****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Summer, Year: 2022), B.Sc. in CSE (Day)**

**Course Title: Data Structure Lab**

**Course Code: CSE-106 Section: DD**

**Lab Project Name: Restaurant Management System**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
|  | Md. Sohanur Rahman | 213902106 |

**Submission Date: 11.09.2022**

**Course Teacher’s Name: Md. Sultanul Islam Ovi**

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Project Status**  **Marks: ………………………………… Signature: .....................**  **Comments: .............................................. Date: ..............................** |

Table of Contents

**Chapter 1 Introduction 3**

1.1 Introduction 3

1.2 Design Goals/Objective 3

**Chapter 2 Design/Development/Implementation of the Project 4**

2.1 Section (Choose the name of this section as appropriate with your project) 4

2.2 Section (Choose the name of this section as appropriate with your project) 4

2.2.1 Subsection 4

**Chapter 3 Performance Evaluation 5**

3.1 Simulation Environment/ Simulation Procedure 5

3.2 Results and Discussions 5

**Chapter 4 Conclusion 6**

4.1 Introduction 6

4.1 Practical Implications 6

4.2 Scope of Future Work 6

**References 7**

# Chapter 1

## 1.1:Introduction

This restaurant management system can be used by employees in a restaurant to handle the clients,their orders and can help them easily find free tables. The restaurant menu is organized bycategories (appetizers, soups, salads, entrees, sides and drinks) of menu items. Each menu itemhas a name, price and associated recipe. A recipe for a menu item has a chef, preparationinstructions and associated ingredients.Restaurant management system is the system for manage the restaurant business. Restaurantmanagement can vary across multiple management styles, however, there is always one common

denominator when it comes to setting goals: maximizing a restaurant’s profitability. In order tomaximize a restaurant’s profitability, one has to always examine and understand a restaurant’soperational costs and how these relate to a restaurant’s productivity and efficiency in delivering

quality service to its customers. Management takes a very important role in controlling andmanipulating the balance of costs and profitability. An effective manager must always concernhimself/herself with restaurant issues that pertain to inventory/stocking, pricing, order-taking, and

## 1.2: Objective

The main objective is to maximize the profit by increasing efficient and decreasing the mistakesthat takes place in the kitchen, this will be done without compromising customer satisfaction. Atthis moment of time, there are still numerous restaurants that still use paper based system to getmessages across between the restaurant and the kitchen, this way of communication is one of theleast efficient method. However, this approach may be implemented and designed in a successful profitable restaurant but there are numerous problems which might be seen as reducing therestaurants efficiency, they are the following:

1.The lack of communication that is caused by handwriting.

2.Uncontrolled order logging (poor order taking).

 3.Unproductive communication between restaurant and kitchen.

 4.Faults with order taking and lack of time management.

5.Lack of good quality stock management.

# 

# 

# Chapter 2

# 2.1: Implimentation

# code:

|  |
| --- |
| //Header file  #include<stdio.h>  #include<stdlib.h>  #include<string.h>  //Represent a node  struct node  {      int number;      char foodname[100];      int quantity;      float price;      struct node \*next;  };  //create some node  struct node \*heada=NULL;  struct node \*taila=NULL;  struct node \*newnode=NULL;  struct node \*headc=NULL;  struct node \*tailc=NULL;  //It show the option for admin  void admin\_menu()  {      printf("\n");      printf("1.View order menu\n");      printf("2.Add item in the menu\n");      printf("3.Delete item from the menu\n");      printf("4.Back to main menu\n");      printf("Enter your choose --> ");  }  //It show the option for customer  void customer\_menu()  {      printf("1.View order menu\n");      printf("2.Press for order food items\n");      printf("3.Delete item from your order\n");      printf("4.View total bill\n");      printf("5.Back to main menu\n");      printf("Enter your choose --> ");  }  //create order menu by admin  struct node \* create\_admin(struct node \*head ,int number,char foodname[50],float price)  {      struct node \*temp1=NULL;      newnode=(struct node \*)malloc(sizeof(struct node));      temp1=head;      newnode->number=number;      newnode->quantity=0;      strcpy(newnode->foodname,foodname);      newnode->price=price;      newnode->next=NULL;      if(temp1==NULL){          taila=heada=newnode;      }      else{          while(temp1->next!=NULL)          {              temp1=temp1->next;          }          taila=temp1;          temp1->next=newnode;      }      return heada;  }  //Create food menu which customer want  struct node \*create\_customer(struct node \*head,int number,int quantity)  {     struct node \*temp;     int flag=0;     temp=heada;     newnode=(struct node \*)malloc(sizeof(struct node));     while(temp!=NULL)     {      if(temp->number==number)      {          flag=1;          break;      }      temp=temp->next;     }     if(flag==1)     {      newnode->number=number;      newnode->quantity=quantity;      strcpy(newnode->foodname,temp->foodname);      newnode->price=quantity\*(temp->price);      newnode->next=NULL;      struct node \*temp1;      temp1=head;      if(temp1==NULL)      {          tailc=headc=newnode;      }      else{          while(temp1->next!=NULL)          {              temp1=temp1->next;          }          tailc=temp1;          temp1->next=newnode;      }     }     else     {      printf("Please Enter a Valid Number !!\n");     }     return headc;  }  //It can be use for both admin and customer for delete  struct node \* delete(struct node \*head,int number,struct node \*tail)  {       if(head==NULL)       {          printf("Menu is Empty !!\n");       }       else       {          struct node \*temp;        if(head->number==number)       {          temp=head;          head=head->next;          free(temp);       }       else if(tail->next->number==number){          temp=tail->next;          tail->next=NULL;          free(temp);       }       else{          struct node \*temp1;          temp=head;             while(temp->number!=number){              temp1=temp;              temp=temp->next;             }             temp1->next=temp->next;             free(temp);       }     }     return head;  }  //Admin can delete a food item  void delete\_admin()  {      printf("Enter the number of food item for delete -->");      int number;      scanf("%d",&number);      struct node \*temp;      temp=heada;      while(temp!=NULL){          if(temp->number==number){          heada=delete(heada,number,taila);          }          temp=temp->next;      }  }  //Customer can delete from his order  void delete\_customer()  {      printf("Enter the number of food item for delete -->");      int number;      scanf("%d",&number);      struct node \*temp;      temp=headc;      while(temp!=NULL){          if(temp->number==number){          headc=delete(headc,number,tailc);          }          temp=temp->next;      }  }  //It count customer total bill  void customer\_bill()  {      struct node \*temp;      temp=headc;      float total\_bill=0;      while(temp!=NULL)      {          total\_bill=total\_bill+temp->price;          temp=temp->next;      }      printf("\nYOUR TOTAL BILL --> %0.2f\n",total\_bill);  }  //Sort Function  void sort(struct node \*head)  {      struct node \*ptr1,\*ptr2;      struct node \*temp;      temp=(struct node\*)malloc(sizeof(struct node));      ptr1=head;      while(ptr1!=NULL)      {      ptr2=ptr1->next;      while(ptr2!=NULL)      {          if(ptr1->number>ptr2->number)          {            temp->number=ptr1->number;            temp->quantity=ptr1->quantity;            temp->price=ptr1->price;            strcpy(temp->foodname,ptr1->foodname);            ptr1->number=ptr2->number;            ptr1->quantity=ptr2->quantity;            ptr1->price=ptr2->price;            strcpy(ptr1->foodname,ptr2->foodname);            ptr2->number=temp->number;            ptr2->quantity=temp->quantity;            ptr2->price=temp->price;            strcpy(ptr2->foodname,temp->foodname);          }          ptr2=ptr2->next;      }      ptr1=ptr1->next;      }  }  //Display the order menu or food menu  void display\_menu(struct node \*ptr)  {      struct node \*temp;      temp=ptr;      if(temp==NULL){          printf("Menu is Empty!!\n");      }      else{          while(temp!=NULL){              if(temp->quantity==0)              {              printf("\n%d %s  %0.2f",temp->number,temp->foodname,temp->price);              }              else{       printf("%d %s %d  %0.2f\n",temp->number,temp->foodname,temp->quantity,temp->price);              }              temp=temp->next;          }      }      printf("\n");  }  //It is the starting point of RMS  void main\_menu()  {      printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");      printf("\t  WELCOME TO RESTAURANT MANAGMENT SYSTEM\n");      printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");      printf("\n");      printf("1.ADMIN MENU\n");      printf("2.CUSTOMER MENU\n");      printf("3.EXIT\n");      printf("Enter your choose --> ");  }  //This section admin can choose which he want  void admin()  {      printf("\n");      printf("---------------------------------\n");      printf("       ADMIN SECTION\n");      printf("----------------------------------\n");      printf("\n");      while(1)//Infinite loop      {        admin\_menu();        int option;        scanf("%d",&option);        if(option==4){          break;        }        switch(option)        {          case 1:          printf("\n");          printf("\n\*\*\*\*\*\*\*\*ORDER MENU\*\*\*\*\*\*\*\*\n");          display\_menu(heada);          break;          case 2:          printf("\nEnter serial number of food item -->");          int num,flag=0;          char name[50];          float price;          scanf("%d",&num);          struct node \*temp;          temp=heada;          while(temp!=NULL)          {              if(temp->number==num)              {                  printf("Wrong Serial Number !!\n");                  flag=1;                  break;              }              temp=temp->next;          }          if(flag==1)          break;          printf("Enter food item name --> ");          fflush(stdin);          gets(name);          printf("Enter price of food item -->");          scanf("%f",&price);          heada=create\_admin(heada,num,name,price);          printf("\nNew food item added in the menu\n");          printf("\n");          break;          case 3:          delete\_admin();          printf("\n\*\*\*UPDATED FOOD MENU\*\*\*\n");          printf("\n");          display\_menu(heada);          printf("\n");          break;          default:          printf("Wrong Choose!!\n");          break;        }      }  }  //This section customer can choose which they want  void customer()  {      printf("\n");      printf("-----------------------------------\n");      printf("      CUSTOMER SECTION\n");      printf("----------------------------------- \n");      printf("\n");      while(1) //Infinite loop      {          customer\_menu();          int option;          scanf("%d",&option);          if(option==5)          {              break;          }          switch(option)          {              case 1:              printf("\n");              printf("\n\*\*\*\*\*\*\*\*ORDER MENU\*\*\*\*\*\*\*\*\*\n");              display\_menu(heada);              printf("\n");              break;              case 2:              printf("Please enter the food number you want to order -->");              int number,quantity;              scanf("%d",&number);              printf("please enter the quantity -->");              scanf("%d",&quantity);              headc= create\_customer(headc,number,quantity);              printf("\n\*\*\*YOUR ORDERED FOOD ITEMS\*\*\*\n");              sort(headc);//here called sort function              display\_menu(headc);              printf("\n");              break;              case 3:              delete\_customer();              printf("\*\*\*UPDATED ORDER LIST\*\*\*\n");              printf("\n");              display\_menu(headc);              printf("\n");              break;              case 4:              customer\_bill();              printf("\n");              break;              default:              printf("\nPlease Enter a Valid Option\n");              break;          }      }  }  //Main function  int main()  {      //Initialize food number,name and price      heada=create\_admin(heada,1,"Fried Rice",200);      heada=create\_admin(heada,2,"Korian Soup",30);      heada=create\_admin(heada,3,"Black Coffee",45);      while(1) //It is infinite loop      {      main\_menu();      int choose;      scanf("%d",&choose);      if(choose==3)      {          printf("\n");          printf("\*\*\*\*\*Thank you\*\*\*\*\*\n");          break;      }      switch(choose)      {          case 1:          admin();          break;          case 2:          customer();          break;          case 3:          break;          default:          printf("please enter the valid option");          break;      }      }  } |

# Screen shots :

# 

Figure 1: Main Menu interface

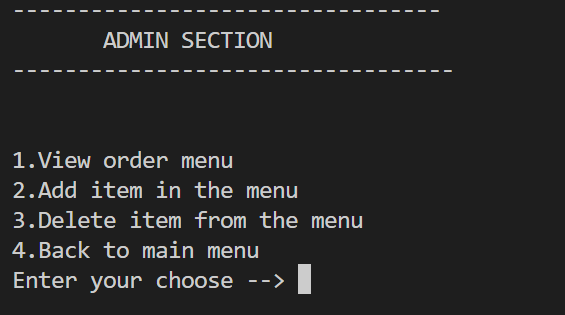


Figure 2: Admin Section

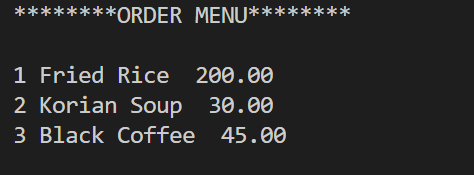


Figure 3: Order menu

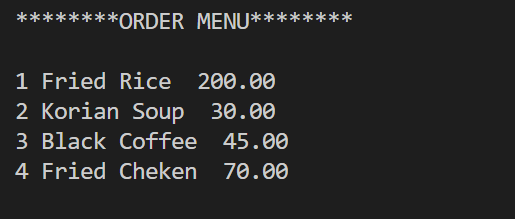


Figure 4: Order menu after added food item

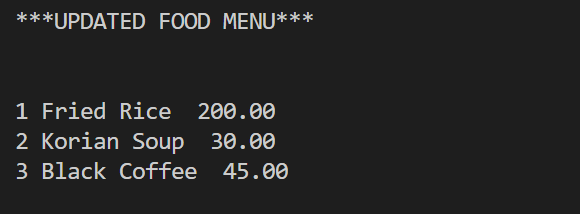


Figure 5: Order menu after added food item

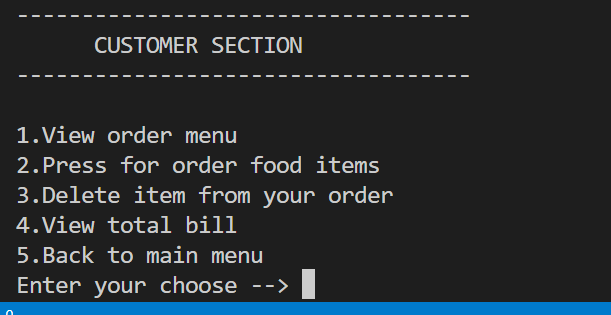


Figure 6: Customer section

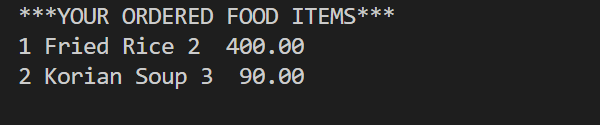


Figure 7: Ordered food item from customer



Figure 8: Total bill of ordered food item

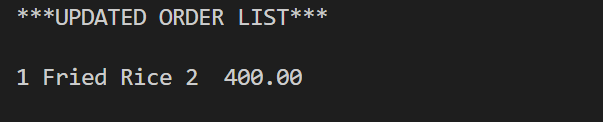


Figure 9: Order menu after delete food item

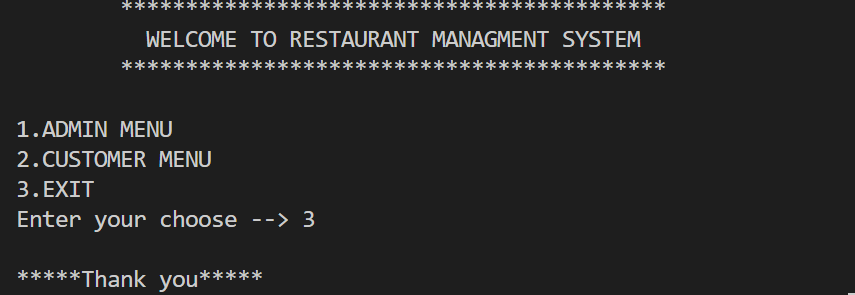


Figure 10: Exit the program

**Chapter 3**

**Conclusions :**

We were able to create a computerized system for Silk Route to maintain billing & Restaurantrecords .This system is able to store billing records securely and retrieve the records wheneverneeded easily.Data entering of customers and employees are also included in this system alongwith the order and the billing process.Customers,restaurant records and employees areinterconnected in order to maintain the accuracy of this system .This system can also be furtherimproved adding many other features and including the other systems as well. Finally we believethat we were able to launch an effective computerized system to the restaurant causing therestaurant to perform well in the future regarding the billing and restaurant records.

**Future Scope:**

1.Enhanced and Rich UI will be Designed.

2.Special Offer will be implemented.

3.Another payment option will be included.

4.User Will get a Discount of 30% in an order of Rs 300 or above

# References

1. Author Initial. Author Surname, Title. City: Publisher, Year Published, p. Pages Used.
2. A. Rezi and M. Allam, ”Techniques in array processing by means of transforma- tions, ” in Control and Dynamic Systems, Vol. 69, Multidemsional Systems, C. T. Leondes, Ed. San Diego: Academic Press, 1995, pp. 133-180.
3. O. B. R. Strimpel, ”Computer graphics,” in McGraw-Hill Encyclopedia of Science and Technology, 8th ed., Vol. 4. New York: McGraw-Hill, 1997, pp. 279-283.
4. K. Schwalbe, Information Technology Project Management, 3rd ed. Boston: Course Technology, 2004.
5. M. N. DeMers, Fundamentals of Geographic Information Systems, 3rd ed. New York: John Wiley, 2005.
6. M. Bell, et al., Universities Online: A survey of online education and services in Australia, Occasional Paper Series 02-A. Canberra: Department of Education, Science and Training, 2002.
7. Bdren, http://vsession.bdren.net.bd/faculty/classSchedules. [Accessed date: 13.09.21]