“Food order management system using SJF”

**Submitted by :**

**Amareswararao Bollimuntha**

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **S. No.** | **Topic** | **Page No.** |
| **1** | Project Summary | **5** |
| **2** | Problem Statement | **5** |
| **3** | Hardware and Software Requirements | **6** |
| **4** | Algorithm | **6** |
| **5** | Implementation | **7** |
| **6** | Results | **18** |
| **7** | Conclusion | **20** |
| **8** | References | **20** |

**1.Project Summary: -**

Food Ordering System is an application which will help restaurant to optimized and control over their restaurants. For the waiters, it is making life easier because they do not have to go kitchen and give the orders to chef easily.

This application helps the restaurants to do all functionalities mare accurately and faster way. Food Ordering System reduces manual works and improves efficiency of restaurant.

This application is helping Food Ordering s to maintain the stock and

there are many more functionalities, like.

* To store orders
* Control orders and services.
* Billings
* Control the order of serving.
* Control multiple branches.

The main goal is to maintain the restaurant's functions in an effective and accurate manner and it is reducing the use of manual entries. This software helps food orders to reach the customers according to the time taken by order given. It is keeping a proper manner for serving the food to the customers and give their bill along the way.

**2.Problem Statement: -**

Many restaurants are storing all of their data in manual way. They have huge number of customers daily. So, because large number of customers, they need the help of some features so they can maintain and stores the records accurately.

They need full-fledged software to maintain their transactions, orders, and regular update on orders.

This application will be helpful for the waiters to take the orders and serve them at order of the time taken to make ordered items and serve to the customers accordingly.

**3.Hardware and software requirements: -**

**Software requirements: -**

The software interface is the operating system, and application programming interface used for the development of the software.

Operating System: Windows XP or higher / Mac OS X 10.5.8 or later / Linux.

Language used: c language.

Compiler and editor used: visual studio code.

**Hardware requirements: -**

Processor : Intel I5 and above

RAM : 4 GB and above

Hard Disk : 50 GB and above

**4.Algorithm: -**

1.Take input of no of orders.

2.Create a while loop.

3.Run it up to no of orders given as input.

4.Then display menu each time order is taken.

5.Take input of the items, flavors (if available) and quantity required.

6.Ask whether any other items are required.

7.If yes repeat step 3 to step 6.

8.Else print their bill.

9.After the end of taking the orders ,all the orders are sorted according to the SJFAlgorithm.

10.Print the list of orders.

**5.Implementation: -**

**Source code-**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int count=0,qty1=0,time=0;

void menu();

struct food

{

char items[300];

int qty[20];

int time;

int cost[20];

}order[20];

char piz1[]="Basic pizza",piz2[]="Chicken Bar BQ pizza",piz3[]="Chicken Fazita pizza",piz4[]="Creamy Max pizza";

char drinks1[]="Mountain Dew.", drinks2[]="Coca Cola.", drinks3[]="Enegry+.";

char burger\_1[]="Zinger Burger.",burger\_2[]="Chicken Burger.",burger\_3[]="Cheesy Chicken Burger.";

char sandwich\_1[]="Club Sandwich.", sandwich\_2[]="Chicken Crispy Sandwich.", sandwich\_3[]="Extream Veg Sandwich.";

char fried1[]="Chicken Fried.", fried2[]="Prawn Fried.", fried3[]="Fish Fried.";

void sjf()

{

int bt[count],p[count],wt[count],tat[count],i=0,j=0,total=0,pos=0,temp=0;

float avg\_wt,avg\_tat;

for(i=1;i<=count;i++)

p[i]=i;

for(i=1;i<=count;i++)

bt[i]=order[i].time;

for(i=1;i<=count;i++)

{

pos=i;

for(j=i+1;j<=count;j++)

{

if(bt[j]<bt[pos])

pos=j;

}

temp=bt[i];

bt[i]=bt[pos];

bt[pos]=temp;

temp=p[i];

p[i]=p[pos];

p[pos]=temp;

}

wt[1]=1;

for(i=2;i<=count;i++)

{

wt[i]=0;

for(j=0;j<i;j++)

wt[i]+=bt[j];

total+=wt[i];

}

avg\_wt=(float)total/count;

total=0;

printf("\nORDER\t Burst Time \tWaiting Time\tTurnaround Time");

for(i=1;i<=count;i++)

{

tat[i]=bt[i]+wt[i];

total+=tat[i];

printf("\n\norder%d\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);

}

avg\_tat=(float)total/count;

printf("\n\nAverage Waiting Time=%.0f",avg\_wt);

printf("\nAverage Turnaround Time=%.0f\n",avg\_tat);

}

void burger()

{

int pizzaoption,itime=0,qty=0,again=0;

reenter:

system("cls");

printf("\n1)%s\n",burger\_1);

printf("2)%s\n",burger\_2);

printf("3)%s\n",burger\_3);

printf("\nPlease Enter which Burger would you like to have?:");

scanf("%d",&pizzaoption);

if(pizzaoption>=1 && pizzaoption<=3)

{

if(pizzaoption==1)

strcat(order[count].items,burger\_1);

else if(pizzaoption==2)

strcat(order[count].items,burger\_2);

else

strcat(order[count].items,burger\_3);

printf("\nPlease Enter Quantity: ");

scanf("%d",&qty);

itime=5;

if(qty>1)

{

for(int t=2;t<=qty;t++)

itime+=5;

}

int price=0;

switch(pizzaoption)

{

case 1: price = 50.00\*qty;

break;

case 2: price = 100.00\*qty;

break;

case 3: price = 150.00\*qty;

break;

}

qty1++;

order[count].qty[qty1]=qty;

order[count].cost[qty1]=price;

}

else

{

printf("Enter the option correctly\n");

goto reenter;

}

order[count].time+=itime;

printf("\nWould you like to order anything? if yes press '1' else press '0':");

scanf("%d",&again);

if(again==1)

menu();

}

void drinks()

{

int pizzaoption,itime=0,qty=0,again=0;

reenter:

system("cls");

printf("\n1)%s\n",drinks1);

printf("2)%s\n",drinks2);

printf("3)%s\n",drinks3);

printf("\nPlease Enter which Drink would you like to have?:");

scanf("%d",&pizzaoption);

if(pizzaoption>=1 && pizzaoption<=3)

{

if(pizzaoption==1)

strcat(order[count].items,drinks1);

else if(pizzaoption==2)

strcat(order[count].items,drinks2);

else

strcat(order[count].items,drinks3);

printf("\nPlease Enter Quantity: ");

scanf("%d",&qty);

itime=1;

if(qty>1)

{

for(int t=2;t<=qty;t++)

itime+=1;

}

int price=0;

switch(pizzaoption)

{

case 1: price = 30.00\*qty;

break;

case 2: price = 30.00\*qty;

break;

case 3: price = 30.00\*qty;

break;

}

qty1++;

order[count].qty[qty1]=qty;

order[count].cost[qty1]=price;

}

else

{

printf("Enter the option correctly\n");

goto reenter;

}

order[count].time+=itime;

printf("\nWould you like to order anything? if yes press '1' else press '0':");

scanf("%d",&again);

if(again==1)

menu();

}

void sandwich()

{

int pizzaoption,itime=0,qty=0,again=0;

reenter:

system("cls");

printf("\n1)%s\n",sandwich\_1);

printf("2)%s\n",sandwich\_2);

printf("3)%s\n",sandwich\_3);

printf("\nPlease Enter which Sandwich would you like to have?:");

scanf("%d",&pizzaoption);

if(pizzaoption>=1 && pizzaoption<=3)

{

if(pizzaoption==1)

strcat(order[count].items,sandwich\_1);

else if(pizzaoption==2)

strcat(order[count].items,sandwich\_2);

else

strcat(order[count].items,sandwich\_3);

printf("\nPlease Enter Quantity: ");

scanf("%d",&qty);

itime=5;

if(qty>1)

{

for(int t=2;t<=qty;t++)

itime+=3;

}

int price =0;

switch(pizzaoption)

{

case 1: price = 70.00\*qty;

break;

case 2: price = 100.00\*qty;

break;

case 3: price = 150.00\*qty;

break;

}

qty1++;

order[count].qty[qty1]=qty;

order[count].cost[qty1]=price;

}

else

{

printf("Enter the option correctly\n");

goto reenter;

}

order[count].time+=itime;

printf("\nWould you like to order anything? if yes press '1' else press '0':");

scanf("%d",&again);

if(again==1)

menu();

}

void fried()

{

int pizzaoption,itime=0,qty=0,again=0;

reenter:

system("cls");

printf("\n1)%s\n",fried1);

printf("2)%s\n",fried2);

printf("3)%s\n",fried3);

printf("\nPlease Enter which Fry would you like to have?:");

scanf("%d",&pizzaoption);

if(pizzaoption>=1 && pizzaoption<=3)

{

if(pizzaoption==1)

strcat(order[count].items,fried1);

else if(pizzaoption==2)

strcat(order[count].items,fried2);

else

strcat(order[count].items,fried3);

printf("\nPlease Enter Quantity: ");

scanf("%d",&qty);

itime=5;

if(qty>1)

{

for(int t=2;t<=qty;t++)

itime+=5;

}

int price=0;

switch(pizzaoption)

{

case 1: price = 100.00\*qty;

break;

case 2: price = 130.00\*qty;

break;

case 3: price = 150.00\*qty;

break;

}

qty1++;

order[count].qty[qty1]=qty;

order[count].cost[qty1]=price;

}

else

{

printf("Enter the option correctly\n");

goto reenter;

}

order[count].time+=itime;

printf("\nWould you like to order anything? if yes press '1' else press '0':");

scanf("%d",&again);

if(again==1)

menu();

}

void pizza()

{

int pizzaoption,pizzaoption1,qty=0,itime=0,again=0;

reenter:

system("cls");

printf("\n1)%s\n",piz1);

printf("2)%s\n",piz2);

printf("3)%s\n",piz3);

printf("4)%s\n",piz4);

printf("\nPlease Enter which Pizza Flavor would you like to have?:");

scanf("%d",&pizzaoption);

if(pizzaoption>=1 && pizzaoption<=4)

{

if(pizzaoption==1)

strcat(order[count].items,piz1);

else if(pizzaoption==2)

strcat(order[count].items,piz2);

else if(pizzaoption==3)

strcat(order[count].items,piz3);

else

strcat(order[count].items,piz4);

reenter1:

printf("\n1) Small Rs:150.00\n2) Regular Rs:250.00\n3) Large Rs:500.00\n");

printf("\nChoose Size Please:");

scanf("%d",&pizzaoption1);

if(pizzaoption1==1)

strcat(order[count].items,"-(size)small.");

else if(pizzaoption1==2)

strcat(order[count].items, "-(size)regular.");

else

strcat(order[count].items, "-(size)large.");

if(pizzaoption1>=1 && pizzaoption1<=3)

{

printf("\nPlease Enter Quantity: ");

scanf("%d",&qty);

itime=7;

if(qty>1)

{

for(int t=2;t<=qty;t++)

itime+=2;

}

int price=0;

switch(pizzaoption1)

{

case 1: price = 150.00\*qty;

break;

case 2: price = 250.00\*qty;

break;

case 3: price = 500.00\*qty;

break;

}

qty1++;

order[count].qty[qty1]=qty;

order[count].cost[qty1]=price;

}

else

{

printf("Enter the option correctly\n");

goto reenter1;

}

}

else

{

printf("Enter the option correctly\n");

goto reenter;

}

order[count].time+=itime;

printf("\nWould you like to order anything? if yes press '1' else press '0':");

scanf("%d",&again);

if(again==1)

menu();

}

void menu()

{

int i=0;

printf("\n\t\t\t|----------------------------------------------------------------|\n");

printf("\t\t\t| Food Menu |\n");

printf("\t\t\t|----------------------------------------------------------------|\n\n");

printf("[Choice 1] Pizzas\n");

printf("[Choice 2] Burgers\n");

printf("[Choice 3] Sandwich\n");

printf("[Choice 4] Drinks\n");

printf("[Choice 5] Fried\n\n");

printf("\nPlease Enter your Choice: ");

scanf("%d",&i);

switch(i)

{

case 1:

pizza();

break;

case 2:

burger();

break;

case 3:

sandwich();

break;

case 4:

drinks();

break;

case 5:

fried();

break;

default :

printf("\n\t\t\tChoose correct Option!!!\n");

break;

}

}

void bill()

{

int len;

system("cls");

printf("\n\t\t\t|----------------------------------------------------------------|\n");

printf("\t\t\t| BILL |\n");

printf("\t\t\t|----------------------------------------------------------------|\n\n");

len=strlen(order[count].items);

int totalc=0;

int l=1;

printf("%d) ",l);

for(int i=0;i<len;i++)

{

if(order[count].items[i]=='.')

{

printf("\t\tqty:%d",order[count].qty[qty1]);

printf("\t\tCost:%d",order[count].cost[qty1]);

totalc+=order[count].cost[qty1];

printf("\n");

l++;

if(l<=qty1)

printf("%d) ",l);

}

else

printf("%c",order[count].items[i]);

}

printf("\n\t\t\t\t\t\t\t\t\t\tTOTAL:%d\n",totalc);

}

void noorder(int n)

{

system("cls");

printf("\n\t\t\t|----------------------------------------------------------------|\n");

printf("\t\t\t| Food - Order Management System |\n");

printf("\t\t\t|----------------------------------------------------------------|\n\n");

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

{

count++;

qty1=0;

menu();

bill();

}

sjf();

}

int main()

{

int n;

system("cls");

printf("\t\t\t|----------------------------------------------------------------|\n");

printf("\t\t\t| Food - Order Management System |\n");

printf("\t\t\t|----------------------------------------------------------------|\n\n");

printf("\t\tEnter number of orders(MAX 20):");

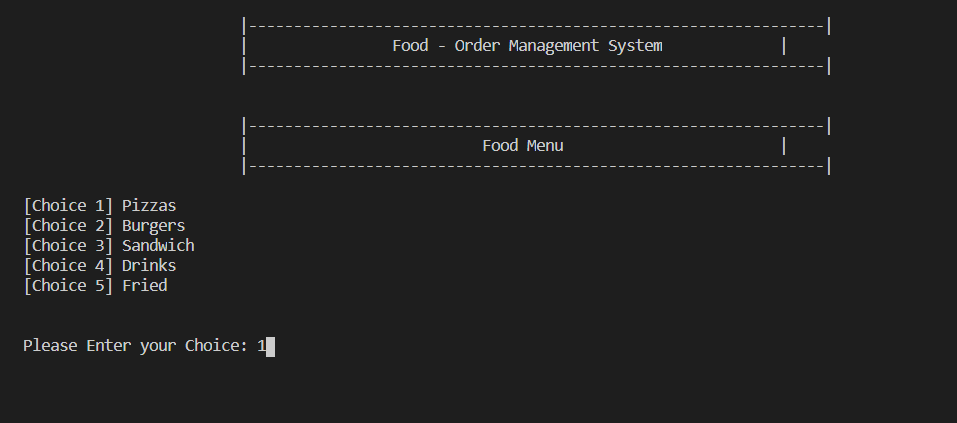
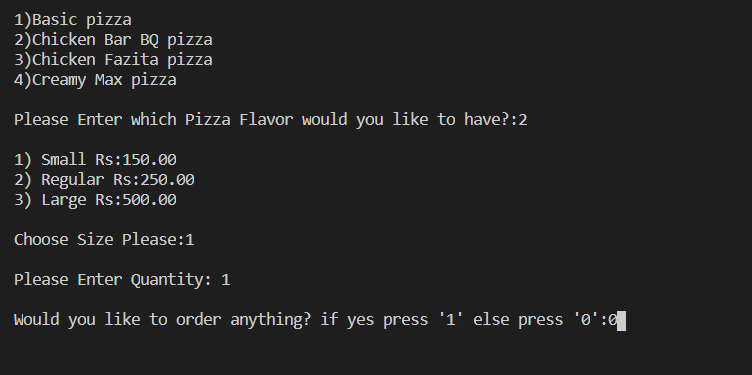
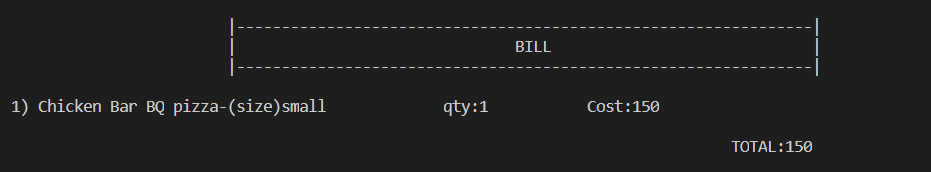
scanf("%d",&n);

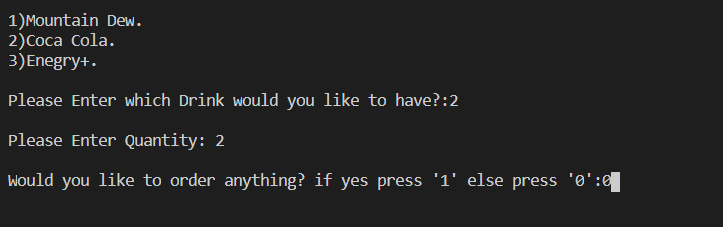
vnoorder(n);

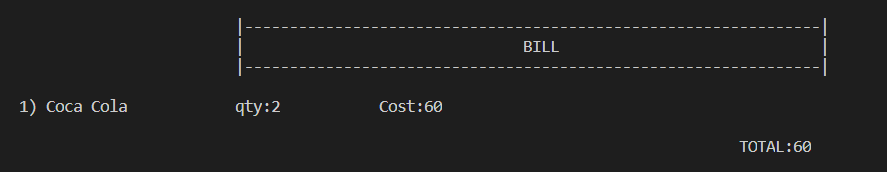
return 0;

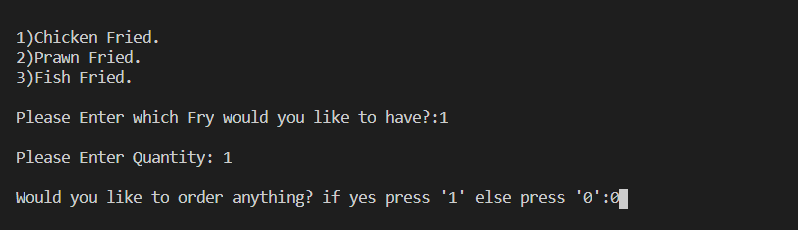
}

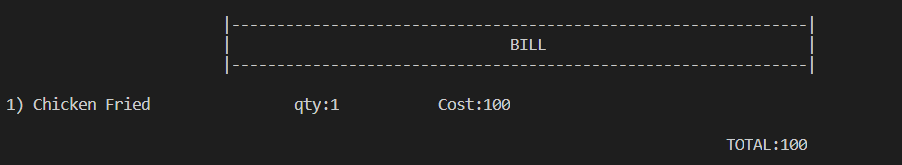
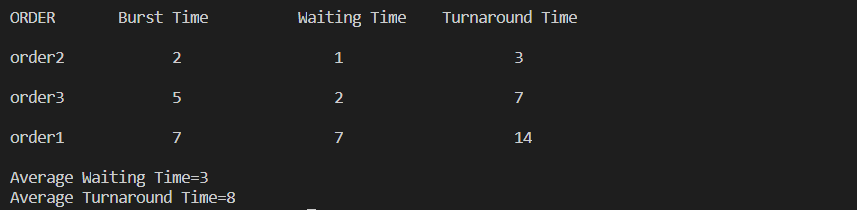
**6.Results(output): -**

**** **** 

****

****

****

****

**7.Conclusion: -**

All the orders are taken from the users and served to the customers according to time taken by their order items. There is no risk or high waiting time for the customers.

Full work is done systematically and computerized.

**8.References: -**

<https://www.slideshare.net/armanreza161/food-ordering-system-75284735>

[https://www.jotform.com/blog/wp-content/uploads/2020/07/food-order- management-01.png](https://www.jotform.com/blog/wp-content/uploads/2020/07/food-order-%20%20%20management-01.png)