数据结构

课程设计报告

题 目： 银行叫号机管理系统

学 院： 计算机与控制工程学院

专业班级： 软件172

学生姓名： 马哲

指导教师： 杜晓昕

2019 年12 月26日

数据结构课程设计评分表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 班级 | 软件172 | | 姓名 | 马哲 | 指导教师 | 杜晓昕 | |
| 题目：银行叫号机管理系统 | | | | | | | |
| 评分标准 | | | | | | | |
| 评分标准 | | 分数权重 | 评分的依据 | | | | 得分 |
| A | | C | |
| 选题 | | 10 | 选题符合大纲要求，题目较新颖，工作量大 | | 选题基本符合大纲要求，工作量适中 | |  |
| 工作态度 | | 10 | 态度端正，能主动认真完成各个环节的工作，不迟到早退，出勤好。 | | 能够完成各环节基本工作，出勤较好。 | |  |
| 存储结构、算法描述 | | 20 | 能正确选择存储结构，定义准确，算法流程图或编程语言描述的算法准确无误 | | 能正确选择存储结构，算法流程图或编程语言描述的算法基本准确 | |  |
| 独立解决问题的能力 | | 10 | 具有独立分析、解决问题能力，有一定的创造性，能够独立完成软件的设计与调试工作，程序结构清晰，逻辑严谨，功能完善。 | | 有一定的分析、解决问题能力。能够在老师指导下完成软件的设计与调试工作，程序功能较完善。 | |  |
| 答辩问题回答 | | 20 | 能准确回答老师提出的问题 | | 能基本准确回答老师提出的问题 | |  |
| 程序运行情况 | | 10 | 程序运行正确、界面清晰，测试数据设计合理。 | | 程序运行正确、界面较清晰，能给出合适的测试数据。 | |  |
| 课程设计报告 | | 20 | 格式规范，层次清晰，设计思想明确，解决问题方法合理，体会深刻。 | | 格式较规范，设计思想基本明确，解决问题方法较合理。 | |  |
| 总分 | | | | | | |  |
| 指导教师（签字）： | | | | | | | |

注：介于A和C之间为B级，低于C为D级和E级。按各项指标打分后，总分在90～100为优，80～89为良，70～79为中，60～69为及格，60分以下不及格。

1 问题描述

目前，在以银行营业大厅为代表的窗口行业，大里客户的拥挤排队已成为了这些企事业单位改善服务品质、提升营业形象的主要障碍。排队(叫号)系统的使用将成为改变这种状况的有力手段。本文设计了一套排队叫号系统。该系统是以排队抽号顺序为核心,客户利用客户端抽号，工作人员利用叫号端叫号;通过显示器及时显示当前所叫号数,客户及时了解排队信息，通过合理的程序结构来执行排队抽号。以提高排队等待效率，解决排队秩序混乱，前拥后挤等现象,实现排队自动化，规范化。通过该系统的使用，客户不必为排队浪费大量精力,便于管理排队秩序，同时适应信息时代管理数字化的要求，提高服务水平与质量。排队叫号系统是针对银行、工商、税务、通讯、政府机构等部门的大厅工作流程设计的，是利用电脑的科学管理客户排队的系统,能够很好地解决客户在服务机构办理业务时所遇到的各种排队、拥挤和混乱现象，为客户办理业务带来莫大的方便和愉悦。本课题研究的是-款无人排队的排队叫号系统，排队系统完全模拟了人群排队全过程，通过取票进队、排队等待、叫号服务等功能，代替了人们站队的辛苦，把顾客排队等待的烦恼变成-段难得的休闲时光，使客户拥有了一个自由的空间和一份美好的心情。建立银行排队系统，改善排队质量，避免不必要的纠纷。通过使用排队系统，有传统的客户站立排队变为取票进队、排队等待、叫号服务，由传统物理的多个队列变为一个逻辑队列，使”先来先服务”的思想得到更好的贯彻。排队叫号系统的功能模块图如图1-1所示。

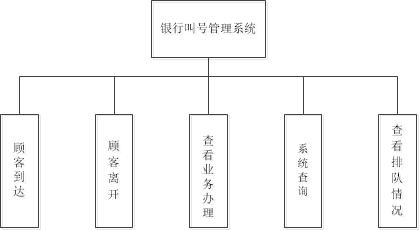


图1-1 功能模块图

顾客用例图如图1-2所示

2 数据结构描述

2.1 数据结构和存储结构

在银行排队叫号软件中，首先要找到一种数据结构来存放顾客所得到的服务号,这些服务号表示客户的请求服务的先后顺序,也表示客户被服务的先后顺序。先

的客户被服务:在这个程序中，主要运用了队列这种结构:

1. 队列

队列是-种特殊的线性表，是-种只允许在表的一端插入操作而在另-端进行删除操作的线性表。进行插入操作的表尾称为队尾(Rear), 进行删除操作的头部称为对头(Front)。当队列中没有数据元素时称为空队列(EmptyQueue)。队列通常记为Q= (a1, a2,，,an) ,a1为对头元素，an为队尾元素。这n个元素是按照a1, a2,，an 的次序依次入队的，出队的顺序与入队顺序相同，a1第一个出队，an最后一个出队。

###### 指针

###### 指针是一个变量，其值为另一个变量的地址，即，内存位置的直接地址。就像其他变量或常量一样，您必须在使用指针存储其他变量地址之前，对其进行声明。

###### 结构体

###### 结构是 C 编程中另一种用户自定义的可用的数据类型，它允许您存储不同类型的数据项。

1. 队列的基本操作:

(1) 入队列操作: EnQueue (T elem);

(2)出队列操作: DeQueue ( );

(3)取队头元素: GetFront ();

(4) 求队列的长度; GetLength ();

(5)判断队列是否为空: IsEmpty ();

(6)清空操作: Clear ();

(7) 判断是否为满: IsFull ();

2.2 主要算法

顺序查找算法是按照序列原有顺序对数组进行遍历比较查询的基本算法。实现简单，所以本系统采用了顺序查找算法对学生信息数据文件进行插入和删除操作。

2.3 ADT

ADT，即抽象数据类型，是一个实现包括储存数据元素的存储结构以及实现基本操作的算法。在这个数据抽象的思想中，数据类型的定义和它的实现是分开的。以下为本系统的ADT。

ADT system{

数据对象{data，\*next，Q.front，Q.rear，sum, core，UE\_score，DS\_score}

基本操作{

#//1.初始化线性

void Initshuzu()

//13.初始化等待队列

void Initqueue()

//11.等待客户进等待队列

void Enqueue(Linkqueue \*Q,int elem)

//12.等待队列客户出队

int Dlqueue(Linkqueue \*Q)

//2.输出办理业务的客户数组

void printl()

//3.输出办理业务排队客户队列

void print2()

//4.顾客到达事件算法

void daoda(int x)

//5.顾客离开事件算法

void likai(int x)

//6.判断输入的柜台号是否正确

int guitai( )

//8.主评分函数

void mygrade()

//9.vip用户认证

void vip(int x)

}

} ADT system

3 算法描述

实现：

设计一个队列类

（1）队列存储有序的项目序列

（2）能够创建空队列

（3）队列空间有一定限制

（3）能够检查队列是否为空

（4）能够检查队列是否已满

（5）能够在队尾添加元素（push操作）

（6）能够在队首删除元素（pop操作）

（7）能够确定队列中的项目数

3.1 .等待客户进排队模块的实现

1．功能

在队列尾添加新的客户

1. 具体语言实现

在本系统中有按照队列的排序方法。在此列出按先后顺序顺序排序算法。

//1.初始化线性

void Initshuzu()

{

for(int i=1;i<=n;i++)

L.A[i]=0;

L.len=0;

}

//13.初始化等待队列

void Initqueue()

{

Q.front=Q.rear=(Lnode \*)malloc(sizeof(Lnode));

Q.front->next=NULL;

}

//11.等待客户进等待队列

void Enqueue(Linkqueue \*Q,int elem)

{

Lnode \*s;

s=(Lnode \*)malloc(sizeof(Lnode));

s->data=elem;

s->next=NULL;

Q->rear->next=s;

Q->rear=s;

}

void Enqueue(Linkqueue \*Q,int elem)

{

Lnode \*s;

s=(Lnode \*)malloc(sizeof(Lnode));

s->data=elem;

s->next=NULL;

Q->rear->next=s;

Q->rear=s;

}

3.2等待队列客户出队模块设计

1. 功能

顾客办理完业务退出系统，客户号码弹出队列

2.算法描述

用函数int Dlqueue(Linkqueue \*Q) 如果队头等于队尾此时队列为空，输出队列为空。否则free(t)

3.具体语言实现

int Dlqueue(Linkqueue \*Q)

{

Lnode \*t;

int x;

if(Q->front==Q->rear)

{

printf("队列为空！\n");

exit(1);

}

else

{

t=Q->front->next;

Q->front->next=t->next;

x=t->data;

free(t);

return x;

}

}

3.3查看业务办理模块设计

1. 功能

界面显示当前正在办理业务的客户的号码

2．算法描述

用函数void printl() 输出每个柜台前顾客号码

3．具体语言实现

void printl()

{

int i;

printf("正在办理业务的顾客编号为： 一号柜台 二号柜台 三号柜台\n");

printf(" ");

for( i=1;i<=L.len;i++)

{

printf("%d号客户 ",L.A[i]);

}

printf("\n");

}

3.4顾客到达事件模块

1．功能

顾客到达，队列元素加一

2．算法描述

用函数void daoda(int x) 来实现，if (L.len<n) 链表元素加一。

3．具体语言实现

在本系统中在此只列出高等数学的排序算法的核心代码。

void daoda(int x)

{

int i=L.len+1;

if (L.len<n)

{ L.A[i]=x;

i++;

L.len++;

}

Else

Enqueue(&Q,x);

}

3.5顾客离开事件模块设计

1. 功能

顾客办理完业务退出系统，客户号码弹出队列

2.算法描述

用函数void likai(int x) 实现， if(x>L.len)则输出"输入有误！\n请重新输入："否则用一个for循环

3.具体语言实现

void likai(int x)

{

int i=0;

do

{

if(x>L.len)

{

printf("输入有误！\n请重新输入：");

scanf("%d",&x);

}

else

for(i=0;i<=L.len;i++)

{

if(i==x)

{

printf("尊敬的%d号顾客您好！\n",x);

L.A[i]=0;

L.len--;

if(Q.front!=Q.rear)

{

int y=Dlqueue(&Q);

L.A[i]=y;

L.len++;

}

}

}

}while(i==0);

}

3.6vip用户认证模块设计

1. 功能

顾客办理完业务输入自己账号密码判断顾客是否为vip顾客。

2.算法描述

用函数void vip(int x) 实现，由输入账号判断顾客是否为vip用户 if(i==100&&ch[0]=='1'&&ch[1]=='1'&&ch[2]=='1'&&ch[3]=='1')......那么输出"尊敬的VIP用户您好，请您直接到VIP区办理业务！"

3.具体语言实现

vip用户认证

void vip(int x)

{

int i,a;

a=x;

char ch[3];

int k=0;

switch(a)

{

case 1:

{

printf("请输入你的卡号：");

scanf("%d",&i);

printf("请您输入五位密码：");

while(ch[k-1]!=' ')

{

ch[k]=getch();

k++;

printf("\*");

}

if(i==100&&ch[0]=='1'&&ch[1]=='1'&&ch[2]=='1'&&ch[3]=='1')

{

printf("\n尊敬的VIP用户您好，请您直接到VIP区办理业务！\n");

vip1++;

}

else if(i==200&&ch[0]=='2'&&ch[1]=='2'&&ch[2]=='2'&&ch[3]=='2')

{

printf("\n尊敬的VIP用户您好，请您直接到VIP区办理业务！\n");

vip1++;

}

else if(i==300&&ch[0]=='3'&&ch[1]=='3'&&ch[2]=='3'&&ch[3]=='3')

{

printf("\n尊敬的VIP用户您好，请您直接到VIP区办理业务！\n");

vip1++;

}

else if(i==400&&ch[0]=='4'&&ch[1]=='4'&&ch[2]=='4'&&ch[3]=='4')

{

printf("\n尊敬的VIP用户您好，请您直接到VIP区办理业务！\n");

vip1++;

}

else if(i==500&&ch[0]=='5'&&ch[1]=='5'&&ch[2]=='5'&&ch[3]=='5')

{

printf("\n尊敬的VIP用户您好，请您直接到VIP区办理业务！\n");

vip1++;

}

else printf("\n你的输入有误！\n");

break;

}

default : break;

}

}

3.7顾客评分模块设计

1. 功能

顾客办理完业务对系统服务进行评分以开发维护人员对系统进行更好的完善。

1. 算法描述

用函数void mygrade() 实现，switch (z) 语句来判断顾客的满意程度，再用printf("")语句输出评价。

1. 具体语言实现

void mygrade()

{

printf("为了改善服务质量，请你对我们的服务进行评价。谢谢！\n");

z=guitai( );

y=pingfeng( );

switch (z)

{

case 1:

{

sum1+=y;

i1++;

ave1=sum1/i1;

printf(" %d号柜台的平均满意度为%0.2f。感谢你的评分，谢谢！！！\n",z,ave1);

break;

}

case 2:

{

sum2+=y;

i2++;

ave2=sum2/i2;

printf(" %d号柜台的平均满意度为%0.2f。感谢你的评分，谢谢！！！\n",z,ave2);

break;

}

case 3:

{

sum3+=y;

i3++;

ave3=sum3/i3;

printf(" %d号柜台的平均满意度为%0.2f。感谢你的评分，谢谢！！！\n",z,ave3);

break;

}

default : printf("你的输入有误，请重新输入！\n");

}

getch();

}

4 时间复杂度分析

算法的时间复杂度是一个函数，它定量描述了该算法的运行时间。这是一个关于代表算法输入值的[字符串](https://baike.so.com/doc/630752-667548.html" \t "https://baike.so.com/doc/_blank)的长度的函数。时间复杂度常用[大O符号](https://baike.so.com/doc/1839210-1944868.html" \t "https://baike.so.com/doc/_blank)表述，不包括这个函数的低阶项和首项系数。

本系统的主要功能是对银行排队管理，采用的是比较简单的线性表的链式队列。线性表的顺序存储结构的算法思想简单明了，容易转换成相对应的程序源码，它们的时间复杂度为T(n)=O(n)。本系统中，对数据的排序采用了冒泡排序算法，其时间复杂度为T(n)= O(n ^2)。但是在对用户排队情况查询使用了选择排序，其时间复杂度为T(n)= O(n ^2)。数据的删除和插入采用顺序查找的方法，先找到相对应的记录，其时间复杂度为T(n)=O(n)。

5 系统测试

5.1 测试数据

开始录入的数据：1 2 3 4 5 6

账号：2017021140 密码：12345

输入各界面对应号码。如1：顾客到达

2：顾客离开

3：查询业务办理

4：查看业务办理情况

5：系统查询

6：退出

5.2 测试输出结果

登入系统后，先输入1客户到达。如图5-1所示。在此页面

1：顾客到达：此时顾客到达，并且得到自己的号码，系统提示输入用户类型等操作。

2：顾客离开：顾客离开顾客号码弹出队列

3：查询业务办理：顾客可通过选择3来查询当前各个窗口业务办理情况，以便判断自己办理业务时间。

4：查看业务办理情况：用户可查询当前办理业务者号码并且提示用户在用户前还有多少人排队

5：系统查询

6：退出：自此页面用户退出查询，系统向用户提示对本次服务进行评分。

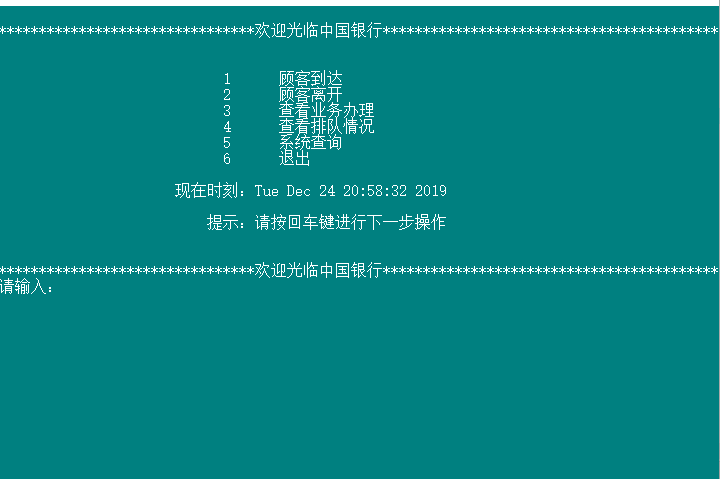


图5-1 用户到达界面

系统将进入菜单界面，如图5-2所示。

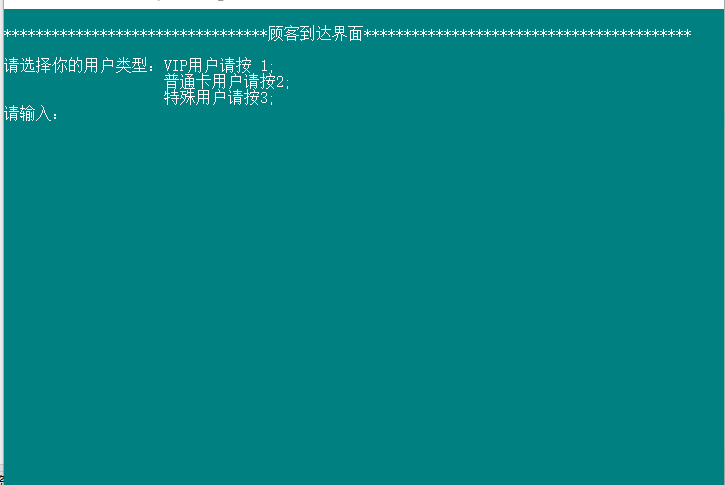


图5-2 菜单界面

在菜单中选择2选择用户类型。如图5-3所示。

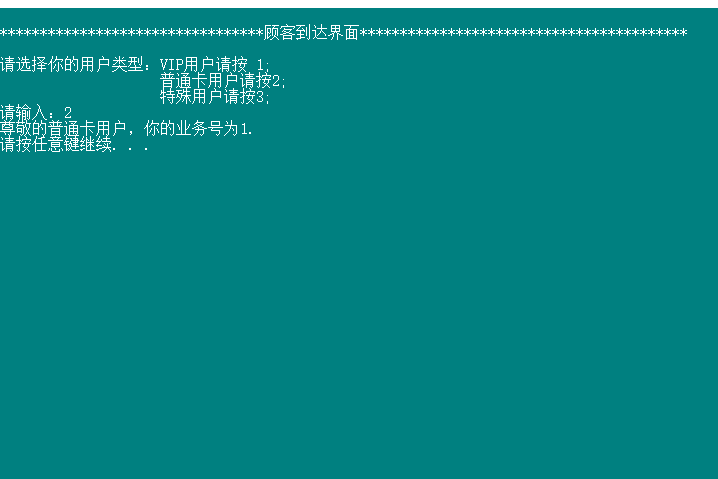


图5-3 用户类型界面

按任意键返回主菜单回到欢迎界面如图5-4所示。

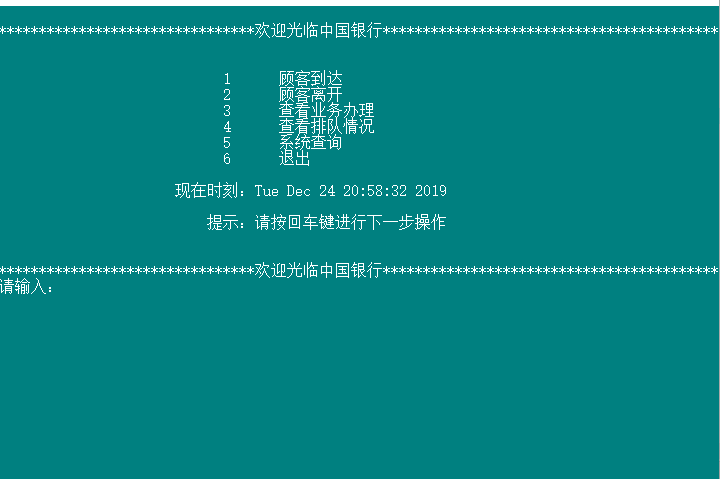


图5-4 学返回主菜单界面

查看当前排队人数如图5-5所示。

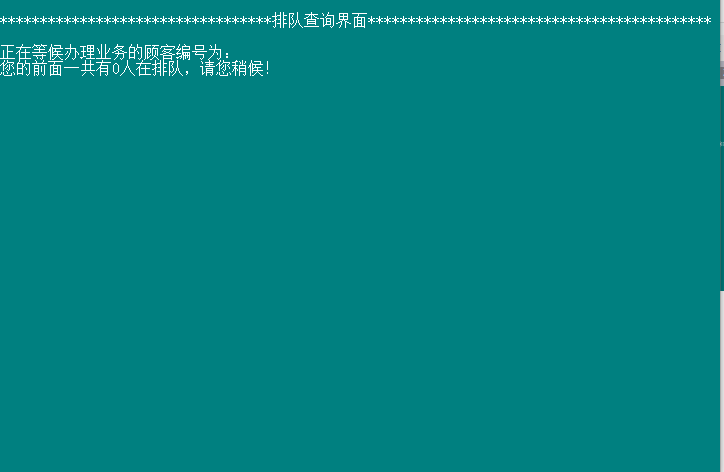


图5-5 排队查询页面

按任意键返回欢迎界面，再次按1用户到达界面第二位顾客到达系统界面如图5-6所示。

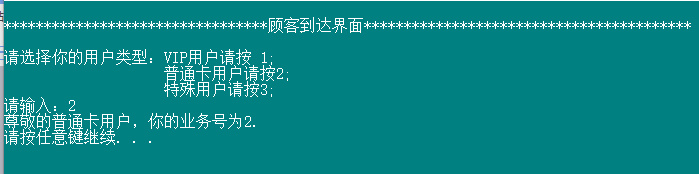


图5-6第二位顾客到达系统界面

按任意键返回欢迎界面，选择4查看当前在前面排的人数系统界面如图5-7所示。

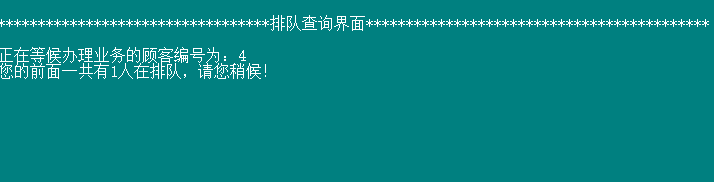


图5-7 查询前面人数界面

按任意键返回欢迎界面，再次按3到达业务查询界面如图5-8所示。

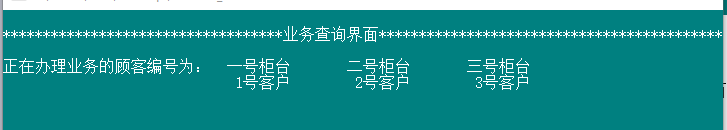


图5-8 业务查询界面

按任意键返回欢迎界面，按5到达顾客离开界面如图5-9所示

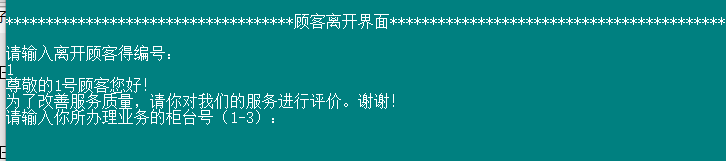


图5-9 顾客离开界面

同时进入用户评分界面，输入柜台号1进行评分如图5-10所示

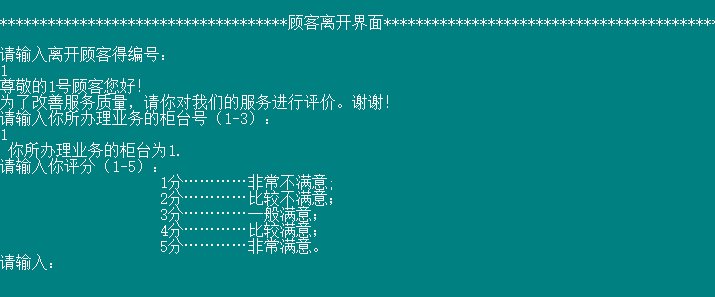


图5-10 评分界面

输入对应评分号码，完成评分。如图5-11所示

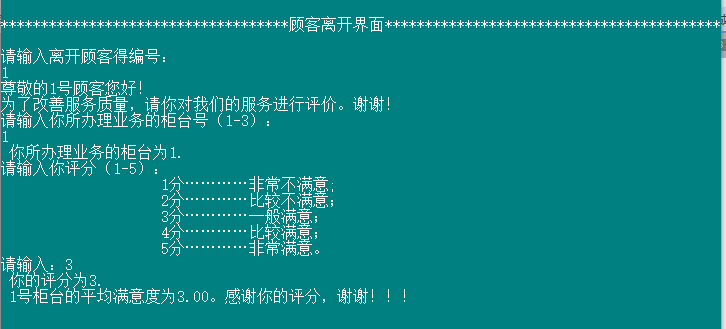


图5-11 完成评分界面

6 总结

次课程设计的顺利完成，得到了老师和同学的很多帮助。这次实现的是银行排队系统，在应用方面可以很好的解决当前银行业务排队的一些问题，更好的为银行客户服务。从课程设计开始到结束，这个过程中，也是从什么都不懂，到系统的最后顺利完工。在课程设计实现的过程中，通过不断地和同学讨论，在讨论中发现问题，最后共同寻求解决方案，从而使得该系统的功能不断完善。

这次课程设计的完成，让我明白了很多，以前觉得总是觉得自己的知识学得不够好，所以一开始没有什么信心，但是在课程设计开始之后，在不断的实现过程中，就开始发现，其实也不是那么难，最后明白了，知识只有在真正地实战操作中才能得以巩固和加强。