



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: 106 Section: DA

Lab Project Name: __ Restaurant Management Software _____

Student Details

Name	ID
Nazmul Hasan	213902003

Submission Date : _ 11/9/22 _____

Course Teacher's Name : ____ Farhana Akter Sunny _____

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

Lab Project Status

Marks:

Signature:

.....

Comments:	Date: 11/09/22

Table of contents

Chapter 1: Introduction3

1.1 Introduction3

1.2 Designs Goals /Objectives.....3

Chapter 2: Design/Development/Implementation of the project4

2.1 Interface.....4

2.2 Algorithm4

2.3 Source code.....6

Chapter 3: Performance Evaluation.....27

3.1 Result and discussion.....27

Chapter 4: Performance Evaluation.....36

4.1 Introduction.....36

4.2 Practical implementation.....36

4.3 Scope of future work.....36

Chapter 1

1.1 Introduction

The name of my project is Restaurant management software. As you know the population is increasing rapidly as population increasing their necessity is also growing among all the necessary food is one of them .among food thing a robust system that plays in the food chain is a restaurant. That's why I make software mainly for this restaurant business. I made this software primarily to save time for people and use more minor workers. So, the computerization of the restaurant management software will not only improve the efficiency of the restaurant workers and help the customers get all things at the right price.

1.2 Goal of my project:

- Our main aim of the project is to get the correct information about particular things that customers Order.
- The main focus of this project is to lessen human efforts.
- To provide a user-friendly environment where users can get serviced better.

Chapter 2

2.1 restaurant management software design:

2.1.1 Interface:

```
*****  
WELCOME TO RESTAURANT MANAGEMENT SYSTEM  
*****  
  
1. ADMIN SECTION-->  
2. CUSTOMER SECTION-->  
3. Exit-->  
  
Enter Your Choice --->
```

2.2 Algorithm:

Step 1: Start.

Step 2: create a structure data type and create pointer data type for head and tail of a linked list.

Step 3: Create admin menu function.

Step 3.1: Create option for admin function

Step 3.1.1: Create view total sales function to create this function we use a linked list and for that we need to initialize a new node and give it size with dynamic memory allocation malloc and set the next and previous value null .and use if and else and loop to linked the node in he linked list .and use printf function to show to show to sales result.

Step 3.1.2: Create add menu function same approach as create total sales function but in end here we will not print we will just linked a node in the list.

Step 3.1.3: Create display function Again same approach as creating total sales function but here we will use a another pointer node with a loop which will run until the position is given and when the position is found it will over right the node .

Step 3.1.4: Create product arrival function To create this function we will need to create a queue and that we will need to initialize an array with two variable front and rare and initialize their value with -1 after that we will create a input function where we will use if and else condition to add item in the queue.

Step 4: Create customer function Here create 5 function

Step 4.1: Place your order function to create this function we will use linked as the previous function and use printf and scanf function to take and print the order.

Step 4.2: View your order items Also use linked list use if and else condition to select the order and use printf function to print the order.

Step 4.3: Delete an item from the order function use linked list use a while loop and a new node and find the positon where you want to delete and over right the node with the given node and size --.

Step 4.4: Display function and finally booked table function

And create delete and display function which will delete and display all the element inside queue.

Step 4.5: Display final bill create another linked list as previous linked list .create another pointer variable and use a while loop until node find null position and print all the node.

Step 4.6: Create a Booked table function here and call the queue insert delete and display function.

Step 5: Create an exit function with if else and break statement.

Step 6: Use a switch statement and call all the function in the switch case.

Step 7: End.

2.3 Source code:

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

// bubble sort:
struct restaurant
{
    int serial;
    char name[100];
};
struct restaurant st[100];
```

```
void addthing(struct restaurant *st, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("\t\t\t\t\tEnter Serial: ");
        int serial;
        scanf("%d", &serial);
        st[i].serial = serial;

        printf("\t\t\t\t\tEnter Name: ");
        char name[100];
        scanf("%s", name);
        strcpy(st[i].name, name);
    }
}

void Bshort(struct restaurant *st, int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        int flag = 0;
        for (int j = 0; j < n - i - 1; j++)
        {
            if (st[j].serial > st[j + 1].serial)
            {
                struct restaurant temp = st[j];
                st[j] = st[j + 1];
                st[j + 1] = temp;
                flag = 1;
            }
        }

        if (flag == 0)
        {
            break;
        }
    }
}

void printlist(struct restaurant *st, int n)
```



```

void insert(int x)
{
    if (rare == N - 1)
    {
        printf("\n\t\t\t\t\t\t\t\t\t\tSorry No Table Is Free For
Booked    :  \n");
    }
    if (rare == -1 && front == -1)
    {
        front = rare = 0;
        queue[rare] = x;
    }
    else
    {
        rare++;
        queue[rare] = x;
    }
}

void Delete()
{
    if (rare == -1 && front == -1)
    {
        printf("\n\t\t\t\t\t\t\t\t\t\tFell Free To Booked Any Table
:  \n");
    }
    else
    {
        printf("\n\t\t\t\t\t\t\t\t\t\tCancled Table No: %d \n",
queue[front]);
        front++;
    }
}

void display()
{
    if (rare == -1 && front == -1)

```

```
        printf("\n\t\t\t\t\t\t\tFell Free To Booked Any Table\n");  
    :   \n");  
  
    else  
    {  
        for (int i = front; i < rare + 1; i++)  
        {  
            printf("\n\t\t\t\t\t\t\tBooked Table No: %d \n",  
queue[i]);  
        }  
    }  
}  
  
struct node  
{  
    char foodname[50];  
    int quantity;  
    float price;  
    int data;  
    struct node *prev;  
    struct node *next;  
};  
  
struct node *headc = NULL, *newnode, *tailc = NULL;  
struct node *heada = NULL, *taila = NULL;  
struct node *head_s;  
  
void adminmenu()  
{  
    printf("\n\t\t\t\t\t\t\t1. View total sales\n");  
    printf("\t\t\t\t\t\t\t2. Add new items in the order menu\n");  
    printf("\t\t\t\t\t\t\t3. Delete items from the order menu\n");  
    printf("\t\t\t\t\t\t\t4. Display order menu\n");  
    printf("\t\t\t\t\t\t\t5. Product Arrival \n");  
    printf("\t\t\t\t\t\t\t6. Back To Main Menu \n\n");  
    printf("\t\t\t\t\t\t\tEnter Your Choice --->");
```

}

```
void customermenu()
```

{

[illegible]

}

```
struct node *createadmin(struct node *head, int data, char
foodname[25], float price)
```

{

```
newnode = (struct node *)malloc(sizeof(struct node));
```

```
newnode->data = data;
```

```
newnode->price = price;
```

```
newnode->quantity = 0;
```

```
strcpy(newnode->foodname, foodname);
```

```
newnode->next = NULL;
```

```
newnode->prev = NULL;
```

```
struct node *temp = head;
```

```
if (temp == NULL)
```

```
heada = taila = newnode;
```

```
else
```

 $\{$

```
while (temp->next != NULL)
```

```
temp = temp->next;
```

```
temp->next = newnode;
```

```
newnode->prev = taila;
```

```
taila = newnode;
```

}

```

        return heada;
    }

    struct node *createcustomer(struct node *head, int data, int
quantity)
{
    newnode = (struct node *)malloc(sizeof(struct node));

    struct node *temp1 = heada;
    int flag = 0;
    while (temp1 != NULL)
    {
        if (temp1->data == data)
        {
            flag = 1;
            break;
        }
        temp1 = temp1->next;
    }

    if (flag == 1)
    {
        newnode->data = data;
        newnode->price = quantity * (temp1->price);
        newnode->quantity = quantity;
        strcpy(newnode->foodname, temp1->foodname);
        newnode->next = NULL;
        newnode->prev = NULL;

        struct node *temp = head;

        if (temp == NULL)
            headc = tailc = newnode;
        else
        {
            while (temp->next != NULL)
                temp = temp->next;

```

```
temp->next = newnode;
newnode->prev = tailc;
tailc = newnode;
}
}
else
{
    printf("\n\t\t\t\t\t\t\tThis item is not present in the menu!\n");
}
return headc;
}

void displayList(struct node *head)
{
    struct node *temp1 = head;
    if (temp1 == NULL)
    {
        printf("\n\t\t\t\t\t\t\tList is empty!!\n\n");
    }
    else
    {
        printf("\n");
        while (temp1 != NULL)
        {
            if (temp1->quantity == 0)
                printf("\t\t\t\t\t\t\t%d\t%s\t%0.2f\n", temp1-
>data, temp1->foodname, temp1->price);
            else
            {
                printf("\t\t\t\t\t\t\t%d\t%s\t%d\t%0.2f\n",
temp1->data, temp1->foodname, temp1->quantity, temp1->price);
            }

            temp1 = temp1->next;
        }
        printf("\n");
    }
}
```

```

    }
}

struct node *totalsales(int data, int quantity)
{
    newnode = (struct node *)malloc(sizeof(struct node));
    int flag = 0;

    struct node *temp1 = heada;
    while (temp1->data != data)
    {
        temp1 = temp1->next;
    }

    newnode->data = data;
    newnode->price = quantity * (temp1->price);
    newnode->quantity = quantity;
    strcpy(newnode->foodname, temp1->foodname);
    newnode->next = NULL;
    newnode->prev = NULL;

    struct node *temp = head_s;

    if (temp == NULL)
        head_s = newnode;
    else
    {
        while (temp->next != NULL)
        {
            if (temp->data == data)
            {
                flag = 1;
                break;
            }
            temp = temp->next;
        }

        if (flag == 1)

```

```

        {
            temp->quantity += newnode->quantity;
            temp->price += newnode->price;
        }
        else
        {
            temp->next = newnode;
        }
    }

    return head_s;
}

void calculatetotsales()
{
    struct node *temp = headc;
    while (temp != NULL)
    {
        head_s = totalsales(temp->data, temp->quantity);
        temp = temp->next;
    }
}

struct node *delete (int data, struct node *head, struct node
*tail)
{
    if (head == NULL)
    {
        printf("\n\t\t\t\t\t\t\tList is empty\n");
    }
    else
    {
        struct node *temp;
        if (data == head->data)
        {
            temp = head;
            head = head->next;
            if (head != NULL)

```

```

        head->prev = NULL;
        free(temp);
    }
    else if (data == tail->data)
    {
        temp = tail;
        tail = tail->prev;
        tail->next = NULL;
        free(temp);
    }
    else
    {
        temp = head;
        while (data != temp->data)
        {
            temp = temp->next;
        }
        (temp->prev)->next = temp->next;
        (temp->next)->prev = temp->prev;
        free(temp);
    }
}
return head;
}

int deleteadmin()
{
    printf("\n\t\t\t\t\tEnter serial no. of the food item which
is to be deleted: ");
    int num;
    scanf("%d", &num);

    struct node *temp = heada;
    while (temp != NULL)
    {
        if (temp->data == num)
        {

```



```

        heada = delete (num, heada, taila);
        return 1;
    }
    temp = temp->next;
}

return 0;
}

int deletecustomer()
{
    printf("\n\t\t\t\t\tEnter serial no. of the food item which
is to be deleted: ");
    int num;
    scanf("%d", &num);

    struct node *temp = headc;
    while (temp != NULL)
    {
        if (temp->data == num)
        {
            headc = delete (num, headc, tailc);
            return 1;
        }
        temp = temp->next;
    }

    return 0;
}

void displaybill()
{
    displayList(headc);
    struct node *temp = headc;
    float total_price = 0;
    while (temp != NULL)
    {
        total_price += temp->price;
    }
}

```

```
temp = temp->next;
}

printf("\t\t\t\t\tTotal price: %0.02f\n", total_price);
}

struct node *deleteList(struct node *head)
{
    if (head == NULL)
    {
        return NULL;
    }
    else
    {
        struct node *temp = head;
        while (temp->next != 0)
        {
            temp = temp->next;
            free(temp->prev);
        }
        free(temp);
        head = NULL;
    }

    return head;
}

void admin()
{
    system("cls");
    printf("\n\t\t\t\t\t-----\n");
    printf("\t\t\t\t\tADMIN SECTION\n");
    printf("\t\t\t\t\t-----\n");
    while (1)
    {
        adminmenu();
```

```

int opt, n1;
scanf("%d", &opt);

if (opt == 6)
    break;

switch (opt)
{
case 1:

    displayList(head_s);
    break;
case 2:

    printf("\n\t\t\t\t\tEnter serial no. of the food
item: ");

    int num, flag = 0;
    char name[50];
    float price;
    scanf("%d", &num);

    struct node *temp = heada;

    while (temp != NULL)
    {
        if (temp->data == num)
        {
            printf("\n\t\t\t\t\tFood item with given
serial number already exists!!\n\n");
            flag = 1;
            break;
        }
        temp = temp->next;
    }

    if (flag == 1)
        break;

```



```

void mainmenu()
{
    system("cls");
    printf("\n
*****
*****\n");
    printf("
WELCOME TO RESTAURANT MANAGEMENT SYSTEM\n");
    printf("
*****
*****\n\n\n");
    printf("\t\t\t\t\t1. ADMIN SECTION--> \n");
    printf("\t\t\t\t\t2. CUSTOMER SECTION--> \n");
    printf("\t\t\t\t\t3. Exit--> \n\n");
    printf("\t\t\t\t\tEnter Your Choice --->");
}

int main()
{
    heada = createadmin(heada, 1, "Hot and Sour Soup", 100);
    heada = createadmin(heada, 2, "Manchow Soup", 200);
    heada = createadmin(heada, 3, "Manchurian Noodles", 150);
    heada = createadmin(heada, 4, "Fried Rice", 180);
    heada = createadmin(heada, 5, "Hakka Noodles", 80);

    while (1)
    {
        system("cls");
        mainmenu();
        int choice;
        scanf("%d", &choice);

        if (choice == 3)
        {
            printf("\n\n\t\t\t\t\t*****Thank
you!!*****\n");
            break;
        }
    }
}

```

```
switch (choice)
{
case 1:
    admin();
    break;
case 2:
    customer();
    break;
case 3:
    break;

default:
    printf("\n\t\t\t\t\tWrong Input !! Please choose
valid option\n");
    break;
}
}
}
```

Chapter3

Performance evaluation:

Front page of the project /main menu:

❖ Here you will see three options :

1. Admin section 2. Customer section and finally 3.Exit section.

```
*****
WELCOME TO RESTAURANT MANAGEMENT SYSTEM
*****

1. ADMIN SECTION-->
2. CUSTOMER SECTION-->
3. Exit-->

Enter Your Choice --->
```

Admin Section:

❖ When you select one as the admin section, you will see another page where you will give another six option:

1. View total sales.
2. Add new items in the order menu.
3. Delete items from the order menu.
4. Display order menu.
5. Product Arrival.

6. Back to Main Menu and finally your choice option.

```
-----  
ADMIN SECTION  
-----
```

1. View total sales
2. Add new items in the order menu
3. Delete items from the order menu
4. Display order menu
5. Product Arrival
6. Back To Main Menu

```
Enter Your Choice --->[ ]
```

Add new items in the order menu option:

- ❖ This option is mainly for the restaurant owner or workers who will choose to add a new item to their order menu.

```

-----
                        ADMIN SECTION
-----

1. View total sales
2. Add new items in the order menu
3. Delete items from the order menu
4. Display order menu
5. Product Arrival
6. Back To Main Menu

Enter Your Choice --->2

Enter serial no. of the food item: 6
Enter food item name: Biryani
Enter price: 120

New food item added to the list!!

```

Delete item in the order menu option:

- ❖ This option is also for the restaurant owner or workers who will choose to delete an item from their order menu.

```

Enter Your Choice --->3

Enter serial no. of the food item which is to be deleted: 5

### Updated list of food items menu ###

1      Hot and Sour Soup      100.00
2      Manchow Soup          200.00
3      Manchurian Noodles     150.00
4      Fried Rice             180.00

```

Display Order menu option:

- ❖ This option will show their order menu.

```

Enter Your Choice --->4

-----
Order Menu
-----

1      Hot and Sour Soup      100.00
2      Manchow Soup          200.00
3      Manchurian Noodles     150.00
4      Fried Rice             180.00

```

Product arrival option:

- ❖ This option will generate another three options 1. Input 2. Delete and finally insert. This will help the owner to keep a record of their daily product and help them input and delete the product from their arrival list.

```

-----
Product Arrival
-----

1.Input product
2.Delete product
3.Insert product
4.Back To Main Menu

Enter Your Choice --->

```

Input product option:

```

-----
                        Product Arrival
-----

1.Input product
2.Delete product
3.Insert product
4.Back To Main Menu

Enter Your Choice --->1
Enter how many product arrive : 3
Enter Serial: 12
Enter Name: Elish
Enter Serial: 11
Enter Name: Cowmeat
Enter Serial: 10
Enter Name: Milk

-----
                        PRINTING LIST
-----

Serial number: 10, Name: Milk
Serial number: 11, Name: Cowmeat
Serial number: 12, Name: Elish
-----

```

Delete product option :

```

Enter Your Choice --->2
Enter how many product arrive : 3
Enter which position product you want to remove

deleted product is 11,∞

-----
                        PRINTING LIST
-----

Serial number: 10, Name: Milk
Serial number: 12, Name: Elish

-----
                        Product Arrival
-----

```

Customer section:

- ❖ This option is basically for the customer here, the customer will get 6 options such as:

1. Place your order

2. View your ordered items
3. Delete an item from an order
4. Display final bill
5. Booked Table
6. Back to the main menu

And choice option:

```

-----
                        CUSTOMER SECTION
-----

1. Place your order
2. View your ordered items
3. Delete an item from order
4. Display final bill
5. Booked Table

6. Back To Main Menu

Enter Your Choice --->

```

Over view output of all 6 option:

View your order items :

```

Enter Your Choice --->1

1      Hot and Sour Soup      100.00
2      Manchow Soup          200.00
3      Manchurian Noodles     150.00
4      Fried Rice             180.00
5      Hakka Noodles          80.00

Enter number corresponding to the item you want to order: 1
Enter quantity: 3

```

List of your order items:


```
Enter Your Choice --->2
```

```
### List of ordered items ###
```

```
1      Hot and Sour Soup      3      300.00
```

Delete an item from your order:

```
Enter Your Choice --->3
```

```
Enter serial no. of the food item which is to be deleted: 1
```

```
### Updated list of your ordered food items ###
```

```
1      Hot and Sour Soup      2      200.00
3      Manchurian Noodles     2      300.00
```

Display final bill:

```
Enter Your Choice --->4
```

```
### Final Bill ###
```

```
1      Hot and Sour Soup      2      200.00
3      Manchurian Noodles     2      300.00
```

```
Total price: 500.00
```

Booked Table:

The booked table option is helpful sometimes; customers come to the restaurant but cannot get a suitable table to sit at, or the restaurant is complete. For this reason, they must go back home to avoid this type of problem. I created this option so they could quickly book a table and did not need to wait or go out of the restaurant to eat. Booked table option will also show the customer another 4 options, such as

1. Enter your desire table No;
2. Canceled your Table No:
3. Watched Booked Table:
4. Back To main menu:

And choice option.

Here is the overview picture of all the options:

Enter your desire table No:

```
TABLE BOOKING
-----
1.Enter your desire table No :
2.Canceled Your Table no  :
3.Watch Bokked Table    :
4.Back To mainmenu      :
   Enter Your Choice --->1

Table No   : 14
```

Canceled your Table No:

```

Enter Your Choice --->2

Cancled Table No: 14

-----
TABLE BOOKING
-----

1.Enter your desire table No :
2.Cancled Your Table no :
3.Watch Bokked Table :
4.Back To mainmenu :
Enter Your Choice --->14

-----
TABLE BOOKING
-----

1.Enter your desire table No :
2.Cancled Your Table no :
3.Watch Bokked Table :
4.Back To mainmenu :
Enter Your Choice --->3

Booked Table No: 23

```

Watched Booked Table:

```

-----
TABLE BOOKING
-----

1.Enter your desire table No :
2.Cancled Your Table no :
3.Watch Bokked Table :
4.Back To mainmenu :
Enter Your Choice --->3

Booked Table No: 14

Booked Table No: 23

```

Exit option:

When the customer is done their ordering work they can quickly get out from the system by choosing the exit option, and here is overview output of the exit function:

```
Enter Your Choice --->3
```

```
*****Thank you!!*****
```

```
*****Nazmul Hasan (ID:213902003)*****
```

```
*****Department : CSE *****
```

```
*****Green University Of Bangladesh*****
```

Chapter 4

Conclusion:

4.1 Introduction:

After completing the project, we are sure that this management software will help restaurant workers and customers order their desired food in time and without any Problem. It will reduce human errors and increase efficiency.

4.2 Practical implementation:

I already use this software in my friend's restaurant, which was not bad. The menu that I used was his restaurant menu, and he was pleased to use it.

4.3 Scope of future work:

Restaurant management software will be the most helpful software for all restaurant owners in the future. We can also use lots of new features like a food delivery system, customer record system, and we can also use it to save the record and collect information of the works .so I think this software reduces time and will serve as many ways it can.

