FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING
UNIVERSITI MALAYSIA PAHANG

UMP
Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

BCI1093 DATA STRUCTURE & ALGORITHMS
Semester 1 2018/2019

Project Title : AIRLINES RESERVATION SYSTEM

Lab Section : 06B

Lecturer Name : ROSDIANA BINTI SHAHRIL

Group Member : TAN MEI YAN
LEE PEI YIN
YEOW KAI ZE

1.0 Introduction

Airline reservation systems (ARS) are part of the so-called passenger service systems (PSS), which are applications supporting the direct contact with the passenger.

ARS eventually evolved into the computer reservations system (CRS). A computer reservation system is used for the reservations of a particular airline and interfaces with a global distribution system (GDS) which supports travel agencies and other distribution channels in making reservations for most major airlines in a single system.Airline reservation systems incorporate airline schedules, fare tariffs, passenger reservations and ticket records. An airline's direct distribution works within their own reservation system, as well as pushing out information to the GDS. The second type of direct distribution channel are consumers who use the internet or mobile applications to make their own reservations. Travel agencies and other indirect distribution channels access the same GDS as those accessed by the airline reservation systems, and all messaging is transmitted by a standardized messaging system that functions on two types of messaging that transmit on SITA's high level network (HLN). These messaging types are called Type A [usually EDIFACT format] for real time interactive communication and Type B [TTY] for informational and booking type of messages. Message construction standards set by IATA and ICAO, are global, and apply to more than air transportation. Since airline reservation systems are business critical applications, and they are functionally quite complex, the operation of an in-house airline reservation system is relatively expensive.

2.0 Case study

Happy Booking is a airline reservation system. Happy Booking allows user to register, book flight, cancel flight, search flight that are already book and view booking list. All flight are from KL. The price of the ticket show in the table below:

Destination	Price	Time	
		Flight A	Flight B
Thailand	309	0300	1900
Singapore	150	0600	2000
Sabah	300	0900	2100
Indonesia	234	1200	2200

Sample list

No	User Name	Destination	Tel No	Seat No	Flight Date
1	Lisa	Thailand	012-9856451	5	
2	Ben	Singapore	018-6662658	42	
3	Tan	Sabah	012-3658942	9	
4	Melisa	Sabah	016-9851236	18	
5	Wong	Singapore	010-9852031	35	
6	Kent	Indonesia	017-9346852	43	
7	Lee	Thailand	011-1254963	11	
8	Cindy	Indonesia	013-9547863	2	
9	Sity	Singapoe	012-5532123	53	
10	Melinda	Indonesia	016-9966521	30	
11	Gray	Sabah	012-3354667	11	
12	Low	Thailand	010-9962154	60	

Seat Plan

1	2	3		4	5	6
7	8	9		10	11	12
13	14	15		16	17	18
19	20	21		22	23	24
25	26	27		28	29	30
31	32	33		34	35	36
37	38	39		40	41	42
43	44	45		46	47	48
49	50	51		52	53	54
55	56	57		58	59	60

Case Study 1 Related : Videcom Airline Reservation Systems

VIDECOM design, develop, host and deliver automation systems to the airline and travel industries including the latest state-of-the-art airline reservation systems which provide modern, flexible reservations and inventory management solutions for airlines including call centre, travel agency and internet ticket sales and bookings. The standard solution provides as integrated Departure Control System with a very easy to use interface for rapid passenger check-in and real time access to reservations.

Inventory, schedule, fares and agent control modules provide a complete solution for airlines to manage their bookings and ticket reservations. A modern, robust and scalable reservations architecture provides a stable platform to airlines of all size and with industry standard interfaces to all external reservation systems including GDSs, other airline CRS systems, SSIM schedules and ATPCO fares.

Source : http://www.videcom.com/

Case Study 2 Related : Blue Skies Airline Reservation System

Blue Skies Airlines, a new airline, offers services between any two of the following cities: Denver; Washington, DC; Los Angeles; New York City; Atlanta; and Cleveland.

When a customer calls Blue Skies to make a flight reservation, the reservation agent first asks him or her for

The desired travel dates

The departure and destination cities

The seat grade desired (first class, business class, or economy)

The reservation agent then informs the customer of all available flights that meet his or her criteria. For each flight, the flight number, departure date and time, arrival date and time, and round-trip price are communicated to the customer. If the customer finds any of the available flights acceptable, he or she may either pay for the ticket via credit card or request that the seat be held for 24 hours. (A specific seat assignment—row and seat number—isn't issued until the seat is paid for.)

A limited number of seats on each flight are earmarked as frequent flyer seats. A customer who is a frequent flyer member may reserve and "pay for" one of these seats by giving the agent his or her frequent flyer membership number. The agent then verifies that the appropriate balance is available in the customer's account before the seat can be confirmed, at which point those miles are deducted from the account.

The customer has two ticketing options: he or she may request that a conventional "paper" ticket be issued and mailed to his or her home address, or an electronic ticket (E-ticket) may instead be assigned, in which case the customer is simply informed of the E-ticket serial number by telephone. (With an E-ticket, the customer simply reports to the airport at the time of his or her departure, and presents suitable ID to a ticket agent at the gate. No paperwork is exchanged.) In either case, the reservation agent records the serial number of the (conventional or electronic) ticket issued to this customer.

The number of seats available for a given flight in each of the seat grade categories is dependent on the type of aircraft assigned to a given flight.

Source :http://www.icodeguru.com/dotnet/Beginning-CSharp-Objects/8880final/LiB0181.html

3.0 Title And Description

No	Title	Description
1	Airlines Reservation System	-New customer for checking flight ticket and registration -Booking ticket -Delete some record of customer when want to change flight ticket -Search the record of the customer by the customer's record. -Display all the data record about the customer.

4.0 Reference

1. https://en.wikipedia.org/wiki/Airline_reservations_system
2. http://www.videcom.com/
3. http://www.icodeguru.com/dotnet/Beginning-CSharp-Objects/8880final/LiB0181.html
4. https://github.com/search?l=Java&q=airline+reservation+system&type=Repositories

5.0 Appendix

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<stdlib.h>
#define TRUE 1
#define FALSE 0

   struct ticket
   {
      int flight;
      char name[99];
      int seat;
      char book_date[15];
      int phone_no;
      struct ticket  *ptrnext;
   };
   struct ticket *headptr, *newptr, *curptr, *prevptr,*ptrnext,*min;

   void flightlist();
   void read();
   void enqueue();
   void deletea();
   void sortbyflight();

   int i;

   int main()
   {
      char ch;
      int choic=TRUE;

          headptr=NULL;

      system("cls");
      printf("\t\t------------------------------------------------\n");
      printf("\t\t\tHappy Booking\n");
      printf("\t\tAirline Reservation System\n");
      printf("\t\t------------------------------------------------\n");
      printf("\n Press any key to continue");
      getch();
      Menu:
                    printf("  \nWelcome to Happy Booking\n");
                    printf("\n");
                    for(i=0;i<80;i++)
                    printf("-");
```

```
           while(choic==TRUE)
  {
    system("cls");
    printf(" \n A  Book a flight");
    printf(" \n B  View flight");
    printf(" \n C  Cancel flight ");
    printf(" \n D  Search ");
    printf(" \n E  Sort by flight ");
    printf(" \n F  Exit");
    printf("\n\n\n\n\t\tPlease enter your choice:\n\n");

    ch=getche();
    switch(toupper(ch))
      {
        case 'A':
          read();enqueue();break;
        case 'B':
          list();
          break;
        case 'C':
          deletea();break;
        case 'D':
          search();break;
        case 'E':
          sortbyflight();break;
        case 'F':
          system("cls");
          printf("\n\n\t *THANK YOU*");
          exit(0);
          break;
        default:
          system("cls");
          printf("Wrong Input");
          printf("\n Press try again");
          goto Menu;
      }
    }
}

void flightlist()
{
  printf("\n");
  printf("\n\t\tFlight List ");
  printf("\n\t Flight No \t Destination \t Price(RM) \t Time");
  printf("\n\t 01        \t Thailand    \t 309       \T 0300");
  printf("\n\t 02        \t Thailand    \t 309       \T 1900");
  printf("\n\t 03        \t Singapore   \t 150       \T 0600");
  printf("\n\t 04        \t Singapore   \t 150       \T 2000");
```

```
    printf("\n\t 05        \t Sabah       \t 300       \T 0900");
    printf("\n\t 06        \t Sabah       \t 300       \T 2100");
    printf("\n\t 07        \t Indonesia   \t 234       \T 1200");
    printf("\n\t 08        \t Indonesia   \t 234       \T 2200");
}

void read()
{
    system("cls");

    flightlist();
    newptr = (struct ticket*)malloc(sizeof(struct ticket));


                      printf("\n Enter flight:");
                      scanf("%d",&newptr->flight);
    if(newptr->flight>=1 && newptr->flight<=8)
        {
            printf("\nSeat Number(1-60): ");
            scanf("%d",&newptr->seat);
            if(newptr->seat >60)
                {
                    printf("\nThis seat is not available.");
                }
            else if(newptr->seat >0 && newptr->seat <= 60)
                {


                    printf("Name: ");
                    scanf("%s",&newptr->name);
                    printf("Enter ticket date(dd/mm/yyyy): \n");
                    scanf("%s",&newptr->book_date);
                    printf("Phone number: ");
                    scanf("%d",&newptr->phone_no);
                    printf("\n\n1 Your flight is successfully booked!!");
                }
        }
        else
        {
            printf("\nFlight not found!");
        }

    getch();
    system("cls");
}
```

```
void enqueue()
{
   if (headptr==NULL)
   {
      headptr=newptr;
      newptr->ptrnext=NULL;
}
   else
   {

      curptr=headptr;
      while(curptr->ptrnext!=NULL)
      {
         curptr=curptr->ptrnext;
      }
      curptr->ptrnext=newptr;
      newptr->ptrnext=NULL;
   }
}

void list()
{
   system("cls");
   printf("\n Press any key to continue");
   getche();
   printf("\n");
   printf("                              Flight Booking List                              ");
   printf("\n");
   for(i=0;i<118;i++)
   printf("-");
   printf("\n");
   printf("\n\tFlight\t");
   printf("\tName\t");
   printf("\tSeat\t");
   printf("\tTicket Date ");
   printf("\t\tPhone Number\n ");
      if (headptr==NULL)
      {
         for(i=0;i<118;i++)
         printf("-");
         printf("\n\nEmpty list.\n");
      }
   else
   {
      curptr=headptr;

      while(curptr!=NULL)
      {
            prevptr=curptr;
            printf("\t%d\t\t",curptr->flight);
            printf("%s\t\t", curptr->name);
            printf("%d\t\t",curptr->seat);
            printf("%s\t\t",curptr->book_date);
            printf("\t%d\t\t\n",curptr->phone_no);
            curptr=curptr->ptrnext;
        }
    }

    printf("\n Press any key to continue");
    getch();
    system("cls");
}

void deletea()
{
    int seatdlt;
    if (headptr==NULL)
    {
        printf("\n\nThe list is empty , cannot delete!!!");
    }
    else
    {
        printf("\nWhat seat you want to delete: ");
        scanf("%d",&seatdlt);

        curptr=headptr;
        while(curptr != NULL)
        {
            if (curptr->seat == seatdlt)
            {

                prevptr->ptrnext=curptr->ptrnext;
                free(curptr);

                break;
            }
        else
            {
                prevptr=curptr;
                curptr=curptr->ptrnext;
            }
        }

    }
}
```

```
void search()
{
   int flightno;

   printf("\n\n\nWhat flight you want to search: ");
   scanf("%d",&flightno);

      curptr=headptr;
      if(curptr != NULL)
      {
         if (curptr->flight == flightno)
         {
            printf("\n\n\nRecord is found!!!");
            printf("\n\n\nFlight: %d", curptr->flight);
            printf("\nName: %s", curptr->name);
            printf("\nSeat: %d",curptr->seat);
            printf("\nticket Date:%s",curptr->book_date);
            printf("\nPhone No:%d\n\n\n\n",curptr->phone_no);
            printf("\n Press any key to continue");
            getche();
         }
         else
         {
            prevptr=curptr;
            curptr=curptr->ptrnext;
         }
      }
      else
      {
         printf("No record found in this flight!");
         printf("\n Press any key to continue");
         getche();
      }
   }


void sortbyflight()
{
   {
   curptr = headptr;

   while(curptr->ptrnext)
      {
         min = curptr;
```

```
            newptr = curptr->ptrnext;

            while(newptr)
            {

                if( min->flight > newptr->flight )
                {
                    min = newptr;
                }

                newptr = newptr->ptrnext;
            }
                int temp = curptr->flight;
                curptr->flight = min->flight;
                min->flight = temp;
            curptr = curptr->ptrnext;
        }
    }
}
```

6.0 Meeting Report

Meeting Report 1

Date: 28 November 2018
Time: 3pm - 5pm
Discussion:
In the first discussion, the case study was being discussed. We research more information about our case study. After that, we collect and compile all the information we get and make finalisation for choosing the mini project title. We also did work distribution.

Meeting Report 2

Date: 5 December 2018
Time: 3pm - 5pm
Discussion:
In this meeting, we discussed all the problem we met and find the solution together.

Meeting Report 3

Date: 12 December2018
Time: 2pm - 5pm
Discussion:
The meeting was started by collecting all the parts we did and make correction in some certain part to make it perfect and smooth. We also do compiling all the parts and finalise our assignment.

7.0 Work distribution

Task	Name
Struct, main,sort, dequeue + organise, search + report	Tan Mei Yan
Branch, insert + compile, analyse + report	Lee Pei Yin
Enqueue, display, search + help check, organisation	Yeoh Kai Ze