



课程设计报告

课设课题： 航空客运订票系统

学 院： 智慧教育学院

专 业： 软件工程

姓 名： 黄暄 王锋 邓浩宇 许嘉诚 陈记德

班级学号： 239007045-38-37-36-49

指导教师： 王树梅

报告日期： 2024-6-19

**目 录**

[一、课题分析](#_Toc342981375)------------------------------------------------------------------1

[1.1 课题的背景和意义](#_Toc342981376)------------------------------------------------------1

1.2 设计题目与要求--------------------------------------------------------1

[二、总体结构设计](#_Toc342981378)--------------------------------------------------------------2

[三、各子模块设计](#_Toc342981379)--------------------------------------------------------------11

[四、程序设计调试情况分析](#_Toc342981388)------------------------------------------------------35

[五、测试结果](#_Toc342981389)------------------------------------------------------------------36

[六、总结](#_Toc342981400)----------------------------------------------------------------------45

[七、附录(源代码)](#_Toc342981402)--------------------------------------------------------------46

# 一、课题分析

## 1.1 课题的背景和意义

**背景：**航空客运订票系统作为航空运输业的重要组成部分，其高效、稳定、便捷的性能对提升航空公司的服务质量、提高客户满意度具有重要意义。然而，传统的航空订票系统往往存在功能繁琐等问题。因此，开发一款基于面向对象思想、功能全面、操作简便、安全可靠的航空客运订票系统显得尤为必要，故选择本课题为研究方向。

**意义：**本课题需要学生运用面向对象思想进行系统设计并实现具体功能，这一过程能够让学生将理论知识与实践相结合，提高他们的动手能力和分析问题、解决问题的能力。同时可以培养团队协作精神，在设计实现航空客运订票系统的过程中，学生需要分工合作、共同完成项目任务，从而培养他们的团队协作精神、沟通能力以及项目管理能力。

## 1.2 设计题目与要求

【问题描述】

实现航空客运订票的主要业务活动，包括浏览和查询航班信息、机票预订和办理退票等。

【设计要求】

（1）航班管理。每条航班所设计的信息有：终点站名、航班号、飞机型号、飞行周日（星期几）、乘员定额、余票量。

（2）客户管理。有关订票的客户信息（姓名、订票量、舱位等级（1,2））以及等候替补的客户名单（姓名、所需票量）。

（3）查询航班。根据旅客提出的终点站名输出航班号、飞机型号3、飞行日期、余票数。

（4）订票业务。根据客户提出的要求（航班号、订票数量）查询该航班票额情况，若有余篇，则为客户办理订票手续，输出座位号；若已满或者余票少于订票数，则需要重新询问客户要求；若需要，可等待排队候补。

（5）退票业务。根据客户提供的情况（日期、航班），为客户办理退票手续，然后查询该航班是否有人排队等候。

**【基本要求】**

1. 每条航线所设计的信息有：航班ID、起点站名、终点站名、飞机号、飞行周日（星期几）、成员定额、总余票量、经济仓余票量、商务舱余票量、已订票的客户名单、经济舱候补客户名单、商务舱候补客户名单。

其中，经济舱候补客户名单和商务舱候补客户名单中的包括以下几个域：姓名、身份证、所需票量、预定票的等级

1. 作为示意系统，数据放在内存中，以链表形式；
2. 系统能实现的操作和功能如下：
3. 查看所有航班信息。
4. 查询模块：输入起点站和终点站进行搜索。
5. 订票模块的功能：

A：输入航班ID进行订票，当航班ID不存在或者不合法时要求重新输入航班ID，然后输入订票数量和等级，数量和等级要合法，订票数目要大于0，等级输入1代表经济舱，输入2代表商务舱。不合法则重新输入，再判断对应等级的票是否足够 ，足够则继续输入客户信息（姓名和身份证），然后订票成功。当对应等级的票数不足时，询问是否改变订票计划，同意改变计划则重新输入航班ID进行相应的订票操作。不同意改变计划则继续询问是否排队等候，不同意排队等候则询问是否需要推荐相同航线的其他航班，询问操作时，输入Y或者y表示同意，输入N，n或者其它数据表示不同意。

B：根据起点和终点查询航班

C：查询所有航班的相关信息

D：返回上一级菜单

1. 退票模块的功能：

A:先输入航班ID，然后输入姓名和身份证号码进行验证，验证成功则办理退票手续。

B：返回上一级菜单

1. 管理员模块的功能：

A：管理员密码验证

B：查询所有航班信息

C：根据航班ID查询客户信息

D：增加航班（航班的各个信息输入要合法，不合法的重新输入）

E：删除航班，根据ID进行删除，ID不存在则删除失败。

F：退出系统

# 二、总体结构设计

**概要设计：**

**1.所要用到的头文件**

#include<iostream>

#include<string>

#include<windows.h>

#include <algorithm>

#include<bits/stdc++.h>

**2.类型定义**//乘客节点

typedef struct CNode

{

char name[20];//姓名

int num;//订票数

char ID[20];//身份证

char day[20];

int dengji;//舱等级

CNode \*next;

} CNode, \*CList;//乘客节点

//候补队列中的节点

typedef struct HbNode

{

char name[20];//姓名

char ID[20]; //身份证

int num;//预定数量

struct HbNode \*next;

} QhbNode, \*Phb;

//候补队列

typedef struct Queue

{

Phb front;//等候替补客户名单域的头指针

Phb rear;//等候替补客户名单域的尾指针

} LinkQueue;

//航班节点

typedef struct hangban

{

char qidian[20];//起点站名

char zhongdian[20];//终点站名

char hbID[20];//航班编号

char hbnum[20];//飞机号

char day[20];//飞行日期（星期几）

int maxnum;//总票数

int less;//总余票量

int lessgrade1; //等级1剩余量

int lessgrade2; //等级2剩余量

hangban \*next;

CList cusLinkList;//乘员名单，指向乘员名单链表的头指针

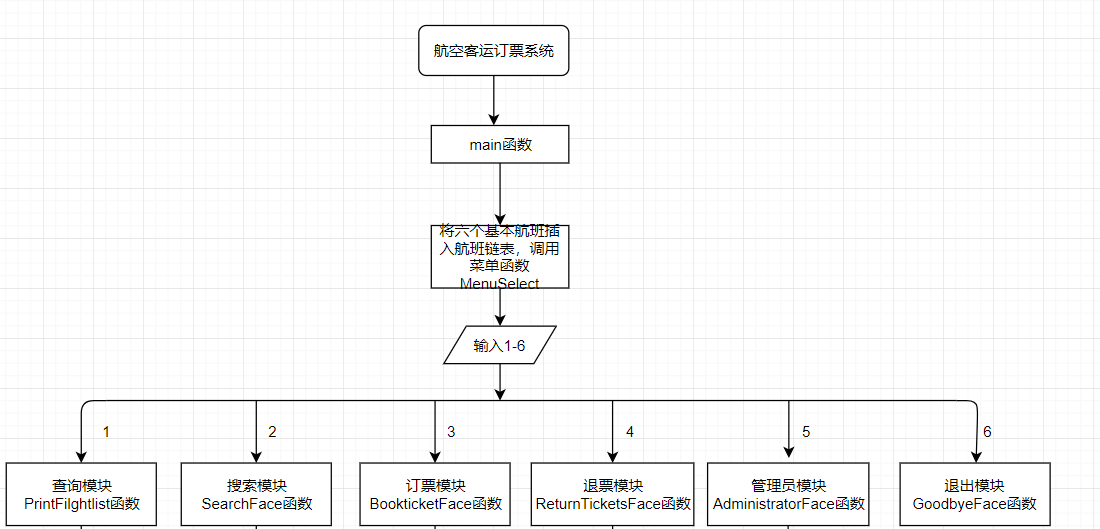
LinkQueue wQgrade1;//等级1候补，等候替补的客户名单域，指向一个队列

LinkQueue wQgrade2;//等级2候补，等候替补的客户名单域，指向一个队列

} hangban, \*Phangban;//航班节点

航班链表采用全局变量：pFlight，指向航班链表的头指针

**4.基本的模块调用关系图：**



**5.函数接口设计：**

ps：

函数的接口采用多行注释

@param表示该函数的参数

@return 表示函数的返回值

/\*\* \*根据自己输入的航班ID查询并以指针形式返回 \* @return 航班指针 \*/hangban \*find();/\*\* \* 初始化带头结点的链队列 \* @param q

\*/void initFlight();/\*\* \* 初始化已订票乘客指针链表 \* @param cusLinkList 航班中乘员链表的头指针 \* @return操作状态

\*/int initCusLinkList(CList &cusLinkList);/\*\* \* 打印info航班的基本信息 \* @param info \*/void fightinfo(hangban \*p);

/\*\* \*根据客户提出的起点和终点站名输出航线信息 \*/void sousuoflight();

/\*\* \* 打印订票乘员名单域的客户名单信息 \*/void pricusInfo();/\*\* \* 订票功能模块 \*/void dingpiao();

/\*\* \* 退票功能模块 \*/void tuipiao();/\*\* \* 搜索界面 \*/void sousuoFace();/\*\* \* 订票界面 \* 属于订票模块 \*/void dingpiaoFace();/\*\* \* 退票模块界面 \*/void tuipiaoFace();

/\*\* \* 查询模块 \* 打印全部航班信息 \* @param flight 传入的是航班链表的头指针 \*/void printfall(hangban \*pflight);

/\*\* \*菜单界面函数 \* @return 返回选择的操作 \*/int menu();// 菜单界面函数

/\*\* \* 管理员界面 \*/int guanliyuanyemian();/\*\* \* 将flight1的4个航班用头插入法插入到pFlight的链表中 \* @param flight1 里面存有六个基本航班 \* @return 返回操作是否成功 \*/int Create(Phangban flight1);/\*\* \* 管理员插入航班时遍历航班，防止航班ID重复 \* @param flight \* @param flightCodeID \* @return 返回是否重复 \*/int duibiFlight(hangban \*flight, char ID[])

/\*\* \* 登录页面

\* @return操作状态

\*/int denglu();

/\*\* \* 增加航班

\*/int zengjiaFlight();

/\*\* \* 删除航班

\*/int shanchuFlight();

/\*\* \* 初始化候补队列 \* @param linkQueue \* @return 是否成功

\*/int initQueue(LinkQueue &q);

/\*\* \* //将订票成功的乘客节点加入到已订票链表 \* @return 返回是否重复 \*/void addlink(CList &head, int need, char name[], char id[],char day[], int grade);

/\*\* \* 用户登录页面

\* @return操作状态

\*/int yonghudenglu();

/\*\* \* 管理员登录页面

\* @return操作状态

\*/int guanliyuandenglu();

**6.main函数**

int main()

{

system("color F1");

initFlight();//初始化航班链表

Create(flight1);//将航标信息插入链表

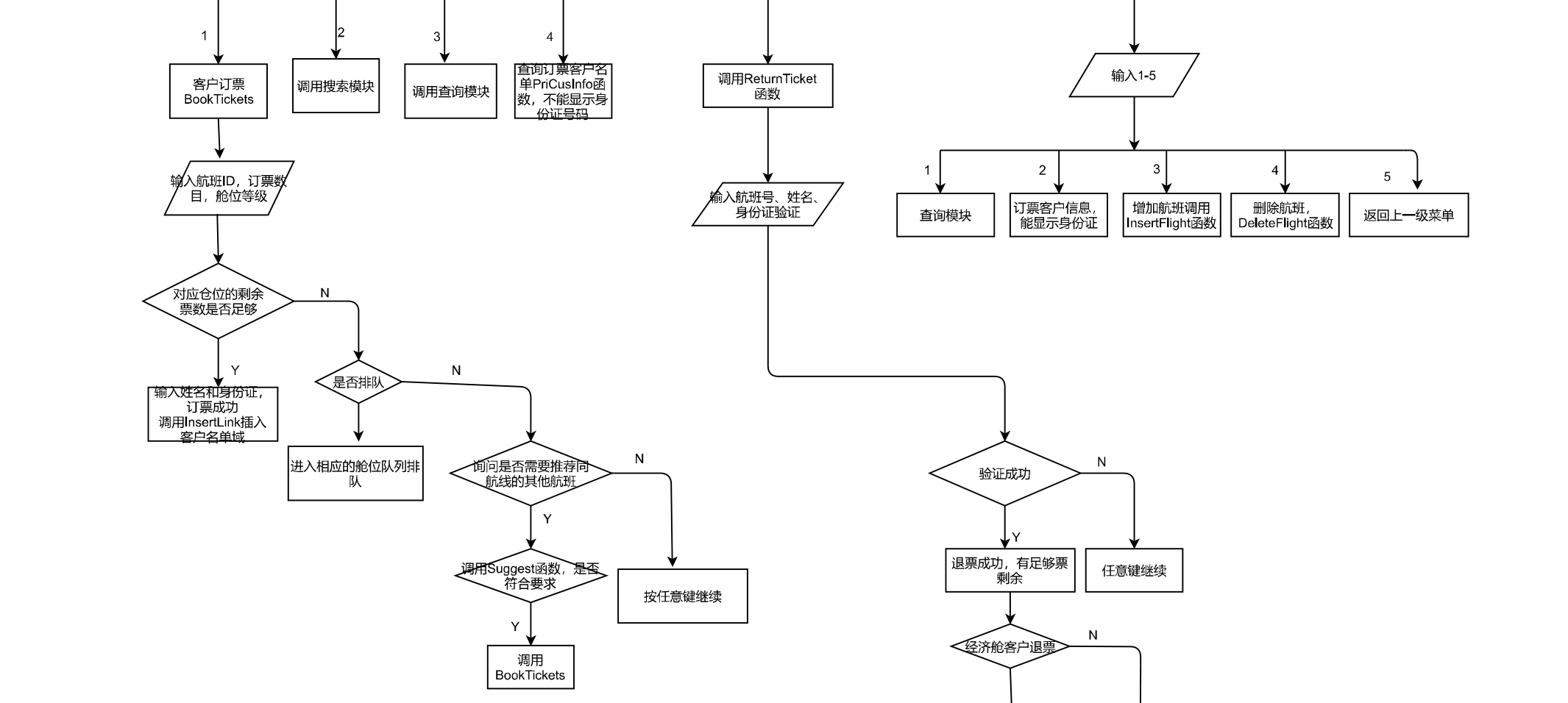
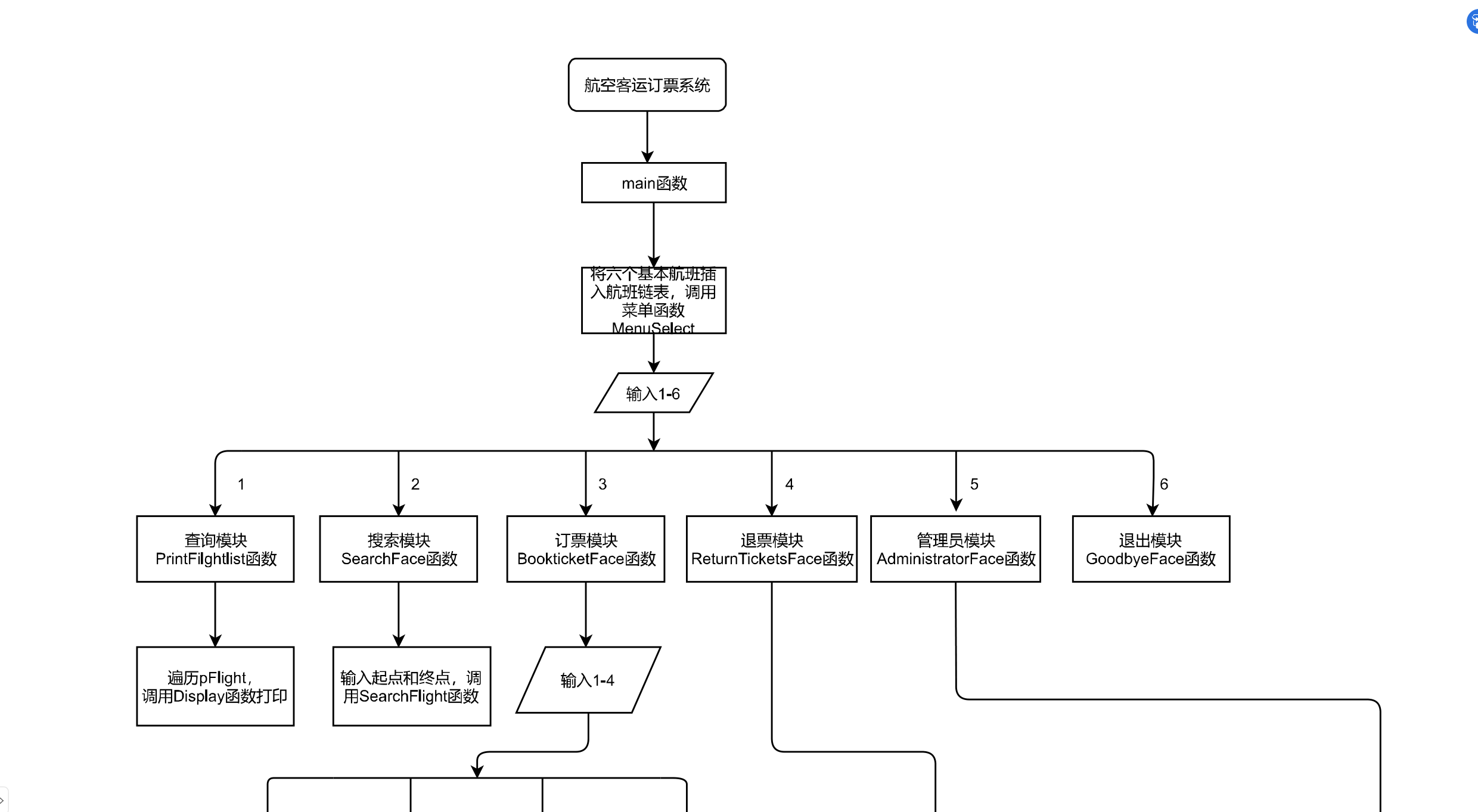
denglu(); //菜单

return 0;

}

# 三、各子模块设计

**函数之间的调用关系，如下图所示（如果看不清楚可查看pdf流程图文档）：**

****

**各模块部分设计：**

//定义全局指针变量pFlight，航班链表的头指针

hangban \*pFlight;//全局节点变量

//初始输入的航班信息

hangban flight1[4] = {

{"徐州", "南京", "1", "A1185", "星期五", 400, 200, 100, 100},

{"苏州", "淮安", "2", "B4521", "星期一", 300, 200, 100, 100},

{"北京", "上海", "3", "C8541", "星期六", 500, 200, 100, 300},

{"扬州", "镇江", "4", "D3652", "星期三", 600, 200, 100, 400},

};

/\*\*

\* 初始化航班链表

\* 目的是要初始化移pFlight为头结点的空航班链表,录入航班信息和增加航班后将航班结点插入该链表

\*/

void initFlight()

{

pFlight = new hangban;

pFlight->next = NULL;

}

/\*\*

\* 将flight1的4个航班用头插入法插入到pFlight的链表中

\* @param flight1 里面存有六个基本航班

\* @return 返回操作是否成功

\*/

int Create(Phangban flight1)

{

hangban \*p = pFlight, \*q;

for (int i=0; i < 4; i++) {

q = new hangban;

if (q == NULL)

return 0;

strcpy(q->qidian, flight1[i].qidian);//利用链表把每个信息录入

strcpy(q->zhongdian, flight1[i].zhongdian);

strcpy(q->hbID, flight1[i].hbID);

strcpy(q->hbnum, flight1[i].hbnum);

strcpy(q->day, flight1[i].day);

q->maxnum = flight1[i].maxnum;

q->less = flight1[i].maxnum;

q->lessgrade1 = flight1[i].lessgrade1;

q->lessgrade2 = flight1[i].maxnum - flight1[i].lessgrade1;

initCusLinkList(q->cusLinkList); //初始化已经订票乘客链表

initQueue(q->wQgrade1);// 将两种等级的票入队 ，方便后面预定排队

initQueue(q->wQgrade2);

q->next = p->next;

p->next = q;

}

return 1;

}

/\*\*

\* 管理员操作

\* 将新的航班结点插入到航班链表中，

\* 采用头插入法

\* @return 返回操作是否成功

\*/

int zengjiaFlight()

{

hangban \*q;

hangban \*p = pFlight;

int y = 1;

cout << "请依次输入以下内容" << endl;

system("pause");

while (y != 0)

{

q = new hangban;

if (q == NULL)

return 0;

cout << "请输入航班编号" << endl;

cin >> q->hbID;

int t = duibiFlight(pFlight, q->hbID);

if (t == 0)

{

cout << "该航班编号已经存在，请重新输入航班编号" << endl;

continue;

}

cout << "请输入起点站名" << endl;

cin >> q->qidian;

cout << "请输入终点站名" << endl;

cin >> q->zhongdian;

cout << "请输入型号" << endl;

cin >> q->hbnum;

cout << "请输入飞行日期" << endl;

cin >> q->day;

cout << "请输入乘客定额" << endl;

cin >> q->maxnum;

q->less = q->maxnum;

cout << "请输入1等票数目" <<endl;

cin >> q->lessgrade1;

q->lessgrade2 = q->maxnum - q->lessgrade1;

q->next = p->next;

p->next = q;

cout << "是否继续录入航班信息（任意数字继续，0表示停止）。\n";

cout << "请输入：";

cin >> y;

}

return 0;

}

/\*\*

\* 管理员操作

\* 删除节点

\* @return 返回操作是否成功

\*/

int shanchuFlight()

{

string ID;

cout << "请输入航班ID" << endl;

cin >> ID;

hangban \*pre = pFlight;

hangban \*p = pre->next;

while (p != NULL)

{

if (ID == p->hbID)

{

pre->next = p->next;

delete p;

return 1;

}

pre = p;

p = p->next;

}

return 0;

}

/\*\*

\* 管理员的查询客户信息操作

\*/

void pricusInfo()

{

CList p;

hangban \*info;

info = find(); // 假设find函数已定义，用于查询航班信息

if (info == NULL)

{

cout << "没有这个航班" << endl;

return;

}

// 头结点的下一个节点开始遍历

p = info->cusLinkList->next;

if (p != NULL)

{

cout << "客户姓名 订票数额 舱位等级 身份证号 " << endl;

while (p)

{

cout << p->name << "\t\t" << p->num << "\t" << p->dengji << "\t" <<p->ID<< endl;

p = p->next;

}

}

else

{

cout << "该航班没有客户信息!!" << endl;

}

}

/\*\*

\* 管理员模块

\* 管理员界面函数

\*/

int guanliyuanyemian()

{

int i;

do{

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"管理员页面"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 1.插入航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 2.删除航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 3.浏览航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 4.查询客户信息 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 5.返回首页 "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入您的选择(1-2,否则无效！)"<<endl;

cout<<" ";

cout<<"您的选择是:";

cin >> i;

switch (i) {

case 1:

zengjiaFlight();

guanliyuanyemian();

break;

break;

case 2:

if (1 == shanchuFlight())

{

cout << "删除成功" << endl;

}

else

{

cout << "没有这个航班，删除失败！" << endl;

}

system("pause");

guanliyuanyemian();

break;

case 3:

printfall(pFlight);

system("pause");

guanliyuanyemian();

break;

case 4:

pricusInfo();

system("pause");

guanliyuanyemian();

break;

case 5:

denglu();

break;

default:

guanliyuanyemian();

}

} while (true);

}

/\*\*

\*菜单界面函数

\* @return 返回选择的操作

\*/

int menu()

{

int i;

do {

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"欢迎来到航空客运订票系统"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" 1.查询航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 2.搜索航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 3.订票 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 4.退票 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 5.退出系统 "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入您的选择(1-5,否则无效！)"<<endl;

cout<<" ";

cout<<"您的选择是:";

cin >> i;

switch (i) {

case 1:

printfall(pFlight);

system("pause");

menu();

break;

case 2:

sousuoFace();

system("pause");

menu();

break;

case 3:

dingpiaoFace(); // 订票界面

system("pause");

menu();

break;

case 4:

tuipiaoFace();

system("pause");

menu();

break;

case 5:

denglu();

default:

menu();

}

} while (true);

}

/\*\*

\* 打印info航班的基本信息

\* @param info

\*/

void fightinfo(hangban \*p)

{

cout << "起点 终点 编号 飞机号 飞行日期 总载客 余票量 1级余量 2级余量" << endl;

cout << p->qidian << " " << p->zhongdian << " " << p->hbID << " " << p->hbnum << " " << p->day << " "

<< p->maxnum << " " << p->less << " " << p->lessgrade1 << " " << p->lessgrade2 << endl;

}

/\*\*

\* 查询模块

\* 打印全部航班信息

\* @param flight 传入的是航班链表的头指针

\*/

void printfall(hangban \*pflight)

{

hangban \*p;

p = pflight->next;

cout << "起点 终点 编号 飞机型号 飞行日期 总载客 余票量 1级余量 2级余量 " << endl;

while (p != NULL)

{

cout << p->qidian << " " << p->zhongdian << " " << p->hbID << " " << p->hbnum << " " << p->day << " "

<< p->maxnum << " " << p->less << " " << p->lessgrade1 << " " << p->lessgrade2 <<" "<< endl;

p = p->next;

}

cout << endl << endl;

}

/\*\*

\*根据客户提出的起点，终点站名输出航线信息

\*/

void sousuoflight()

{

string qidian, zhongdian;

int flag = 0;

cout << "请输入起点站名:";

cin >> qidian;

cout << "请输入终点站名:";

cin >> zhongdian;

hangban \*p = pFlight->next;

cout << "起点 终点 编号 飞机号 飞行日期 总载客 余票量 1级余量 2级余量" << endl;

while (p != NULL) {

if (qidian == p->qidian && zhongdian == p->zhongdian)

{

flag = 1;

cout << p->qidian << " " << p->zhongdian << " " << p->hbID << " " << p->hbnum << " " << p->day << " "

<< p->maxnum << " " << p->less << " " << p->lessgrade1 << " " << p->lessgrade2 << endl;

}

p = p->next;

}

cout <<endl <<endl;

if (flag == 0)

cout << "抱歉！不存在该航班！" << endl;

}

\* 管理员插入航班时遍历航班，防止航班ID重复

\* @param flight

\* @param char ID[]

\* @return 返回是否重复

\*/

int duibiFlight(hangban \*flight, char ID[])

//对比航班的编号，防止增加航班时，出现重复

{

hangban \*p = flight;

while (p != NULL)

{

if (!strcmp(ID, p->hbID))

{

return 0;//航班ID重复

}

p = p->next;

}

//ID不重复

return 1;

}

/\*\*

\*根据自己输入的航班ID查询并以指针形式返回

\* @return 航班指针

\*/

hangban \*find()

{

char number[10];

int i = 0;

cout<<"请输入航班编号"<<endl;

cin>>number;

hangban \*p = pFlight->next; //头结点的下一个节点开始遍历

while (p != NULL)

{

if (!strcmp(number, p->hbID))

return p;

p = p->next;

}

return NULL;

}

/\*\*

\* 订票成功之后，将乘客信息插入到对应航班的订成员名单域中（链表）

\* @param head 乘客名单域头指针

\* @param need 该乘客订票的数量

\* @param name 乘客的姓名

\* @param grade 订票的等级

\* @param id 身份证号码

\* @return 乘客链表头指针

\*/

void addlink(CList &head, int need, char name[], char id[],char day[],int grade)

//插入到已经订票客户链表

{

CList new1;//创建一个新节点

new1 = (CNode \*) malloc(sizeof(CNode));

strcpy(new1->name, name);

strcpy(new1->ID, id);

strcpy(new1->day,day);

new1->num = need;

new1->dengji = grade;

new1->next = head->next;//头插入法加入成员名单域

head->next = new1;

return;

}

/\*\*

\* 初始化已订票乘客指针链表

\* @param cusLinkList 航班中乘员链表的头指针

\* @return

\*/

int initCusLinkList(CList &cusLinkList)

{

CList q = cusLinkList;

cusLinkList = (CNode \*) malloc(sizeof(CNode));

cusLinkList->next = NULL;

}

/\*\*

\* 初始化带头结点的链队列

\* @param q

\*/

int initQueue(LinkQueue &q)

{

QhbNode \*p;

p = new QhbNode;

p->next = NULL;

q.front = q.rear = p;//队空

return 1;

}

/\*\*

\*入队，增加排队等候的客户名单域

\* @param q 带头结点的链队列

\* @param name

\* @param need

\*@param id

\* @return

\*/

LinkQueue enqueue(LinkQueue &q, char name[], int need, char id[])

//入队，增加排队等候的客户名单域

{

Phb new1;

new1 = (Phb) malloc(sizeof(QhbNode));

strcpy(new1->name, name);

strcpy(new1->ID, id);

new1->num = need;

new1->next = NULL;

q.rear->next = new1;

q.rear = new1;

return q;

}

/\*\*

\* 订票模块

\*/

void dingpiao()

{

hangban \*info;

int need, grade;

int num;//剩余的等级1数目或者等级2数目

char name[20];

char id[20];

char day[20];

info = find();//输入想订的航班，看是否还有空位置，并返回复制节点

// 初始化随机数种子

srand(time(0));

int seatCount = 0; // 已分配座位的数量

if (info == NULL)

{

cout<<"不存在该航班"<<endl;

dingpiao();

}

cout<<"请输入您需要的票数"<<endl;

while(1)

{

cin>>need;

if(need<=0)

{

cout<<"请输入正确的订票数"<<endl;

continue;

}

else break;

}

cout<<"请输入您需要的座位等级"<<endl;

while(1)

{

cin>>grade;

if(grade!=1&&grade!=2)

{

cout<<"请输入正确的座位舱等级"<<endl;

continue;

}

else break;

}

if (grade == 1)

num = info->lessgrade1;

else

num = info->lessgrade2;

if (need <= num) //订票数小于剩余数

{

vector<int> allocatedSeats;

int i;

cout<<"请输入您的名字"<<endl;

cin>>name;

cout << "请输入您的身份证号码: ";

while (true)

{

cin>>id;

if (!isValidID(id))

{

cout << "身份证号码格式不正确，请输入18位的数字: ";

continue;

}

else

{

break;

}

}

cout<<"请输入你的飞行时间："<<endl;

cin>>day;

CList head = info->cusLinkList;

addlink(head, need, name, id,day,grade);//订票成功，插入成员名单链表

// 从1开始，因为通常座位号从1开始编号

int nextSeatNumber = 1;

int seatStatus[info->maxnum] = {0}; // 0表示座位未被占用，1表示已被占用

// 循环分配座位号

for (int i = 0; i < need; ++i)

{

int seatNumber;

do {

seatNumber = rand() % info->maxnum + 1;

} while (seatStatus[seatNumber - 1]); // 检查座位号是否已被占用

seatStatus[seatNumber - 1] = 1; // 标记座位号为已占用

cout << name << "的座位号是" << seatNumber << endl;

}

info->less -= need;

if (grade == 1)

info->lessgrade1 -= need;

else

info->lessgrade2 -= need;

cout<<"订票成功"<<endl;

}

else

{

char r;

cout<<"该等级的票不足，以下为该航班乘客信息"<<endl;

fightinfo(info);//输出该航班的订票信息

cout<<"是否改变订票计划？Y/N"<<endl;

cin>>r;

if (r == 'Y' || r == 'y') //改变计划，重新选择航班

{

dingpiao();//返回订票主页面

}

else

{

cout<<"您需要排队等候吗?(Y/N)"<<endl;

cin>>r;

if (r == 'Y' || r == 'y')

{

//不改变计划，排队候票

cout<<"请输入您的姓名（排队订票）:"<<endl;

cin>>name;

cout<<"请输入您的身份证"<<endl;

cin>>id;

if (grade == 1) //进入1等舱排队队列

{

info->wQgrade1 = enqueue(info->wQgrade1, name, need, id);

}

else //进入2等舱排队队列

{

info->wQgrade2 = enqueue(info->wQgrade2, name, need, id);

}

cout<<"排队成功"<<endl;

}

}

}

}

/\*

\* 退票功能模块

\*/

void tuipiao() //退票功能

{

hangban \*info;

int grade;

CNode \*p1, \*p2, \*head;//p1为遍历指针，p2指向p1的前驱

char name[20];//客户姓名

char id[20];//客户身份证

char day[20];

info = find();//复制节点信息给info，find函数根据航班编号返回该航班节点的指针

while (info == NULL)

{

cout<<"没有这个航班，请重新输入"<<endl;

tuipiao();

}

head = info->cusLinkList;//head为该航班的的乘员名单域的头指针

p1 = head->next; //带头结点的指针，head->next 开始遍历

cout<<"请输入你的姓名: "<<endl;

cin>>name;

cout<<"请输入你的身份证号码:"<<endl;

cin>>id;

cout<<"请输入飞行时间"<<endl;

cin>>day;

p2 = head;

while (p1 != NULL)

{

//对比姓名和身份证

if ((strcmp(name, p1->name) == 0) && (strcmp(id, p1->ID) == 0)&&(strcmp(day, p1->day) == 0)) break;

p2 = p1;

p1 = p1->next;

}

if (p1 == NULL)

{

cout<<"对不起，你没有订过票或姓名和身份证,与飞行周期不对应"<<endl;

return;

}

else

{//退票成功

//从乘客名单域中移除该节点

grade = p1->dengji;

p2->next = p1->next;

//加回该航班的剩余票

info->less += p1->num;

if (grade == 1)

{

info->lessgrade1 += p1->num;

}

else

{

info->lessgrade2 += p1->num;

}

cout<<name<<"退票成功"<<endl;

}

}

/\*\*

\* 搜索界面

\*/

void sousuoFace()

{

int i;

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"搜索窗口"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 1.搜索航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 2.返回上一级 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入您的选择(1-2,否则无效！)"<<endl;

cout<<" ";

cout<<"您的选择是:";

cin >> i;

switch (i) {

case 1:

sousuoflight();

system("pause");

sousuoFace();

break;

case 2:

menu();

break;

default:

sousuoFace();

}

}

/\*\*

\* 订票界面

\* 属于订票模块

\*/

void dingpiaoFace()

{

int i;

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"订票窗口"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" 1.确认订票 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 2.搜索航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 3.浏览航班 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 4.返回上级菜单 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入您的选择(1-4,否则无效！)"<<endl;

cout<<" ";

cout<<"您的选择是:";

cin >> i;

switch (i) {

case 1:

dingpiao();

system("pause");

dingpiaoFace();

break;

case 2:

sousuoflight();

system("pause");

dingpiaoFace();

break;

case 3:

printfall(pFlight);

system("pause");

dingpiaoFace();

break;

case 4:

menu();

break;

default:

dingpiaoFace();

}

}

/\*\*

\* 退票模块界面

\*/

void tuipiaoFace()//退票模块界面

{

int i;

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"退票窗口"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 1.办理退票 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 2.返回上级菜单 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入您的选择(1-2,否则无效！)"<<endl;

cout<<" ";

cout<<"您的选择是:";

cin >> i;

switch (i)

{

case 1:

tuipiao();

system("PAUSE");

tuipiaoFace();

break;

case 2:

menu();

break;

default:

tuipiaoFace();

}

}

/\*\*

\* 退出程序模块

\*/

void GoodbyeFace() {

system("cls");

printf("\n");

printf("\n");

printf(" ---------------------------------------------\n");

printf(" | 感谢使用航空航班订票系统 \n");

printf(" | |\n");

printf(" | GoodBye！ \n");

printf(" --------------------------------------------\n");

}

/\*\*

\* pFlight为全局变量，航班链表的头指针

\* 先初始化pFlight航班链表，再把flight1的四个你接班数据插入pFlight链表

\* 调用菜单函数

\*/

int main() {

//初始化pFlight，pFlight为全局变量。

InitFlight();

//用flight1中的数据创建初始航班链表，里面含有四个节点

Create(flight1);

//菜单函数

MenuSelect();

return 0;

}

/\*\* \* 用户登录页面

\* @return操作状态

\*/

int yonghudenglu()

{

string account1="黄暄";

string passwords1="239007045";

string account2="王峰";

string passwords2="239007038";

string account3="邓浩宇";

string passwords3="239007049";

string account4="陈记德";

string passwords4="239007037";

string account5="徐嘉诚";

string passwords5="239007036";

string account;

string passwords;

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"用户登录页面"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 请输入正确的账号与密码以进入系统 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" (返回首页请输入exit) "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入账号:"<<endl;

cout<<" ";

cout<<"account:";

cin>>account;

cout<<" ";

if(account=="exit")

{

denglu();

}

cout<<"请输入密码:"<<endl;

cout<<" ";

cout<<"password:";

cin>>passwords;

if((account==account1&&passwords==passwords1)||(account==account2&&passwords==passwords2)||

(account==account3&&passwords==passwords3)||(account==account4&&passwords==passwords4)||

(account==account5&&passwords==passwords5))

{

menu();

}

else

{

cout<<" ";

cout<<"输入错误，按任意键重新输入"<<endl;

}

}

/\*\* \* 管理员登录页面

\* @return操作状态

\*/

int guanliyuandenglu()

{

string passwords="123456";

string shuru;

cout<<endl<<endl<<endl<<endl<<endl;

cout<<" ";

cout<<"管理员登录页面"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" 请输入管理员密码以验证身份 "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" (返回首页请输入exit) "<<"|"<<endl;

cout<<" ";

cout<<"|"<<" "<<"|"<<endl;

cout<<" ";

cout<<"-----------------------------------------------"<<endl;

cout<<" ";

cout<<"请输入密码:"<<endl;

cout<<" ";

cout<<"password:";

cin>>shuru;

if(shuru==passwords)

{

cout<<" ";

cout<<"管理员身份验证成功！"<<endl;

guanliyuanyemian();

}

else if(shuru=="exit")

{

denglu();

}

else

{

cout<<" ";

cout<<"身份验证失败，按任意键重新输入"<<endl;

system("pause");

guanliyuandenglu();

}

}

# 四、程序设计调试情况分析

1. 航班采用单链表连接起来，采用头插入法，插入的时间复杂度为O(1)，删除和查询的时间复杂度为O(n),空间复杂度都为O（1）；
2. 候补客户使用含有头指针和尾指针的结构体队列实现，所以入队和出队的时间复杂度和空间复杂度都为O（1）。
3. 航班和候补队列都要用到单链表，一开始我使用尾插入法的，后来觉得头插入法更简单，所以最后均采用了头插入法插入，为了插入和删除节点方便，使用了带头结点的链表和链队列。
4. 关于舱位等级和剩余票问题，一开始我是采用了票数和舱位等级无关，自己随意输入舱位等级都行，但是后来发现这样并不合理，后来就将总剩余票数分为两部分，一部分是经济舱剩余票数，一部分是商务舱剩余票，这样，在候补客户排队订票时，就也需要分开两条队列，一条是在经济舱队列排队，一条是在商务舱队列排队。
5. 在链表或者队列的最后一个节点的next指针域一定要置为NULL，否则就会出现野指针，可能会导致实验出现错误。
6. 在输入的过程中，要注意参数值的合法性，比如票数不能出现负数，不合法的参数要求重新输入。
7. 在退票的时候原本我是采用了输入航班ID和姓名就能退票了，后来觉得不合理，增加了身份证号码验证，验证成功才能退票，并且身份证号码格式要正确。
8. 当票数不足时，要采取多种判断来为客户服务，这就涉及到了很多if else的嵌套。

# 五、测试结果

**用户说明**

1. 用户运行程序后可以看到登录页面，包括用户登录与管理员登录两大部分。

2.用户可以选择适当的途径登录。

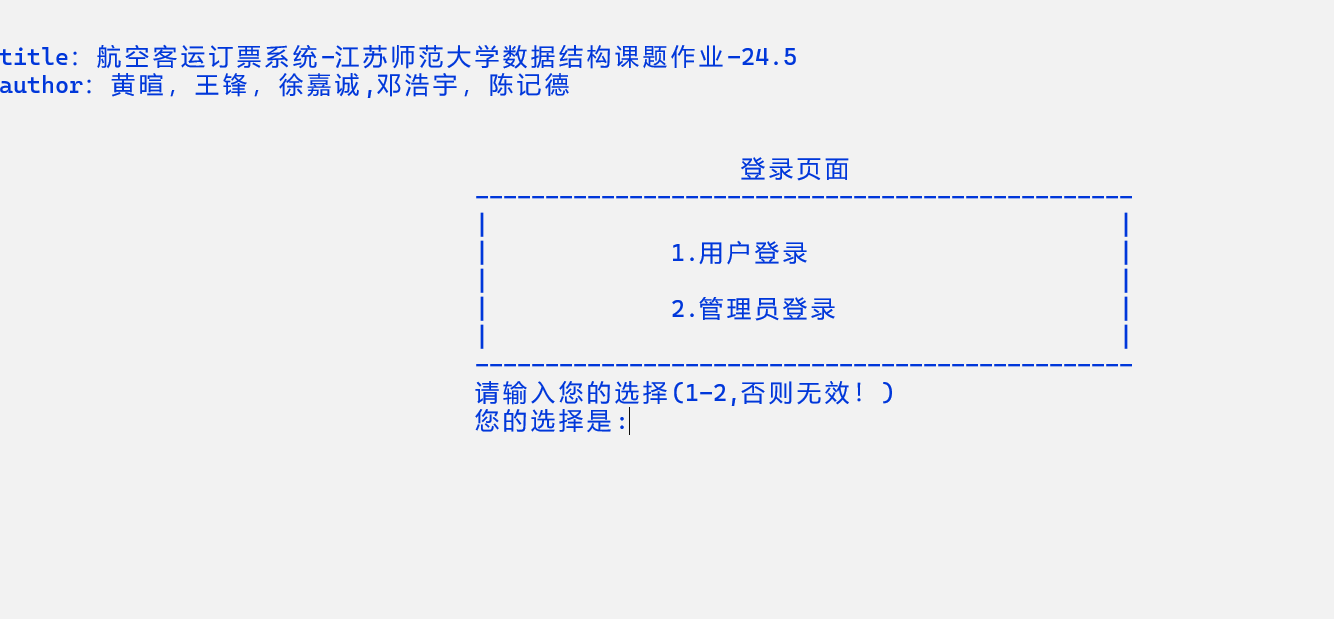
3. 普通用户可以进行用户登录（账号为作者姓名，密码为作者学号，如账号：黄暄；密码：239007045），可以使用的功能包括全部航班查询、搜索航班、订票以及退票。

4. 管理员登录需要密码验证（密码为123456），管理员可以对航班进行增加删除操作，还可以对客户订票情况进行查询.

5.退票需要验证航班ID，姓名和身份证，验证成功才能退票。

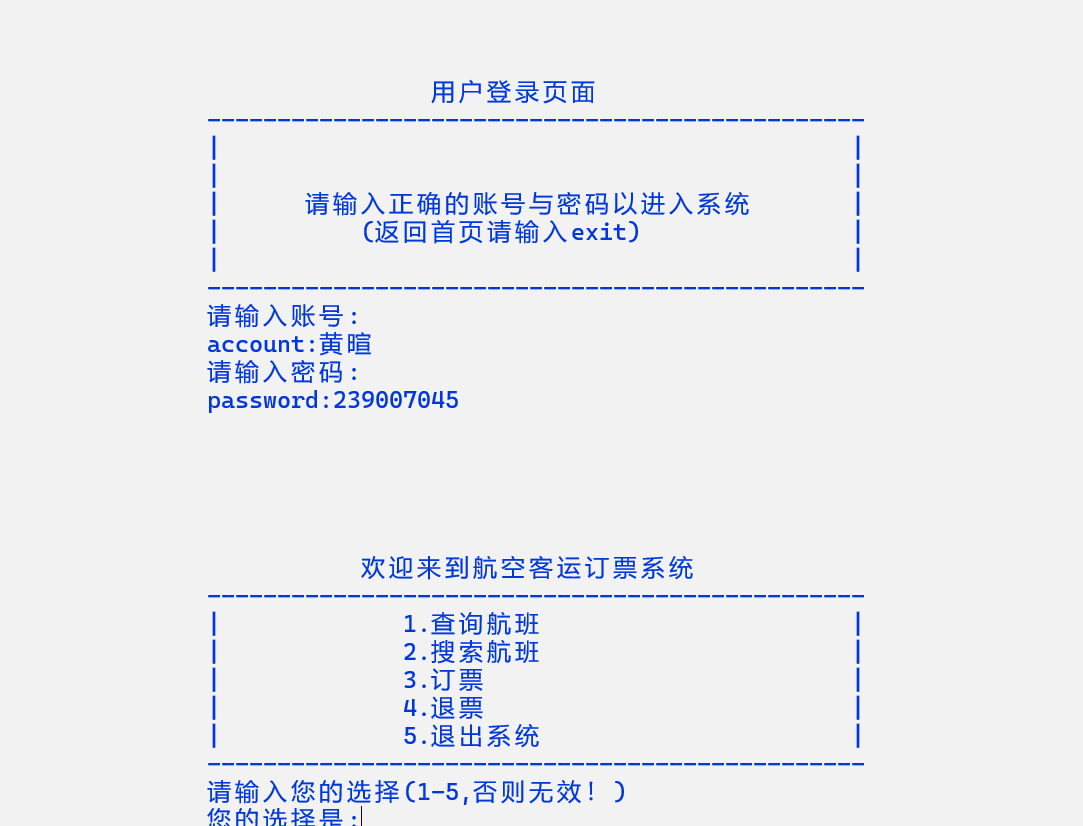
6.当票数不足时可以选择排队或者重新订其他票。

1. 登录界面界面：选择用户登录或管理员登录



用户登录：

输入账号密码登录用户页面



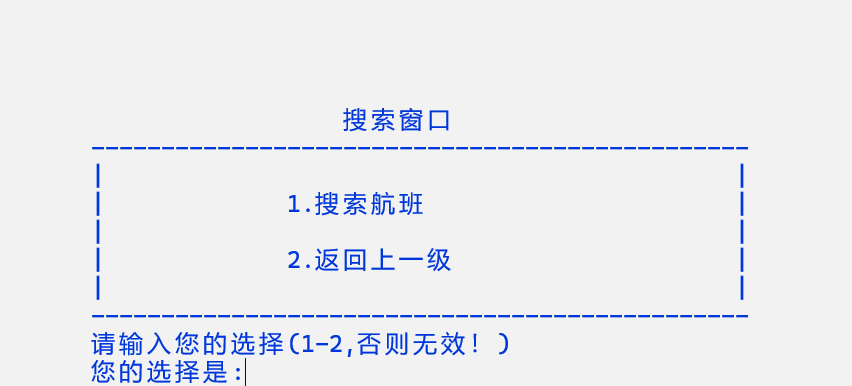
模块一查询模块：

从菜单中输入1：调用了printFall函数，进入查询模块，将显示所有航班的相关信息，查看完之后按任意键会返回菜单界面。

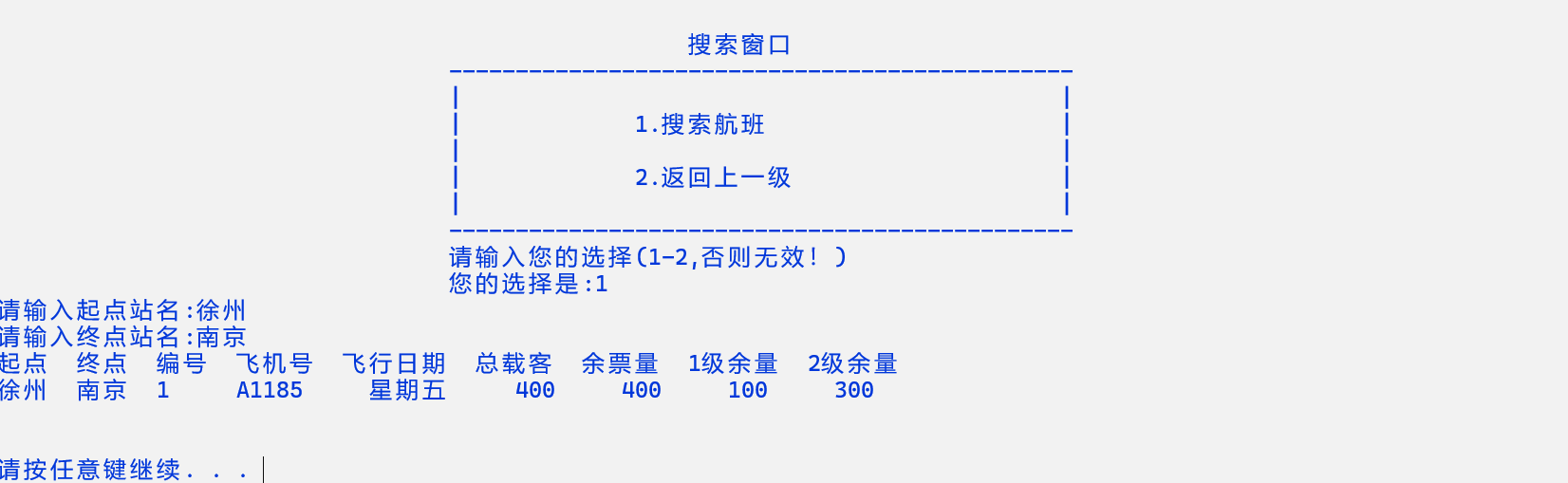


模块二-搜索函数：

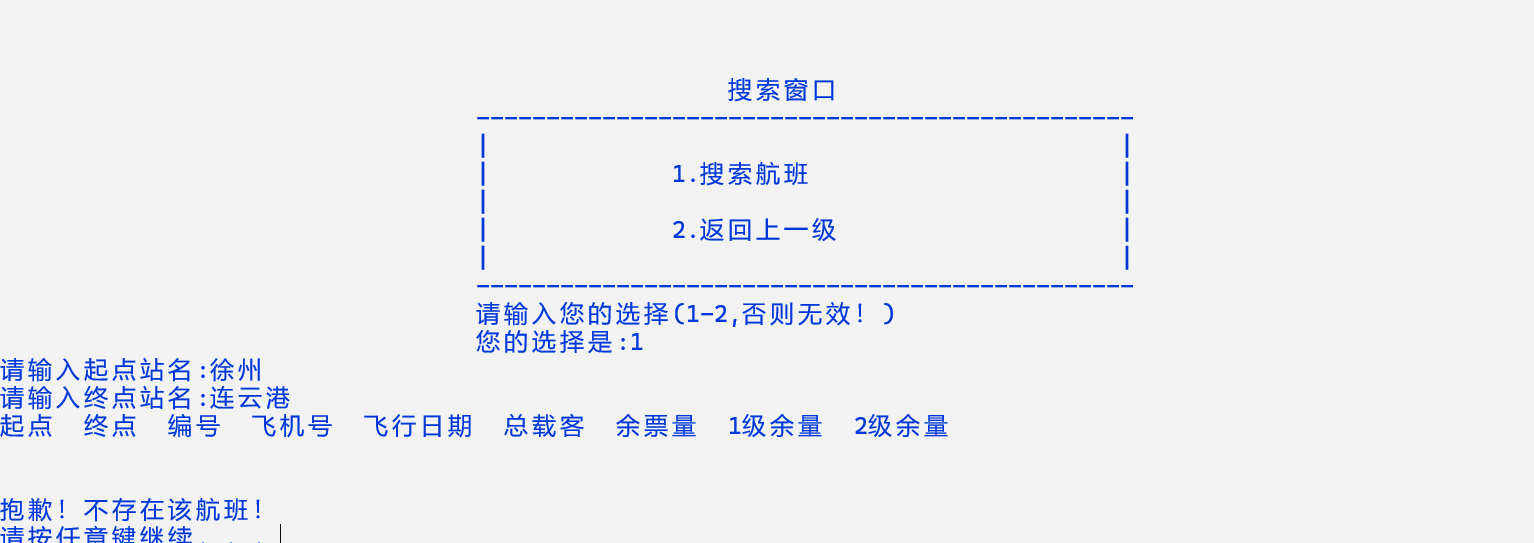
从主菜单中输入2：调用了sousuoFace函数，进入搜索模块



* 1. 输入1，搜索航班信息，需要输入起点站和终点站名字

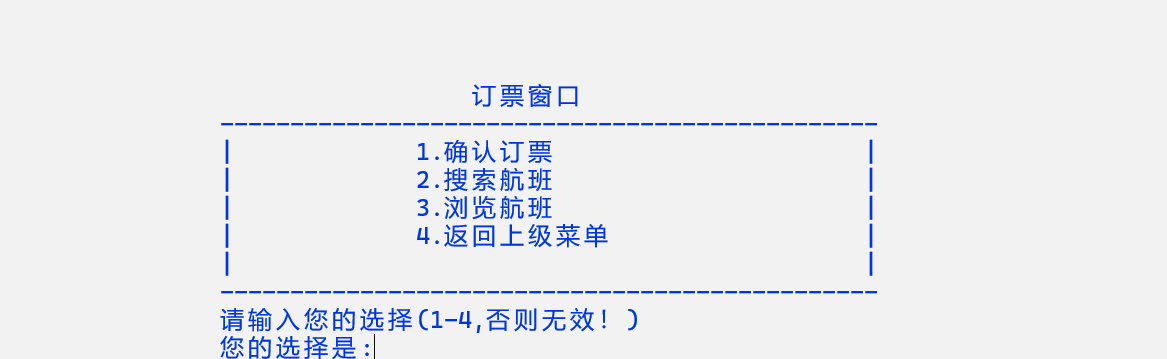


* 1. 输入起点站名和终点站名搜索航班，若是在该航班链表中不存在，则提示该航班未找到。

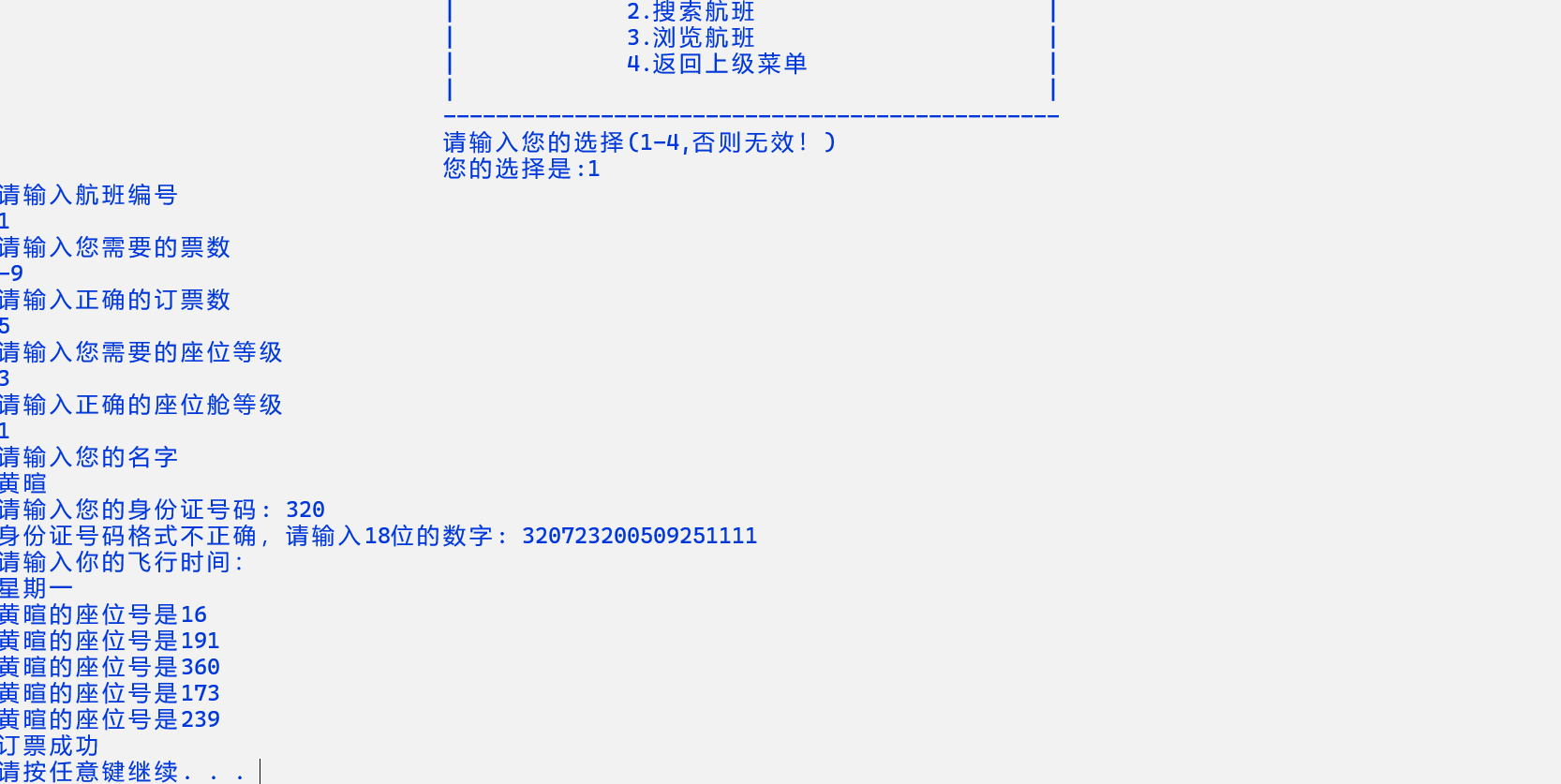


模块三-订票模块：

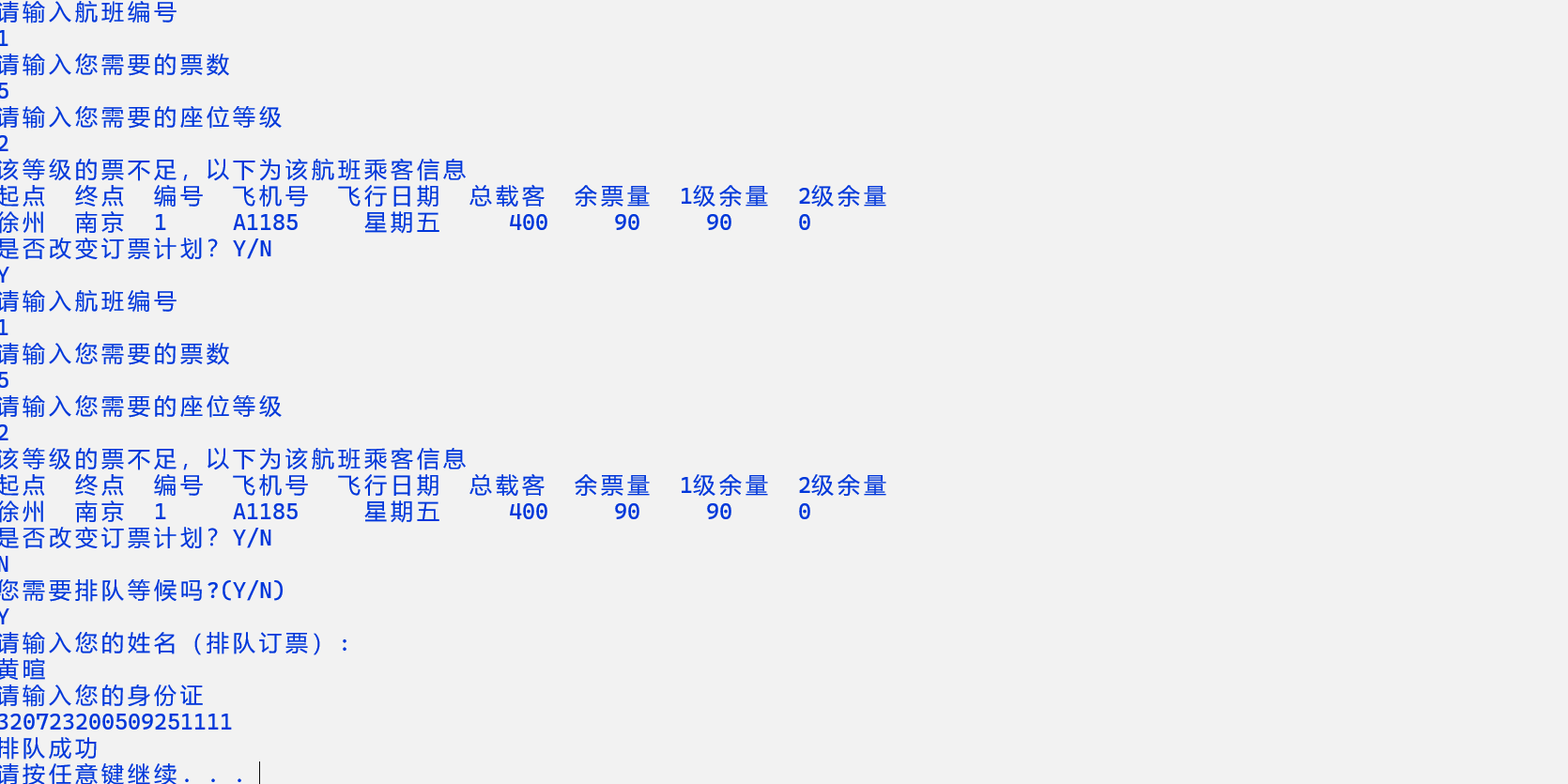
从主菜单中输入3，调用dingpiaoface函数，进入订票模块。



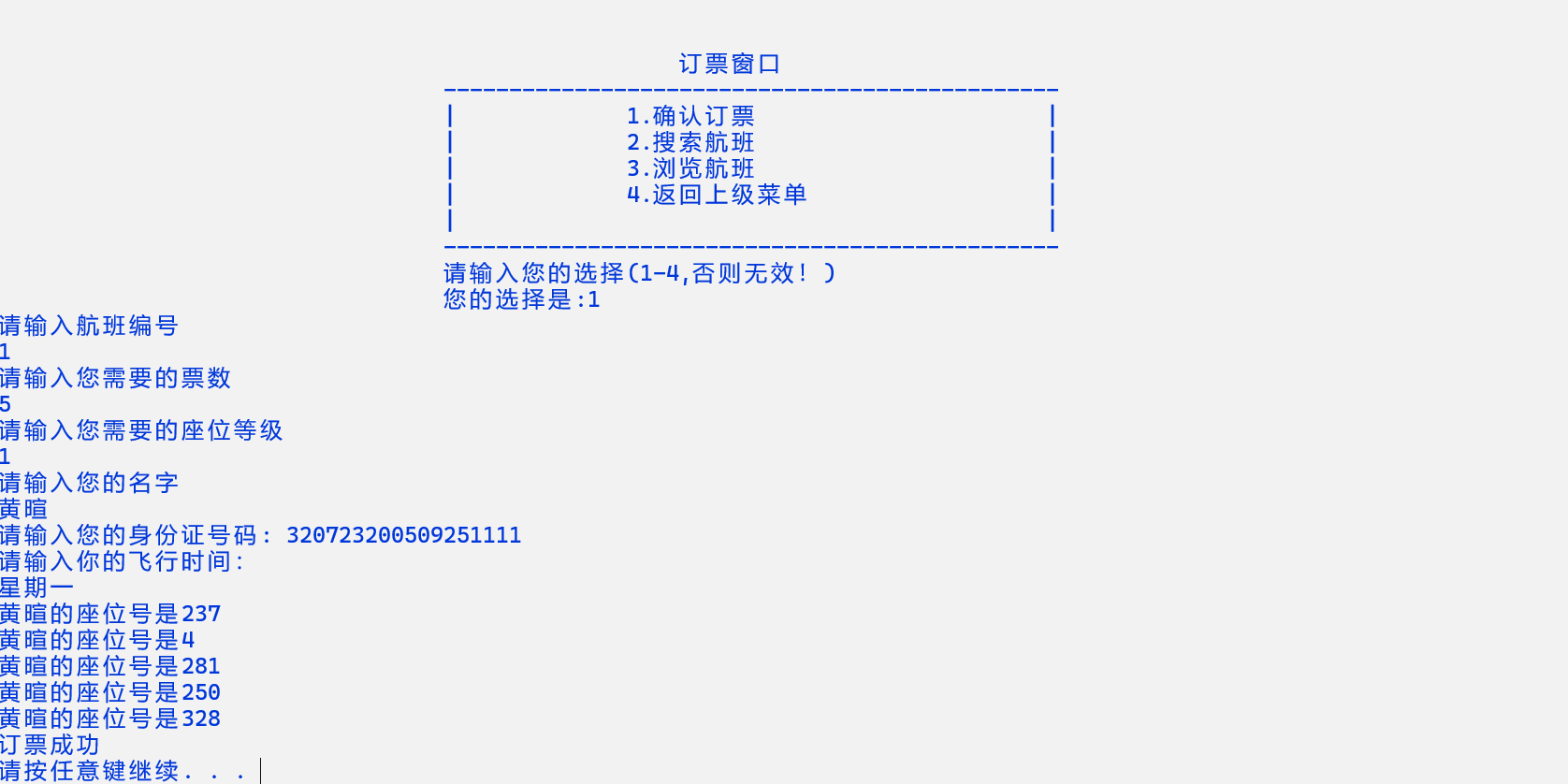
1. 输入1，进行订票，航班ID要正确输入，不存在该航班ID会提示没有这个航班（例如输入aa），则要求重新输入。当航班ID正确时输入订票数目，数目要合法，输入负数要求重新输入（例如下图输入了-1和-100就不行），正确输入票数之后，需要输入舱位等级，此时，如果对应仓位剩余的票数足够，则显示订票成功，并告诉你相应的随机座位号,。



以下是订票失败选择排队的案例：1号航班的经济票已经售完，然后两个客户，黄暄先后再去订1号航班的经济舱的票，订票失败，但是黄暄和王锋都选择了排队。



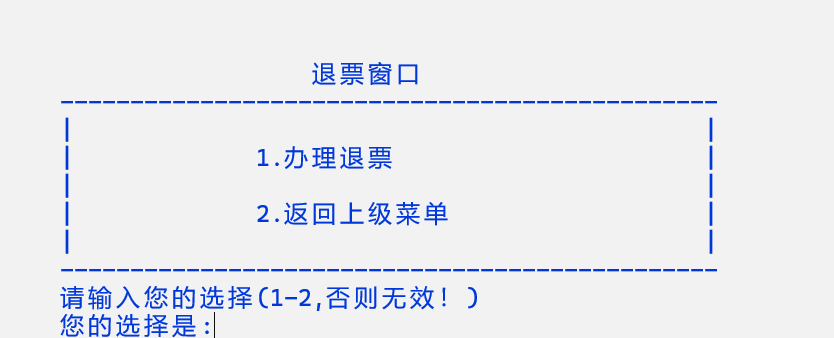
1号的经济舱票数已经为0，但是商务舱票数仍然有剩余，此时，黄暄前去订1号航班的商务舱的机票，则订票成功。



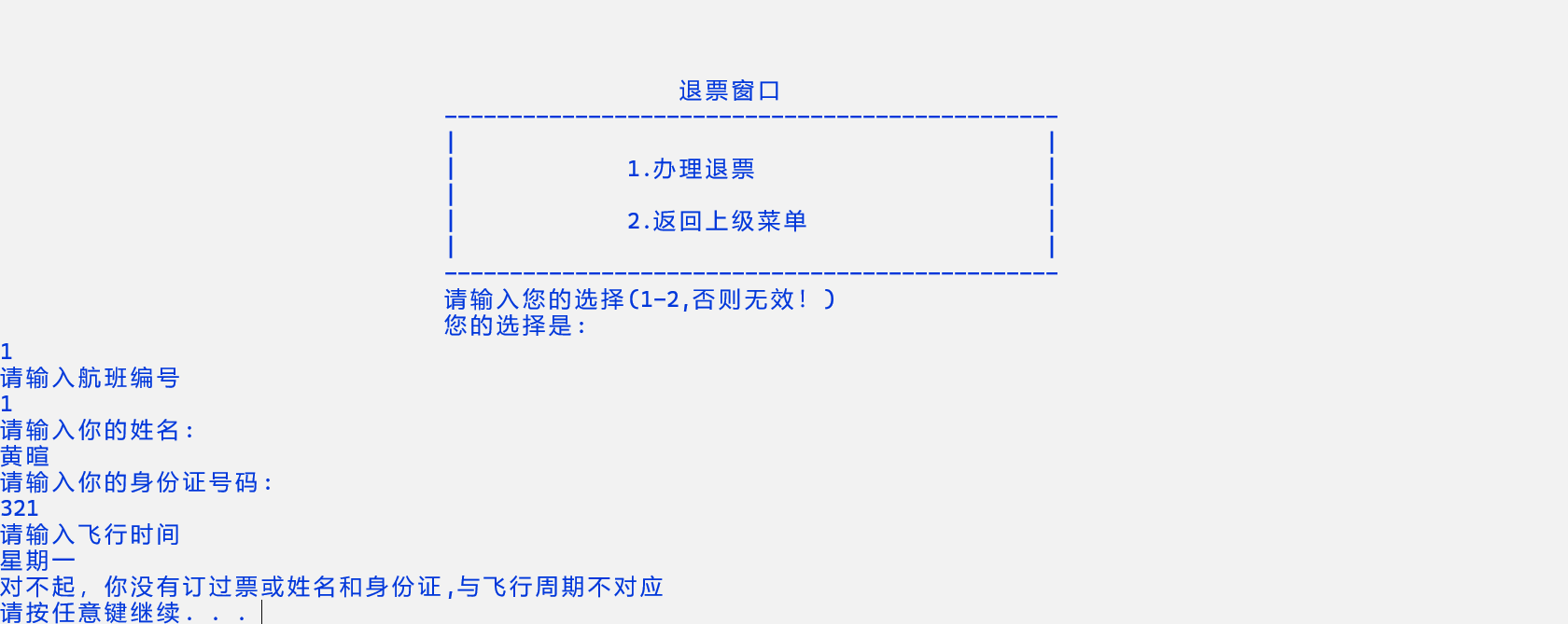
1. 订票模块中输入2和3分别时调用搜索和查询模块，前面已经测试过了，这里不再重复测试。

模块四-退票模块：

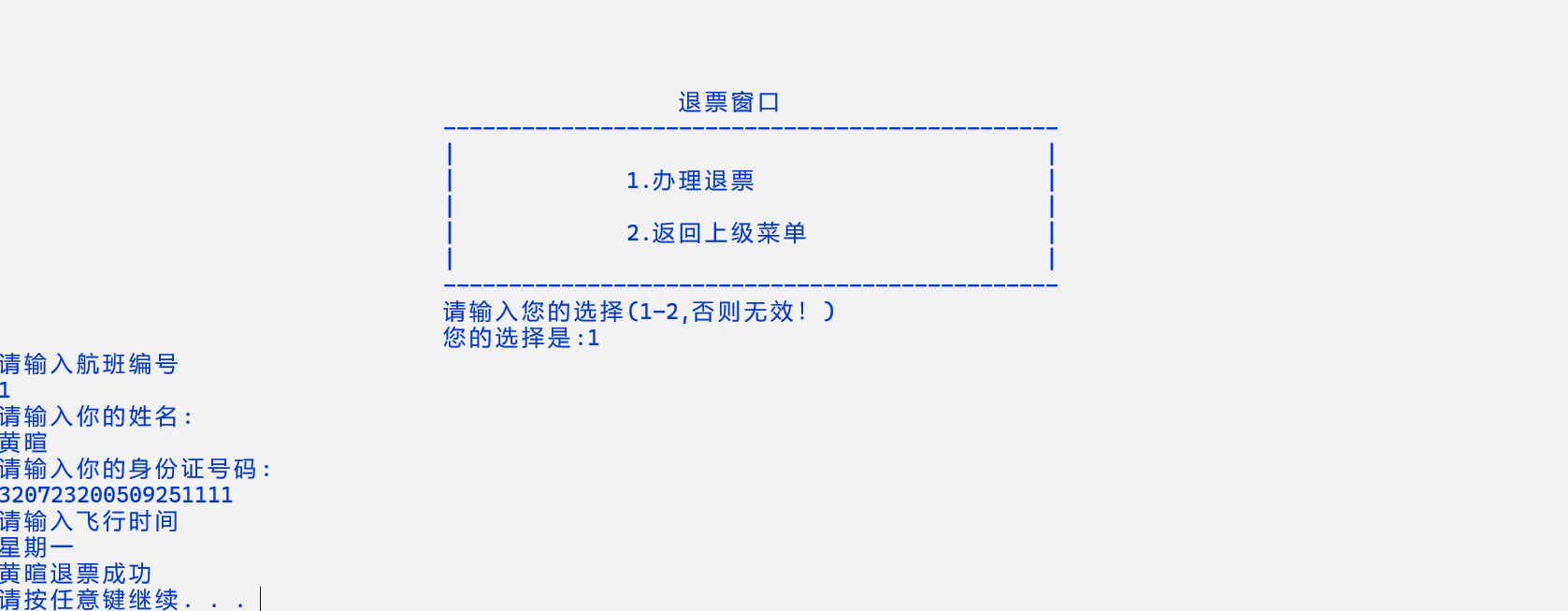
从主菜单中输入4，进入退票模块：



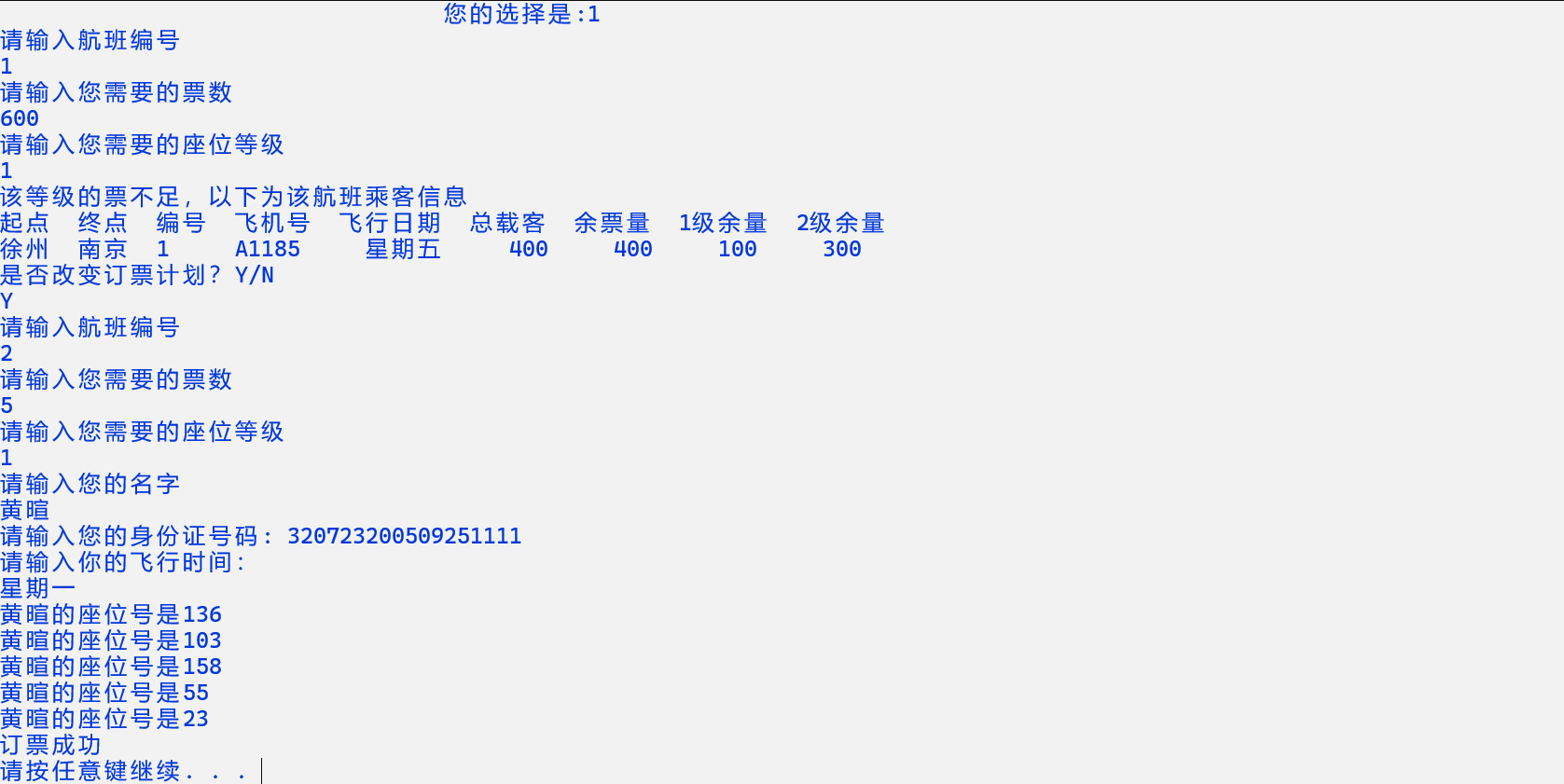
输入1，办理瑞票手续，当没有订票或者输入的航班号、姓名和身份证号码没有一一对应，都将退票失败。



黄暄订了5张1号航班的经济舱的票，然后，退票，退票成功。

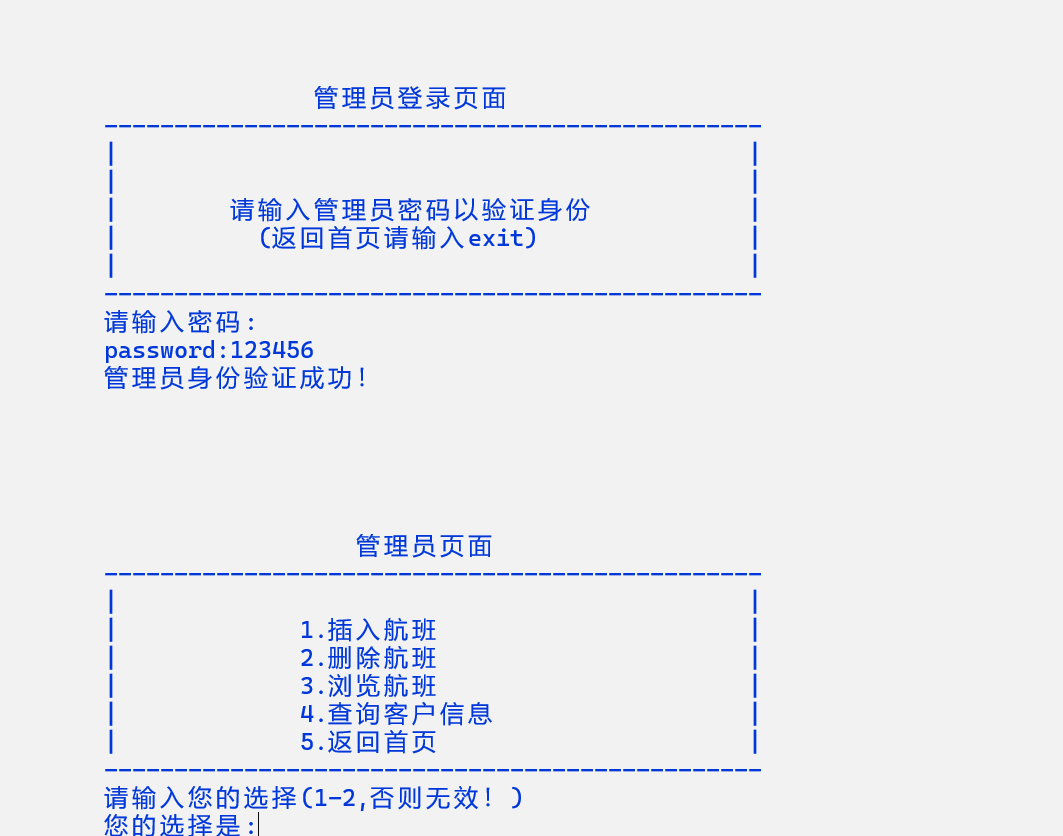


下面测试订票失败，不选择排队而再次重新订票的例子。

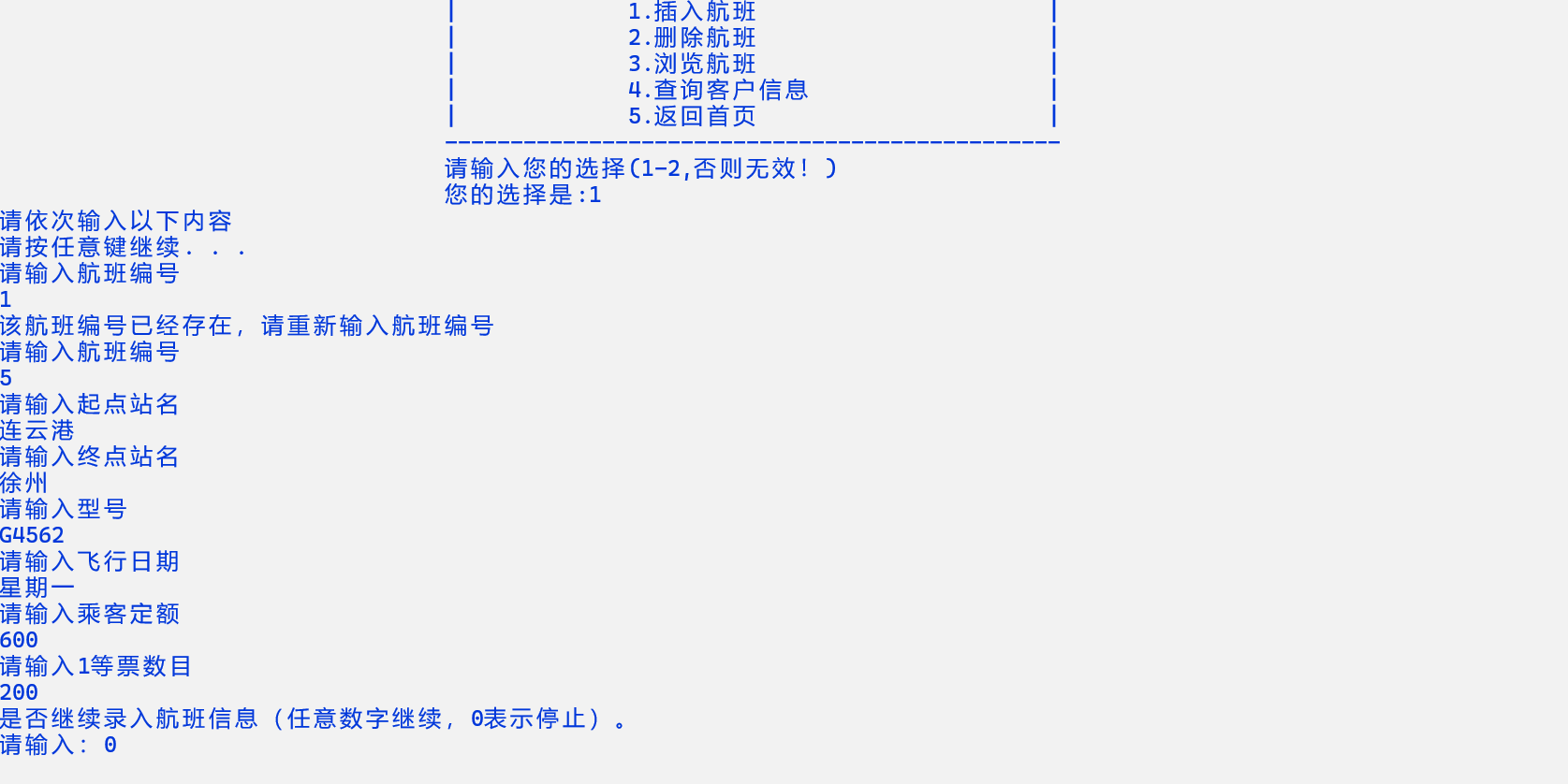


**管理员登录**

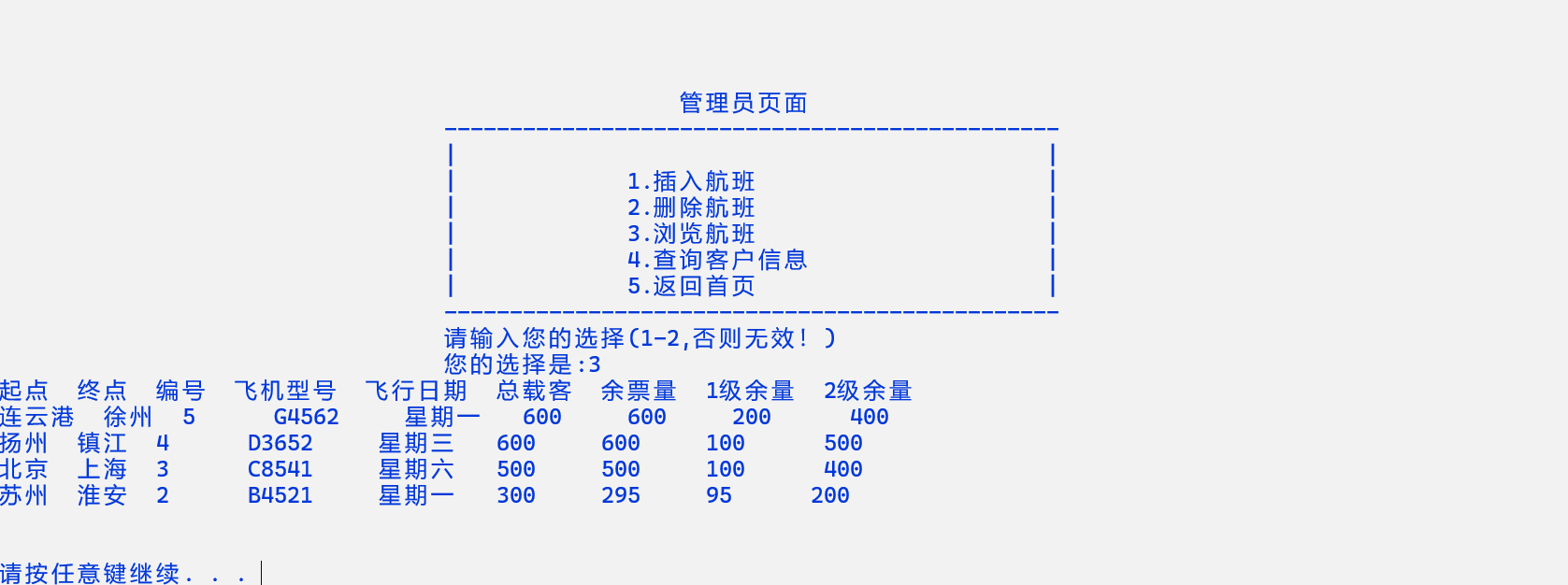
从登录页面中选择管理员登录：进入管理员模块，先验证密码（123456）



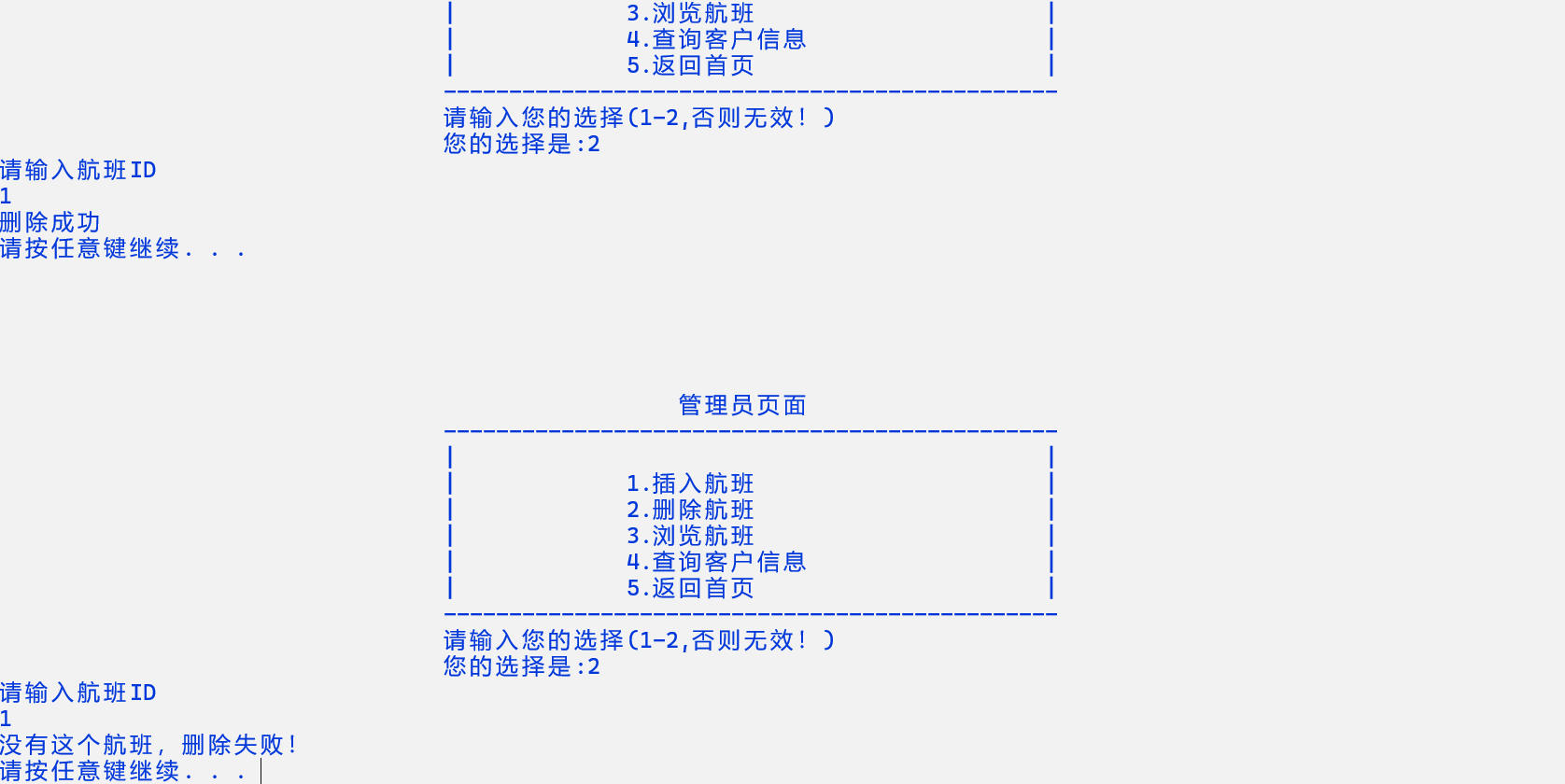
1. 选择1，增加航班，航班ID，不能重复，重复要求重新输入。

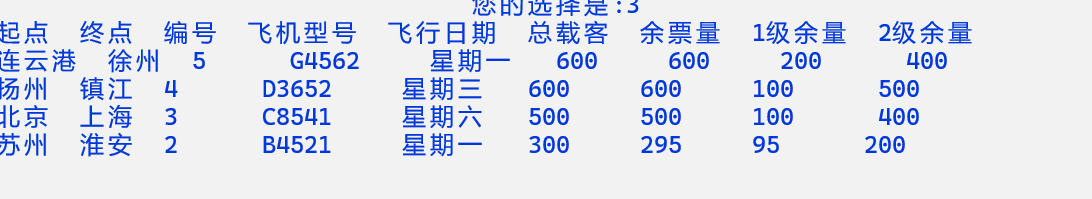


插入成功后，调用查询模块查询，可以看到刚刚添加的5号航班。



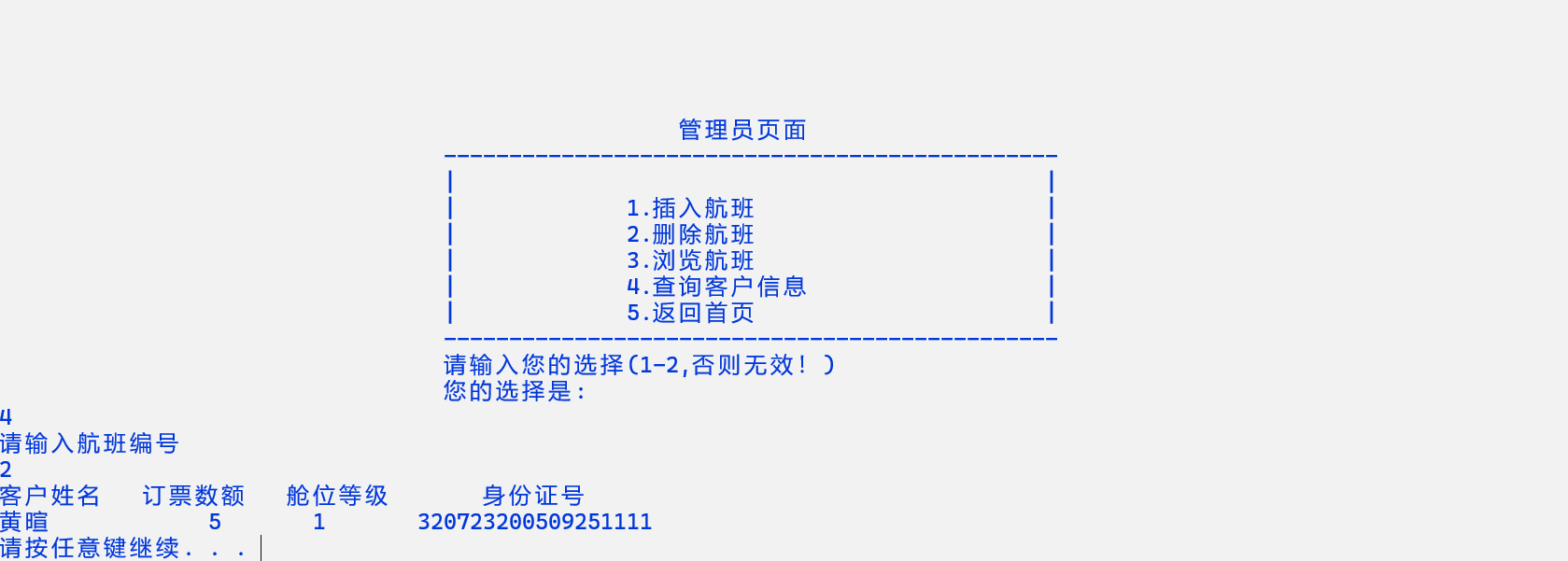
1. 选择2：删除航班：如果航班不存在，则删除失败

 删成功后查询，可以看到1号航班已经不存在了。



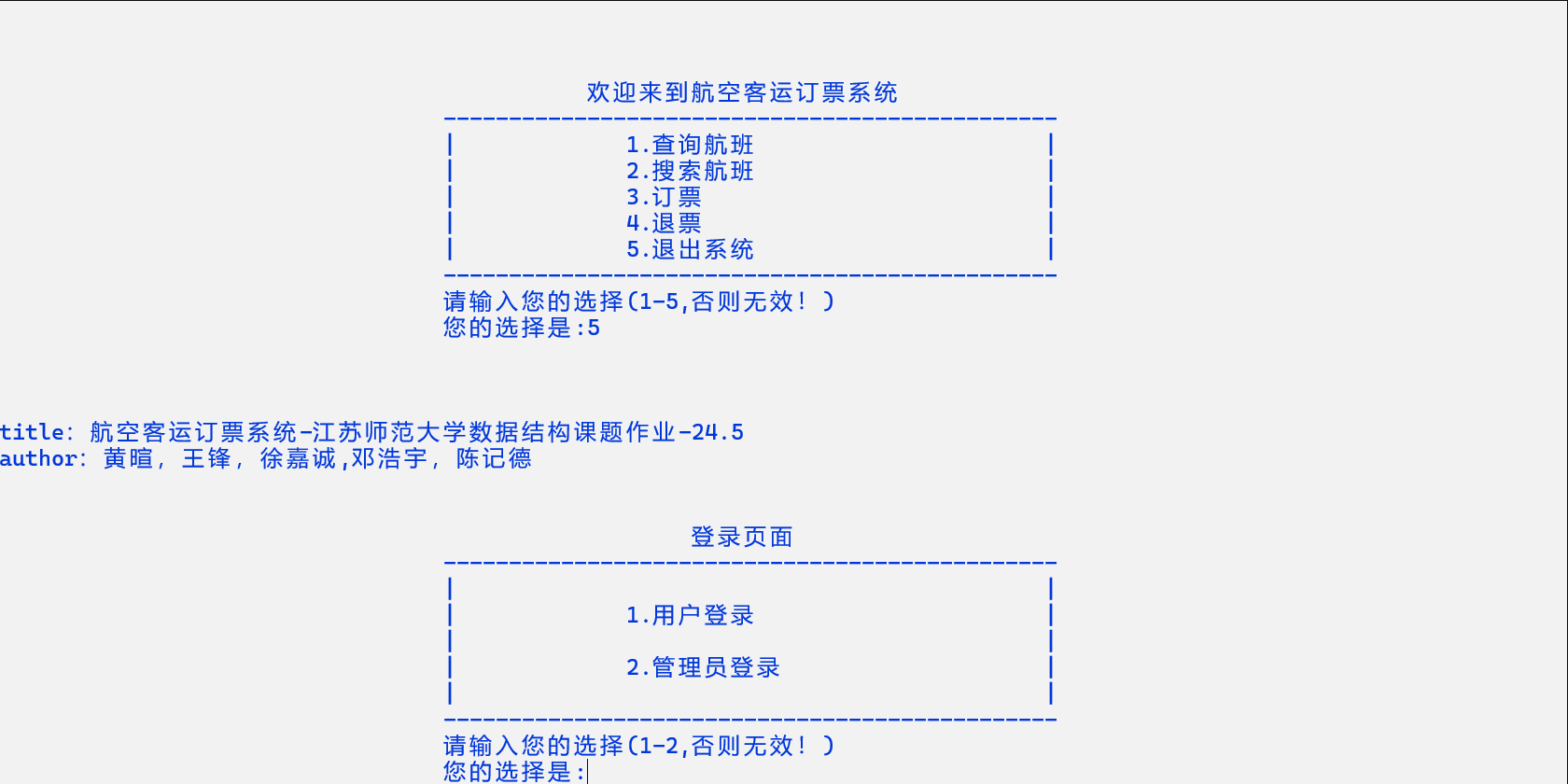
1. 选择3：调用航班查询模块，不再重复测试。

1. 通过航班号查询客户信息，能显示身份证号码。



模块六：退出系统

退出系统回到登录页面



# 六、总结

该航空客运订票系统是我学习C++以来第一个付出实践的较大项目，为此付出了较多心血，在设计该系统的过程中，我深刻体会到了C++的魅力，以及数据结构与算法在实际场景的应用，看着这一系统一步一步的完善，一直向真正的航空订票系统靠近，心中充满激动，尽管我深知它目前不过是一个比较简易的小项目，但我仍然以它为骄傲，自豪。。

本项目以单链表为存储结构，通过使用本学期在数据结构与算法课程上的学习，熟练掌握了对单链表的插入与删除，并学会使用指针来由针对性的访问它们，我认为这是我值得骄傲的地方，在乘客候补问题上使用由结构体构成的队列来进行存储候补乘客的信息，这在时间复杂度的较为高效，本项目所使用的数据结构与算法方面较为简单，仅仅只有单链表的增删与候补队列的存储，希望日后能够有机会使用更高阶的方法来进一步完善我的系统。

在本项目的搭建中，本人体会到了模块化编程的思想，通过各各函数之间的层层调用，才使得这一系统能够流畅运行，我相信模块化编程的思想将会在我日后的学习与工作中发挥着不可或缺的作用！

非常感谢王树梅老师对本项目提出了若干指导性意见，如身份号码的验证，座位号的随机生成等，王树梅老师是一位极为认真负责的老师，能够及时帮助同学们解决问题，我非常喜欢听他上课，也非常庆幸能够在大学阶段遇到这样一位亦师亦友的老师。

在这次写代码的过程中，出现过很多问题，如指针错误，空指针异常，野指针等等，在处理这些Bug的时候，我深刻认识到了断点调试的重要性，只有断点调试，才能快速找出Bug并加以改正。

通过这次课设，我加深了对指针、链表和队列的理解，同时也自我感觉动手能力和算法能力有了明显的提升。

# 七、附录(源代码)

1. 源代码：航空客运订票系统.cpp
2. 可执行文件：航空客运订票系统.exe
3. 课程设计报告电子版：航空客运订票系统-数据结构课程设计报告.docx
4. 流程图电子版：航空客运订票系统流程图.pdf
5. 开题报告ppt
6. 期末答辩ppt
7. 课程设计任务书

**完整源码如下：**

1. #include<iostream>
2. #include<string>
3. #include<windows.h>
4. #include <algorithm>
5. #include<bits/stdc++.h>
6. using namespace std;
7. typedef struct CNode
8. {
9. char name[20];*//姓名*
10. int num;*//订票数*
11. char ID[20];*//身份证*
12. char day[20];
13. int dengji;*//舱等级*
14. CNode \*next;
15. } CNode, \*CList;*//乘客节点*
16. typedef struct HbNode
17. {
18. char name[20];*//姓名*
19. char ID[20]; *//身份证*
20. int num;*//预定数量*
21. struct HbNode \*next;
22. } QhbNode, \*Phb;*//候补队列中的节点*
23. typedef struct Queue
24. {
25. Phb front;*//等候替补客户名单域的头指针*
26. Phb rear;*//等候替补客户名单域的尾指针*
27. } LinkQueue;*//候补队列*
28. typedef struct hangban {
29. char qidian[20];*//起点站名*
30. char zhongdian[20];*//终点站名*
31. char hbID[20];*//航班编号*
32. char hbnum[20];*//飞机号*
33. char day[20];*//飞行日期（星期几）*
34. int maxnum;*//总票数*
35. int less;*//总余票量*
36. int lessgrade1; *//等级1剩余量*
37. int lessgrade2; *//等级2剩余量*
38. hangban \*next;
39. CList cusLinkList;*//乘员名单，指向乘员名单链表的头指针*
40. LinkQueue wQgrade1;*//等级1候补，等候替补的客户名单域，指向一个队列*
41. LinkQueue wQgrade2;*//等级2候补，等候替补的客户名单域，指向一个队列*
42. } hangban, \*Phangban;*//航班节点*
43. *//中期答辩*
44. int guanliyuanyemian();*//管理员页面*
45. int denglu(); *//登录*
46. void initFlight(); *//初始化航班链表*
47. void sousuoflight();*//由起点终点搜索航班*
48. void sousuoFace();*// 搜索界面*
49. void printfall(hangban \*pflight);*//打印全部航班信息*
50. int menu();*// 菜单界面函数*
51. int zengjiaFlight();*//增加航班*
52. int shanchuFlight();*// 删除航班*
53. void sousuoFlight();*//由起点和终点搜索对应航班*
54. int Create(Phangban flight1);*//将航班信息插入到链表中*
55. int duibiFlight(hangban \*flight, char flightCodeID[]);*// 对比插入航班标号，防止标号重复*
56. *//期末答辩*
57. hangban \*find(); *// 输入编号返回对应信息*
58. int initQueue(LinkQueue &q);*//初始化队列*
59. int initCusLinkList(CList &cusLinkList);*//初始化订票乘客链表*
60. void addlink(CList &head, int need, char name[], char id[],char day[], int grade);*//将订票成功的乘客节点加入到已订票链表*
61. void pricusInfo();*//输出已经订票的函数*
62. void dingpiao();*// 订票函数*
63. void fightinfo(hangban \*p);*//输出对应节点信息*
64. LinkQueue enqueue(LinkQueue &q, char name[], int need, char id[]);*//将等待的客户入队*
65. void tuipiao();*// 退票*
66. void dingpiaoFace();*//  订票界面*
67. void tuipiaoFace(); *//退票界面*
68. hangban \*pFlight;*//全局节点变量*
69. *//初始输入的航班信息*
70. hangban flight1[4] = {
71. {"徐州", "南京", "1", "A1185", "星期五", 400, 200, 100, 100},
72. {"苏州", "淮安", "2", "B4521", "星期一", 300, 200, 100, 100},
73. {"北京", "上海", "3", "C8541", "星期六", 500, 200, 100, 300},
74. {"扬州", "镇江", "4", "D3652", "星期三", 600, 200, 100, 400},
75. };
76. *//初始化航班链表*
77. void initFlight()
78. {
79. pFlight = new hangban;
80. pFlight->next = NULL;
81. }
82. *// 将已有航班信息插入链表*
83. int Create(Phangban flight1)
84. {
85. hangban \*p = pFlight, \*q;
86. for (int i=0; i < 4; i++) {
87. q = new hangban;
88. if (q == NULL)
89. return 0;
90. strcpy(q->qidian, flight1[i].qidian);*//利用链表把每个信息录入*
91. strcpy(q->zhongdian, flight1[i].zhongdian);
92. strcpy(q->hbID, flight1[i].hbID);
93. strcpy(q->hbnum, flight1[i].hbnum);
94. strcpy(q->day, flight1[i].day);
95. q->maxnum = flight1[i].maxnum;
96. q->less = flight1[i].maxnum;
97. q->lessgrade1 = flight1[i].lessgrade1;
98. q->lessgrade2 = flight1[i].maxnum - flight1[i].lessgrade1;
99. initCusLinkList(q->cusLinkList); *//初始化已经订票乘客链表*
100. initQueue(q->wQgrade1);*// 将两种等级的票入队 ，方便后面预定排队*
101. initQueue(q->wQgrade2);
102. q->next = p->next;
103. p->next = q;
104. }
105. return 1;
106. }
107. *//初始化已订票乘客指针链表*
108. int initCusLinkList(CList &cusLinkList)
109. {
110. CList q = cusLinkList;
111. cusLinkList = (CNode \*) malloc(sizeof(CNode));
112. cusLinkList->next = NULL;
113. }
114. *//初始化带头结点的链队列*
115. int initQueue(LinkQueue &q)
116. {
117. QhbNode \*p;
118. p = new QhbNode;
119. p->next = NULL;
120. q.front = q.rear = p;*//队空*
121. return 1;
122. }
123. bool isValidID(const char id[])
124. {
125. int length = strlen(id);
126. *// 检查身份证号码长度是否为18位数字*
127. return (length == 18) && all\_of(id, id + length, ::isdigit); *// 使用 C++11 的 all\_of 函数检查所有字符是否为数字*
128. }
129. *// 订票函数*
130. void dingpiao()
131. {
132. hangban \*info;
133. int need, grade;
134. int num;*//剩余的等级1数目或者等级2数目*
135. char name[20];
136. char id[20];
137. char day[20];
138. info = find();*//输入想订的航班，看是否还有空位置，并返回复制节点*
139. *// 初始化随机数种子*
140. srand(time(0));
141. int seatCount = 0; *// 已分配座位的数量*
142. if (info == NULL)
143. {
144. cout<<"不存在该航班"<<endl;
145. dingpiao();
146. }
147. cout<<"请输入您需要的票数"<<endl;
148. while(1)
149. {
150. cin>>need;
151. if(need<=0)
152. {
153. cout<<"请输入正确的订票数"<<endl;
154. continue;
155. }
156. else break;
157. }
158. cout<<"请输入您需要的座位等级"<<endl;
159. while(1)
160. {
161. cin>>grade;
162. if(grade!=1&&grade!=2)
163. {
164. cout<<"请输入正确的座位舱等级"<<endl;
165. continue;
166. }
167. else break;
168. }
169. if (grade == 1)
170. num = info->lessgrade1;
171. else
172. num = info->lessgrade2;
173. if (need <= num) *//订票数小于剩余数*
174. {
175. vector<int> allocatedSeats;
176. int i;
177. cout<<"请输入您的名字"<<endl;
178. cin>>name;
179. cout << "请输入您的身份证号码: ";
180. while (true)
181. {
182. cin>>id;
183. if (!isValidID(id))
184. {
185. cout << "身份证号码格式不正确，请输入18位的数字: ";
186. continue;
187. }
188. else
189. {
190. break;
191. }
192. }
193. cout<<"请输入你的飞行时间："<<endl;
194. cin>>day;
195. CList head = info->cusLinkList;
196. addlink(head, need, name, id,day,grade);*//订票成功，插入成员名单链表*
197. *// 从1开始，因为通常座位号从1开始编号*
198. int nextSeatNumber = 1;
199. int seatStatus[info->maxnum] = {0}; *// 0表示座位未被占用，1表示已被占用*
200. *// 循环分配座位号*
201. for (int i = 0; i < need; ++i)
202. {
203. int seatNumber;
204. do {
205. seatNumber = rand() % info->maxnum + 1;
206. } while (seatStatus[seatNumber - 1]); *// 检查座位号是否已被占用*
207. seatStatus[seatNumber - 1] = 1; *// 标记座位号为已占用*
208. cout << name << "的座位号是" << seatNumber << endl;
209. }
210. info->less -= need;
211. if (grade == 1)
212. info->lessgrade1 -= need;
213. else
214. info->lessgrade2 -= need;
215. cout<<"订票成功"<<endl;
216. }
217. else
218. {
219. char r;
220. cout<<"该等级的票不足，以下为该航班乘客信息"<<endl;
221. fightinfo(info);*//输出该航班的订票信息*
222. cout<<"是否改变订票计划？Y/N"<<endl;
223. cin>>r;
224. if (r == 'Y' || r == 'y') *//改变计划，重新选择航班*
225. {
226. dingpiao();*//返回订票主页面*
227. }
228. else
229. {
230. cout<<"您需要排队等候吗?(Y/N)"<<endl;
231. cin>>r;
232. if (r == 'Y' || r == 'y')
233. {
234. *//不改变计划，排队候票*
235. cout<<"请输入您的姓名（排队订票）:"<<endl;
236. cin>>name;
237. cout<<"请输入您的身份证"<<endl;
238. cin>>id;
239. if (grade == 1) *//进入1等舱排队队列*
240. {
241. info->wQgrade1 = enqueue(info->wQgrade1, name, need, id);
242. }
243. else *//进入2等舱排队队列*
244. {
245. info->wQgrade2 = enqueue(info->wQgrade2, name, need, id);
246. }
247. cout<<"排队成功"<<endl;
248. }
249. }
250. }
251. }
252. *//根据自己输入的航班标号查询是否存在并返回节点*
253. hangban \*find()
254. {
255. char number[10];
256. int i = 0;
257. cout<<"请输入航班编号"<<endl;
258. cin>>number;
259. hangban \*p = pFlight->next; *//头结点的下一个节点开始遍历*
260. while (p != NULL)
261. {
262. if (!strcmp(number, p->hbID))
263. return p;
264. p = p->next;
265. }
266. return NULL;
267. }
268. void addlink(CList &head, int need, char name[], char id[],char day[],int grade)
269. *//插入到已经订票客户链表*
270. {
271. CList new1;*//创建一个新节点*
272. new1 = (CNode \*) malloc(sizeof(CNode));
273. strcpy(new1->name, name);
274. strcpy(new1->ID, id);
275. strcpy(new1->day,day);
276. new1->num = need;
277. new1->dengji = grade;
278. new1->next = head->next;*//头插入法加入成员名单域*
279. head->next = new1;
280. return;
281. }
282. void fightinfo(hangban \*p)
283. {
284. cout << "起点  终点  编号  飞机号  飞行日期  总载客  余票量  1级余量  2级余量" << endl;
285. cout << p->qidian << "  " << p->zhongdian << "  " << p->hbID << "     " << p->hbnum << "     " << p->day << "     "
286. << p->maxnum << "     " << p->less << "     " << p->lessgrade1 << "     " << p->lessgrade2 << endl;
287. }
288. LinkQueue enqueue(LinkQueue &q, char name[], int need, char id[])
289. *//入队，增加排队等候的客户名单域*
290. {
291. Phb new1;
292. new1 = (Phb) malloc(sizeof(QhbNode));
293. strcpy(new1->name, name);
294. strcpy(new1->ID, id);
295. new1->num = need;
296. new1->next = NULL;
297. q.rear->next = new1;
298. q.rear = new1;
299. return q;
300. }
301. void tuipiao() *//退票功能*
302. {
303. hangban \*info;
304. int grade;
305. CNode \*p1, \*p2, \*head;*//p1为遍历指针，p2指向p1的前驱*
306. char name[20];*//客户姓名*
307. char id[20];*//客户身份证*
308. char day[20];
309. info = find();*//复制节点信息给info，find函数根据航班编号返回该航班节点的指针*
310. while (info == NULL)
311. {
312. cout<<"没有这个航班，请重新输入"<<endl;
313. tuipiao();
314. }
315. head = info->cusLinkList;*//head为该航班的的乘员名单域的头指针*
316. p1 = head->next;    *//带头结点的指针，head->next 开始遍历*
317. cout<<"请输入你的姓名: "<<endl;
318. cin>>name;
319. cout<<"请输入你的身份证号码:"<<endl;
320. cin>>id;
321. cout<<"请输入飞行时间"<<endl;
322. cin>>day;
323. p2 = head;
324. while (p1 != NULL)
325. {
326. *//对比姓名和身份证*
327. if ((strcmp(name, p1->name) == 0) && (strcmp(id, p1->ID) == 0)&&(strcmp(day, p1->day) == 0)) break;
328. p2 = p1;
329. p1 = p1->next;
330. }
331. if (p1 == NULL)
332. {
333. cout<<"对不起，你没有订过票或姓名和身份证,与飞行周期不对应"<<endl;
334. return;
335. }
336. else
337. {*//退票成功*
338. *//从乘客名单域中移除该节点*
339. grade = p1->dengji;
340. p2->next = p1->next;
341. *//加回该航班的剩余票*
342. info->less += p1->num;
343. if (grade == 1)
344. {
345. info->lessgrade1 += p1->num;
346. }
347. else
348. {
349. info->lessgrade2 += p1->num;
350. }
351. cout<<name<<"退票成功"<<endl;
352. }
353. }
354. int zengjiaFlight()
355. {
356. hangban \*q;
357. hangban \*p = pFlight;
358. int y = 1;
359. cout << "请依次输入以下内容" << endl;
360. system("pause");
361. while (y != 0)
362. {
363. q = new hangban;
364. if (q == NULL)
365. return 0;
366. cout << "请输入航班编号" << endl;
367. cin >> q->hbID;
368. int t = duibiFlight(pFlight, q->hbID);
369. if (t == 0)
370. {
371. cout << "该航班编号已经存在，请重新输入航班编号" << endl;
372. continue;
373. }
374. cout << "请输入起点站名" << endl;
375. cin >> q->qidian;
376. cout << "请输入终点站名" << endl;
377. cin >> q->zhongdian;
378. cout << "请输入型号" << endl;
379. cin >> q->hbnum;
380. cout << "请输入飞行日期" << endl;
381. cin >> q->day;
382. cout << "请输入乘客定额" << endl;
383. cin >> q->maxnum;
384. q->less = q->maxnum;
385. cout << "请输入1等票数目" <<endl;
386. cin >> q->lessgrade1;
387. q->lessgrade2 = q->maxnum - q->lessgrade1;
388. q->next = p->next;
389. p->next = q;
390. cout << "是否继续录入航班信息（任意数字继续，0表示停止）。\n";
391. cout << "请输入：";
392. cin >> y;
393. }
394. return 0;
395. }
396. int shanchuFlight()
397. {
398. string ID;
399. cout << "请输入航班ID" << endl;
400. cin >> ID;
401. hangban \*pre = pFlight;
402. hangban \*p = pre->next;
403. while (p != NULL)
404. {
405. if (ID == p->hbID)
406. {
407. pre->next = p->next;
408. delete p;
409. return 1;
410. }
411. pre = p;
412. p = p->next;
413. }
414. return 0;
415. }
416. int duibiFlight(hangban \*flight, char ID[])
417. *//对比航班的编号，防止增加航班时，出现重复*
418. {
419. hangban \*p = flight;
420. while (p != NULL)
421. {
422. if (!strcmp(ID, p->hbID))
423. {
424. return 0;*//航班ID重复*
425. }
426. p = p->next;
427. }
428. *//ID不重复*
429. return 1;
430. }
431. void sousuoFace()
432. {
433. int i;
434. cout<<endl<<endl<<endl<<endl<<endl;
435. cout<<"                                                    ";
436. cout<<"搜索窗口"<<endl;
437. cout<<"                                  ";
438. cout<<"-----------------------------------------------"<<endl;
439. cout<<"                                  ";
440. cout<<"|"<<"                                             "<<"|"<<endl;
441. cout<<"                                  ";
442. cout<<"|"<<"             1.搜索航班                      "<<"|"<<endl;
443. cout<<"                                  ";
444. cout<<"|"<<"                                             "<<"|"<<endl;
445. cout<<"                                  ";
446. cout<<"|"<<"             2.返回上一级                    "<<"|"<<endl;
447. cout<<"                                  ";
448. cout<<"|"<<"                                             "<<"|"<<endl;
449. cout<<"                                  ";
450. cout<<"-----------------------------------------------"<<endl;
451. cout<<"                                  ";
452. cout<<"请输入您的选择(1-2,否则无效！)"<<endl;
453. cout<<"                                  ";
454. cout<<"您的选择是:";
455. cin >> i;
456. switch (i) {
457. case 1:
458. sousuoflight();
459. system("pause");
460. sousuoFace();
461. break;
462. case 2:
463. menu();
464. break;
465. default:
466. sousuoFace();
467. }
468. }
469. void sousuoflight()
470. {
471. string qidian, zhongdian;
472. int flag = 0;
473. cout << "请输入起点站名:";
474. cin >> qidian;
475. cout << "请输入终点站名:";
476. cin >> zhongdian;
477. hangban \*p = pFlight->next;
478. cout << "起点  终点  编号  飞机号  飞行日期  总载客  余票量  1级余量  2级余量" << endl;
479. while (p != NULL) {
480. if (qidian == p->qidian && zhongdian == p->zhongdian)
481. {
482. flag = 1;
483. cout << p->qidian << "  " << p->zhongdian << "  " << p->hbID << "     " << p->hbnum << "     " << p->day << "     "
484. << p->maxnum << "     " << p->less << "     " << p->lessgrade1 << "     " << p->lessgrade2 << endl;
485. }
486. p = p->next;
487. }
488. cout <<endl <<endl;
489. if (flag == 0)
490. cout << "抱歉！不存在该航班！" << endl;
491. }
492. void dingpiaoFace()
493. {
494. int i;
495. cout<<endl<<endl<<endl<<endl<<endl;
496. cout<<"                                                    ";
497. cout<<"订票窗口"<<endl;
498. cout<<"                                  ";
499. cout<<"-----------------------------------------------"<<endl;
500. cout<<"                                  ";
501. cout<<"|"<<"             1.确认订票                      "<<"|"<<endl;
502. cout<<"                                  ";
503. cout<<"|"<<"             2.搜索航班                      "<<"|"<<endl;
504. cout<<"                                  ";
505. cout<<"|"<<"             3.浏览航班                      "<<"|"<<endl;
506. cout<<"                                  ";
507. cout<<"|"<<"             4.返回上级菜单                  "<<"|"<<endl;
508. cout<<"                                  ";
509. cout<<"|"<<"                                             "<<"|"<<endl;
510. cout<<"                                  ";
511. cout<<"-----------------------------------------------"<<endl;
512. cout<<"                                  ";
513. cout<<"请输入您的选择(1-4,否则无效！)"<<endl;
514. cout<<"                                  ";
515. cout<<"您的选择是:";
516. cin >> i;
517. switch (i) {
518. case 1:
519. dingpiao();
520. system("pause");
521. dingpiaoFace();
522. break;
523. case 2:
524. sousuoflight();
525. system("pause");
526. dingpiaoFace();
527. break;
528. case 3:
529. printfall(pFlight);
530. system("pause");
531. dingpiaoFace();
532. break;
533. case 4:
534. menu();
535. break;
536. default:
537. dingpiaoFace();
538. }
539. }
540. void tuipiaoFace()*//退票模块界面*
541. {
542. int i;
543. cout<<endl<<endl<<endl<<endl<<endl;
544. cout<<"                                                    ";
545. cout<<"退票窗口"<<endl;
546. cout<<"                                  ";
547. cout<<"-----------------------------------------------"<<endl;
548. cout<<"                                  ";
549. cout<<"|"<<"                                             "<<"|"<<endl;
550. cout<<"                                  ";
551. cout<<"|"<<"             1.办理退票                      "<<"|"<<endl;
552. cout<<"                                  ";
553. cout<<"|"<<"                                             "<<"|"<<endl;
554. cout<<"                                  ";
555. cout<<"|"<<"             2.返回上级菜单                  "<<"|"<<endl;
556. cout<<"                                  ";
557. cout<<"|"<<"                                             "<<"|"<<endl;
558. cout<<"                                  ";
559. cout<<"-----------------------------------------------"<<endl;
560. cout<<"                                  ";
561. cout<<"请输入您的选择(1-2,否则无效！)"<<endl;
562. cout<<"                                  ";
563. cout<<"您的选择是:";
564. cin >> i;
565. switch (i)
566. {
567. case 1:
568. tuipiao();
569. system("PAUSE");
570. tuipiaoFace();
571. break;
572. case 2:
573. menu();
574. break;
575. default:
576. tuipiaoFace();
577. }
578. }
579. void pricusInfo()
580. {
581. CList p;
582. hangban \*info;
583. info = find(); *// 假设find函数已定义，用于查询航班信息*
584. if (info == NULL)
585. {
586. cout << "没有这个航班" << endl;
587. return;
588. }
589. *// 头结点的下一个节点开始遍历*
590. p = info->cusLinkList->next;
591. if (p != NULL)
592. {
593. cout << "客户姓名   订票数额   舱位等级       身份证号  " << endl;
594. while (p)
595. {
596. cout << p->name << "\t\t" << p->num << "\t" << p->dengji << "\t" <<p->ID<< endl;
597. p = p->next;
598. }
599. }
600. else
601. {
602. cout << "该航班没有客户信息!!" << endl;
603. }
604. }
605. void printfall(hangban \*pflight)
606. {
607. hangban \*p;
608. p = pflight->next;
609. cout << "起点  终点  编号  飞机型号  飞行日期  总载客  余票量  1级余量  2级余量 " << endl;
610. while (p != NULL)
611. {
612. cout << p->qidian << "  " << p->zhongdian << "  " << p->hbID << "      " << p->hbnum << "     " << p->day << "   "
613. << p->maxnum << "     " << p->less << "     " << p->lessgrade1 << "      " << p->lessgrade2 <<"      "<< endl;
614. p = p->next;
615. }
616. cout << endl << endl;
617. }
618. int yonghudenglu()
619. {
620. string account1="黄暄";
621. string passwords1="239007045";
622. string account2="王峰";
623. string passwords2="239007038";
624. string account3="邓浩宇";
625. string passwords3="239007049";
626. string account4="陈记德";
627. string passwords4="239007037";
628. string account5="徐嘉诚";
629. string passwords5="239007036";
630. string account;
631. string passwords;
632. cout<<endl<<endl<<endl<<endl<<endl;
633. cout<<"                                                  ";
634. cout<<"用户登录页面"<<endl;
635. cout<<"                                  ";
636. cout<<"-----------------------------------------------"<<endl;
637. cout<<"                                  ";
638. cout<<"|"<<"                                             "<<"|"<<endl;
639. cout<<"                                  ";
640. cout<<"|"<<"                                             "<<"|"<<endl;
641. cout<<"                                  ";
642. cout<<"|"<<"      请输入正确的账号与密码以进入系统       "<<"|"<<endl;
643. cout<<"                                  ";
644. cout<<"|"<<"          (返回首页请输入exit)               "<<"|"<<endl;
645. cout<<"                                  ";
646. cout<<"|"<<"                                             "<<"|"<<endl;
647. cout<<"                                  ";
648. cout<<"-----------------------------------------------"<<endl;
649. cout<<"                                  ";
650. cout<<"请输入账号:"<<endl;
651. cout<<"                                  ";
652. cout<<"account:";
653. cin>>account;
654. cout<<"                                  ";
655. if(account=="exit")
656. {
657. denglu();
658. }
659. cout<<"请输入密码:"<<endl;
660. cout<<"                                  ";
661. cout<<"password:";
662. cin>>passwords;
663. if((account==account1&&passwords==passwords1)||(account==account2&&passwords==passwords2)||
664. (account==account3&&passwords==passwords3)||(account==account4&&passwords==passwords4)||
665. (account==account5&&passwords==passwords5))
666. {
667. menu();
668. }
669. else
670. {
671. cout<<"                                  ";
672. cout<<"输入错误，按任意键重新输入"<<endl;
673. }
674. }
675. int guanliyuandenglu()
676. {
677. string passwords="123456";
678. string shuru;
679. cout<<endl<<endl<<endl<<endl<<endl;
680. cout<<"                                                 ";
681. cout<<"管理员登录页面"<<endl;
682. cout<<"                                  ";
683. cout<<"-----------------------------------------------"<<endl;
684. cout<<"                                  ";
685. cout<<"|"<<"                                             "<<"|"<<endl;
686. cout<<"                                  ";
687. cout<<"|"<<"                                             "<<"|"<<endl;
688. cout<<"                                  ";
689. cout<<"|"<<"        请输入管理员密码以验证身份           "<<"|"<<endl;
690. cout<<"                                  ";
691. cout<<"|"<<"          (返回首页请输入exit)               "<<"|"<<endl;
692. cout<<"                                  ";
693. cout<<"|"<<"                                             "<<"|"<<endl;
694. cout<<"                                  ";
695. cout<<"-----------------------------------------------"<<endl;
696. cout<<"                                  ";
697. cout<<"请输入密码:"<<endl;
698. cout<<"                                  ";
699. cout<<"password:";
700. cin>>shuru;
701. if(shuru==passwords)
702. {
703. cout<<"                                  ";
704. cout<<"管理员身份验证成功！"<<endl;
705. guanliyuanyemian();
706. }
707. else if(shuru=="exit")
708. {
709. denglu();
710. }
711. else
712. {
713. cout<<"                                  ";
714. cout<<"身份验证失败，按任意键重新输入"<<endl;
715. system("pause");
716. guanliyuandenglu();
717. }
718. }
719. int guanliyuanyemian()
720. {
721. int i;
722. do{
723. cout<<endl<<endl<<endl<<endl<<endl;
724. cout<<"                                                    ";
725. cout<<"管理员页面"<<endl;
726. cout<<"                                  ";
727. cout<<"-----------------------------------------------"<<endl;
728. cout<<"                                  ";
729. cout<<"|"<<"                                             "<<"|"<<endl;
730. cout<<"                                  ";
731. cout<<"|"<<"             1.插入航班                      "<<"|"<<endl;
732. cout<<"                                  ";
733. cout<<"|"<<"             2.删除航班                      "<<"|"<<endl;
734. cout<<"                                  ";
735. cout<<"|"<<"             3.浏览航班                      "<<"|"<<endl;
736. cout<<"                                  ";
737. cout<<"|"<<"             4.查询客户信息                  "<<"|"<<endl;
738. cout<<"                                  ";
739. cout<<"|"<<"             5.返回首页                      "<<"|"<<endl;
740. cout<<"                                  ";
741. cout<<"-----------------------------------------------"<<endl;
742. cout<<"                                  ";
743. cout<<"请输入您的选择(1-2,否则无效！)"<<endl;
744. cout<<"                                  ";
745. cout<<"您的选择是:";
746. cin >> i;
747. switch (i) {
748. case 1:
749. zengjiaFlight();
750. guanliyuanyemian();
751. break;
752. break;
753. case 2:
754. if (1 == shanchuFlight())
755. {
756. cout << "删除成功" << endl;
757. }
758. else
759. {
760. cout << "没有这个航班，删除失败！" << endl;
761. }
762. system("pause");
763. guanliyuanyemian();
764. break;
765. case 3:
766. printfall(pFlight);
767. system("pause");
768. guanliyuanyemian();
769. break;
770. case 4:
771. pricusInfo();
772. system("pause");
773. guanliyuanyemian();
774. break;
775. case 5:
776. denglu();
777. break;
778. default:
779. guanliyuanyemian();
780. }
781. } while (true);
782. }
783. int denglu()
784. {
785. int i;
786. do {
787. cout<<endl<<endl<<endl;
788. cout<<"title：航空客运订票系统-江苏师范大学数据结构课题作业-24.5"<<endl;
789. cout<<"author：黄暄，王锋，徐嘉诚,邓浩宇，陈记德";
790. cout<<endl<<endl<<endl;
791. cout<<"                                                     ";
792. cout<<"登录页面"<<endl;
793. cout<<"                                  ";
794. cout<<"-----------------------------------------------"<<endl;
795. cout<<"                                  ";
796. cout<<"|"<<"                                             "<<"|"<<endl;
797. cout<<"                                  ";
798. cout<<"|"<<"             1.用户登录                      "<<"|"<<endl;
799. cout<<"                                  ";
800. cout<<"|"<<"                                             "<<"|"<<endl;
801. cout<<"                                  ";
802. cout<<"|"<<"             2.管理员登录                    "<<"|"<<endl;
803. cout<<"                                  ";
804. cout<<"|"<<"                                             "<<"|"<<endl;
805. cout<<"                                  ";
806. cout<<"-----------------------------------------------"<<endl;
807. cout<<"                                  ";
808. cout<<"请输入您的选择(1-2,否则无效！)"<<endl;
809. cout<<"                                  ";
810. cout<<"您的选择是:";
811. cin >> i;
812. switch (i) {
813. case 1:
814. yonghudenglu();
815. system("pause");
816. break;
817. case 2:
818. guanliyuandenglu();
819. system("pause");
820. break;
821. default:
822. denglu();
823. }
824. } while (true);
825. }
826. int menu()
827. {
828. int i;
829. do {
830. cout<<endl<<endl<<endl<<endl<<endl;
831. cout<<"                                             ";
832. cout<<"欢迎来到航空客运订票系统"<<endl;
833. cout<<"                                  ";
834. cout<<"-----------------------------------------------"<<endl;
835. cout<<"                                  ";
836. cout<<"|"<<"             1.查询航班                      "<<"|"<<endl;
837. cout<<"                                  ";
838. cout<<"|"<<"             2.搜索航班                      "<<"|"<<endl;
839. cout<<"                                  ";
840. cout<<"|"<<"             3.订票                          "<<"|"<<endl;
841. cout<<"                                  ";
842. cout<<"|"<<"             4.退票                          "<<"|"<<endl;
843. cout<<"                                  ";
844. cout<<"|"<<"             5.退出系统                      "<<"|"<<endl;
845. cout<<"                                  ";
846. cout<<"-----------------------------------------------"<<endl;
847. cout<<"                                  ";
848. cout<<"请输入您的选择(1-5,否则无效！)"<<endl;
849. cout<<"                                  ";
850. cout<<"您的选择是:";
851. cin >> i;
852. switch (i) {
853. case 1:
854. printfall(pFlight);
855. system("pause");
856. menu();
857. break;
858. case 2:
859. sousuoFace();
860. system("pause");
861. menu();
862. break;
863. case 3:
864. dingpiaoFace();  *// 订票界面*
865. system("pause");
866. menu();
867. break;
868. case 4:
869. tuipiaoFace();
870. system("pause");
871. menu();
872. break;
873. case 5:
874. denglu();
875. default:
876. menu();
877. }
878. } while (true);
879. }
880. int main()
881. {
882. system("color F1");
883. initFlight();*//初始化航班链表*
884. Create(flight1);*//将航标信息插入链表*
885. denglu();    *//菜单*
886. return 0;
887. }