杭州电子科技大学

HANGZHOU DIANZI UNIVERSITY

数据结构课程设计报告

(2021-2022-2 学期)

题目	航空客运订票系统			
学院	理学院			
专业	信息与计算科学			
组号	第十六组	学号	姓名	分工
		20071226	童繁	流程图
		20071227	王瀚功	数据
		20071228	王赛豪	文案
		20071229	吴政豪	调试
		20071230	武琦	代码
时间	2022 年 4 月 21 日			

目录

1. 需求分析

(1) 设计一个航空客运订票系统，实现查询航线、客票预定和办理退票等功能。

(2) 航线信息包括站名、机场、航空公司、航班号、飞机号、飞行时间、乘员定额、余票量、已订票的客户名单以及等候替补的客户名单。

(3) 查询航线：根据旅客提出的起飞地-目的地和起飞日期输出航班信息，并内嵌订票功能。

(4) 承办订票业务：根据客户要求，若有余票，办理订票手续并输出座位号；若已满员或余票额少于订票额，则需重新询问客户要求。若需要，可登记候补。

(5) 承办退票业务：根据客户的身份证号，输出所有的订票信息，客户选择退票航班，为客户办理退票手续。然后查询该航班是否有人排队候补，首先询问排在第一的客户，若所退票额能满足他的要求，则为他办理订票手续，否则依次询问剩余候补的客户。

(6) 当客户订票要求不能满足时，系统可向客户提供中转航线。

2. 项目亮点

(1) 建立了航线信息库和客户名单库两个 csv 文件，方便管理员调用和修改信息；

(2) 建立了独立的类型库和功能库，使得项目的调用更加清晰合理；

(3) 在查询功能中内嵌了订票功能，整个系统遵循现实订票的逻辑和流程；

(4) 在订票时提供了中转的路线推荐；

(5) 客户可以根据自己的身份证号查询自己的所有订单，并选择想退的订单；

(6) 解决了客户订票时与自己所有订单的时间冲突问题。

(7) 客户可以组团订票。

3. 概要设计

利用 airline.csv 存储航线信息，利用 client.csv 存储客户信息：航线序号、姓名、身份证号、飞机舱位。每次打开系统首先读取两个文件，每次关闭系统前重新录入更新后的信息。

订单中转则是找出满足地点条件和日期条件的一组航班（包含两次航班，一个中转站）。当前一个航班的到达时间与下一个航班的起飞时间在同一天且间隔时间大于一小时时，则为可行方案，待用户选择后分别下单这两次航班即可。

时间冲突问题是客户下单时选择的航班与客户已经订票成功的航班之间有时间交集，客户不可能同时在两趟航班上乘坐，若时间不冲突，则客户可以正常下单。

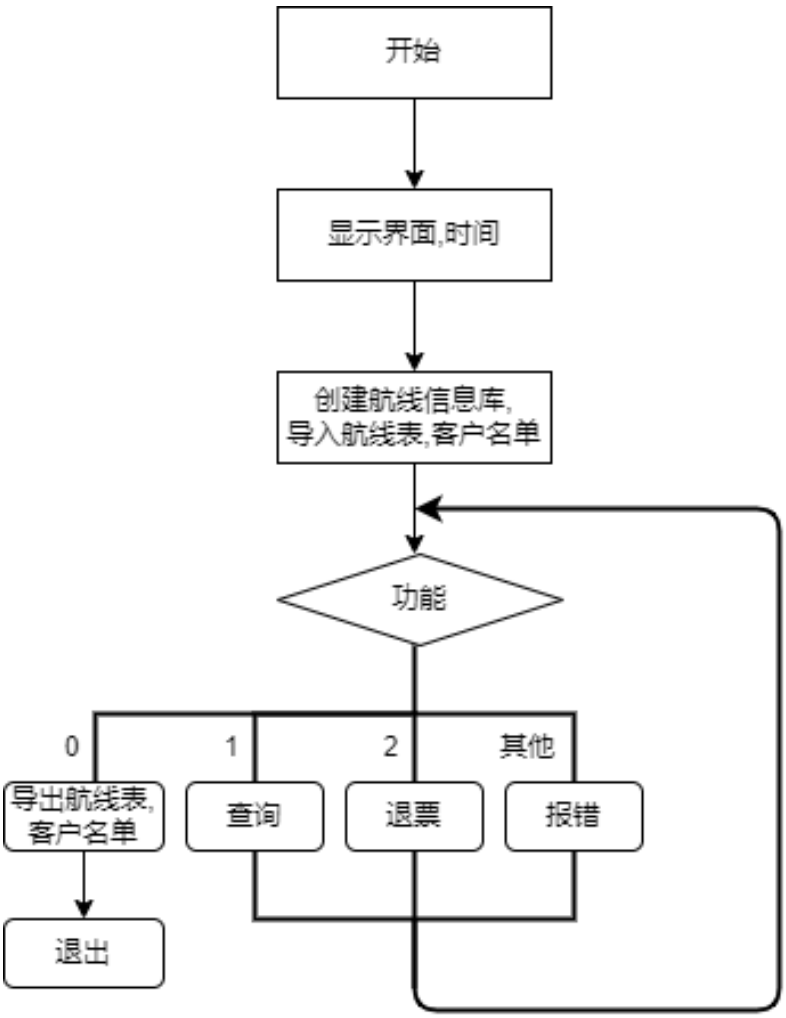

图 1: 主函数流程图

4. 详细设计

4.1 定义

```
#ifndef DEFINITION
#define DEFINITION
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

#define OK 1
#define ERROR 0
#define OVERFLOW -1
FILE *fp;//文件指针
typedef int Status;//函数封装
```

```
13
14 typedef struct passenger
15 {
16     char name[50];//姓名
17     char ID[50];//身份证号码
18     char grade;//飞机舱位 头等舱F,商务舱C,经济舱Y
19 }Client;//乘客信息
20
21 typedef struct node
22 {
23     Client C;
24     struct node *next;
25 }node,*List;//乘客名单
26
27 typedef struct
28 {
29     List front;
30     List rear;
31 }Queue;//候补队列
32
33 typedef struct
34 {
35     int number;//序号
36     char flightNum[10];//航班号
37     char planeNum[10];//飞机号
38     char departure[20];//始发站
39     char destination[20]; //目的地
40     char company[20];//航空公司
41     char airport[20];//起飞机场
42     char nextairport[20];//降落机场
43     int price;//经济舱基础票价（头等舱价格为基础票价150%，商务舱价格为基础票价的130%）
44     int month;
45     int day;
46     int hour;//起飞时间
47     int min;
48     int nexthour;//抵达时间
49     int nextmin;
50     int date;//星期
51     int capacity;//载客量 一条航线的舱位中有0.1的头等舱，0.2的商务舱，其余为经济舱
52     int F;//头等舱余票量
53     int C;//商务舱余票量
54     int Y;//经济舱余票量
55     List L;//乘客名单
56     Queue Q;//等候替补队列
57 }Airline;//航线信息
58
59 //List相关函数
```

```
60  Status Create(List *B);//创建乘客名单
61  Status OrderInSert(List B,Client E);//客户信息放入名单
62  Status Search(List B,char id[],Client *E);//查询客户订票信息
63  Status Delete(List B,Client E);//名单移除订票信息
64  Status EntryList();//录入客户名单
65  Status OutList();//存储客户名单
66
67  //Queue相关函数
68  Status InitQueue(Queue *Q);//构造一个空队列
69  Status EnQueue(Queue *Q,Client E);//插入队尾元素
70  Status DeQueue(Queue *Q,Client *E);//删除队头元素
71  Status QueueEmpty(Queue *Q);//判断队列为空
72  Status CheckQueue(Queue *Q,char space,Client
        *E);//检查退票是否满足候补客户需求，并排队候补
73
74  //Airline相关函数
75  int airnum;//航线数
76  Airline transit[2][1000]={0};//中转航班序号库
77  Airline A[10000]={0};//航线信息库
78  char *week[7]={"星期天","星期一","星期二","星期三","星期四","星期五","星期六"};//星期
79  Status EntryAirline();//录入航线信息
80  Status OutAirline();//存储航线信息
81  Status PrintAirline(Airline *A);//打印航线信息
82  Status AirlineReady();//构建航线对应的线性表和队列
83  Status AirlineBack(Airline *B,char C);//退票后补位
84
85  //inquire函数
86  Status Query();//航线查询
87  Status Booking(char place1[15],char place2[15]);//订票
88  Status Number();//选择序号
89  Status PlaceOrder(Airline *B);//下单
90  Status FillBlank(Client *C);//客户信息录入系统
91  Status Transit(char place1[],char place2[],int m,int n);//订单中转
92  Status Day(Airline *B,Airline *D);//判断中转时到达时间和起飞时间是否在同一天
93  Status MonthDays(int month,int day);//计算日期在一年中的天数
94  Status TransitNum(int k);//中转订票
95  Status Conflict(char id[],Airline *B);//判断订票时间是否冲突
96  Status DayConflit(Airline *B,Airline *D);//判断时间冲突
97  Status TranMin(Airline *B,int min[2]);//时间转换分钟
98
99  //refund函数
100 Status Information();//显示信息
101 Status Refund(char id[]);//退票
102 #endif
```

4.2 类型库

4.2.1 List 库

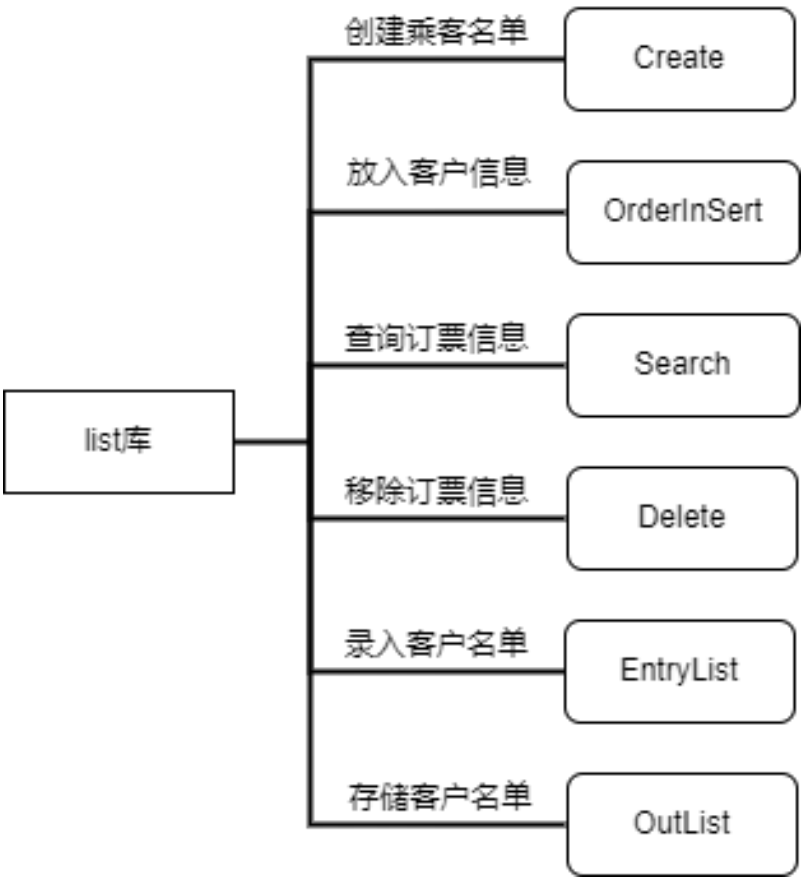

图 2: List 库示意图

```
Status Create(List *B)//创建乘客名单
{
    (*B)=(List)malloc(sizeof(node));
    (*B)->next=NULL;
    return OK;
}

Status OrderInSert(List B,Client E)//客户信息放入名单
{
    List L=B;
    while(L->next!=NULL)
    {
        L=L->next;

    }
```

```
16      List S=(List)malloc(sizeof(node));
17      S->C=E;
18      S->next=NULL;
19      L->next=S;
20      return OK;
21  }
22
23  Status Search(List B,char id[],Client *E)//查询客户订票信息
24  {
25      List L=B->next;
26      int flag=0;
27      while(L!=NULL)
28      {
29          if(strcmp(L->C.ID,id)==0)
30          {
31              *E=L->C;
32              flag=1;
33              break;
34          }
35          L=L->next;
36      }
37      if(flag==0) return ERROR;
38      return OK;
39  }
40
41  Status Delete(List B,Client E)//名单移除订票信息
42  {
43      List L=B;
44      while(L->next!=NULL)
45      {
46          List M=L->next;
47          if(strcmp(E.ID,M->C.ID)==0)
48          {
49              L->next=M->next;
50              free(M);
51              printf("成功办理退票!\n");
52              break;
53          }
54          L=L->next;
55      }
56      return OK;
57  }
58
59  Status EntryList()//录入客户名单
60  {
61      Client C;
62     char *line,*record,buffer[1024];
```

```
63    if((fp=fopen("client.csv","r+"))==NULL) return ERROR;
64    fseek(fp,29L,SEEK_SET);
65    while((line=fgets(buffer,sizeof(buffer),fp))!=NULL)
66    {
67       record=strtok(line,",");
68         int i=atoi(record);
69         record=strtok(NULL, ",");
70       if(record!=NULL)
71       {
72          sprintf(C.name,"%s",record);record=strtok(NULL, ",");
73             sprintf(C.ID,"%s",record);record=strtok(NULL, ",");
74             C.grade=record[0];record=strtok(NULL, ",");
75       }
76         OrderInSert(A[i-1].L,C);
77    }
78    fclose(fp);
79    return OK;
80 }
81
82 Status OutList()//存储客户名单
83 {
84     if((fp=fopen("client.csv","w+"))==NULL) return ERROR;
85     fprintf(fp,"序号,姓名,身份证号,飞机舱位\n");
86     for(int i=0;i<airnum;i++)
87     {
88        List L=A[i].L->next;
89        while(L!=NULL)
90        {
91           fprintf(fp,"%d,%s,%s,%c\n",A[i].number,L->C.name,L->C.ID,L->C.grade);
92           L=L->next;
93        }
94     }
95     return OK;
96 }
```

4.2.2 Queue 库

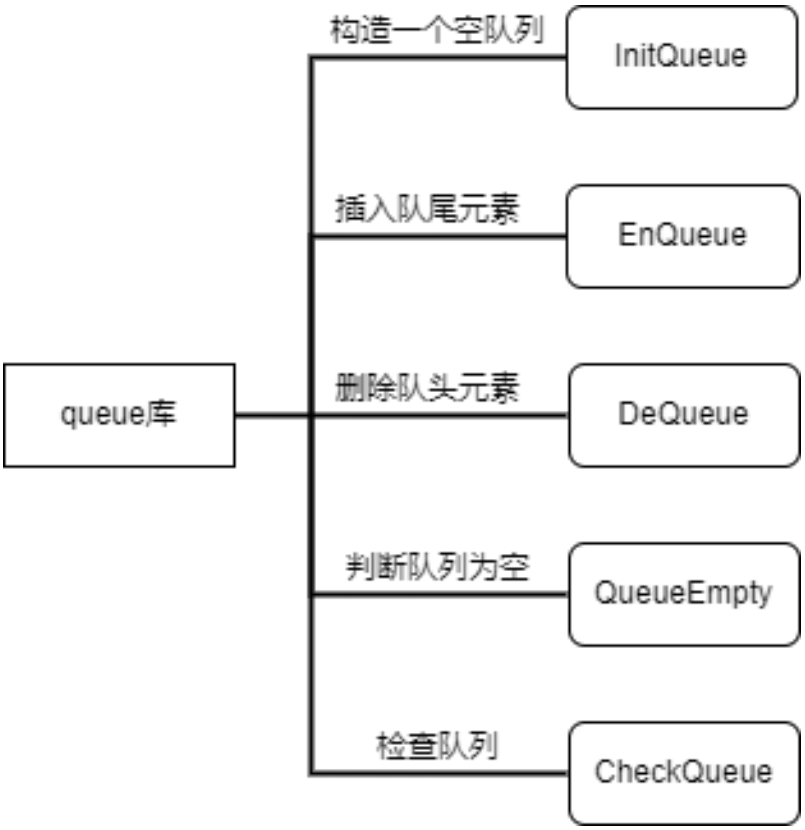

图 3: Queue 库示意图

```
Status InitQueue(Queue *Q)//构造一个空队列
{
    Q->front=(List)malloc(1*sizeof(node));
    if(!Q->front) exit(OVERFLOW);
    Q->rear=Q->front;
    Q->front->next=NULL;
    return OK;
}

Status EnQueue(Queue*Q,Client E)//插入元素E为Q的新的队尾元素
{
    List new=(List)malloc(sizeof(node));
    if(!new) exit(OVERFLOW);
    new->C=E;
    new->next=NULL;
    Q->rear->next=new;
    Q->rear=new;
    return OK;
}

```

```
21 Status DeQueue(Queue *Q,Client
       *E)//若队列不空，则删除Q的队头元素，用E返回其值，并返回OK；否则返回ERROR
22 {
23     if(Q->front==Q->rear) return ERROR;
24     List head=Q->front->next;
25     *E=head->C;
26     Q->front->next=head->next;
27     if(Q->rear==head) Q->rear=Q->front;
28     free(head);
29     return OK;
30 }
31
32 Status QueueEmpty(Queue *Q)//若队列Q为空队列，则返回OK；否则返回ERROR
33 {
34     if(Q->front==Q->rear) return OK;
35     return ERROR;
36 }
37
38 Status CheckQueue(Queue *Q,char space,Client
       *E)//检查退票是否满足候补客户需求，并排队候补
39 {
40     Queue S;
41     InitQueue(&S);
42     int flag=0;
43     while(!QueueEmpty(Q))
44     {
45         Client D;
46         DeQueue(Q,&D);
47         if(D.grade==space)
48         {
49             *E=D;
50             flag=1;
51             space='\0';
52         }
53         else
54             EnQueue(&S,D);
55     }
56     *Q=S;
57     if(flag==0) return ERROR;
58     return OK;
59 }
```

4.2.3 Airline 库

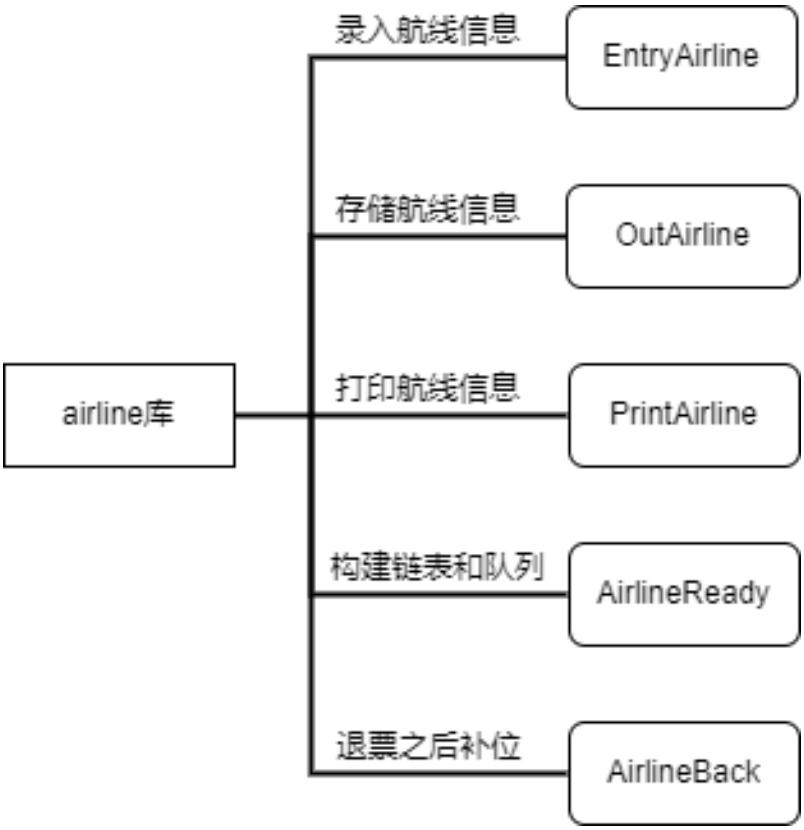

图 4: Airline 库示意图

```
Status EntryAirline()//录入航线信息
{
   char *line,*record,buffer[1024];
    int i=0;
   if((fp=fopen("airline.csv","r+"))==NULL) return ERROR;
   fseek(fp,127L,SEEK_SET);
   while((line=fgets(buffer,sizeof(buffer),fp))!=NULL)
   {
       record=strtok(line,",");
      if(record!=NULL)
      {
           A[i].number=atoi(record);record=strtok(NULL, ",");
           sprintf(A[i].departure,"%s",record);record=strtok(NULL, ",");
           sprintf(A[i].destination,"%s",record);record=strtok(NULL, ",");
           sprintf(A[i].flightNum,"%s",record);record=strtok(NULL, ",");
           sprintf(A[i].company,"%s",record);record=strtok(NULL, ",");
           sprintf(A[i].planeNum,"%s",record);record=strtok(NULL, ",");
        A[i].price=atoi(record);record=strtok(NULL, ",");
          A[i].capacity=atoi(record);record=strtok(NULL, ",");
        A[i].F=atoi(record);record=strtok(NULL, ",");
```

```
        A[i].C=atoi(record);record=strtok(NULL, ",");
        A[i].Y=atoi(record);record=strtok(NULL, ",");
           sprintf(A[i].airport,"%s",record);record=strtok(NULL, ",");
           sprintf(A[i].nextairport,"%s",record);record=strtok(NULL, ",");
        A[i].month=atoi(record);record=strtok(NULL, ",");
        A[i].day=atoi(record);record=strtok(NULL, ",");
           A[i].date=atoi(record);record=strtok(NULL, ",");
        A[i].hour=atoi(record);record=strtok(NULL, ",");
        A[i].min=atoi(record);record=strtok(NULL, ",");
           A[i].nexthour=atoi(record);record=strtok(NULL, ",");
           A[i].nextmin=atoi(record);record=strtok(NULL, ",");
      }
      i++;
   }
    airnum=i;
   fclose(fp);
   return OK;
}

Status OutAirline()//存储航线信息
{
    int i;
    if((fp=fopen("airline.csv","w+"))==NULL) return ERROR;
   fprintf(fp,"序号,出发地,抵达地,航班号,航空公司,飞机号,起步价,载客量,头等舱,商务舱,
    经济舱,起飞机场,降落机场,月,日,星期,时,分,抵达时,抵达分\n");
    for(i=0;i<airnum;i++) fprintf(fp,"%d,%s,%s,%s,%s,%s,%d,%d,%d,%d,%d,%s,%s,%d,
    %d,%d,%d,%d,%d,%d\n",A[i].number,A[i].departure,A[i].destination,A[i].flightNum,
    A[i].company,A[i].planeNum,A[i].price,A[i].capacity,A[i].F,A[i].C,A[i].Y,
    A[i].airport,A[i].nextairport,A[i].month,A[i].day,A[i].date,A[i].hour,A[i].min,
    A[i].nexthour,A[i].nextmin);
    return OK;
}

Status PrintAirline(Airline *A)//打印航线信息
{
    printf("%s\t%s\t%s\t%s\t",A->departure,A->destination,A->flightNum,A->company);
    printf("%02d月%02d号\t%02d:%02d\t\t%02d:%02d\t\t",A->month,A->day,A->hour,A->min,
    A->nexthour,A->nextmin);
    printf("%-d\t%-d\t%-d\t",A->F,A->C,A->Y);
    printf("%-d元\t%-d\t",A->price,A->F+A->C+A->Y);
    int i=A->date;
    printf("%-s\t\n",week[i]);
    return OK;
}

Status AirlineReady()//构建航线对应的线性表和队列
{
```

```
68      for(int i=0;i<airnum;i++)
69      {
70          Create(&(A[i].L));
71          InitQueue(&(A[i].Q));
72      }
73      return OK;
74  }
75
76  Status AirlineBack(Airline *B,char C)//退票后补位
77  {
78      switch(C)
79      {
80          case 'F':B->F++;break;
81          case 'C':B->C++;break;
82          case 'Y':B->Y++;break;
83      }
84      return OK;
85  }
```

4.3 功能库

4.3.1 查询库

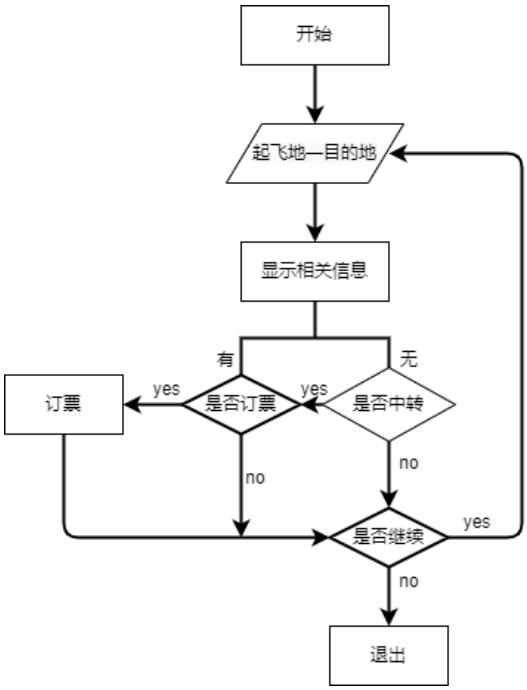

图 5: 查询流程图

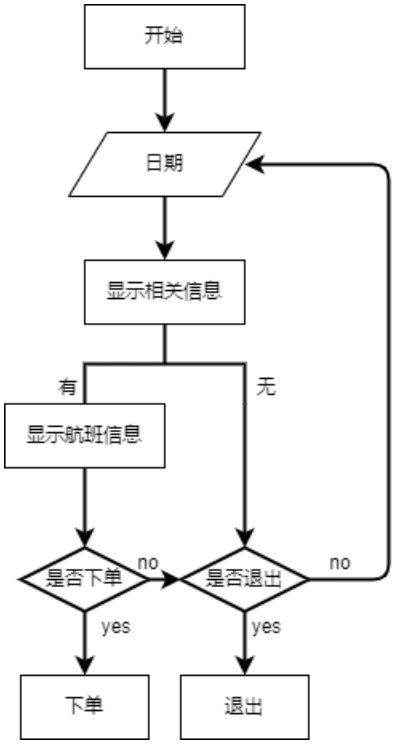

图 6: 订票流程图

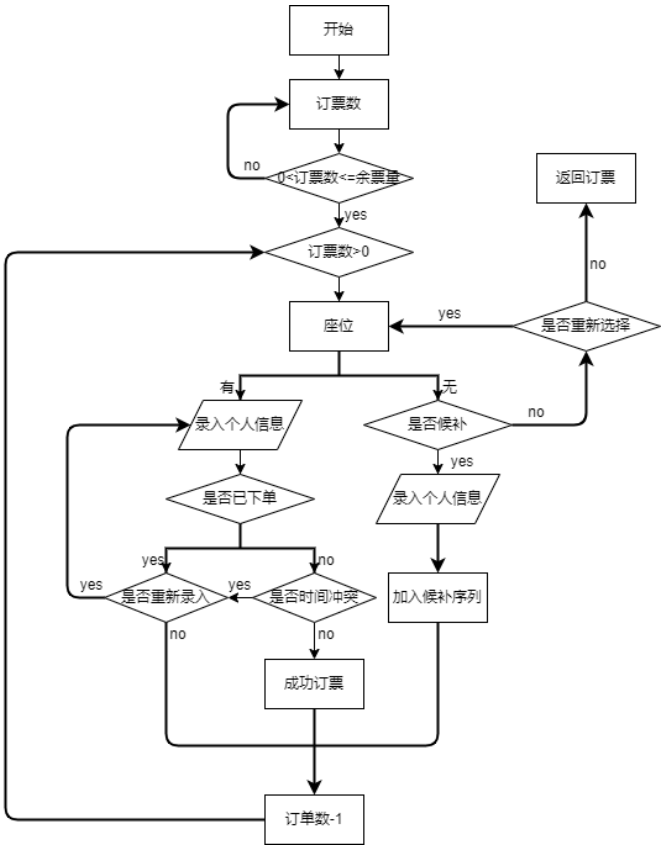

图 7: 下单流程图

```
Status Query()//航线查询
{
    while(1)
    {
        char place1[15],place2[15],choice;
        int i,count=0;
        printf("请输入起飞地:");
        scanf("%s",place1);
        printf("请输入目的地:");
        scanf("%s",place2);
        //printf("出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t头等舱\t
        //商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
        for(i=0;i<airnum;i++)
        {
            if(strcmp(place1,A[i].departure)==0&&
            strcmp(place2,A[i].destination)==0&&A[i].F+A[i].C+A[i].Y>0)
            {
                //PrintAirline(&A[i]);
                count++;
            }
        }
        if(count>0)
        {
            printf("最近共有%d趟航班从%s直接飞往%s\n",count,place1,place2);
            printf("是否订票(y|n):");
            scanf(" %c",&choice);
            if(choice=='y') Booking(place1,place2);
        }
        else
        {
            printf("最近没有从%s直接飞往%s的航班\n",place1,place2);
            printf("是否考虑中转航班(y|n):");
            scanf(" %c",&choice);
            if(choice=='y') Booking(place1,place2);
        }
        printf("是否继续查询(y|n):");
        scanf(" %c",&choice);
        if(choice=='y') continue;
        else break;
    }
    return OK;
}

Status Booking(char place1[15],char place2[15])//订票
{
    printf("<<<<<<<<<<<<<<<您正在办理订票业务>>>>>>>>>>>>\n");
    while(1)
```

```
48      {
49          int m,n,i,k,count=0;
50          char choice;
51          printf("请输入日期(00-00):");
52          scanf("%d-%d",&m,&n);
53          printf("序号\t出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t
54          头等舱\t商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
55          for(i=0;i<airnum;i++)
56          {
57              if(m==A[i].month&&n==A[i].day&&strcmp(place1,A[i].departure)==0&&
58              strcmp(place2,A[i].destination)==0)
59              {
60                  printf("%d\t",A[i].number);
61                  PrintAirline(&A[i]);
62                  count++;
63              }
64          }
65          k=Transit(place1,place2,m,n);
66          count+=k;
67          if(count>0)
68          {
69              if(k)
70              {
71                  printf("是否选择中转航班(y|n):");
72                  scanf(" %c",&choice);
73                  if(choice=='y') TransitNum(k);
74                  else Number();
75              }
76              else Number();
77          }
78          else printf("最近没有%d月%d日的目标航班\n",m,n);
79          printf("是否退出订票(y|n):");
80          scanf(" %c",&choice);
81          if(choice=='n') continue;
82          else break;
83      }
84      return OK;
85  }
86
87  Status Number()//选择序号
88  {
89      int i,flag=0;
90      char choice;
91      while(1)
92      {
93          int num;
94          printf("请输入订票序号:");
```

```
95          scanf("%d",&num);
96          for(i=0;i<airnum;i++)
97          {
98              if(num==A[i].number)
99              {
100                 flag=1;
101                 break;
102             }
103         }
104         if(flag)
105         {
106             printf("|头等舱%d席|商务舱%d席|经济舱%d席|\n",A[i].F,A[i].C,A[i].Y);
107             printf("是否下单(y|n):");
108             scanf(" %c",&choice);
109             if(choice=='y')
110             {
111                 PlaceOrder(&A[i]);
112                 break;
113             }
114             else break;
115         }
116         else
117         {
118             printf("\n是否愿意选择该日期的其他航班(y|n):");
119             scanf(" %c",&choice);
120             if(choice=='y') continue;
121             else break;
122         }
123     }
124     return OK;
125 }
126
127 Status PlaceOrder(Airline *B)//下单
128 {
129     char choice;
130     Client C,E;
131     int count;
132     while(1)
133     {
134         printf("请输入订票数:");
135         scanf("%d",&count);
136         if(B->F+B->C+B->Y<=count)
137         {
138             printf("当前余票%d张,是否重新输入订票数(y|n):",B->F+B->C+B->Y);
139             scanf(" %c",&choice);
140             if(choice=='y') continue;
141             else
```

```
142             {
143                 printf("人数过多,请考虑其他航班\n");
144                 return OK;
145             }
146         }
147         else if(count<=0)
148         {
149             printf("输入错误!\n");
150             continue;
151         }
152         else break;
153     }
154     while(count)
155     {
156         int flag=0;
157         printf("请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):");
158         scanf(" %c",&C.grade);
159         switch(C.grade)
160         {
161             case 'F':if(B->F==0) printf("当前没有头等舱席位\n");else
                    {B->F--;flag=1;}break;
162             case 'C':if(B->C==0) printf("当前没有商务舱席位\n");else
                    {B->C--;flag=1;}break;
163             case 'Y':if(B->Y==0) printf("当前没有经济舱席位\n");else
                    {B->Y--;flag=1;}break;
164             default:printf("输入有误!\n");flag=2;break;
165         }
166         if(flag==1)
167         {
168             while(1)
169             {
170                 FillBlank(&C);
171                 if(Search(B->L,C.ID,&E))
172                 {
173                     printf("身份证号为%s的顾客已下单本次航班,是否重新录入顾客信息(y|n):",
174                     C.ID);
175                     scanf(" %c",&choice);
176                     if(choice=='y') continue;
177                     else break;
178                 }
179                 else
180                 {
181                     if(Conflict(C.ID,B))
182                     {
183                         OrderInSert(B->L,C);
184                         printf("您成功订票!\n");
185                         break;
```

```
186                 }
187                 else
188                 {
189                     printf("\n本次订单与您的这次订单时间冲突,请选择其他航班!\n");
190                     break;
191                 }
192             }
193         }
194     }
195     else if(flag>1) break;
196     else
197     {
198         printf("是否继续愿意登记候补(y|n):");
199         scanf(" %c",&choice);
200         if(choice=='y')
201         {
202             FillBlank(&C);
203             EnQueue(&(B->Q),C);
204             printf("您目前在排队候补!\n");
205         }
206         else
207         {
208             printf("是否重新选择席位(y|n):");
209             scanf(" %c",&choice);
210             if(choice=='y') continue;
211             else break;
212         }
213     }
214     count--;
215 }
216 return ERROR;
217 }
218
219 Status FillBlank(Client *C)//客户信息录入系统
220 {
221     printf("请填写个人信息:\n");
222     printf("请输入您的姓名:");
223     scanf("%s",C->name);
224     printf("请输入您的身份证号码:");
225     scanf("%s",C->ID);
226     return OK;
227 }
228
229 Status Transit(char place1[],char place2[],int m,int n)//订单中转
230 {
231     int i,j,k=0;
232     printf("\n中转号\t出发\t中转\t抵达\t起飞日期\t起飞时间\t到达时间\t中转日期\t
```

```
233     中转起飞时间\t中转到达时间\n");
234     for(i=0;i<airnum;i++)
235     {
236         if(strcmp(A[i].departure,place1)==0&&A[i].month==m&&A[i].day==n)
237         {
238             for(j=0;j<airnum;j++)
239             {
240                 if(strcmp(A[i].destination,A[j].departure)==0&&Day(&A[i],&A[j])==1
241                 &&strcmp(A[j].destination,place2)==0)
242                 {
243                     printf("%d\t",k+1);
244                     printf("%s\t%s\t%s\t",A[i].departure,A[i].destination,
245                     A[j].destination);
246                     printf("%02d月%02d日\t%02d:%02d\t\t%02d:%02d\t\t%02d月%02d日\t
247                     %02d:%02d\t\t%02d:%02d\n",A[i].month,A[i].day,A[i].hour,A[i].min,
248                     A[i].nexthour,A[i].nextmin,A[j].month,A[j].day,A[j].hour,A[j].min,
249                     A[j].nexthour,A[j].nextmin);
250                     transit[0][k]=A[i];
251                     transit[1][k]=A[j];
252                     k++;
253                 }
254             }
255         }
256     }
257     if(k) printf("共有%d趟中转航班\n",k);
258     else printf("没有中转航班!\n");
259     return k;
260 }
261
262 Status Day(Airline *B,Airline *D)//判断中转时到达时间和起飞时间是否在同一天
263 {
264     int day;
265     if(B->nexthour>=B->hour) day=B->day;
266     else day=B->day+1;
267     if(MonthDays(B->month,day)==MonthDays(D->month,D->day))
268     {
269         if(B->nexthour+1<D->hour) return OK;
270         else if(B->nexthour+1==D->hour)
271         {
272             if(B->nextmin<D->min) return OK;
273         }
274     }
275     return ERROR;
276 }
277
278 Status MonthDays(int month,int day)//计算日期在一年中的天数
279 {
```

```
280     int months[12]={31,28,31,30,31,30,31,31,30,31,30,31};
281     int i,days=0;
282     for(i=0;i<month-1;i++)
283         days+=months[i];
284     days+=day;
285     return days;
286 }
287 
288 Status TransitNum(int k)//中转订票
289 {
290     int i,x;
291     char choice;
292     printf("请注意中转时间,以防误机!\n");
293     while(1)
294     {
295         printf("请输入中转号:");
296         scanf("%d",&x);
297         if(x<=k&&x>=1)
298         {
299             printf("序号\t出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t
300             头等舱\t商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
301             printf("%d\t",transit[0][x-1].number);
302             PrintAirline(&transit[0][x-1]);
303             printf("%d\t",transit[1][x-1].number);
304             PrintAirline(&transit[1][x-1]);
305             Number();
306             Number();
307             break;
308         }
309         else
310         {
311             printf("输入有误,是否重新输入(y|n):");
312             if(choice=='y') continue;
313             else break;
314         }
315     }
316     return OK;
317 }
318 
319 Status Conflict(char id[],Airline *B)//判断订票时间是否冲突
320 {
321     int i;
322     Client E;
323     for(i=0;i<airnum;i++)
324     {
325         if(Search(A[i].L,id,&E))
326         {
```

```
327             if(DayConflit(B,&A[i]))
328             {
329                 printf("出发:%s\t抵达:%s\t起飞日期:%02d月%02d日\t起飞时间:%02d:%02d\t
330                 到达时间:%02d:%02d\t",A[i].departure,A[i].destination,A[i].month,
331                 A[i].day,A[i].hour,A[i].min,A[i].nexthour,A[i].nextmin);
332                 return ERROR;
333             }
334         }
335     }
336     return OK;
337 }
338 
339 Status DayConflit(Airline *B,Airline *D)//判断时间冲突
340 {
341     int min1[2],min2[2];
342     TranMin(B,min1);
343     TranMin(D,min2);
344     if(min1[0]<min2[0]&&min1[1]>min2[0]||min2[0]<min1[0]&&min2[1]>min1[0]) return OK;
345     return ERROR;
346 }
347 
348 Status TranMin(Airline *B,int min[2])//时间转换分钟
349 {
350     int hour;
351     min[0]=24*60*MonthDays(B->month,B->day)+B->hour*60+B->min;
352     if(B->hour>=B->nexthour) hour=B->nexthour+24;
353     else hour=B->nexthour;
354     min[1]=24*60*MonthDays(B->month,B->day)+hour*60+B->nextmin;
355     return OK;
356 }
```

4.3.2 退票库

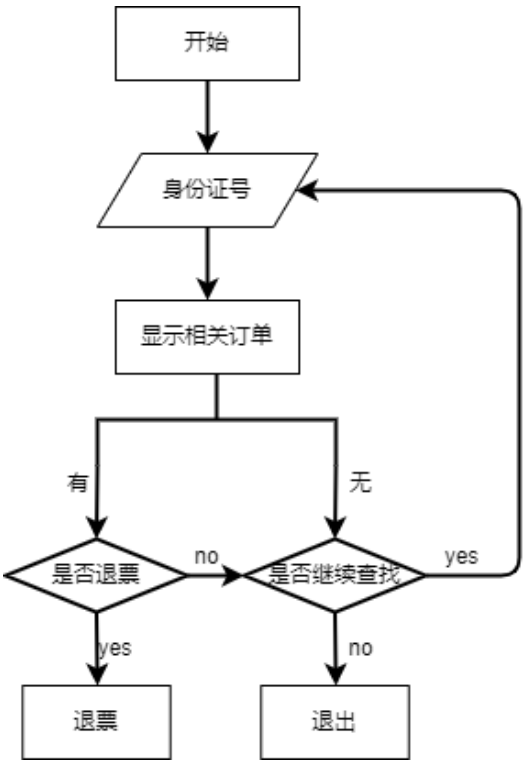

图 8: 个人信息检索流程图

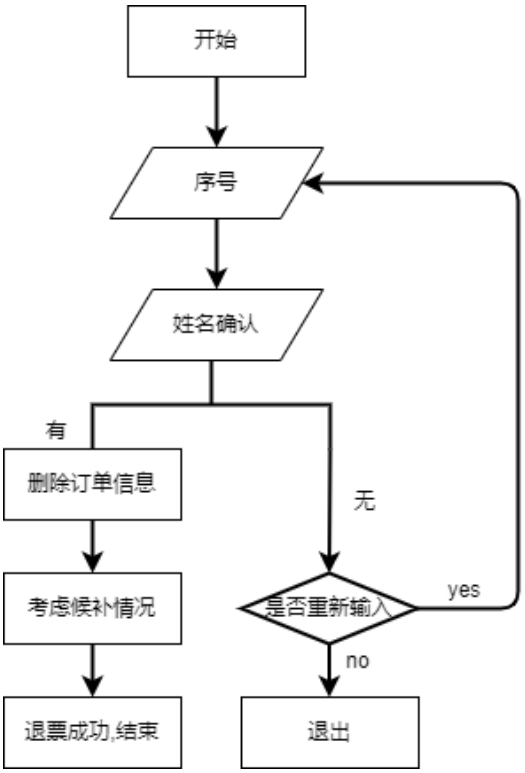

图 9: 退票流程图

```
Status Information()//显示信息
{
    char id[20],choice;
    int i,price,date,count=0;
    Client C;
    printf("请输入您的身份证号:");
    scanf("%s",id);
    printf("序号\t起飞机场\t降落机场\t出发\t抵达\t航班号\t飞机号\t起飞日期\t
    起飞时间\t到达时间\t舱位\t价格\t星期\n");
    for(i=0;i<airnum;i++)
    {
        if(Search(A[i].L,id,&C))
        {
            printf("%d\t%s\t%s\t%s\t%s\t%s\t%s\t%02d月%02d号\t%02d:%02d\t\t
            %02d:%02d\t\t",A[i].number,A[i].airport,A[i].nextairport,
            A[i].departure,A[i].destination,A[i].flightNum,A[i].planeNum,
            A[i].month,A[i].day,A[i].hour,A[i].min,A[i].nexthour,A[i].nextmin);
            price=A[i].price;
            if(A[i].L->C.grade=='F') {price*=1.5;printf("头等舱");}
            else if(A[i].L->C.grade=='C') {price*=1.3;printf("商务舱");}
            else printf("经济舱");
            printf("\t%-d元",price);
            date=A[i].date;
            printf("\t%s\n",week[date]);
            count++;
        }
    }
    if(count)
    {
        while(1)
        {
            printf("是否退票(y|n):");
            scanf(" %c",&choice);
            if(choice=='y') Refund(id);
            if(count>1)
            {
                printf("是否继续退票(y|n):");
                scanf(" %c",&choice);
                if(choice=='y') continue;
                else break;
            }
            else break;
        }

    }
    else
```

```
47      {
48          printf("未找到该身份证号的相关订票信息,是否继续查找(y|n):");
49          scanf(" %c",&choice);
50          if(choice=='y') Information();
51      }
52      return OK;
53  }
54
55  Status Refund(char id[])//退票
56  {
57      int num,i,flag=0;
58      char name[12],choice;
59      Client C;
60      printf("<<<<<<<<<<<<<<<您正在办理退票业务>>>>>>>>>>>>>\n");
61      while(1)
62      {
63          printf("请选择序号:");
64          scanf("%d",&num);
65          printf("是否确认退票,请输入您的姓名确认:");
66          scanf("%s",name);
67          for(i=0;i<airnum;i++)
68          {
69              if(num==A[i].number)
70              {
71                  List L=A[i].L->next;
72                  while(L!=NULL)
73                  {
74                      if(strcmp(name,L->C.name)==0&&strcmp(id,L->C.ID)==0)
75                      {
76                          C=L->C;
77                          flag=1;
78                          break;
79                      }
80                      L=L->next;
81                  }
82              }
83          }
84          if(flag)
85          {
86              printf("请您稍等,正在为您办理退票!\n");
87              Delete(A[num-1].L,C);
88              if(!QueueEmpty(&(A[num-1].Q)))
89              {
90                  Client E;
91                  if(CheckQueue(&(A[num-1].Q),C.grade,&E)==0)
                        AirlineBack(&A[num-1],C.grade);
92                  else OrderInSert(A[num-1].L,E);
```

```
            }
            else
            {
                AirlineBack(&A[num-1],C.grade);
                return OK;
            }
        }
        else
        {
            printf("输入错误,是否重新输入(y|n):");
            scanf(" %c",&choice);
            if(choice=='y') continue;
            else break;
        }
    }
    printf("未成功退票!\n");
    return OK;
}
```

5. 用户手册

5.1 界面

```
<<<<<<<<<<----------------------------------------------------->>>>>>>>>>
                    欢迎使用中国航空客运订票系统
<<<<<<<<<<----------------------------------------------------->>>>>>>>>>
            当前时间是北京时间:【2022年4月20日星期3 23时56分5秒】

                <<<<<<<<<<<<----功能菜单---->>>>>>>>>>>>

                    0.退出系统    1.查询航线    2.个人信息

请输入选择的功能(1查询|2退票):1
```

图 10: 用户界面

5.2 订票

```
请输入选择的功能(1查询|2退票):1
请输入起飞地:成都
请输入目的地:上海
最近共有38趟航班从成都直接飞往上海
是否订票(y|n):y
:<<<<<<<<<<<<<<您正在办理订票业务>>>>>>>>>>>
请输入日期(00-00):10-31
序号    出发    抵达    航班号  航空公司        起飞日期        起飞时间        到达时间        头等舱  商务舱  经济舱  起步价  余票量  星期
943     成都    上海    MU3161  东方航空        10月31号        21:10           23:55           0       0       68      2673元  68      星期天

中转号  出发    中转    抵达    起飞日期        起飞时间        到达时间        中转日期        中转起飞时间    中转到达时间
没有中转航班!
请输入订票序号:943
|头等舱0席|商务舱0席|经济舱68席|
是否下单(y|n):y
请输入订票数:1
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):C
当前没有商务舱席位
是否继续愿意登记候补(y|n):n
是否重新选择席位(y|n):y
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):Y
请填写个人信息:
请输入您的姓名:wer
请输入您的身份证号码:008
您成功订票!
是否退出订票(y|n):y
是否继续查询(y|n):n
请输入选择的功能(1查询|2退票):0

>>>>>>>感谢您的使用,下次再会!<<<<<<<<<
```

图 11: 订票测试

5.3 候补

```
请输入日期(00-00):8-10
序号    出发    抵达    航班号  航空公司        起飞日期        起飞时间        到达时间        头等舱  商务舱  经济舱  起步价  余票量  星期
451     北京    深圳    ZH9102  深圳航空        08月10号        21:30           01:05           0       7       50      2700元  57      星期二

中转号  出发    中转    抵达    起飞日期        起飞时间        到达时间        中转日期        中转起飞时间    中转到达时间
1       北京    兰州    深圳    08月10日        06:55           09:35           08月10日        18:00           20:55
共有1趟中转航班
是否选择中转航班(y|n):n
请输入订票序号:451
|头等舱0席|商务舱7席|经济舱50席|
是否下单(y|n):y
请输入订票数:3
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):C
请填写个人信息:
请输入您的姓名:aaa
请输入您的身份证号码:777
您成功订票!
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):F
当前没有头等舱席位
是否继续愿意登记候补(y|n):y
请填写个人信息:
请输入您的姓名:bbb
请输入您的身份证号码:233
您目前在排队候补!
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):F
当前没有头等舱席位
是否继续愿意登记候补(y|n):n
是否重新选择席位(y|n):y
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):Y
请填写个人信息:
请输入您的姓名:ccc
请输入您的身份证号码:999
您成功订票!
```

图 12: 订票不同情况测试

5.4 中转

```
<<<<<<<<<<<<<<<您正在办理订票业务>>>>>>>>>>>>>>>
请输入日期(00-00):8-10
序号    出发    抵达    航班号  航空公司        起飞日期        起飞时间        到达时间        头等舱  商务舱  经济舱  起步价  余票量  星期
451     北京    深圳    ZH9102  深圳航空        08月10号        21:30           01:05           0       7       50      2700元  57      星期二

中转号  出发    中转    抵达    起飞日期        起飞时间        到达时间        中转日期        中转起飞时间    中转到达时间
1       北京    兰州    深圳    08月10日        06:55           09:35           08月10日        18:00           20:55
共有1趟中转航班
是否选择中转航班(y|n):y
请注意中转时间,以防误机!
请输入中转号:1
序号    出发    抵达    航班号  航空公司        起飞日期        起飞时间        到达时间        头等舱  商务舱  经济舱  起步价  余票量  星期
296     北京    兰州    MU2129  东方航空        08月10号        06:55           09:35           0       4       27      1356元  31      星期二
3298    兰州    深圳    HU7266  海南航空        08月10号        18:00           20:55           0       5       21      2622元  26      星期二
请输入订票序号:
```

图 13: 中转订票测试

5.5 时间冲突

```
请输入起飞地:北京
请输入目的地:大连
最近共有29趟航班从北京直接飞往大连
是否订票(y|n):y
<<<<<<<<<<<<<<<您正在办理订票业务>>>>>>>>>>>>>>>
请输入日期(00-00):4-8
序号    出发    抵达    航班号  航空公司        起飞日期        起飞时间        到达时间        头等舱  商务舱  经济舱  起步价  余票量  星期
99      北京    大连    NZ3937  新西兰航空      04月08号        22:25           23:55           0       0       124     869元   124     星期一

中转号  出发    中转    抵达    起飞日期        起飞时间        到达时间        中转日期        中转起飞时间    中转到达时间
没有中转航班!
请输入订票序号:99
|头等舱0席|商务舱0席|经济舱124席|
是否下单(y|n):y
请输入订票数:1
请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):Y
请填写个人信息:
请输入您的姓名:user
请输入您的身份证号码:666
出发:北京       抵达:成都       起飞日期:04月08日       起飞时间:22:40  到达时间:01:45
本次订单与您的这次订单时间冲突,请选择其他航班!
是否退出订票(y|n):
```

图 14: 时间冲突测试

5.6 退票

```
请输入选择的功能(1查询|2退票):2
请输入您的身份证号:666
序号    起飞机场        降落机场        出发    抵达    航班号  飞机号  起飞日期        起飞时间        到达时间        舱位    价格    星期
29      首都机场        双流机场        北京    成都    ZH9163  A320    04月08号        22:40           01:45           经济舱  1358元  星期一
943     双流机场        浦东机场        成都    上海    MU3161  A320    10月31号        21:10           23:55           经济舱  2673元  星期天
2380    萧山机场        首都机场        杭州    北京    JD5352  A320    04月27号        19:15           21:40           经济舱  1560元  星期三
是否退票(y|n):y
<<<<<<<<<<<<<<<您正在办理退票业务>>>>>>>>>>>>>>>
请选择序号:29
是否确认退票,请输入您的姓名确认:user
请您稍等,正在为您办理退票!
成功办理退票!
是否继续退票(y|n):n
请输入选择的功能(1查询|2退票):
```

图 15: 退票测试

6. 心得体会

这次课程设计的心得体会通过实践我们的收获如下：

1. 在这次的航空客运订票系统的过程中，我们更深刻的了解队列的特点与用法。

2. 在不断的修改程序 bug 的过程中，我们对程序运行的细节更加明了，提高了我们的查错，纠错能力。

3. 这个课程设计考察的内容是线性表、线性链表，以及队列的综合应用，我们学会了运用伪数据库来处理数据，数据结构的理念得到了强化。

4. 这个项目的难点不在于程序语法和算法，而是在于对整个程序架构的把握，如何摆布函数，如何理清系统的运行逻辑。

7. 附录

7.1 definition.h

```
#ifndef DEFINITION
#define DEFINITION
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

#define OK 1
#define ERROR 0
#define OVERFLOW -1
FILE *fp;//文件指针
typedef int Status;//函数封装

typedef struct passenger
{
    char name[50];//姓名
    char ID[50];//身份证号码
    char grade;//飞机舱位 头等舱F,商务舱C,经济舱Y
}Client;//乘客信息

typedef struct node
{
    Client C;
    struct node *next;
}node,*List;//乘客名单

typedef struct
{
    List front;
    List rear;
}Queue;//候补队列

typedef struct
{
    int number;//序号
    char flightNum[10];//航班号
    char planeNum[10];//飞机号
    char departure[20];//始发站
    char destination[20]; //目的地
    char company[20];//航空公司
    char airport[20];//起飞机场
    char nextairport[20];//降落机场
```

```
43      int price;//经济舱基础票价（头等舱价格为基础票价150%，商务舱价格为基础票价的130%）
44      int month;
45      int day;
46      int hour;//起飞时间
47      int min;
48      int nexthour;//抵达时间
49      int nextmin;
50      int date;//星期
51      int capacity;//载客量 一条航线的舱位中有0.1的头等舱，0.2的商务舱，其余为经济舱
52      int F;//头等舱余票量
53      int C;//商务舱余票量
54      int Y;//经济舱余票量
55      List L;//乘客名单
56      Queue Q;//等候替补队列
57  }Airline;//航线信息
58
59  //List相关函数
60  Status Create(List *B);//创建乘客名单
61  Status OrderInSert(List B,Client E);//客户信息放入名单
62  Status Search(List B,char id[],Client *E);//查询客户订票信息
63  Status Delete(List B,Client E);//名单移除订票信息
64  Status EntryList();//录入客户名单
65  Status OutList();//存储客户名单
66
67  //Queue相关函数
68  Status InitQueue(Queue *Q);//构造一个空队列
69  Status EnQueue(Queue *Q,Client E);//插入队尾元素
70  Status DeQueue(Queue *Q,Client *E);//删除队头元素
71  Status QueueEmpty(Queue *Q);//判断队列为空
72  Status CheckQueue(Queue *Q,char space,Client
        *E);//检查退票是否满足候补客户需求，并排队候补
73
74  //Airline相关函数
75  int airnum;//航线数
76  Airline transit[2][1000]={0};//中转航班序号库
77  Airline A[10000]={0};//航线信息库
78  char *week[7]={"星期天","星期一","星期二","星期三","星期四","星期五","星期六"};//星期
79  Status EntryAirline();//录入航线信息
80  Status OutAirline();//存储航线信息
81  Status PrintAirline(Airline *A);//打印航线信息
82  Status AirlineReady();//构建航线对应的线性表和队列
83  Status AirlineBack(Airline *B,char C);//退票后补位
84
85  //inquire函数
86  Status Query();//航线查询
87  Status Booking(char place1[15],char place2[15]);//订票
88  Status Number();//选择序号
```

```
89  Status PlaceOrder(Airline *B);//下单
90  Status FillBlank(Client *C);//客户信息录入系统
91  Status Transit(char place1[],char place2[],int m,int n);//订单中转
92  Status Day(Airline *B,Airline *D);//判断中转时到达时间和起飞时间是否在同一天
93  Status MonthDays(int month,int day);//计算日期在一年中的天数
94  Status TransitNum(int k);//中转订票
95  Status Conflict(char id[],Airline *B);//判断订票时间是否冲突
96  Status DayConflit(Airline *B,Airline *D);//判断时间冲突
97  Status TranMin(Airline *B,int min[2]);//时间转换分钟
98
99  //refund函数
100 Status Information();//显示信息
101 Status Refund(char id[]);//退票
102 #endif
```

7.2 main.c

```
1  #include "definition.h"
2  #include "list.c"
3  #include "queue.c"
4  #include "airline.c"
5  #include "inquire.c"
6  #include "refund.c"
7
8  Status main()
9  {
10     system("Color F3");
11     int color,x;
12     time_t T;
13     time(&T);
14     struct tm *now;
15     now=localtime(&T);
16     printf("<<<<<<<<<<----------------------------------------------------->>>>>>>>>>\n");
17     printf("                  欢迎使用中国航空客运订票系统                    \n");
18     printf("<<<<<<<<<<----------------------------------------------------->>>>>>>>>>\n");
19     printf("        当前时间是北京时间:【%d年%d月%d日星期%d
           %d时%d分%d秒】\n",now->tm_year+1900,now->tm_mon+1,now->tm_mday,now->tm_wday,
20     now->tm_hour,now->tm_min,now->tm_sec);
21     EntryAirline();AirlineReady();EntryList();
22     printf("\n            <<<<<<<<<<<<----功能菜单---->>>>>>>>>>>>            \n\n");
23     printf("              0.退出系统  1.查询航线  2.个人信息\n\n");
24     while(1)
25     {
26        printf("请输入选择的功能(1查询|2退票):");
27        scanf("%d",&x);
```

```
28          if(x!=0&&x!=1&&x!=2) {printf("输入有误!!\n");continue;}
29          if(x==0) break;
30          switch(x)
31          {
32              case 1:Query();break;//查询航线
33              case 2:Information();break;//个人信息
34          }
35      }
36      OutAirline();OutList();
37      printf("\n>>>>>>>>感谢您的使用,下次再会!<<<<<<<<<\n");
38      return OK;
39  }
```

7.3 list.c

```
1   Status Create(List *B)//创建乘客名单
2   {
3       (*B)=(List)malloc(sizeof(node));
4       (*B)->next=NULL;
5       return OK;
6   }
7
8   Status OrderInSert(List B,Client E)//客户信息放入名单
9   {
10      List L=B;
11      while(L->next!=NULL)
12      {
13          L=L->next;
14
15      }
16      List S=(List)malloc(sizeof(node));
17      S->C=E;
18      S->next=NULL;
19      L->next=S;
20      return OK;
21  }
22
23  Status Search(List B,char id[],Client *E)//查询客户订票信息
24  {
25      List L=B->next;
26      int flag=0;
27      while(L!=NULL)
28      {
29          if(strcmp(L->C.ID,id)==0)
30          {
```

```
31            *E=L->C;
32            flag=1;
33            break;
34        }
35        L=L->next;
36     }
37     if(flag==0) return ERROR;
38     return OK;
39 }
40 
41 Status Delete(List B,Client E)//名单移除订票信息
42 {
43     List L=B;
44     while(L->next!=NULL)
45     {
46         List M=L->next;
47         if(strcmp(E.ID,M->C.ID)==0)
48         {
49             L->next=M->next;
50             free(M);
51             printf("成功办理退票!\n");
52             break;
53         }
54         L=L->next;
55     }
56     return OK;
57 }
58 
59 Status EntryList()//录入客户名单
60 {
61     Client C;
62    char *line,*record,buffer[1024];
63    if((fp=fopen("client.csv","r+"))==NULL) return ERROR;
64    fseek(fp,29L,SEEK_SET);
65    while((line=fgets(buffer,sizeof(buffer),fp))!=NULL)
66    {
67       record=strtok(line,",");
68         int i=atoi(record);
69         record=strtok(NULL, ",");
70       if(record!=NULL)
71       {
72          sprintf(C.name,"%s",record);record=strtok(NULL, ",");
73             sprintf(C.ID,"%s",record);record=strtok(NULL, ",");
74             C.grade=record[0];record=strtok(NULL, ",");
75       }
76         OrderInSert(A[i-1].L,C);
77    }
```

```
78    fclose(fp);
79    return OK;
80  }
81
82  Status OutList()//存储客户名单
83  {
84      if((fp=fopen("client.csv","w+"))==NULL) return ERROR;
85      fprintf(fp,"序号,姓名,身份证号,飞机舱位\n");
86      for(int i=0;i<airnum;i++)
87      {
88          List L=A[i].L->next;
89          while(L!=NULL)
90          {
91              fprintf(fp,"%d,%s,%s,%c\n",A[i].number,L->C.name,L->C.ID,L->C.grade);
92              L=L->next;
93          }
94      }
95      return OK;
96  }
```

7.4 queue.c

```
1   Status InitQueue(Queue *Q)//构造一个空队列
2   {
3       Q->front=(List)malloc(1*sizeof(node));
4       if(!Q->front) exit(OVERFLOW);
5       Q->rear=Q->front;
6       Q->front->next=NULL;
7       return OK;
8   }
9
10  Status EnQueue(Queue*Q,Client E)//插入元素E为Q的新的队尾元素
11  {
12      List new=(List)malloc(sizeof(node));
13      if(!new) exit(OVERFLOW);
14      new->C=E;
15      new->next=NULL;
16      Q->rear->next=new;
17      Q->rear=new;
18      return OK;
19  }
20
21  Status DeQueue(Queue *Q,Client
        *E)//若队列不空，则删除Q的队头元素，用E返回其值，并返回OK；否则返回ERROR
22  {
```

```
    if(Q->front==Q->rear) return ERROR;
    List head=Q->front->next;
    *E=head->C;
    Q->front->next=head->next;
    if(Q->rear==head) Q->rear=Q->front;
    free(head);
    return OK;
}

Status QueueEmpty(Queue *Q)//若队列Q为空队列，则返回OK；否则返回ERROR
{
    if(Q->front==Q->rear) return OK;
    return ERROR;
}

Status CheckQueue(Queue *Q,char space,Client
     *E)//检查退票是否满足候补客户需求，并排队候补
{
    Queue S;
    InitQueue(&S);
    int flag=0;
    while(!QueueEmpty(Q))
    {
        Client D;
        DeQueue(Q,&D);
        if(D.grade==space)
        {
            *E=D;
            flag=1;
            space='\0';
        }
        else
            EnQueue(&S,D);
    }
    *Q=S;
    if(flag==0) return ERROR;
    return OK;
}
```

7.5 airline.c

```
Status EntryAirline()//录入航线信息
{
   char *line,*record,buffer[1024];
    int i=0;
```

```
5    if((fp=fopen("airline.csv","r+"))==NULL) return ERROR;
6    fseek(fp,127L,SEEK_SET);
7    while((line=fgets(buffer,sizeof(buffer),fp))!=NULL)
8    {
9        record=strtok(line,",");
10     if(record!=NULL)
11     {
12          A[i].number=atoi(record);record=strtok(NULL, ",");
13          sprintf(A[i].departure,"%s",record);record=strtok(NULL, ",");
14          sprintf(A[i].destination,"%s",record);record=strtok(NULL, ",");
15          sprintf(A[i].flightNum,"%s",record);record=strtok(NULL, ",");
16          sprintf(A[i].company,"%s",record);record=strtok(NULL, ",");
17          sprintf(A[i].planeNum,"%s",record);record=strtok(NULL, ",");
18        A[i].price=atoi(record);record=strtok(NULL, ",");
19          A[i].capacity=atoi(record);record=strtok(NULL, ",");
20        A[i].F=atoi(record);record=strtok(NULL, ",");
21        A[i].C=atoi(record);record=strtok(NULL, ",");
22        A[i].Y=atoi(record);record=strtok(NULL, ",");
23          sprintf(A[i].airport,"%s",record);record=strtok(NULL, ",");
24          sprintf(A[i].nextairport,"%s",record);record=strtok(NULL, ",");
25        A[i].month=atoi(record);record=strtok(NULL, ",");
26        A[i].day=atoi(record);record=strtok(NULL, ",");
27          A[i].date=atoi(record);record=strtok(NULL, ",");
28        A[i].hour=atoi(record);record=strtok(NULL, ",");
29        A[i].min=atoi(record);record=strtok(NULL, ",");
30          A[i].nexthour=atoi(record);record=strtok(NULL, ",");
31          A[i].nextmin=atoi(record);record=strtok(NULL, ",");
32     }
33     i++;
34   }
35    airnum=i;
36   fclose(fp);
37   return OK;
38 }
39
40 Status OutAirline()//存储航线信息
41 {
42    int i;
43    if((fp=fopen("airline.csv","w+"))==NULL) return ERROR;
44   fprintf(fp,"序号,出发地,抵达地,航班号,航空公司,飞机号,起步价,载客量,头等舱,商务舱,
45    经济舱,起飞机场,降落机场,月,日,星期,时,分,抵达时,抵达分\n");
46    for(i=0;i<airnum;i++) fprintf(fp,"%d,%s,%s,%s,%s,%s,%d,%d,%d,%d,%d,%s,%s,%d,
47    %d,%d,%d,%d,%d,%d\n",A[i].number,A[i].departure,A[i].destination,A[i].flightNum,
48    A[i].company,A[i].planeNum,A[i].price,A[i].capacity,A[i].F,A[i].C,A[i].Y,
49    A[i].airport,A[i].nextairport,A[i].month,A[i].day,A[i].date,A[i].hour,A[i].min,
50    A[i].nexthour,A[i].nextmin);
51    return OK;
```

```
52 }
53
54 Status PrintAirline(Airline *A)//打印航线信息
55 {
56     printf("%s\t%s\t%s\t%s\t",A->departure,A->destination,A->flightNum,A->company);
57     printf("%02d月%02d号\t%02d:%02d\t\t%02d:%02d\t\t",A->month,A->day,A->hour,A->min,
58     A->nexthour,A->nextmin);
59     printf("%-d\t%-d\t%-d\t",A->F,A->C,A->Y);
60     printf("%-d元\t%-d\t",A->price,A->F+A->C+A->Y);
61     int i=A->date;
62     printf("%-s\t\n",week[i]);
63     return OK;
64 }
65
66 Status AirlineReady()//构建航线对应的线性表和队列
67 {
68     for(int i=0;i<airnum;i++)
69     {
70         Create(&(A[i].L));
71         InitQueue(&(A[i].Q));
72     }
73     return OK;
74 }
75
76 Status AirlineBack(Airline *B,char C)//退票后补位
77 {
78     switch(C)
79     {
80         case 'F':B->F++;break;
81         case 'C':B->C++;break;
82         case 'Y':B->Y++;break;
83     }
84     return OK;
85 }
```

7.6 inquire.c

```
1 Status Query()//航线查询
2 {
3     while(1)
4     {
5         char place1[15],place2[15],choice;
6         int i,count=0;
7         printf("请输入起飞地:");
8         scanf("%s",place1);
```

```
9         printf("请输入目的地:");
10        scanf("%s",place2);
11        //printf("出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t头等舱\t
12        //商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
13        for(i=0;i<airnum;i++)
14        {
15            if(strcmp(place1,A[i].departure)==0&&
16            strcmp(place2,A[i].destination)==0&&A[i].F+A[i].C+A[i].Y>0)
17            {
18                //PrintAirline(&A[i]);
19                count++;
20            }
21        }
22        if(count>0)
23        {
24            printf("最近共有%d趟航班从%s直接飞往%s\n",count,place1,place2);
25            printf("是否订票(y|n):");
26            scanf(" %c",&choice);
27            if(choice=='y') Booking(place1,place2);
28        }
29        else
30        {
31            printf("最近没有从%s直接飞往%s的航班\n",place1,place2);
32            printf("是否考虑中转航班(y|n):");
33            scanf(" %c",&choice);
34            if(choice=='y') Booking(place1,place2);
35        }
36        printf("是否继续查询(y|n):");
37        scanf(" %c",&choice);
38        if(choice=='y') continue;
39        else break;
40    }
41    return OK;
42 }
43
44 Status Booking(char place1[15],char place2[15])//订票
45 {
46    printf("<<<<<<<<<<<<<<<您正在办理订票业务>>>>>>>>>>>>>\n");
47    while(1)
48    {
49        int m,n,i,k,count=0;
50        char choice;
51        printf("请输入日期(00-00):");
52        scanf("%d-%d",&m,&n);
53        printf("序号\t出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t
54        头等舱\t商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
55        for(i=0;i<airnum;i++)
```

```
56          {
57              if(m==A[i].month&&n==A[i].day&&strcmp(place1,A[i].departure)==0&&
58              strcmp(place2,A[i].destination)==0)
59              {
60                  printf("%d\t",A[i].number);
61                  PrintAirline(&A[i]);
62                  count++;
63              }
64          }
65          k=Transit(place1,place2,m,n);
66          count+=k;
67          if(count>0)
68          {
69              if(k)
70              {
71                  printf("是否选择中转航班(y|n):");
72                  scanf(" %c",&choice);
73                  if(choice=='y') TransitNum(k);
74                  else Number();
75              }
76              else Number();
77          }
78          else printf("最近没有%d月%d日的目标航班\n",m,n);
79          printf("是否退出订票(y|n):");
80          scanf(" %c",&choice);
81          if(choice=='n') continue;
82          else break;
83      }
84      return OK;
85  }
86
87  Status Number()//选择序号
88  {
89      int i,flag=0;
90      char choice;
91      while(1)
92      {
93          int num;
94          printf("请输入订票序号:");
95          scanf("%d",&num);
96          for(i=0;i<airnum;i++)
97          {
98              if(num==A[i].number)
99              {
100                 flag=1;
101                 break;
102             }
```

```
103         }
104         if(flag)
105         {
106             printf("|头等舱%d席|商务舱%d席|经济舱%d席|\n",A[i].F,A[i].C,A[i].Y);
107             printf("是否下单(y|n):");
108             scanf(" %c",&choice);
109             if(choice=='y')
110             {
111                 PlaceOrder(&A[i]);
112                 break;
113             }
114             else break;
115         }
116         else
117         {
118             printf("\n是否愿意选择该日期的其他航班(y|n):");
119             scanf(" %c",&choice);
120             if(choice=='y') continue;
121             else break;
122         }
123     }
124     return OK;
125 }
126 
127 Status PlaceOrder(Airline *B)//下单
128 {
129     char choice;
130     Client C,E;
131     int count;
132     while(1)
133     {
134         printf("请输入订票数:");
135         scanf("%d",&count);
136         if(B->F+B->C+B->Y<=count)
137         {
138             printf("当前余票%d张,是否重新输入订票数(y|n):",B->F+B->C+B->Y);
139             scanf(" %c",&choice);
140             if(choice=='y') continue;
141             else
142             {
143                 printf("人数过多,请考虑其他航班\n");
144                 return OK;
145             }
146         }
147         else if(count<=0)
148         {
149             printf("输入错误!\n");
```

```
150             continue;
151         }
152         else break;
153     }
154     while(count)
155     {
156         int flag=0;
157         printf("请输入你想预订的舱位等级( 头等舱 F | 商务舱 C | 经济舱 Y ):");
158         scanf(" %c",&C.grade);
159         switch(C.grade)
160         {
161             case 'F':if(B->F==0) printf("当前没有头等舱席位\n");else
                    {B->F--;flag=1;}break;
162             case 'C':if(B->C==0) printf("当前没有商务舱席位\n");else
                    {B->C--;flag=1;}break;
163             case 'Y':if(B->Y==0) printf("当前没有经济舱席位\n");else
                    {B->Y--;flag=1;}break;
164             default:printf("输入有误!\n");flag=2;break;
165         }
166         if(flag==1)
167         {
168             while(1)
169             {
170                 FillBlank(&C);
171                 if(Search(B->L,C.ID,&E))
172                 {
173                     printf("身份证号为%s的顾客已下单本次航班,是否重新录入顾客信息(y|n):",
174                     C.ID);
175                     scanf(" %c",&choice);
176                     if(choice=='y') continue;
177                     else break;
178                 }
179                 else
180                 {
181                     if(Conflict(C.ID,B))
182                     {
183                         OrderInSert(B->L,C);
184                         printf("您成功订票!\n");
185                         break;
186                     }
187                     else
188                     {
189                         printf("\n本次订单与您的这次订单时间冲突,请选择其他航班!\n");
190                         break;
191                     }
192                 }
193             }
```

```
194         }
195         else if(flag>1) break;
196         else
197         {
198             printf("是否继续愿意登记候补(y|n):");
199             scanf(" %c",&choice);
200             if(choice=='y')
201             {
202                 FillBlank(&C);
203                 EnQueue(&(B->Q),C);
204                 printf("您目前在排队候补!\n");
205             }
206             else
207             {
208                 printf("是否重新选择席位(y|n):");
209                 scanf(" %c",&choice);
210                 if(choice=='y') continue;
211                 else break;
212             }
213         }
214         count--;
215     }
216     return ERROR;
217 }
218
219 Status FillBlank(Client *C)//客户信息录入系统
220 {
221     printf("请填写个人信息:\n");
222     printf("请输入您的姓名:");
223     scanf("%s",C->name);
224     printf("请输入您的身份证号码:");
225     scanf("%s",C->ID);
226     return OK;
227 }
228
229 Status Transit(char place1[],char place2[],int m,int n)//订单中转
230 {
231     int i,j,k=0;
232     printf("\n中转号\t出发\t中转\t抵达\t起飞日期\t起飞时间\t到达时间\t中转日期\t
233     中转起飞时间\t中转到达时间\n");
234     for(i=0;i<airnum;i++)
235     {
236         if(strcmp(A[i].departure,place1)==0&&A[i].month==m&&A[i].day==n)
237         {
238             for(j=0;j<airnum;j++)
239             {
240                 if(strcmp(A[i].destination,A[j].departure)==0&&Day(&A[i],&A[j])==1
```

```
241                 &&strcmp(A[j].destination,place2)==0)
242                 {
243                     printf("%d\t",k+1);
244                     printf("%s\t%s\t%s\t",A[i].departure,A[i].destination,
245                     A[j].destination);
246                     printf("%02d月%02d日\t%02d:%02d\t\t%02d:%02d\t\t%02d月%02d日\t
247                     %02d:%02d\t\t%02d:%02d\n",A[i].month,A[i].day,A[i].hour,A[i].min,
248                     A[i].nexthour,A[i].nextmin,A[j].month,A[j].day,A[j].hour,A[j].min,
249                     A[j].nexthour,A[j].nextmin);
250                     transit[0][k]=A[i];
251                     transit[1][k]=A[j];
252                     k++;
253                 }
254             }
255         }
256     }
257     if(k) printf("共有%d趟中转航班\n",k);
258     else printf("没有中转航班!\n");
259     return k;
260 }
261
262 Status Day(Airline *B,Airline *D)//判断中转时到达时间和起飞时间是否在同一天
263 {
264     int day;
265     if(B->nexthour>=B->hour) day=B->day;
266     else day=B->day+1;
267     if(MonthDays(B->month,day)==MonthDays(D->month,D->day))
268     {
269         if(B->nexthour+1<D->hour) return OK;
270         else if(B->nexthour+1==D->hour)
271         {
272             if(B->nextmin<D->min) return OK;
273         }
274     }
275     return ERROR;
276 }
277
278 Status MonthDays(int month,int day)//计算日期在一年中的天数
279 {
280     int months[12]={31,28,31,30,31,30,31,31,30,31,30,31};
281     int i,days=0;
282     for(i=0;i<month-1;i++)
283         days+=months[i];
284     days+=day;
285     return days;
286 }
287
```

```
288 Status TransitNum(int k)//中转订票
289 {
290     int i,x;
291     char choice;
292     printf("请注意中转时间,以防误机!\n");
293     while(1)
294     {
295         printf("请输入中转号:");
296         scanf("%d",&x);
297         if(x<=k&&x>=1)
298         {
299             printf("序号\t出发\t抵达\t航班号\t航空公司\t起飞日期\t起飞时间\t到达时间\t
300             头等舱\t商务舱\t经济舱\t起步价\t余票量\t星期\t\n");
301             printf("%d\t",transit[0][x-1].number);
302             PrintAirline(&transit[0][x-1]);
303             printf("%d\t",transit[1][x-1].number);
304             PrintAirline(&transit[1][x-1]);
305             Number();
306             Number();
307             break;
308         }
309         else
310         {
311             printf("输入有误,是否重新输入(y|n):");
312             if(choice=='y') continue;
313             else break;
314         }
315     }
316     return OK;
317 }
318
319 Status Conflict(char id[],Airline *B)//判断订票时间是否冲突
320 {
321     int i;
322     Client E;
323     for(i=0;i<airnum;i++)
324     {
325         if(Search(A[i].L,id,&E))
326         {
327             if(DayConflit(B,&A[i]))
328             {
329                 printf("出发:%s\t抵达:%s\t起飞日期:%02d月%02d日\t起飞时间:%02d:%02d\t
330                 到达时间:%02d:%02d\t",A[i].departure,A[i].destination,A[i].month,
331                 A[i].day,A[i].hour,A[i].min,A[i].nexthour,A[i].nextmin);
332                 return ERROR;
333             }
334         }
```

```
335     }
336     return OK;
337 }
338 
339 Status DayConflit(Airline *B,Airline *D)//判断时间冲突
340 {
341     int min1[2],min2[2];
342     TranMin(B,min1);
343     TranMin(D,min2);
344     if(min1[0]<min2[0]&&min1[1]>min2[0]||min2[0]<min1[0]&&min2[1]>min1[0]) return OK;
345     return ERROR;
346 }
347 
348 Status TranMin(Airline *B,int min[2])//时间转换分钟
349 {
350     int hour;
351     min[0]=24*60*MonthDays(B->month,B->day)+B->hour*60+B->min;
352     if(B->hour>=B->nexthour) hour=B->nexthour+24;
353     else hour=B->nexthour;
354     min[1]=24*60*MonthDays(B->month,B->day)+hour*60+B->nextmin;
355     return OK;
356 }
```

7.7 refund.c

```
1  Status Information()//显示信息
2  {
3      char id[20],choice;
4      int i,price,date,count=0;
5      Client C;
6      printf("请输入您的身份证号:");
7      scanf("%s",id);
8      printf("序号\t起飞机场\t降落机场\t出发\t抵达\t航班号\t飞机号\t起飞日期\t
9      起飞时间\t到达时间\t舱位\t价格\t星期\n");
10     for(i=0;i<airnum;i++)
11     {
12         if(Search(A[i].L,id,&C))
13         {
14             printf("%d\t%s\t%s\t%s\t%s\t%s\t%s\t%02d月%02d号\t%02d:%02d\t\t
15             %02d:%02d\t\t",A[i].number,A[i].airport,A[i].nextairport,
16             A[i].departure,A[i].destination,A[i].flightNum,A[i].planeNum,
17             A[i].month,A[i].day,A[i].hour,A[i].min,A[i].nexthour,A[i].nextmin);
18             price=A[i].price;
19             if(A[i].L->C.grade=='F') {price*=1.5;printf("头等舱");}
20             else if(A[i].L->C.grade=='C') {price*=1.3;printf("商务舱");}
```

```
21            else printf("经济舱");
22            printf("\t%-d元",price);
23            date=A[i].date;
24            printf("\t%s\n",week[date]);
25            count++;
26        }
27    }
28    if(count)
29    {
30        while(1)
31        {
32            printf("是否退票(y|n):");
33            scanf(" %c",&choice);
34            if(choice=='y') Refund(id);
35            if(count>1)
36            {
37                printf("是否继续退票(y|n):");
38                scanf(" %c",&choice);
39                if(choice=='y') continue;
40                else break;
41            }
42            else break;
43        }
44
45    }
46    else
47    {
48        printf("未找到该身份证号的相关订票信息,是否继续查找(y|n):");
49        scanf(" %c",&choice);
50        if(choice=='y') Information();
51    }
52    return OK;
53 }
54
55 Status Refund(char id[])//退票
56 {
57    int num,i,flag=0;
58    char name[12],choice;
59    Client C;
60    printf("<<<<<<<<<<<<<<<您正在办理退票业务>>>>>>>>>>>>\n");
61    while(1)
62    {
63        printf("请选择序号:");
64        scanf("%d",&num);
65        printf("是否确认退票,请输入您的姓名确认:");
66        scanf("%s",name);
67        for(i=0;i<airnum;i++)
```

```
68          {
69              if(num==A[i].number)
70              {
71                  List L=A[i].L->next;
72                  while(L!=NULL)
73                  {
74                      if(strcmp(name,L->C.name)==0&&strcmp(id,L->C.ID)==0)
75                      {
76                          C=L->C;
77                          flag=1;
78                          break;
79                      }
80                      L=L->next;
81                  }
82              }
83          }
84          if(flag)
85          {
86              printf("请您稍等,正在为您办理退票!\n");
87              Delete(A[num-1].L,C);
88              if(!QueueEmpty(&(A[num-1].Q)))
89              {
90                  Client E;
91                  if(CheckQueue(&(A[num-1].Q),C.grade,&E)==0)
                        AirlineBack(&A[num-1],C.grade);
92                  else OrderInSert(A[num-1].L,E);
93              }
94              else
95              {
96                  AirlineBack(&A[num-1],C.grade);
97                  return OK;
98              }
99          }
100         else
101         {
102             printf("输入错误,是否重新输入(y|n):");
103             scanf(" %c",&choice);
104             if(choice=='y') continue;
105             else break;
106         }
107     }
108     printf("未成功退票!\n");
109     return OK;
110 }
```