine(int routeID, int ID,int operation) |
| 输入描述 | [routeID:待查询路径编号][ID:操作者用户标识码][operation:标识查询字段] |
| 返回值描述 | 查询成功时，返回待查询字段。失败时返回null。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeLine(int routeID, int ID, int operation, char\* text) |
| 输入描述 | [routeID:待修改路径编号][ID:操作者用户标识码][operation:标识修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int addInventory(int invID, int ID, char\* text) |
| 输入描述 | [routeID:待添加仓库编号][ID:操作者用户标识码][text:仓库信息(JSON格式)] |
| 返回值描述 | 添加成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int delInventory(int invID, int ID) |
| 输入描述 | [routeID:待删除仓库编号][ID:操作者用户标识码] |
| 返回值描述 | 删除成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。同时只有在没有经过这个仓库的路径时才能删除仓库。 |

|  |  |
| --- | --- |
| 函数原型 | char\* askInventory(int invID, int ID, int operation) |
| 输入描述 | [invID:待查询路径编号][ID:操作者用户标识码][operation:标识查询字段] |
| 返回值描述 | 查询成功时，返回待查询字段。失败时返回null。 |
| 备注 | 需要有足够的权限。 |

|  |  |
| --- | --- |
| 函数原型 | int changeInventory(int invID, int ID, int operation, char\* text) |
| 输入描述 | [invID:待修改仓库编号][ID:操作者用户标识码][operation:标识修改字段][text:修改字段] |
| 返回值描述 | 修改成功时，返回0。失败时返回-1。 |
| 备注 | 需要有足够的权限。 |