设计报告

一、项目概述

仓库管理是许多企业和组织日常运营的重要环节。传统的手工管理方式存在着效率低下、易出错等问题。因此，开发一个自动化的仓库管理系统能够提高工作效率、减少错误，并为用户提供更好的数据分析和决策支持。本项目的开发旨在满足这些需求，使仓库管理变得更加简单和可靠。

本仓库管理系统使用C语言编写，通过结构体和文件操作来存储和管理数据。用户可以通过命令行界面与系统进行交互，输入相应的指令实现具体的功能。本项目的功能包括注册、登录、进货、退货、查找货物、商品销售、显示库存以及显示操作者信息、供应商信息和商品销量等。

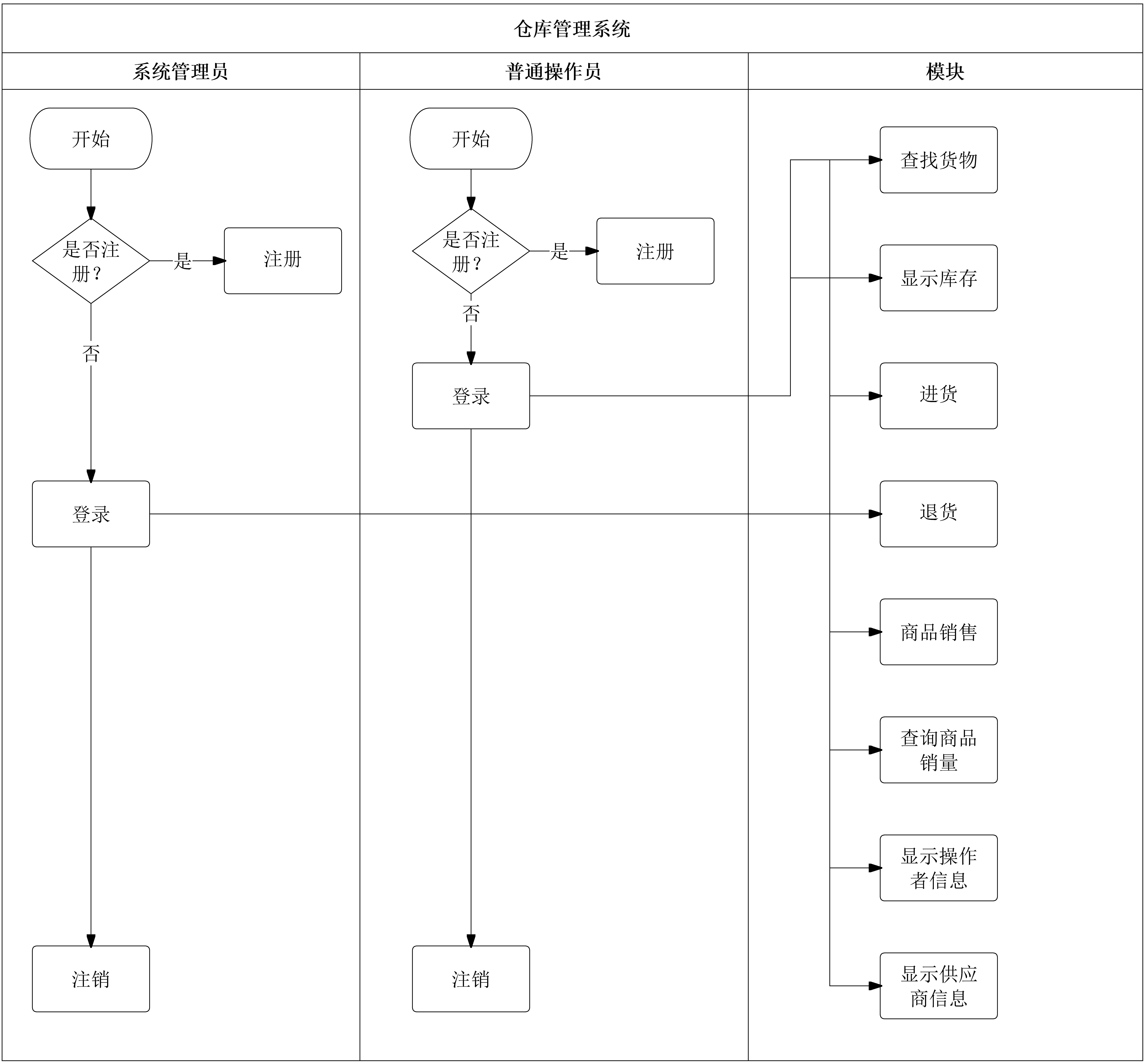
该仓库管理系统提供了一个方便、高效的方式来管理仓库中的货物。它可以帮助用户准确记录进货、退货和销售的货物信息，实时更新库存情况，并提供查找功能来快速定位特定的货物。此外，通过显示操作者信息、供应商信息和商品销量等统计数据，用户可以更好地了解仓库的运营情况，做出相应的决策。

二、项目系统业务流程分析

本项目面向的服务对象是仓库管理员。系统将仓库管理员分为两类，一类是系统管理员，一类是普通操作员。普通操作员可以使用的功能是系统管理员的简略版。

首先来分析系统管理员的业务流程。如果没有注册，可以先注册一个系统管理员账号。如果已注册，可以登录成功后选择不同的功能模块。系统管理员的功能模块包括进货、退货、查找货物、商品销售、显示库存、显示操作者信息、供应商信息、商品销量。

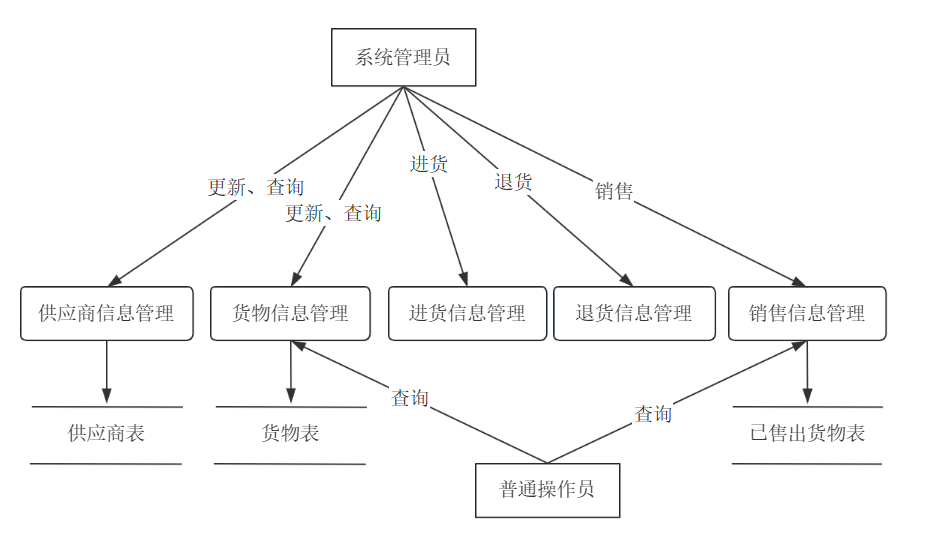
普通操作员与系统管理员的业务流程一致，但是可选的功能模块只有查找货物和显示库存。整个仓库管理系统的业务流程泳道图如下图所示：



**图2.1 业务流程泳道图**

三、系统数据分析描述

1．数据流程分析

****

**图3.1 数据流程分析图**

2．数据对象描述

**1）数据对象的属性及其对应的结构体类型**

本系统定义admin、user、cargo、cargoSold、supplier共5种数据对象，接下来以代码形式呈现各种数据对象的结构体类型和属性，关于属性的说明以注释形式呈现在代码中。

数据对象admin：

// 定义admin数据对象，对每个系统管理员的账户信息进行抽象

struct admin {

char admin\_id[20]; // 管理员编号 主码

char admin\_account[20]; // 管理员账号

char admin\_password[20]; // 管理员密码

char admin\_phone[12]; // 管理员电话

};

数据对象user：

// 定义user数据对象，对每个普通操作员的账户信息进行抽象

struct user {

char user\_id[20]; // 操作员编号 主码

char user\_account[20]; // 操作员账号

char user\_password[20]; // 操作员密码

char user\_phone[12]; // 操作员电话

};

数据对象cargo：

// 定义cargo数据对象，对每个货物信息进行抽象

struct cargo {

char cargo\_id[20]; // 货物编号 主码

char cargo\_name[20]; // 货物名称

int cargo\_quantity; // 货物数量

char cargo\_unit[20]; // 货物单位

char supplier\_id[20]; // 供应商编号 外码

};

数据对象cargoSold：

// 定义cargo数据对象，对每个已售出货物信息进行抽象

struct cargoSold {

char cargoSold\_id[20]; // 已售出货物编号 主码

char cargoSold\_name[20]; // 已售出货物名称

int cargoSold\_quantity; // 已售出货物数量

char cargoSold\_unit[20]; // 已售出货物单位

char cargoSold\_place[20]; // 货物销售地点

char cargo\_id[20]; // 货物编号 外码

char admin\_id[20]; // 管理员编号 外码

char user\_id[20]; // 操作员编号 外码

};

数据对象supplier：

// 定义supplier数据对象，对每个供应商信息进行抽象

struct supplier {

char supplier\_id[20]; // 供应商编号 主码

char supplier\_name[60]; // 供应商名称

char supplier\_phone[12]; // 供应商电话

char supplier\_address[60]; // 供应商地址

};

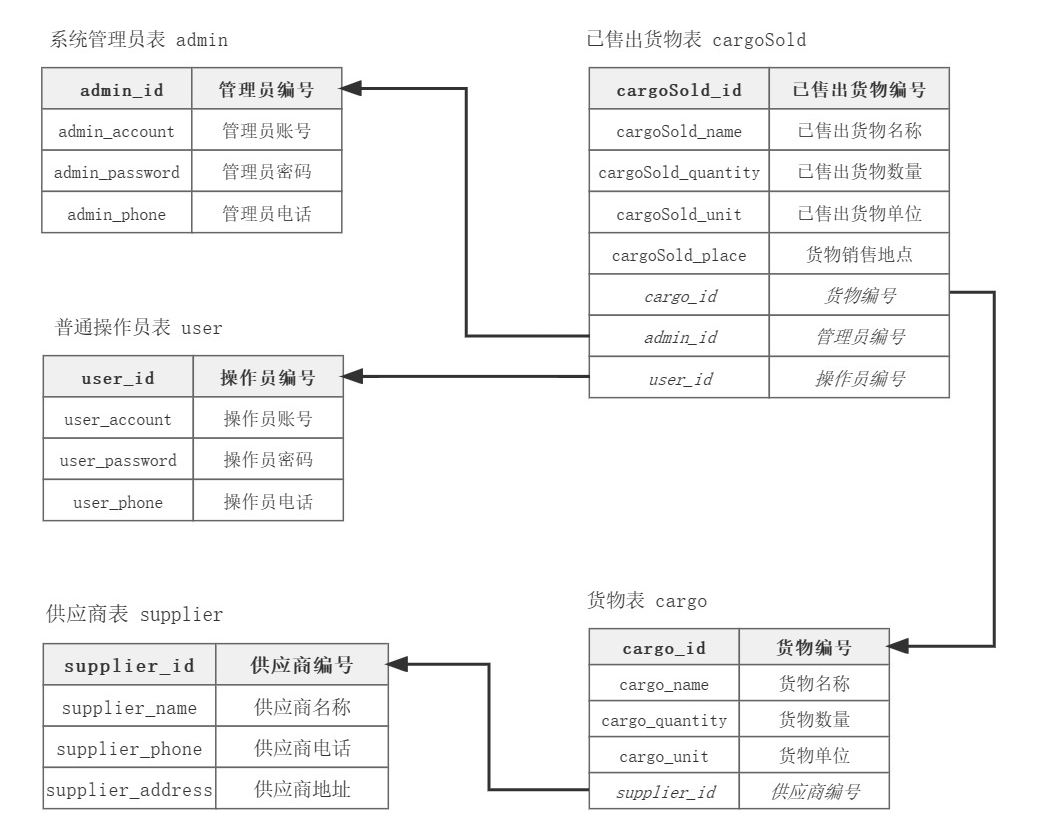
**2）数据对象之间的联系**

系统管理员与已售出货物之间存在一对多的联系，已售出货物表中的外码admin\_id参照系统管理员表中的主码admin\_id。

普通操作员与已售出货物之间存在一对多的联系，已售出货物表中的外码user\_id参照普通操作员表中的主码user\_id。

货物与已售出货物之间存在一对多的联系，已售出货物表中的外码cargo\_id参照货物表中的主码cargo\_id。

供应商与货物之间存在一对多的联系，货物表中的外码supplier\_id参照供应商表中的主码supplier\_id。



**图3.2 关系模式导航图**

3．数据对象实例

系统管理员表实例数据如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **admin\_id** | admin\_account | admin\_password | admin\_phone |
| **ad2291100** | root | abc123.. | 18618660000 |
| **ad2291101** | c | abc123.. | 18618660001 |
| **ad2291102** | l | abc123.. | 18618660002 |
| **ad2291103** | f | abc123.. | 18618660003 |
| **ad2291104** | cyn | abc123.. | 18618660004 |
| **ad2291105** | ly | abc123.. | 18618660005 |
| **ad2291106** | y | abc123.. | 18618660006 |
| **ad2291107** | n | abc123.. | 18618660007 |
| **ad2291108** | t | abc123.. | 18618660008 |
| **ad2291109** | q | abc123.. | 18618660009 |

普通操作员表实例数据如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **user\_id** | user\_account | user\_password | user\_phone |
| **u002291101** | 马德胜 | abc123.. | 17618660001 |
| **u002291102** | 李勤 | abc123.. | 17618660002 |
| **u002291103** | 龚彪 | abc123.. | 17618660003 |
| **u002291104** | 林木华 | abc123.. | 17618660004 |
| **u002291105** | 沈墨 | abc123.. | 17618660005 |
| **u002291106** | 傅卫军 | abc123.. | 17618660006 |
| **u002291107** | 隋东 | abc123.. | 17618660007 |
| **u002291108** | 王响 | abc123.. | 17618660008 |
| **u002291109** | 王阳 | abc123.. | 17618660009 |
| **u002291110** | 王北 | abc123.. | 17618660010 |

供应商表实例数据如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **supplier\_id** | supplier\_name | supplier\_ phone | supplier\_address |
| **s002291101** | 江西铜业集团有限公司 | 15618660001 | 江西省南昌市西湖区站前西路289号三星大厦 |
| **s002291102** | 江铃汽车集团有限公司 | 15618660002 | 江西省南昌市解放西路226号 |
| **s002291103** | 正邦集团有限公司 | 15618660003 | 江西省南昌市青山湖区洪都中大道18号 |
| **s002291104** | 江西方大钢铁集团有限公司 | 15618660004 | 江西省南昌市青山湖区洪都中大道孙家亭路4号 |
| **s002291105** | 双胞胎（集团）股份有限公司 | 15618660005 | 江西省南昌市北京西路194号 |
| **s002291106** | 新余钢铁集团有限公司 | 15618660006 | 江西省南昌市省府北二路68号 |
| **s002291107** | 晶科能源有限公司 | 15618660007 | 江西省南昌市右营街19号 |
| **s002291108** | 江西省建工集团有限责任公司 | 15618660008 | 江西省南昌市东湖区沿江北大道39号 |
| **s002291109** | 江西博能实业集团有限公司 | 15618660009 | 江西省南昌市东湖区阳明路207号 |
| **s002291110** | 江西济民可信集团有限公司 | 15618660010 | 江西省南昌市五纬路382号 |

货物表实例数据如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cargo\_id** | cargo\_name | cargo\_quantity | cargo\_unit | supplier\_id |
| **2023070601** | 文件夹 | 100 | 个 | s002291101 |
| **2023070602** | 票据夹 | 30 | 个 | s002291102 |
| **2023070603** | 档案袋 | 30 | 个 | s002291103 |
| **2023070604** | 名片盒 | 100 | 个 | s002291104 |
| **2023070605** | 公事包 | 30 | 个 | s002291105 |
| **2023070606** | 拉链袋 | 30 | 个 | s002291106 |
| **2023070607** | 文件柜 | 100 | 个 | s002291107 |
| **2023070608** | 资料架 | 30 | 个 | s002291108 |
| **2023070609** | 图纸夹 | 30 | 个 | s002291109 |
| **2023070610** | 订书机 | 100 | 个 | s002291110 |

已售出货物实例数据如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **cargoSold\_id** | cargoSold\_name | cargoSold\_quantity | cargoSold\_unit | cargoSold\_place | user\_id |
| **2023070601** | 文件夹 | 2 | 个 | 北京 | u002291101 |
| **2023070602** | 票据夹 | 2 | 个 | 北京 | u002291102 |
| **2023070603** | 档案袋 | 2 | 个 | 北京 | u002291103 |
| **2023070604** | 名片盒 | 10 | 个 | 南京 | u002291104 |
| **2023070605** | 公事包 | 10 | 个 | 南京 | u002291105 |
| **2023070606** | 拉链袋 | 10 | 个 | 南京 | u002291106 |
| **2023070607** | 文件柜 | 5 | 个 | 天津 | u002291107 |
| **2023070608** | 资料架 | 5 | 个 | 天津 | u002291108 |
| **2023070609** | 图纸夹 | 5 | 个 | 天津 | u002291109 |
| **2023070610** | 订书机 | 40 | 个 | 哈尔滨 | u002291110 |
| **2023070611** | 文件夹 | 10 | 个 | 哈尔滨 | u002291101 |
| **2023070612** | 票据夹 | 10 | 个 | 哈尔滨 | u002291102 |
| **2023070613** | 档案袋 | 10 | 个 | 天津 | u002291103 |
| **2023070614** | 名片盒 | 5 | 个 | 天津 | u002291104 |
| **2023070615** | 公事包 | 5 | 个 | 天津 | u002291105 |
| **2023070616** | 拉链袋 | 5 | 个 | 天津 | u002291106 |
| **2023070617** | 文件柜 | 40 | 个 | 南京 | u002291107 |
| **2023070618** | 资料架 | 10 | 个 | 南京 | u002291108 |
| **2023070619** | 图纸夹 | 40 | 个 | 南京 | u002291109 |
| **2023070620** | 订书机 | 10 | 个 | 北京 | u002291110 |

四、系统结构及功能菜单设计

1．功能菜单设计

**图4.1 系统功能菜单结构图**

系统功能菜单结构图如上图所示，图中展示的功能均已实现。菜单的最大深度为5层，其中最复杂的功能模块是系统管理员登录的模块，普通管理员登录的子功能是系统管理员登录子功能的真子集。操作员注册分为系统和普通，两者的结构基本相同。颜色主题是非业务功能模块，通过简单的交互实现命令行样式的改变。

2．主要功能描述

**1）注册模块**

注册模块使用strcmp函数检查用户名是否已被注册。还要求密码是含有数字和字母的六位及以上的字符串，用strlen检查密码长度，用包含C库函数int isdigit(int c)、void isalpha(int c)的循环结构检查密码是否同时含有数字和字母。注册成功后，根据时间生成ID：用time函数获取当前日期和时间，使用localtime函数将now转换为结构体形式，使用sprintf函数将年、月、日和序号格式化成字符串形式，并将结果存储在用户结构体的id数组中。

**2）登录模块**

使用strcmp函数进行字符串处理，在已有的用户结构体中，匹配输入的登录名、密码。相同即账户验证成功，并进入相应的操作界面。不同则打印登陆失败。

**3）进货模块**

通过fileread()函数读取"采购清单.txt"的文件内容，并将其存储在字符数组buffer中。使用fopen()函数打开"采购清单.txt"文件，使用fseek()和ftell()函数获取文件的大小，根据文件大小动态分配字符数组buffer的内存空间，使用fgetc()函数逐个字符读取文件内容，并将其存储在buffer数组中。

循环判断：循环遍历货物数组cargos[]，如果某个货物的名称为空字符串，则执行以下操作：

a. 输出继续选择操作的菜单界面。

b. 接收用户输入的选择。

c. 判断用户选择的操作，如果选择进货（sel==1），则执行以下操作：

输入进货货物名称。

使用KMP算法进行匹配，判断该货物是否在采购清单中。

如果匹配成功，则继续输入进货货物数量和单位，并更新货物的数量。

如果匹配失败，则输出错误信息。

d. 如果选择退出进货功能（sel==2），则返回到管理员菜单。

**4）退货模块**

使用kmp算法对用户输入的货物名进行验证，查看仓库中是否有该商品。如果输入不在范围内，则要求用户重新输入。使用strcmp函数验证用户输入的退货销售单号、货货物名称是否正确。如果退货单的货物名与输入货物名相等，程序会要求用户输入退货货物数量，并进行判断：如果退货数量超出销售数量，则退货失败；如果库存不足，则退货失败；如果库存充足，则更新库存数量并显示退货成功。最后，调用filesave函数保存数据，并返回到管理员菜单界面。

**5）查找货物模块**

调用fileread()函数读取“货物信息.txt”。用户可以通过货物名称、货物编号进行查找，使用KMP算法在货物数组中查找匹配的货物。用户输入选择的操作，并进行错误检查，然后打印相应的货物信息。

**6）商品销售模块**

商品销售模块调用fileread()函数读取文件中的数据，打印销售功能的欢迎界面和可售货物列表，用户选择操作：填写销售记录或退出销售功能。

如果用户选择填写销售记录，则需要输入售出货物的名称、数量、地址、编号和职工ID。用strcmp函数匹配货物名，在货物列表中查找对应的货物，并判断库存是否足够；验证库存是否充足，如果库存不足，则提示销售失败；如果库存足够，则更新货物的库存数量，并将销售记录保存到“销售记录.txt”中。

**7）显示库存模块**

显示库存可以选择默认排序、按货物ID显示排序和按库存数量显示排序。

以按ID显示排序为例，使用一个临时数组temp[]来存储需要排序的货物信息，并统计有效货物的数量count和总库存数量sum。然后将原始货物数组cargos[]中的非空货物复制到临时数组temp[]中，并累加库存数量到总和sum。使用冒泡排序算法对临时数组temp[]按照货物编号进行升序排序。最后，按照一定格式输出排序后的货物信息。

**8）显示操作者信息、供应商信息、商品销量模块**

这些模块使用循环结构展示了相应的结构体数组信息，不涉及较为复杂的算法和控制流程，这些功能模块的具体细节将在第五部分详细阐述。

五、功能模块实现函数描述

1．系统文件清单

**1）源程序文件**

仓库管理系统.c

源程序文件“仓库管理系统.c”包含了仓库管理系统中所有头文件的引用和测试函数，并在该程序中导入测试数据实现了本仓库管理系统的主要功能，包括系统管理员、普通操作员、货物、销售记录、供应商、等一系列数据文件的打开、写入和关闭。

**2）头文件**

颜色主题.h

结构体.h

文件读取.h

文件保存.h

显示库存.h

货物调拨.h

查找货物.h

进货.h

退货.h

销售记录.h

操作者管理.h

供应商管理.h

系统管理界面.h

普通操作界面.h

管理员注册.h

管理员登录.h

操作员注册.h

操作员登录.h

主菜单.h

“颜色主题.h”该头文件定义了不同颜色主题黑色、白色、蓝色、黄色、绿色、紫色、灰色、红色八种不同颜色的主题函数，这些函数在设置颜色主题后会调用system("pause")函数以暂停程序的执行，并调用menu()函数在用户按下任意键后返回到主菜单，还有一个名为 color\_style 的函数，它显示了一个欢迎界面，并提供用户选择不同颜色主题的选项。用户可以通过控制台命令 color 来实现颜色的切换。并且使用了 #pragma once 来确保头文件只被包含一次，这是一种头文件保护方式，可以避免重复包含导致的编译错误。

"结构体.h"首先定义了三个宏常量：M、N 和 P，它们的值都为 100，可以灵活地控制数组的大小。然后定义了以下几个结构体，每个结构体都有不同的成员变量。struct admin：表示管理员信息，包括管理员ID、账号、密码和电话号码。定义了一个名为 admins 的数组，长度为 M，用来存储管理员的信息。struct user：表示用户信息，包括用户ID、账号、密码和电话号码。定义了一个名为 users 的数组，长度为 N，用来存储用户的信息。struct cargo：表示货物信息，包括货物ID、名称、数量、单位和供应商ID。定义了一个名为 cargos 的数组，长度为 P，用来存储货物的信息。struct cargoSold：表示已售出的货物信息，包括货物ID、名称、数量、单位、交易地点和用户ID。定义了一个cargoSolds 的数组，长度为 P，用来存储已售出的货物信息struct supplier：表示供应商信息，包括供应商ID、名称、电话号码和地址。定义了一个名为 suppliers 的数组，长度为 P，用来存储供应商的信息。

"文件读取.h"定义了一个名为 fileread 的函数，用于读取文件中的数据并将其存储到各个结构体数组中。首先，通过 fopen 函数以只读模式打开名为 "系统管理员.txt" 的文件，并将返回的文件指针赋值给 fp。接着，通过检查文件指针是否为 NULL 来判断文件是否成功打开。如果打开失败，会输出错误信息并使用 exit 函数退出程序。然后，使用 fread 函数从文件中读取结构体 admin 的数据，每次读取 sizeof(struct admin) 大小的数据，最多读取 M 个元素，从而将文件中的管理员信息存储到 admins 数组中。最后，使用 fclose 函数关闭文件指针 fp。接下来，类似地，依次打开并读取名为 "普通操作员.txt"、"供应商.txt"、"货物.txt" 和 "销售记录.txt" 的文件，并将对应的数据读取到相应的结构体数组中。最后，输出一条成功读取文件的提示信息。

"文件保存.h"与文件读取头文件类似，这段代码定义了一个名为 filesave 的函数，用于将数据保存到文件中。首先，通过 fopen 函数以写入模式打开名为 "系统管理员.txt" 的文件，并将返回的文件指针赋值给 fp。接着，通过检查文件指针是否为 NULL 来判断文件是否成功打开。如果打开失败，会输出错误信息并使用 exit 函数退出程序。然后，使用 fwrite 函数将 admins 数组中的管理员数据写入到文件中。每次写入 sizeof(struct admin) 大小的数据，共写入 M 个元素。最后，使用 fclose 函数关闭文件指针 fp。

类似地，依次打开名为 "普通操作员.txt"、"供应商.txt"、"货物.txt" 和 "销售记录.txt" 的文件，并将对应的结构体数组中的数据写入到文件中。最后，输出一条信息保存成功的提示信息，并使用 system("pause") 暂停程序的执行，等待用户按下任意键继续。

"显示库存.h"定义了三个函数：defaultSort()、quantitySort()和idSort()。defaultSort()函数用于按照货物编号对货物列表进行排序并显示排序后的列表。它首先通过循环遍历数组 cargos[] ，将非空的货物信息存入临时数组 temp[] 中，并统计货物数量与库存总和。然后使用冒泡排序对 temp[] 进行按照货物编号升序排序，最后使用循环打印排序后的列表及统计信息。quantitySort()函数用于按照货物库存数量对货物列表进行排序并显示排序后的列表。它与 defaultSort() 类似，不同之处在于排序的标准是货物的库存数量而不是货物编号。而idSort()函数是根据货物编号（cargo ID）对货物清单进行升序排序，并显示排序后的列表。

"货物调拨.h"用于显示当前可销售货物列表的内容，仓库管理员可以在库存允许的情况下对货物进行调拨并填写销售记录，当库存不足或货物不存在时系统会显示错误提示。函数void sold\_cargos()调用了fileread()函数用于文件读取，调用filesave()函数用于文件保存。

"查找货物.h"定义了函数void search\_cargo()，用于用户根据货名或者货物号对仓库内的货物信息进行查找。使用了KMP算法查找函数，函数原型为int kmpSearch(char text[], char pattern[]) 。首先计算模式字符串的部分匹配值，保存在next数组中。初始化指针i和j，分别指向文本字符串和模式字符串的起始位置。在循环中，依次比较text[i]和pattern[j]：如果匹配成功，则将指针i和j向后移动一位。如果匹配失败，根据next[j]的值更新模式字符串的位置j，实现部分匹配的跳转。重复之前的步骤，直到找到完全匹配的模式字符串或者遍历完整个文本字符串。如果找到了完全匹配的模式字符串，返回匹配的起始位置。如果没有找到匹配的内容，返回-1。KMP算法通过预先计算模式字符串的部分匹配值，在匹配过程中避免无效的比较，提高了匹配的效率。

"进货.h"定义了computeLPSArray函数计算模式字符串的部分匹配表（Longest Proper Prefix which is also Suffix）。该函数使用了KMP算法中的核心思想，通过遍历模式字符串并比较字符，将每个位置的部分匹配值存储在lps数组中。

KMPSearch函数是KMP算法的实现，用于在文本字符串中查找模式字符串。它首先调用computeLPSArray函数计算模式字符串的部分匹配表，然后使用指针i和j遍历文本字符串和模式字符串进行比较，根据匹配成功与否以及指针移动的条件进行相应操作，最终返回匹配结果。

stock\_cargos函数是一个功能选项，用于进货管理。函数首先调用fileread函数读取采购清单文件。然后，打开文件，获取文件大小并动态分配字符数组buffer用于存储文件内容。接下来，通过逐字符读取文件内容，并将其存储在buffer数组中。最后，输出仓库管理系统的欢迎信息并进行进一步操作。

"退货.h"定义了一个名为return\_cargos()的函数。可以显示销售记录列表，包括销售单号、货物名称、售出数量、销售单位、售往地址和售货员ID。用户选择操作，可以是退货操作（1）或退出退货功能（2）。如果用户选择退货（1），用户输入退货销售单号。根据输入的销售单号，代码搜索相应的销售记录。如果找到匹配的销售记录，标记flag设置为1，并保存匹配的索引。如果销售记录存在，用户输入退货货物名称。根据输入的货物名称搜索仓库中是否有该商品。如果找到匹配的货物记录，保存匹配的索引。比较销售记录和退货记录的货物名称。如果相等，说明退货单的货物名与输入的货物名匹配。如果仓库中存在该商品，要求用户输入退货货物数量。检查退货数量是否超出销售数量，如果超出则采取相应的处理步骤，否则处理退货的商品并更新相关数据结构或变量。如果仓库中不存在请求的商品（j = -1），或者退货单的货物名称与销售记录中的货物名称不匹配、销售单号未找到匹配的销售记录等不符合要求的情况，采取相应的处理步骤。

"销售记录.h"定义了函数void salesrecords()，显示当前仓库的销售记录列表。它通过遍历销售记录数组并逐行打印每条记录的相关信息来实现这一功能。清除控制台屏幕、打印欢迎信息和销售记录查询功能标题，然后打印销售记录列表的表头。接下来，它会遍历销售记录数组，并检查每个记录的货物名称是否为空字符串。如果不为空，代码会打印记录的编号、货物名称、售出数量、销售单位、售往地址和售货员ID。循环结束后，代码会打印销售记录列表的表尾，并暂停程序的执行，等待用户按任意键继续。

"操作者管理.h"定义了函数void show\_user()主要功能是显示当前仓库的销售记录列表。它通过遍历销售记录数组并逐行打印每条记录的相关信息来实现这一功能。它会检查每个记录的货物名称是否为空字符串。如果不为空，会打印记录的编号、货物名称、售出数量、销售单位、售往地址和售货员ID。

"供应商管理.h"定义了函数void show\_supplier()，它的功能是显示仓库中普通操作员的列表。代码通过循环遍历操作员数组，并检查每个操作员的用户名是否为空字符串。如果不为空，代码会打印操作员的编号、用户名、手机号和密码。如果用户名为空字符串，则结束循环。

"系统管理界面.h"定义了一个名为admin\_menu的管理员菜单函数，它提供了以下操作选项：显示当前库存：调用printCargo()函数来显示当前的库存情况。商品销售：调用sold\_cargos()函数来进行商品销售操作。进货：调用stock\_cargos()函数来进行商品进货操作。退货：调用return\_cargos()函数来进行商品退货操作。查询商品销量：调用salesrecords()函数来查询商品的销售记录。显示操作者信息：调用show\_user()函数来显示操作者的信息。显示供应商信息：调用show\_supplier()函数来显示供应商的信息。查找货物：调用search\_cargo()函数来查找指定的货物。注销登录：调用menu()函数返回到主菜单。

"普通操作界面.h"定义了一个普通操作员菜单函数，它提供了以下操作选项：显示当前库存：调用printCargo()函数来显示当前的库存情况。查找货物：调用search\_cargo()函数来查找指定的货物。注销登录：调用menu()函数返回到主菜单。

"管理员注册.h"定义了管理员注册函数。其主要功能是让管理员进行注册操作，包括填写注册信息并完成注册。在注册流程中，要求管理员填写注册信息，包括用户名、手机号、以及要求包含数字和字母的六位及以上密码。

验证填写的用户名是否已被注册过，如果是，则要求重新填写用户名和其他信息。在存储管理员信息的数组中找到一个空闲位置，将新的管理员信息保存到该位置。对填写的密码进行长度和要求的字符类型（数字和字母）验证，如果不符合要求，则要求重新填写密码和其他信息。注册成功后，生成管理员的ID，包括当前日期和时间信息。

"管理员登录.h"定义了admin\_login函数用于验证管理员登陆。代码将用户输入的用户名和密码与存储在admins数组中的管理员凭据进行比较。如果找到匹配项，将flag设置为1，显示登录成功的消息，使用Sleep函数等待片刻，然后调用admin\_menu函数。如果没有找到匹配项，显示错误消息，提示用户选择重新输入登录信息或返回上一级菜单。验证用户的选择，循环直到输入有效的选择，并根据选择采取适当的操作。完成登录过程后，调用filesave函数将相关数据保存到文件中。

"操作员注册.h"定义了一个用于用户注册的函数 user\_registers。代码会遍历已注册的用户数组users，检查新输入的用户名是否已被注册。如果用户名已被注册，显示错误消息，提示用户重新填写注册信息。如果用户名可用，继续注册流程。完成注册后，返回到主界面。

"操作员登录.h"用于实现仓库管理系统中的操作员登录功能。用户输入登录信息，包括用户名和密码。然后通过循环遍历已注册的用户列表，检查输入的用户名和密码是否与列表中的某个用户匹配。如果匹配成功，将标志位flag设为1，并调用user\_menu()函数显示操作员菜单。如果标志位flag仍为0，则表示输入的用户名或密码有误。根据用户选择的操作，可以重新输入登录信息或返回主菜单。最后，函数调用filesave()函数保存文件。

"主菜单.h"定义了一个菜单函数，实现了一个简单的仓库管理系统的菜单界面和功能选择。让用户可以通过输入选择不同的功能：登录、注册、选择颜色主题等。根据用户的选择，调用相应的函数进行处理。

**3）数据文件**

系统管理员.txt：记录系统管理员的用户名和密码。

普通操作员.txt：记录普通操作员的信息，包括用户名、密码等。

货物.txt：记录货物编号、货物名称、库存数量、单位等信息。

销售记录.txt：记录货物的销售记录的信息，包括编号、货物名称、售出数量、销售单位、售往地址、售货员ID等信息。

供应商.txt：记录供应商的信息，包括供应商编号、供应商名称、地址、电话。

采购清单.txt：记录采购清单信息，包括货物名称。

2．函数原型及功能描述

源程序文件“仓库管理系统.c”中：

原型函数为void testData()的函数用于导入测试数据，包括写入系统管理员、普通操作员、货物、销售记录、供应商数据文件。在主函数中调用该函数进行测试。

原型函数为int menu()的函数用于显示一个菜单并接收用户的选择。步骤如下：system("cls")清屏，清除之前的输出。打印欢迎信息和菜单的选项。使用scanf函数读取用户的选择，并存储在变量sel中。使用getchar()函数读取并处理输入缓冲区中的换行符。在while循环中，检查用户的选择是否合法（范围为0-5）。如果选择不合法，再次要求用户输入正确的选择。使用switch语句根据用户的选择执行相应的操作：

如果选择为1，调用admin\_login()函数，该函数用于系统管理员登录。

如果选择为2，调用user\_login()函数，该函数用于普通操作员登录。

如果选择为3，调用admin\_registers()函数，该函数用于系统管理员注册，然后通过调用menu()函数继续显示菜单。

如果选择为4，调用user\_registers()函数，该函数用于用户注册，然后通过调用menu()函数继续显示菜单。

如果选择为5，调用color\_style()函数，该函数用于颜色主题的选择，然后通过调用menu()函数继续显示菜单。

原型函数为void fileread()的函数在"文件读取.h"中定义，用于文件的打开、读取和关闭。可以打开五个文件（“系统管理员.txt”、“普通操作员.txt”、“供应商.txt”、“货物.txt”、“销售记录.txt”），分别用于读取系统管理员、普通操作员、供应商、货物和销售记录的数据。

原型函数为void filesave()的函数在"文件保存.h"中定义，在主函数中被调用。该函数的功能是将程序中定义的结构体数组（例如admin,user,cargo,supplier,cargoSold等）的数据保存到对应的文件中。每个结构体数组的数据项数量根据代码中的常量M，N，P等指定。

六、总结

完成仓库管理系统项目的过程不是一帆风顺的，我遇到了许多问题，也收获到了解决这些问题的能力，借此总结我的一些感受和收获：

熟悉软件开发流程：在项目中，我需要进行需求分析、设计、编码和测试等多个阶段。这使我对软件开发流程有了更清晰的认识，并学会了如何合理地组织代码和进行调试。

提高问题解决能力：在项目开发过程中，我遇到了各种各样的问题，例如逻辑错误、编译错误和运行时错误等。通过不断调试和查找解决方案，我学会了快速定位问题并解决它们的能力。

加强团队合作意识：在项目中，我与同学一起合作完成任务，我们相互协作、交流和分享经验。这锻炼了我的团队合作意识和沟通能力，也让我明白了团队合作的重要性。

然而，我也意识到项目中存在一些不足和有待改进的问题：

缺乏完善的错误处理机制：在项目中，我没有充分考虑各种异常情况的处理，导致程序在遇到错误时可能会崩溃或产生不可预料的结果。这是一个需要进一步解决的问题，可以通过增加错误处理代码来提高系统的稳定性和健壮性。

用户界面设计有待改进：目前仓库管理系统的用户界面比较简单，缺乏友好的交互和良好的用户体验。在未来的改进中，可以考虑优化界面设计，使用户能够更方便地操作系统。

数据安全性需要加强：在当前的系统中，对于用户信息和数据的安全性保护还不够完善。为了防止数据泄露和非法访问，可以加强密码加密、权限控制等方面的安全措施。

综上所述，通过完成仓库管理系统项目，我不仅学到了很多关于C语言和软件开发的知识，还提高了问题解决能力和团队合作意识。同时，我也认识到项目中存在的不足和需要改进的地方，希望在今后的学习和实践中能够进一步完善和提升这个仓库管理系统。

附件：程序清单

源文件：

文件仓库管理系统.c

头文件：

操作员登录.h

操作员注册.h

操作者管理.h

查找货物.h

供应商管理.h

管理员登录.h

管理员注册.h

货物调拨.h

结构体.h

进货.h

普通操作界面.h

退货.h

文件保存.h

文件读取.h

系统管理界面.h

显示库存.h

销售记录.h

颜色主题.h

主菜单.h

数据文件：

采购清单.txt

供应商.txt

货物.txt

普通操作员.txt

系统管理员.txt

销售记录.txt