PIMPRI CHINCHWAD EDUCATION TRUST'S

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Project on

Hotel Management Software

(HMS)

By

Siddhesh Patil (SYITB222)

Manthan Patil (SYITB224)

Kalpesh Somwanshi (SYITB248)

Chaitanya Sonawane (STITB250)

Under The Guidance of

Meera Thorat

CONTENTS

1. Abstract……03

2. Introduction…….....04

3. Purpose and Scope……...05

4. Objective……..05

5. Details Of Design…….06

   a. Singly Linked List…….06
   b. Doubly Linked List……...06
   c. File Handling…….……....07
   d. Functions Used……..09

6. Experimental Work……...11

   a. Code……..11
   b. Output……24

1.Abstract

Customers are performing the big role while anyone thinking about the hotel. The success of hotel is mainly dependent on the customer's satisfaction because it affects both directly and indirectly to the hotel management business. In hotels currently available systems facilities are traditional they are lacking most of the expected facilities of modern and new world customer. As the time is more valuable for the modern customer, they demand to give good service which involve less time consumption. In order to maintain hotel reputation and Customer demands, we build the new software which helps hotel manager to maintain time consumption and give appropriate order to customers. The main role of this software is to satisfy the customer. This software helps the manager to manage the whole day bills of customers. By improving customer's satisfaction, they will attract more customer towards the hotels and due to these hotel management and economical status improves automatically

2.Introduction

In Today's condition, Hotels are the fast-growing industries. But due to small issues they lose their customers. Then our software is the solution on it to handle those problems and increase hotels economic condition. Hotel management software (HMS) is contained two section manager section and customer section. Fist is manager section it includes View total sales of the hotel, Add new items in the order menu, Delete items from the order menu, Display order menu and second is customer section it includes Place order, view ordered items by customers, delete items from order and display final bill which include GST.

HMS helps customer to delete items from order or add in it. The bill of customer is seen by both customer as well as manager, because of this time of both manager and customer saved.

In this project we include the concepts like linked lists, file handling, etc. linked list helps mostly in it as it easily shrink or grow any time as per our menu as it is a dynamic data structure . File handling is used in it to store the data of the food items. This file is accessible only for the manager, he can made changes in it like adding or deleting food items in it.

3.Purpose & Scope

1. The system can cope with the ordering and automatic billing system.
2. The system can keep reservation records and all menu items.
3. The system can produce several reliable reports.
4. The system can control cost of operation.
5. The system can improve the customer satisfaction.

4.Objective

- Hotel Management Software (HMS) is designed to handle all the primary information of hotel.
- It uses the linked lists which helps us to access data in easier and faster way.
- It is helpful for calculate final bill and total sales.
- It is also helpful for updating menu-card and order.
- It helps to improve efficiency and effectiveness of hotel.

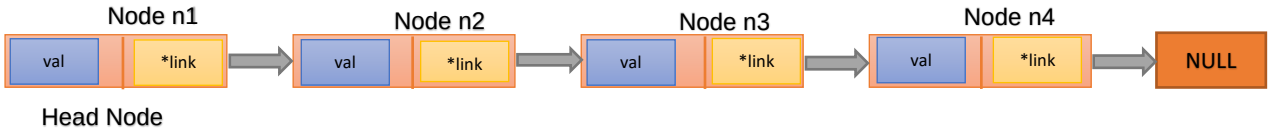
5.Details Of Design

❖ Singly Linked List :

SLL is Linear Data Structure which has nodes with a data field and a next link field.

It is similar to array but in array the size of array is fixed but in linked list we can change or update the size.

SLL can only traverse in one direction.

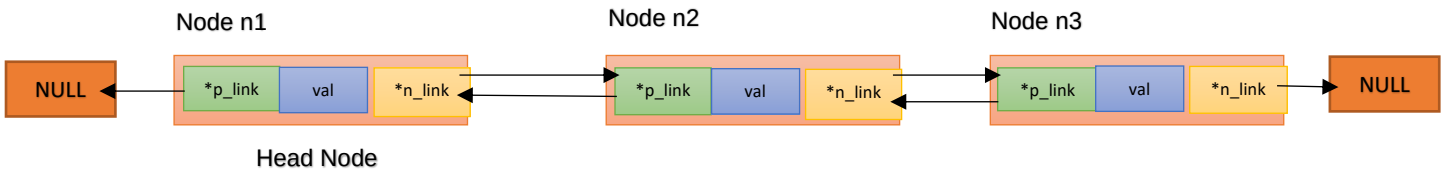


❖ Doubly Linked List :

DLL is special variation of linked list. It is also linear data structure which has node that node contains next pointer, previous pointer and data.

DLL is similar to SLL the difference is only that node also points to its previous node.

Using doubly linked list can traverse both direction front as well as backward.



Some Standard Linked List Operations :

- Insert: Add a new node in linked list at a particular position.
- Traverse: Start visiting from head node to the last node.
- Prepend: Places the node to the beginning of linked list.
- Append: Places the node to the end.
- Count: No of nodes in linked list.
- Delete: Pop out a node from the list.

❖ File Handling:

In our day-to-day life we don't require large amount of memory for processing as a very small amount of data is under consideration. But sometimes it happens that data to be processed is too large to be stored in primary memory. Therefore files come in use in this case as data can be stored in the file in text or binary format and the data can also be retrieved from the file for processing. Luckily C language provides us with the concept of file handling with the help of which we can perform various operations in file.

- **Opening a file:**

fopen() function is used to open a file in C but before opening the file we need to learn about the required access modes with the help of which we can open the file.

The various file modes using which we can open the files in C are given below:

- **"r" –** If we open the file in "r" mode first of all file is searched in disk. If there is enough space on the disk to perform the operations fopen() transfer it into buffer. If the file is not present on the disk fopen() returns NULL. It also returns NULL if file is not able to open due to insufficient space in disk.
- **"w" –** If we open the file in "w" mode first of all file is searched in disk. The contents of file are overwritten and the pointer is set to the begin of the file, if the file is present on the disk. If the file is not present in the disk, a new file is created and pointer is set to beginning. NULL value is returned if file is not opening due to space issues on the disk.

- **“a” –** When we open the file in “a” mode first of all file is searched in disk. After opening the file fopen( ) loads the file into buffer and adds the data to the end of file. If the file is not present on the disk due to some insufficient space on disk, a new file is created.

In addition to all this access modes we also have “r+”, “w+” and “a+” modes which provide some utilities in addition to their normal functions.

- **Reading from a file:**

  We can perform read operations in file using fgetc(), fscanf() or fgets() functions. fgetc() is used to read the characters from the file one by one. fgets() on the other hand is used read strings from the file and has a additional parameter which defines the length up to which the string should be read. fscanf() is used to read a whole record from text or binary files.

  Syntax:

```
FILE * fp1;

FILE * fp;

char ch = fgetc(fp);

fp1 = fopen(“data.dat”, “r+”);

fscanf(fp, "%s %s %s", pune, mumbai, delhi);
```

- **Writing a file:**

  We can perform write operations in file using fputc(), fprintf() or fputs() functions. fputc() is used to write the characters one by one to the file. fputs()is used write strings to the file and also has a additional parameter which defines the length up to which the string should be written. fprintf() is used to write a whole record to text or binary files.

- **Closing a file**:

  We should close the file every time after using and operating on it. To close a file fclose() function is used.

  Code for closing a file whose name is “data” is given below

```
FILE *Dhule;

fclose(Dhule);
```

❖ Functions Used:

a) **struct node *createmenu(struct node *head, int d, char foodname[25], float price):** Used to create a node for manager’s linked list where all the menu items are stored

b) **struct node *createorder(struct node *head, int d, int quantity):** Used to create a node for customer’s linked list where all the menu items are ordered by customer are stored

c) **void displayMenu(struct node *head):** This function displays the entire food menu to the customer by taking input from file.

d) **void displayOrder(struct node *head):** Displays the food items tht were ordered by the customer from food menu.

e) **struct node *totalsales(int d, int quantity):** This function maintains the track of total sales which is done by maintaining another linked list which keeps a track of total sales made to each customer represented by each node.

f) **void calculatetotsales():** Performs the task of calculating total sales for each customer.

g) **struct node *delete (int dt, struct node *head, struct node *tail):** This function performs the task of deleting the orders from customer linked list i.e. Customer’s ordered food item.

h) **struct node *deleteM(int dt, struct node *head, struct node *tail)**: This function performs the task of deleting a particular food item from the Restaurant menu file and it also deletes particular node from manager linked list.

i) **void displaybill():** Displays the total bill of food items ordered by the customer in the restaurant.

j) **struct node *deleteList(struct node *head):** This function performs the task of deleting the entire linked list of the customer after the total bill of food items ordered by customer is processed.

k) **void Manager():** This function opens up the manager section and provides it's interface and various functionalities.

l) **void customer():** This function opens up the customer section and provides it's interface and various functionalities.

6.Experimental Work

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct node
{
    char F_NM[50];
    int qt;
    float price;
    int d;
    struct node *pre;
    struct node *nx;
}FM;

FM *h_c = NULL, *n_n, *t_c = NULL;
FM *h_a = NULL, *t_a = NULL;
FM *h_s;
FM *createmenu(FM *h, int d, char F_NM[25], float price);
FM *createorder(FM *h, int d, int qt);
void displayMenu(FM *h);
void displayOrder(FM *h);
FM *totalsales(int d, int qt);
void calculatetotsales();
FM *delete (int dt, FM *h, FM *tail);
FM *deleteM(int dt, FM *h, FM *tail);
int deleteMenu();
int deleteOrder();
void displaybill();
FM *deleteList(FM *h);
void Manager();
void customer();

int main()
{
    h_a = createmenu(h_a, 101, "Cheese-Pizza", 330);
    h_a = createmenu(h_a, 102, "Special-Pav-Bhaji", 210);
    h_a = createmenu(h_a, 103, "Manchurian-Noodles", 400);
    h_a = createmenu(h_a, 104, "Cheese-Burger", 250);
    h_a = createmenu(h_a, 105, "Cherry-Pineapple", 350);
```

```
    FM *demo = h_a;

    FILE *f_p;
    f_p = fopen("Menu.txt", "w");
    if (f_p == NULL)
        printf("\nError");
    else
    {
        while (demo != NULL)
        {
            fprintf(f_p, "%d \t%s \t%f\n", demo->d, demo->F_NM,
demo->price);
            demo = demo->nx;
        }
    }
    fclose(f_p);
    int choice;
    do
    {
        printf("\n==============================================
==========\n");
        printf("           WELCOME TO TAJ HOTEL\n");
        printf("==============================================
=========\n\n\n");
        printf("          1. MANAGER DESK   \n");
        printf("          2. CUSTOMER DESK \n");
        printf("          3. Exit \n\n");
        printf("          Enter Your Choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            Manager();
            break;

        case 2:
            customer();
            break;

        case 3:
            printf("          *THANK YOU FOR VISITING*\n");
```

```
            break;

        default:
            printf("          Wrong Input, Choose valid
option\n");
            break;
        }
    } while (choice != 3);
}

FM *createmenu(FM *h, int d, char F_NM[25], float price)
{

    n_n = (FM *)malloc(sizeof(FM));
    n_n->d = d;
    n_n->price = price;
    n_n->qt = 0;
    strcpy(n_n->F_NM, F_NM);
    n_n->nx = NULL;
    n_n->pre = NULL;

    FM *demo = h_a;

    if (demo == NULL)
        h_a = t_a = n_n;
    else
    {
        while (demo->nx != NULL)
        {
            demo = demo->nx;
        }
        demo->nx = n_n;
        n_n->pre = t_a;
        t_a = n_n;
    }

    return h_a;
}

FM *createorder(FM *h, int d, int qt)
{
    n_n = (FM *)malloc(sizeof(FM));
```

```
    FM *demo1 = h_a;
    int flag = 0;
    while (demo1 != NULL)
    {
        if (demo1->d == d)
        {
            flag = 1;
            break;
        }
        demo1 = demo1->nx;
    }

    if (flag == 1)
    {
        n_n->d = d;
        n_n->price = qt * (demo1->price);
        n_n->qt = qt;
        strcpy(n_n->F_NM, demo1->F_NM);
        n_n->nx = NULL;
        n_n->pre = NULL;

        FM *demo = h;

        if (demo == NULL)
            h_c = t_c = n_n;
        else
        {
            while (demo->nx != NULL)
                demo = demo->nx;

            demo->nx = n_n;
            n_n->pre = t_c;
            t_c = n_n;
        }
    }
    else
    {
        printf("          This food item is not available in the
menu card!\n");
    }
```

```
    return h_c;
}

void displayMenu(FM *h)
{
    int da;
    float price;
    char name[100];
    FILE *f_p;
    f_p = fopen("Menu.txt", "r");
    int n = 0;
    while (!feof(f_p))
    {
        fscanf(f_p, "%d %s %f", &da, name, &price);
        n++;
    }
    rewind(f_p);

    printf("\n");
    printf("        Sr.no.\t    FoodName  \t       Price\n");
    printf("     ------------------------------------------------
-----\n");
    if (f_p == NULL)
        printf("\nError");
    else
        for (int i = 0; i < n - 1; i++)
        {
            fscanf(f_p, "%d %s %f", &da, name, &price);
            printf("          %d\t%s     \t%0.2f\n", da, name,
price);
        }
    fclose(f_p);
}

void displayOrder(FM *h)
{
    FM *demo1 = h;
    if (demo1 == NULL)
    {
        printf("\n          No Order!!\n\n");
    }
    else
```

```
    {
        printf("\n");
        printf("        Sr.no.\t    FoodName  \t    Quantity\tPri
ce\n");
        printf("     ----------------------------------------
---------\n");

        while (demo1 != NULL)
        {
            if (demo1->qt == 0)
                printf("          %d\t%s     \t%0.2f\n", demo1-
>d, demo1->F_NM, demo1->price);
            else
            {
                printf("          %d\t%s     \t%d     \t%0.2f\n",
demo1->d, demo1->F_NM, demo1->qt, demo1->price);
            }

            demo1 = demo1->nx;
        }
        printf("\n");
    }
}

FM *totalsales(int d, int qt)
{
    n_n = (FM *)malloc(sizeof(FM));
    int flag = 0;

    FM *demo1 = h_a;
    while (demo1->d != d)
    {
        demo1 = demo1->nx;
    }

    n_n->d = d;
    n_n->price = qt * (demo1->price);
    n_n->qt = qt;
    strcpy(n_n->F_NM, demo1->F_NM);
    n_n->nx = NULL;
    n_n->pre = NULL;
```

```
    FM *demo = h_s;

    if (demo == NULL)
        h_s = n_n;
    else
    {
        while (demo->nx != NULL)
        {
            if (demo->d == d)
            {
                flag = 1;
                break;
            }
            demo = demo->nx;
        }

        if (flag == 1)
        {
            demo->qt += n_n->qt;
            demo->price += n_n->price;
        }
        else
        {
            demo->nx = n_n;
        }
    }

    return h_s;
}

void calculatetotsales()
{
    FM *demo = h_c;
    while (demo != NULL)
    {
        h_s = totalsales(demo->d, demo->qt);
        demo = demo->nx;
    }
}

FM *delete (int dt, FM *h, FM *tail)
{
```

```
    int a;
    if (h == NULL)
    {
        printf("\n          No order available!!\n");
    }
    else
    {
        FM *demo;
        if (dt == h->d)
        {
            demo = h;
            if (h != NULL && h->qt == 1)
            {
                h = h->nx;
                h->pre = NULL;
                free(demo);
            }
            (h->qt)--;
        }
        else if (dt == tail->d)
        {
            demo = tail;
            if (tail->qt == 1)
            {
                tail = tail->pre;
                tail->nx = NULL;
                free(demo);
            }
            (tail->qt)--;
        }
        else
        {
            demo = h;
            while (dt != demo->d)
            {
                demo = demo->nx;
            }
            if (demo->qt == 1)
            {
                (demo->pre)->nx = demo->nx;
                (demo->nx)->pre = demo->pre;
                free(demo);
```

```
            }

            (demo->qt)--;
        }
    }
    return h;
}

FM *deleteM(int dt, FM *h, FM *tail)
{
    int a;
    if (h == NULL)
    {
        printf("\n          No Order Available\n");
    }
    else
    {
        FM *demo;
        if (dt == h->d)
        {
            demo = h;
            h = h->nx;
            if (h != NULL)
            {
                h->pre = NULL;
                free(demo);
            }
        }
        else if (dt == tail->d)
        {
            demo = tail;
            tail = tail->pre;
            tail->nx = NULL;
            free(demo);
        }
        else
        {
            demo = h;
            while (dt != demo->d)
            {
                demo = demo->nx;
            }
```

```
            (demo->pre)->nx = demo->nx;
            (demo->nx)->pre = demo->pre;
            free(demo);
        }
    }
    return h;
}

int deleteMenu()
{
    printf("\n          Enter Sr.No. of the food item which you
want to delete: ");

    int num;
    int a = 0;
    scanf("%d", &num);
    FM *demo = h_a;
    while (demo != NULL)
    {
        if (demo->d == num)
        {
            h_a = deleteM(num, h_a, t_a);
            a = 1;
            break;
        }
        demo = demo->nx;
    }

    char nm[100];
    int t = 0;
    int dat;
    float p;
    FILE *f_p, *ft;
    f_p = fopen("Menu.txt", "r");
    ft = fopen("data1.txt", "wb+");
    while (!feof(f_p))
    {
        fscanf(f_p, "%d %s %f", &dat, nm, &p);
        t++;
    }
    rewind(f_p);
    for (int i = 0; i < t - 1; i++)
```

```
    {
        fscanf(f_p, "%d %s %f", &dat, nm, &p);

        if (dat != num)
        {
            fprintf(ft, "%d\t %s\t %f\n", dat, nm, p);
        }
    }

    fclose(f_p);
    fclose(ft);
    remove("Menu.txt");
    rename("data1.txt", "Menu.txt");

    if (a == 1)
        return 1;
    return 0;
}

int deleteOrder()
{

    int num, Qty;
    printf("\n          Enter Sr.No. of the food item which you
want to delete: ");
    scanf("%d", &num);
    printf("\n          Enter qt want to delete: ");
    scanf("%d", &Qty);

    FM *demo = h_c;
    while (demo != NULL)
    {
        if (demo->d == num)
        {
            if (demo->qt == Qty)
            {
                h_c = delete (num, h_c, t_c);
            }
            else
            {
                demo->qt = demo->qt - Qty;
                demo->price = demo->price - (demo->price / Qty);
```

```
            }
            return 1;
        }
        demo = demo->nx;
    }
    return 0;
}

void displaybill()
{
    displayOrder(h_c);
    FM *demo = h_c;
    float total_price = 0;
    while (demo != NULL)
    {
        total_price += demo->price;
        demo = demo->nx;
    }

    printf("-------------------------------------------------------
------------------\n");
    printf("\n          Price :-\t\t %0.02f       \n",
total_price);
    float gst = (((105 * total_price) / 100) - total_price);
    printf("          Additional GST :-\t %.2f\n", gst);
    printf("          -------------------------------\n");
    printf("           Total Price :-\t %.2f\n", gst +
total_price);
    printf("          -------------------------------\n");
}

FM *deleteList(FM *h)
{
    if (h == NULL)
    {
        return NULL;
    }
    else
    {
        FM *demo = h;
        while (demo->nx != 0)
        {
```

```
            demo = demo->nx;
            free(demo->pre);
        }
        free(demo);
        h = NULL;
    }

    return h;
}

void Manager()
{
    printf("\n-----------------------------------------------\n");
    printf("          MANAGER DESK\n");
    printf("-----------------------------------------------\n");
    while (1)
    {
        printf("\n        1. View total orders\n");
        printf("         2. Add new food items in the menu
card\n");
        printf("         3. Delete food items from the menu
card\n");
        printf("         4. Display menu card\n");
        printf("         5. Back To Main Menu \n\n");
        printf("         Enter Your Choice : ");

        int opt;
        scanf("%d", &opt);

        if (opt == 5)
            break;

        switch (opt)
        {
        case 1:
            printf("\n====================== Total Orders
======================\n");

            displayOrder(h_s);
            break;
        case 2:
```

```
            printf("\n          Enter Sr.No of the food item: ");
            int num, flag = 0;
            char name[50];
            float price;
            scanf("%d", &num);

            FM *demo = h_a;

            while (demo != NULL)
            {
                if (demo->d == num)
                {
                    printf("\n         This food item already
exists in menu card!!\n\n");
                    flag = 1;
                    break;
                }
                demo = demo->nx;
            }

            if (flag == 1)
                break;

            printf("          Enter name of food item: ");
            scanf("%s", name);
            printf("          Enter price: ");
            scanf("%f", &price);
            h_a = createmenu(h_a, num, name, price);

            FILE *f_p;
            f_p = fopen("Menu.txt", "a");
            if (f_p == NULL)
                printf("\nError");
            else
                fprintf(f_p, "%d \t%s \t%f\n", num, name, price);
            printf("\n          New food item has been added to
the menu card!!\n\n");
            fclose(f_p);
            break;
        case 3:
            if (deleteMenu())
            {
```

```
                printf("\n======================== Updated menu
card ======================\n");
                displayMenu(h_a);
            }
            else
                printf("\n          Food item with given Sr.No
doesn't available!!\n\n");

            break;
        case 4:
            printf("\n======================== Order menu
========================\n");
            displayMenu(h_a);
            break;

        default:
            printf("\n          Wrong Input !! PLease choose
valid option\n");
            break;
        }
    }
}

void customer()
{
    int flag = 0, j = 1;
    char ch, l;
    printf("\n-----------------------------------------------\n");
    printf("              CUSTOMER DESK\n");
    printf("-----------------------------------------------\n");
    int opt;
    do
    {
        printf("\n          1. Give order\n");
        printf("          2. View ordered food items\n");
        printf("          3. Cancel a food item from order\n");
        printf("          4. Display final bill\n");
        printf("          5. Back To Main Menu \n\n");
        printf("          Enter Your Choice : ");
        scanf("%d", &opt);

        if (opt == 5)
```

```
            break;

        switch (opt)
        {
        case 1:
            do
            {
                displayMenu(h_a);
                printf("\n          Enter Sr.No of food item
which you want to order: ");
                int n;
                scanf("%d", &n);
                printf("          Enter qt: ");
                int qt;
                scanf("%d", &qt);
                h_c = createorder(h_c, n, qt);
                printf("          Do you want to continue your
order (y/n) : ");
                scanf(" %c", &l);
            } while (l != 'n');
            break;
        case 2:
            printf("\n========================== Your Order
==========================\n");
            displayOrder(h_c);
            break;
        case 3:
            if (deleteOrder())
            {
                printf("\n====================== Your Updated
order ========================\n");
                displayOrder(h_c);
            }
            else
                printf("\n          Food item with given Sr.No
doesn't available!!\n");
            break;
        case 4:
            calculatetotsales();
            printf("\n========================== Final Bill
==============================\n");
            displaybill();
```

```
            h_c = deleteList(h_c);
            printf("\n          Press any key to return to main
menu:");
            fflush(stdin);
            ch = fgetc(stdin);
            flag = 1;
            break;
        case 5:
            break;
        default:
            printf("\n          Wrong Input !! PLease choose
valid option\n");
            break;
        }
        if (flag == 1)
            break;
    } while (opt != 5);
}
```

Output:

```
=====================================================
          WELCOME TO TAJ HOTEL
=====================================================


         1. MANAGER DESK
         2. CUSTOMER DESK
         3. Exit

         Enter Your Choice : 2

-----------------------------------------------
             CUSTOMER DESK
-----------------------------------------------

         1. Give order
         2. View ordered food items
         3. Cancel a food item from order
         4. Display final bill
         5. Back To Main Menu

         Enter Your Choice : 1

       Sr.no.      FoodName           Price
     -----------------------------------------------------
         101   Cheese-Pizza            330.00
         102   Special-Pav-Bhaji       210.00
         103   Manchurian-Noodles      400.00
         104   Cheese-Burger           250.00
         105   Cherry-Pineapple        350.00

         Enter Sr.No of food item which you want to order: 101
         Enter qt: 3
         Do you want to continue your order (y/n) : y

       Sr.no.      FoodName           Price
     -----------------------------------------------------
         101   Cheese-Pizza            330.00
         102   Special-Pav-Bhaji       210.00
         103   Manchurian-Noodles      400.00
         104   Cheese-Burger           250.00
         105   Cherry-Pineapple        350.00

         Enter Sr.No of food item which you want to order: 102
         Enter qt: 4
         Do you want to continue your order (y/n) : y
```

```
        Sr.no.      FoodName           Price
     ---------------------------------------------------
          101   Cheese-Pizza            330.00
          102   Special-Pav-Bhaji       210.00
          103   Manchurian-Noodles      400.00
          104   Cheese-Burger           250.00
          105   Cherry-Pineapple        350.00

          Enter Sr.No of food item which you want to order: 104
          Enter qt: 2
          Do you want to continue your order (y/n) : n

          1. Give order
          2. View ordered food items
          3. Cancel a food item from order
          4. Display final bill
          5. Back To Main Menu

          Enter Your Choice : 2

========================= Your Order ========================

        Sr.no.      FoodName        Quantity    Price
     ---------------------------------------------------
          101   Cheese-Pizza            3       990.00
          102   Special-Pav-Bhaji       4       840.00
          104   Cheese-Burger           2       500.00


          1. Give order
          2. View ordered food items
          3. Cancel a food item from order
          4. Display final bill
          5. Back To Main Menu

          Enter Your Choice : 3

          Enter Sr.No. of the food item which you want to delete: 101

          Enter qt want to delete: 2

====================== Your Updated order ====================

        Sr.no.      FoodName        Quantity    Price
     ---------------------------------------------------
          101   Cheese-Pizza            1       495.00
          102   Special-Pav-Bhaji       4       840.00
          104   Cheese-Burger           2       500.00
```

```
          1. Give order
          2. View ordered food items
          3. Cancel a food item from order
          4. Display final bill
          5. Back To Main Menu

          Enter Your Choice : 4

========================== Final Bill ================================

        Sr.no.      FoodName        Quantity    Price
      ------------------------------------------------------
          101   Cheese-Pizza            1       495.00
          102   Special-Pav-Bhaji       4       840.00
          104   Cheese-Burger           2       500.00

----------------------------------------------------------------------

          Price :-              1835.00
          Additional GST :-     91.75
          ------------------------------
           Total Price :-       1926.75
          ------------------------------

          Press any key to return to main menu:5

=========================================================
           WELCOME TO TAJ HOTEL
=========================================================


          1. MANAGER DESK
          2. CUSTOMER DESK
          3. Exit

          Enter Your Choice : 1

---------------------------------------------
          MANAGER DESK
---------------------------------------------

        1. View total orders
          2. Add new food items in the menu card
          3. Delete food items from the menu card
          4. Display menu card
          5. Back To Main Menu
```

```
          Enter Your Choice : 2

          Enter Sr.No of the food item: 106
          Enter name of food item: Manchurian_Noodels
          Enter price: 290

          New food item has been added to the menu card!!


        1. View total orders
          2. Add new food items in the menu card
          3. Delete food items from the menu card
          4. Display menu card
          5. Back To Main Menu

          Enter Your Choice : 4

 ======================= Order menu =======================

        Sr.no.      FoodName           Price
     -----------------------------------------------------
          101   Cheese-Pizza            330.00
          102   Special-Pav-Bhaji       210.00
          103   Manchurian-Noodles      400.00
          104   Cheese-Burger           250.00
          105   Cherry-Pineapple        350.00
          106   Manchurian_Noodels      290.00

        1. View total orders
          2. Add new food items in the menu card
          3. Delete food items from the menu card
          4. Display menu card
          5. Back To Main Menu

          Enter Your Choice : 3

          Enter Sr.No. of the food item which you want to delete: 102

 ======================= Updated menu card =======================

        Sr.no.      FoodName           Price
     -----------------------------------------------------
          101   Cheese-Pizza            330.00
          103   Manchurian-Noodles      400.00
          104   Cheese-Burger           250.00
          105   Cherry-Pineapple        350.00
          106   Manchurian_Noodels      290.00
```

```
        1. View total orders
          2. Add new food items in the menu card
          3. Delete food items from the menu card
          4. Display menu card
          5. Back To Main Menu

          Enter Your Choice : 5

 ========================================================
           WELCOME TO TAJ HOTEL
 ========================================================


          1. MANAGER DESK
          2. CUSTOMER DESK
          3. Exit

          Enter Your Choice : 3
          *THANK YOU FOR VISITING*
```