A Report

on

FoodOnn

Restaurant discovery and billing software

For partial fulfillment of the requirement for the Mini project Laboratory

---

COMPUTER SCIENCE AND ENGINEERING
(2016-2020)
By

Mohammed Areebuddin (1604-16-733-046)

Faizan Mohiuddin (1604-16-733-038)

Mohammed Muzakkir Qadri(1604-16-733-045)

Department of Computer Science and Engineering
Muffakham Jah College of Engineering and Technology
(Affiliated to Osmania University)
Hyderabad

FoodOnn

A project report on

Restaurant discovery and billing software

COMPUTER SCIENCE AND ENGINEERING
(2016-2020)
By

Mohammed Areebuddin (1604-16-733-046)

Faizan Mohiuddin (1604-16-733-038)

Mohammed Muzakkir Qadri(1604-16-733-045)

Under the supervision of

Mr. Meer Arshad Ali
Project Supervisor

Mr.J.Srinivas
Project Coordinator

Department of Computer Science and Engineering
Muffakham Jah College of Engineering and Technology
(Affiliated to Osmania University)
Hyderabad

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY
(Established By Sultan-Ul-Uloom Education Society in 1980)
(Affiliated to Osmania University, Hyderabad)
(Approved by the AICTE & Accredited by NBA)

CERTIFICATE

This is to certify that the Dissertation entitled "FoodOnn" is a bonafide work done and submitted

By

Mohammed Areebuddin (1604-16-733-046)

Faizan Mohiuddin (1604-16-733-038)

Mohammed Muzakkir Qadri(1604-16-733-045)

During the academic year 2018 from the college **Muffakham Jah College of Engineering and Technology**, affiliated to **Osmania University, Hyderabad** is a record of bonafide work carried out by them under our guidance and supervision. The results presented in this dissertation have been verified and are found to be satisfactory. The results embodied in this dissertation have not been submitted to any other University for the award of any other degree or diploma.

Project Supervisor	Project Coordinator	Head of the Department
Mr. MEER ARSHAD ALI	Mr. J. SRINIVAS	Dr. A. A. MOIZ QYSER
Professor	Assistant Professor	Professor and Head
C.S.E. DEPT	C.S.E DEPT	C.S.E DEPT
M.J.C.ET	M.J.C.E.T	M.J.C.E.T

8-2-249 to 267, " Mount Pleasant " Road No.3, Banjara Hills, Post Box No.14,
Hyderabad – 500 155 Phone : 040-23350523, 23352084, Fax : 040-2335 3428
Website: www.mjcollege.ac.in , e-mail : principal@mjcollege.ac.in;

DECLARATION

We hereby declare that the work presented in this report has been carried out by us under the supervision of Mr. MEER ARSHAD ALI at Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Hyderabad. We declare, to the best of our knowledge, that no part of this report has been submitted for the award of a research degree of any university.

Mohammed Areebuddin **(1604-16-733-046)**

Mohammed Faizan Mohiuddin **(1604-16-733-038)**

Mohammed Muzakkir Qadri **(1604-16-733-045)**

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of my project work. Whatever we have done is only due to their guidance and assistance and we would not forget to thank them.

We heartily thank Dr. ABDUL AHMED MOIZ QYSER, Head of Department of Computer Science and Engineering, for his guidance and suggestions during this project work.

We respect and express our great indebtedness to Mr. MEER ARSHAD ALI Assistant Professor for encouragement and keen interest shown throughout the project. We wish to express our deep sense of gratitude to her for suggesting us with this topic and guiding us all through the project.

Table of contents:

1.ABSTRACT

This report is focusing on the development of a computer based software for searching restaurants based on location with pincode manually entered as input. The software will provide a list of restaurants based on the location provided. The user can choose a restaurant for it's details.

The main feature of the software is to search restaurant details. The details provided will be the exact address,working hours,contact number and Google reviews. The user has an option to view the menu of the desired restaurant.

Another important feature of this software is billing. The user must provide his/her budget initially. From the menu provided, the user must select the items corresponding to the item numbers. These items will be added to the cart and a bill is generated which is inclusive of all taxes. The budget which was provided is compared with the bill. This will help the user to acknowledge the bill before consumption. If the budget is exceeded, the user must reduce the items in the cart.

2.SURVEY

2.1 Existing Systems:

Before starting this project we have surveyed few similar applications like Zomato,Swiggy,Foodpanda etc.

Survey details have been provided below.

Zomato:

Zomato is an Indian restaurant search and discovery service founded in 2008 by Deepinder Goyal and Pankaj Chaddah. It currently operates in 23 countries, including Australia and United States.[5] It provides information and reviews on restaurants, including images of menus where the restaurant does not have its own website.

Deepinder Goyal, Founder & CEO, Zomato, is the Founder and CEO of Zomato. Prior to starting Zomato, Deepinder worked as a management consultant with Bain and Company in New Delhi. It was at Bain that Deepinder conceived the idea of an online restaurant information service after seeing the demand for menu cards among his colleagues. He left Bain in 2008 to start Zomato (then foodiebay) out of his apartment and has since overseen strategy and product development. Deepinder graduated with a Mathematics and Computing degree from IIT Delhi in 2005 and hails from Muktsar in Punjab.

Zomato, founded in 2008, is India’s largest restaurant guide listing over 42,000 restaurants across 12 cities in the country – Delhi NCR, Mumbai, Bangalore, Chennai, Kolkata, Pune, Hyderabad, Ahmedabad,

Jaipur, Chandigarh, Lucknow and Indore. Zomato was started by IIT Delhi alumni in July 2008 for Delhi NCR and has expanded its services over a span of 4 years to 12 cities in India. Info Edge has invested over $6.5M in Zomato since 2010. Zomato has recently forayed into print as well with the Citibank Zomato Restaurant Guide 2012. Zomato is headquartered in New Delhi and currently employs 150 people. Zomato has expanded into the international markets with the launch of their Dubai section and will expand to more cities in the Middle East and South East Asia by the end of 2012.

Swiggy:

Swiggy provides on-demand food delivery platform designed to provide food from neighborhood restaurants to the people. The Company's platform utilizes a smartphone application that has local restaurants and their online ordering menu listed, from which it provides delivery using a fleet of personnel, enabling people to order food at their convenient places.

Swiggy was founded by the trio of Rahul Jaimini, Sriharsha Majety and Nandan Reddy. Swiggy began its initial round of operations in Bengaluru, they focused specifically on Koramangala which is one of the upcoming neighborhoods in Bengaluru. They initially started delivering with just six delivery executives and with only 25 restaurants on its platform. Over a period of time, their business grew steadily with currently 6,000 delivery executives operating across Delhi, Mumbai, Hyderabad, Chennai, Kolkata and Pune.

One of the main reasons for their success is their business model. Swiggy has its own set of delivery executives who are equipped with a smart phone and a Swiggy app which thereby helps the user to track their delivery through map integration. Swiggy doesn't have minimum order policy, which makes it more affordable for users apart from that they charge a delivery fee.

On orders higher than the original amount, Swiggy takes commission from the restaurant. Usually, Swiggy lists restaurants on their website which gives only 15-25% commission on orders.Swiggy differentiates itself from other competitors like Zomato and Food Panda by specializing in food delivery services rather than food ordering services. Minimum delivery time for Swiggy is just 37 minutes which is pretty high and they stand just next to Domino's who deliver in 30 minutes.

Swiggy's success has lead to an increase in start-ups who want to specialize in this area of food delivery services. Many clone script companies are creating replicas of Swiggy's website and revenue model. Agriya, who are experts in developing Clone scripts have recently launched OFOS - Online Food ordering Script, a Swiggy clone script. One of the advantages with clone script is it can be customized according to your needs and requirements. If you are looking forward to capitalizing in this sector, this is guaranteed to ensure sustainability in the market

2.2 Disadvantages of the above applications

- Budget is not asked during the order.
- Approximate bill is not generated.
- Since budget is not being asked, the bill may exceed your budget.
- Meticulous details of restaurants are not provided like contact number,working hours etc.

2.3 Advantages of FoodOnn over its contemporaries

- Comparison is done between the budget initially taken as input from the user and bill generated .
- If budget is exceeded , it notifies the user to make a new bill.
- Provides the user with multiple restaurants in a desired location.
- Provides a precised menu for each restaurant from which the user can select food items.
- Details of the selected restaurant which includes working hours,contact information etc is provided.
- Billing mode being one of the features, allows the user to add food items and generate a bill which inclusive of all taxes.

3.SYSTEM ANALYSIS

3.1 System Requirement Specification:

3.1.1 Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

Operating system : Windows(Any version) or LINUX.

- Coding Language : C++

3.1.2 Functional Requirements

In Software engineering and Systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs.

The official definition of 'a functional requirement' is that it essentially **specifies something the system should do.**

The following are the modules that are going to play an important role in the application being developed.

Modules

- Home
- Rank Module

- Cryptography
- Encryption and Decryption Module
- Architecture And Implementation
- Assumptions in Application Contexts
- Piggybacking Client Information
- I/O Access Prediction
- Modeling Disk I/O Access Patterns
- Chaotic Time Series Prediction

3.1.3 Non-Functional Requirements

The definition for a non-functional requirement is that it essentially specifies **how the system should behave** and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes for of a system.

For this application, the following are the non-functional requirements.

Usability:

The ease of use and training the end users of the system is called usability. The application being developed should have qualities like learning ability, efficiency, affect, control etc. The main aim of the project is to increase the scope of page designer to design a page and to reduce the rework of the programmer.

Performance:

The response time, utilization and throughput behaviour of the system must be acceptable. Care must be taken to ensure a system with comparatively high performance.

Modifiability:

The ease with which this application can accommodate changes to its software must be relatively high. The application must be easily adaptable for changes that is useful for the application in future to withstand the needs of the users.

Portability:

The application being developed must have the ability to run under different computing environments. The environment types can be either hardware of software, but is usually a combination of two.

Reusability:

The application should be developed in a way that the extent to which the modules can be reused in new applications is large. Its modules must be reusable a number of times without any bigger technical difficulties.

Maintainability:

All the modules of the application must be clearly separated to allow different user interfaces to be developed in future. Through thoughtful and effective software engineering, all steps of the software development process must be well documented to ensure maintainability

of the product throughout its life time.

Security:

The application must have redundant factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Security must be ensured in the application by involving authentication of users who access the services.

3.1.4 Hardware Requirements

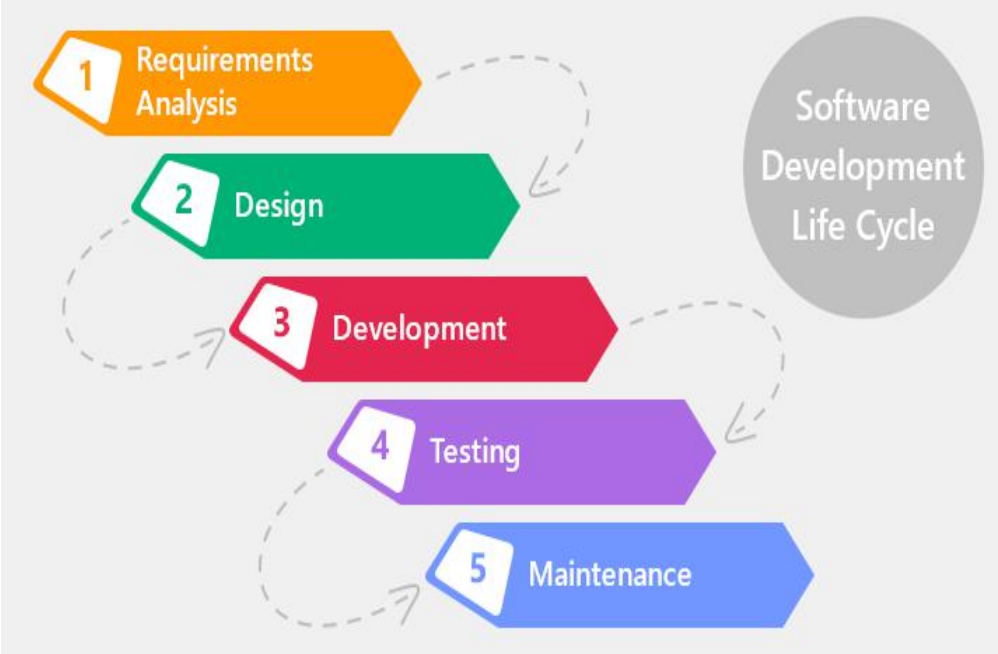
The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Monitor : 14’ Colour Monitor.
- Mouse : Optical Mouse.
- Ram : 512 Mb.

3.2 Software Development Life Cycle (SDLC):

The Software Development Life Cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systemsand software engineering to describe a process for planning, creating, testing, and deploying an information system.

The Software Development Life Cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.



3.3 Time Estimation:

The total duration provided for completion of this project was 14 weeks. The total days utilized to complete this project was 11 weeks.

The total number of lines of code is approximately 400.

This project is done by 3 individuals as a team.

The work was equally contributed in the development of this software .

TASK	DURATION
1. Requirement analysis	2 weeks
2. Design	3 weeks
3. Development	4 weeks
4. Testing	2 weeks

3.4 Feasibilty Study:

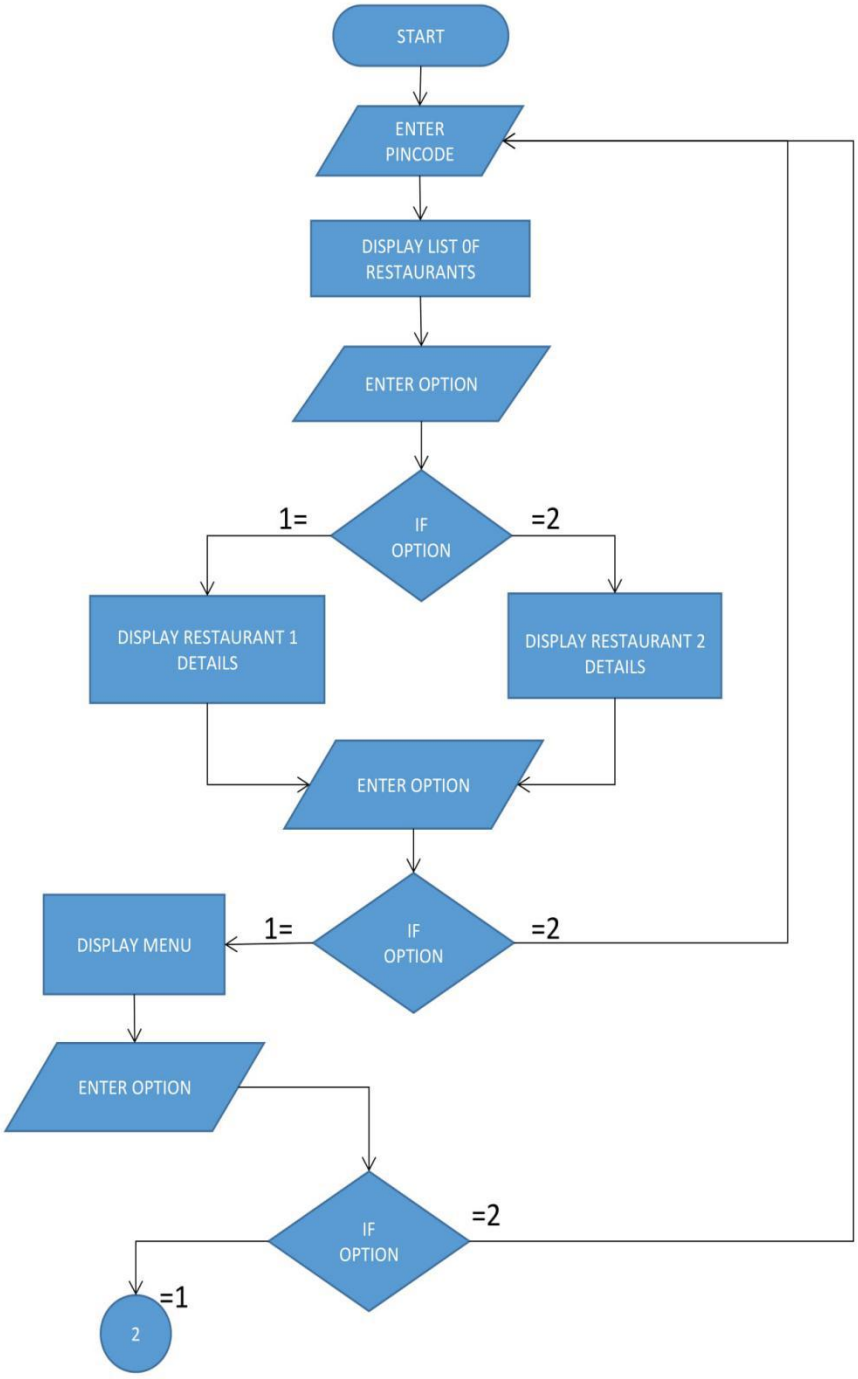
The total duration provided for completion of this project was 14 weeks.

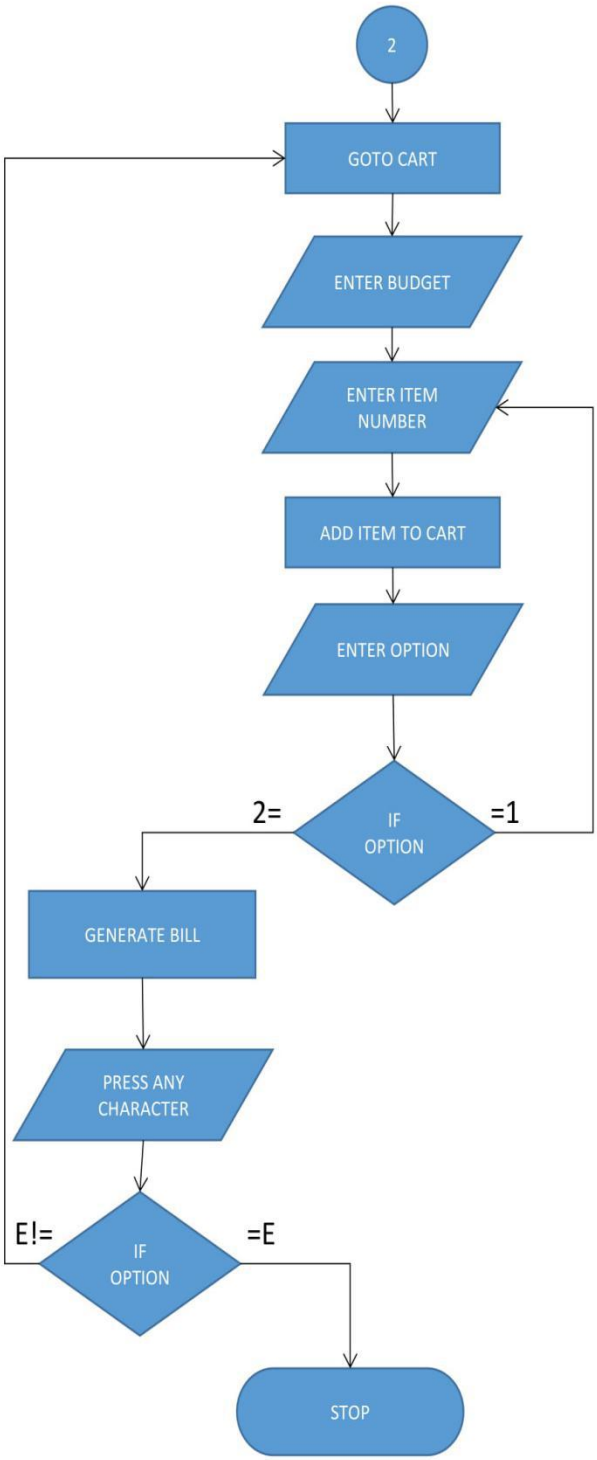
The total days utilized to complete this project was 11 weeks.

Hence the software is profoundly feasible.

4. SOFTWARE DESIGN

Flow Chart




2
GOTO CART
ENTER BUDGET
ENTER ITEM NUMBER
ADD ITEM TO CART
ENTER OPTION
IF OPTION
2=
=1
GENERATE BILL
PRESS ANY CHARACTER
IF OPTION
E!=
=E
STOP

5.SOFTWARE IMPLEMENTATION

5.1 Source code:

```
#include<iostream.h>
#include<fstream.h>
#include<string>
using namespace std;
class Main
{
string s2;
string s3;
string pin;
int budget;
string str,str2,str3,path1,path2,path3,path4,path5;
public:
void Screen1()
{
fstream ifile2;
path2="main/restaurants/Screen1.txt";
char *c2=(char*)path2.c_str();
system("clear");
cout<<endl;
cout<<endl;
cout<<endl;
cout<<"
    **********Welcome**********"<<endl;
cout<<"          FoodONN"<<endl;
cout<<endl;
```

```
cout<<endl;
cout<<endl;
ifile2.open(c2);
if(!ifile2)
{
exit(0);
}
while(!ifile2.eof())
{
getline(ifile2,str2);
cout<<str2<<endl;
}
ifile2.close();
cout<<"\n\n"<<endl;
loop1:cout<<"                    Enter the Location(500001-25):";
cin>>pin;
if(pin.compare("500001")<0||pin.compare("500025")>0)
{
cout<<"                          Invalid PIN"<<endl;
goto loop1;
}
}
void Screen2()
{
loop8:system("clear");
cout<<endl;
cout<<endl;
```

```
cout<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
cout<<"
**********Welcome**********"<<endl;
cout<<"       FoodONN"<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
fstream ifile;
path1=(string)"main/restaurants/"+pin+(string)"/"+pin+(string)".txt
";
char *c=(char*)path1.c_str();
ifile.open(c);
if(!ifile)
{
cout<<"Error in opening file..!!";
exit(0);
}
while(!ifile.eof())
{
getline(ifile,str);
cout<<str<<endl;
}
ifile.close();
cout<<endl;
```

```
cout<<endl;
loop7:cout<<"                    Choose the Restaurant:";
cin>>s2;
if(s2.compare("1")<0||s2.compare("2")>0)
{
goto loop7;
}
}
void Screen3()
{
system("clear");
cout<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
cout<<"
**********Welcome**********"<<endl;
cout<<"
FoodONN"<<endl;
cout<<endl;
cout<<endl;
cout<<endl;
cout<<"                          Restaurant Details"<<endl;
fstream ifile;
path3=(string)"main/restaurants/"+pin+(string)"/"+pin+s2+(string)
".txt";
```

```
char *c=(char*)path3.c_str();
ifile.open(c);
if(!ifile)
{
cout<<"Error in opening file..!!";
exit(0);
}
while(!ifile.eof())
{
getline(ifile,str);
cout<<str<<endl;
}
ifile.close();
cout<<endl;
cout<<endl;
}
void Screen4()
{
fstream ifile3;
path4="main/restaurants/menu.txt";
char *c3=(char*)path4.c_str();
system("clear");
cout<<endl;
cout<<endl;
cout<<"
**********Welcome**********"<<endl;
cout<<"         FoodONN"<<endl;
```

```
cout<<endl;
cout<<endl;
ifile3.open(c3);
if(!ifile3)
{
cout<<"Error in opening file..!!";
exit(0);
}
while(!ifile3.eof())
{
getline(ifile3,str2);
cout<<str2<<endl;
}
ifile3.close();
}
void Screen5()
{
   fstream ifile5;
path5="main/restaurants/menu.txt";
char *c4=(char*)path5.c_str();
system("clear");
cout<<endl;
cout<<endl;
cout<<"
**********Welcome**********"<<endl;
cout<<"          FoodONN"<<endl;
cout<<endl;
```

```
cout<<endl;
ifile5.open(c4);
if(!ifile5)
{
cout<<"Error in opening file..!!";
exit(0);
}
while(!ifile5.eof())
{
getline(ifile5,str3);
cout<<str3<<endl;
}
ifile5.close();
   int invoice[50][4];
   int i=0;
   char more;
loop2:cout<<"            Enter the Budget(50-10000):";
cin>>budget;
if(budget>10000)
{
cout<<"                    Budget exceeded"<<endl;
goto loop2;
}
else if(budget<35)
{
cout<<"                                                Low Budget"<<endl;
goto loop2;
```

```
}
//system("clear");
   do {
      cout << "\n                    Item Number: ";
      cin >> invoice[i][0];
      cout << "                     Quantity : ";
      cin >> invoice[i][1];
      invoice[i][2] = getUnitprice(invoice[i][0]);
      invoice[i][3] = invoice[i][1] * invoice[i][2];
      cout << "          1)Add Item to cart          2)Bill\n "<<endl;
loop9:cout<<"                     Enter Option: ";
      cin >> more;
i++;
   }while(more=='1');
   cout<<"     ItemCode Quantity  UnitPrice  TotalPrice\n\n";
   int tot=0;
   for(int k=0;k<i;k++)
   {                  for(int l=0; l<4; l++)
                       {
                            cout <<"     "<< invoice[k][l]    << "\t";
                       }
                       cout << endl;
                       tot = tot + invoice[k][3];
            }
   double tota=tot*1.05;
  cout << "\n\n             Total(inclusive of taxes) : " << tota<<endl;
      if (tota>budget)
```

```
{
cout<<"                  Total has exceeded Budget"<<endl;
}
/*    cout << "\n              Press any key to refresh cart\n";
cout<<"                  or press E to exit";
char x;
cin>>x;*/
  // return 0;
}
int getUnitprice(int itemCode){
   int price;
   switch (itemCode)
   {
case 1: price = 250;
   break;
case 2: price = 300;
   break;
case 3: price = 550;
   break;
case 4: price = 850;
   break;
case 5: price = 850;
   break;
case 6: price = 200;
   break;
case 7: price = 120;
   break;
```

```
case 8: price = 140;
   break;
case 9: price = 150;
   break;
case 10: price = 153;
   break;
case 11: price = 100;
   break;
case 12: price = 150;
   break;
case 13: price = 180;
   break;
case 14: price = 150;
   break;
case 15: price = 150;
   break;
case 16: price = 200;
   break;
case 17: price = 220;
   break;
case 18: price = 185;
   break;
case 19: price = 140;
   break;
case 20: price = 210;
   break;
case 21: price = 320;
```

```
  break;
case 22: price = 380;
    break;
case 23: price = 190;
    break;
case 24: price = 160;
    break;
case 25: price = 190;
    break;
case 26: price = 140;
    break;
case 27: price = 150;
    break;
case 28: price = 190;
    break;
case 29: price = 200;
    break;
case 30: price = 210;
    break;
case 31: price = 320;
    break;
case 32: price = 330;
    break;
case 33: price = 330;
    break;
case 34: price = 350;
    break;
```

```
case 35: price = 350;
    break;
case 36: price = 35;
    break;
case 37: price = 35;
    break;
case 38: price = 40;
    break;
case 39: price = 45;
    break;
case 40: price = 40;
    break;
case 41: price = 125;
    break;
case 42: price = 125;
    break;
case 43: price = 200;
    break;
case 44: price = 140;
    break;
case 45: price = 150;
    break;
case 46: price = 170;
    break;
case 47: price = 80;
    break;
case 48: price = 45;
```

```
    break;
  case 49: price = 70;
    break;
  case 50: price = 60;
    break;
      default: price = 0;
         break;
   }
   return price;
}
};
 main()
{
Main m;
string s4,s5;
int i=0;
loop3:m.Screen1();
loop4:m.Screen2();
loop5:m.Screen3();
cout<<"       1)MENU                                    2)Go Back"<<endl;
cout<<"\n\n                  Enter Option:";
cin>>s4;
if(s4=="1")
{
goto loop6;
}
else if(s4=="2")
```

```
{
goto loop3;
}
loop6:m.Screen4();
cout<<"    1)Add items to Cart                    2)Go Back"<<endl;
cout<<"\n\n                          Enter Option:";
cin>>s4;
if(s4=="1")
{
goto loop7;
}
else if(s4=="2")
{
goto loop3;
}
loop7:m.Screen5();
   cout << "\n                          Press any key to refresh cart\n";
cout<<"                                 or press E to exit:";
char x;
cin>>x;
if (x=='E'||x=='e')
{
exit(0);
}
Else
goto loop7;
     }
```

6.SOFTWARE TESTING

Software testing is a process of executing a program or application with the intent of finding the software bugs.

- It can also be stated as the process of validating and verifying that a software program or application or product:
  - Meets the business and technical requirements that guided it's design and development
  - Works as expected
  - Can be implemented with the same characteristic.

Process: Testing is a process rather than a single activity.

All Life Cycle Activities:

Testing is a process that's take place throughout the Software Development Life Cycle (SDLC).

- The process of designing tests early in the life cycle can help to prevent defects from being introduced in the code. Sometimes it's referred as "verifying the test basis via the test design".
- The test basis includes documents such as the requirements and design specifications.

Planning:

We need to plan as what we want to do. We control the test activities, we report on testing progress and the status of the software under test.

Preparation:

We need to choose what testing we will do, by selecting test conditions and designing test cases.

Evaluation:

During evaluation we must check the results and evaluate the software under test and the completion criteria, which helps us to decide whether we have finished testing and whether the software product has passed the tests.

6.1 Types of Testing

1) **Static Testing:**

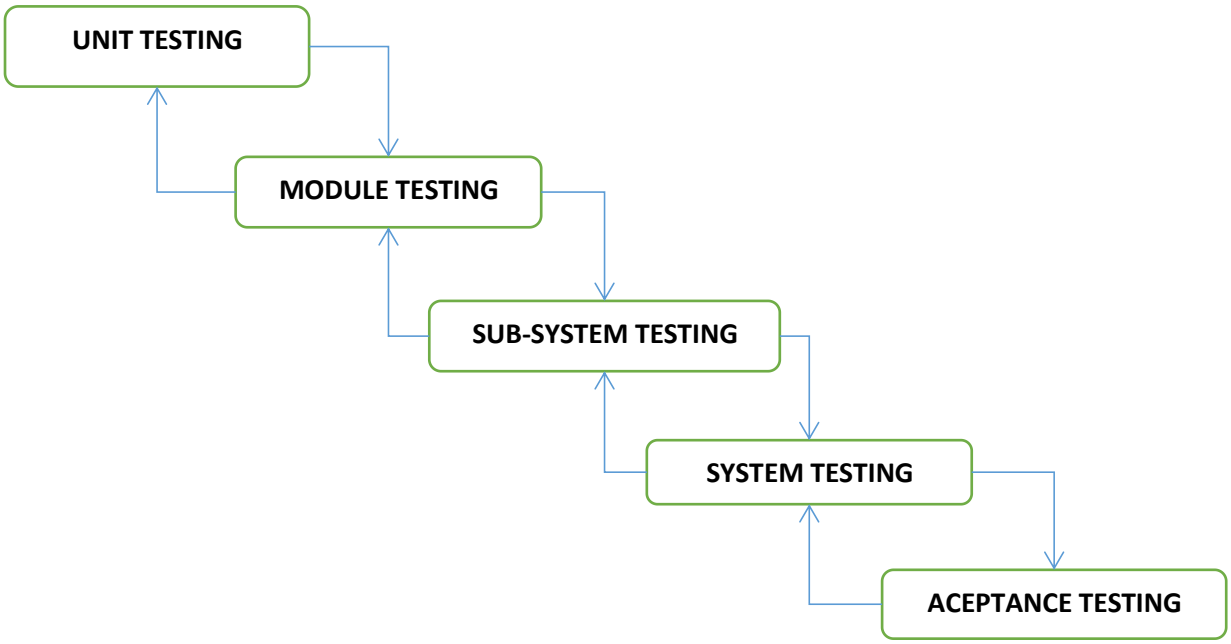
It can test and find defects without executing code. Static Testing is done during verification process. This testing includes reviewing of the documents (including source code) and static analysis. This is useful and cost effective way of testing. For example: reviewing, walkthrough, inspection, etc.

2) **Dynamic Testing:**

In dynamic testing the software code is executed to demonstrate the result of running tests. It's done during validation process. For example: unit testing, integration testing, system testing, etc.

3) **Integration Testing:**

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.



integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

4) **Component integration testing**: Testing performed to expose defects in the interfaces and interaction between integrated components.

5) S**ystem integration testing**: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Analogy

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. For example, whether the cap fits into the body or not.

Method

Any of Black Box Testing, White Box Testing and Gray Box Testing methods can be used. Normally, the method depends on your definition of 'unit'.

When is Integration Testing performed?

Integration Testing is the second level of testing performed after Unit Testing and before System Testing.

Who performs Integration Testing?

Developers themselves or independent testers perform Integration Testing.

Approaches

- Big Bang is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach

is taken when the testing team receives the entire software in a bundle. So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.

- Top Down is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

- Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.
- Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

6.2 TEST CONDITIONS:

TEST NO.	INPUT	EXPECTED BEHAVIOR	OBSERVED BEHAVIOR	STATUS P=PASS F=FAIL
1	Enter location & press enter	Display list of restaurants	List of restaurants is displayed	P
2	Select restaurant	Display details of restaurant	Details are displayed	P
3	Selecting (1) Menu	Display menu	Menu displayed	P

	(2) Go Back	Go to Home screen	Home screen displayed	P
4	Selecting (1)Add to cart	Display billing screen	Billing screen displayed	P
	(2)Go Back	Go to Home screen	Home screen displayed	P
5	selecting (1)Enter item no. and quantity	Items must be added to cart	Items are added to cart	P
	(2)Bill	Bill must be displayed	Bill is displayed	P
6	(1)Press any key to refresh	Cart should be refreshed	Cart is refreshed	P
	(2)Press E to exit	Should terminate program	Program is terminated	P

7. EXECUTION

HOMESCREEN

Fig. Homescreen with list of locations ranging between 500001-500025.

Fig. Screen showing list of Restaurants

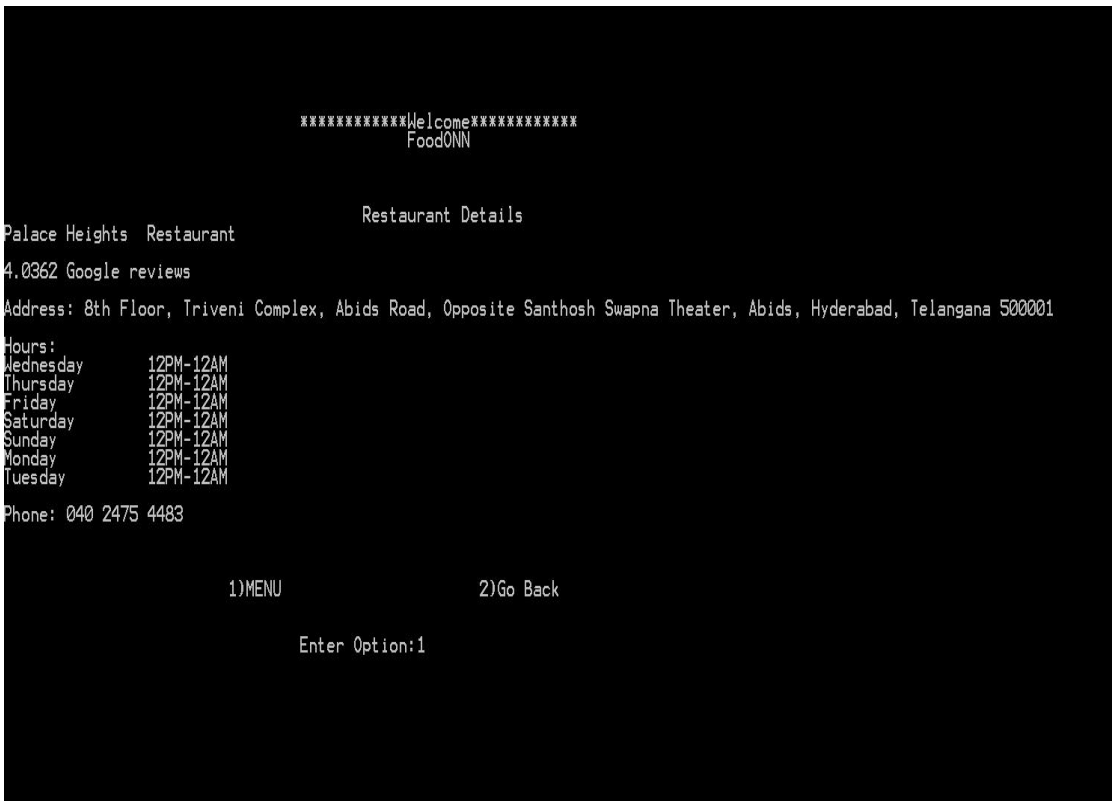


Fig. Screen showing details of selected restaurant

Fig. Screen displaying the menu of selected restaurant

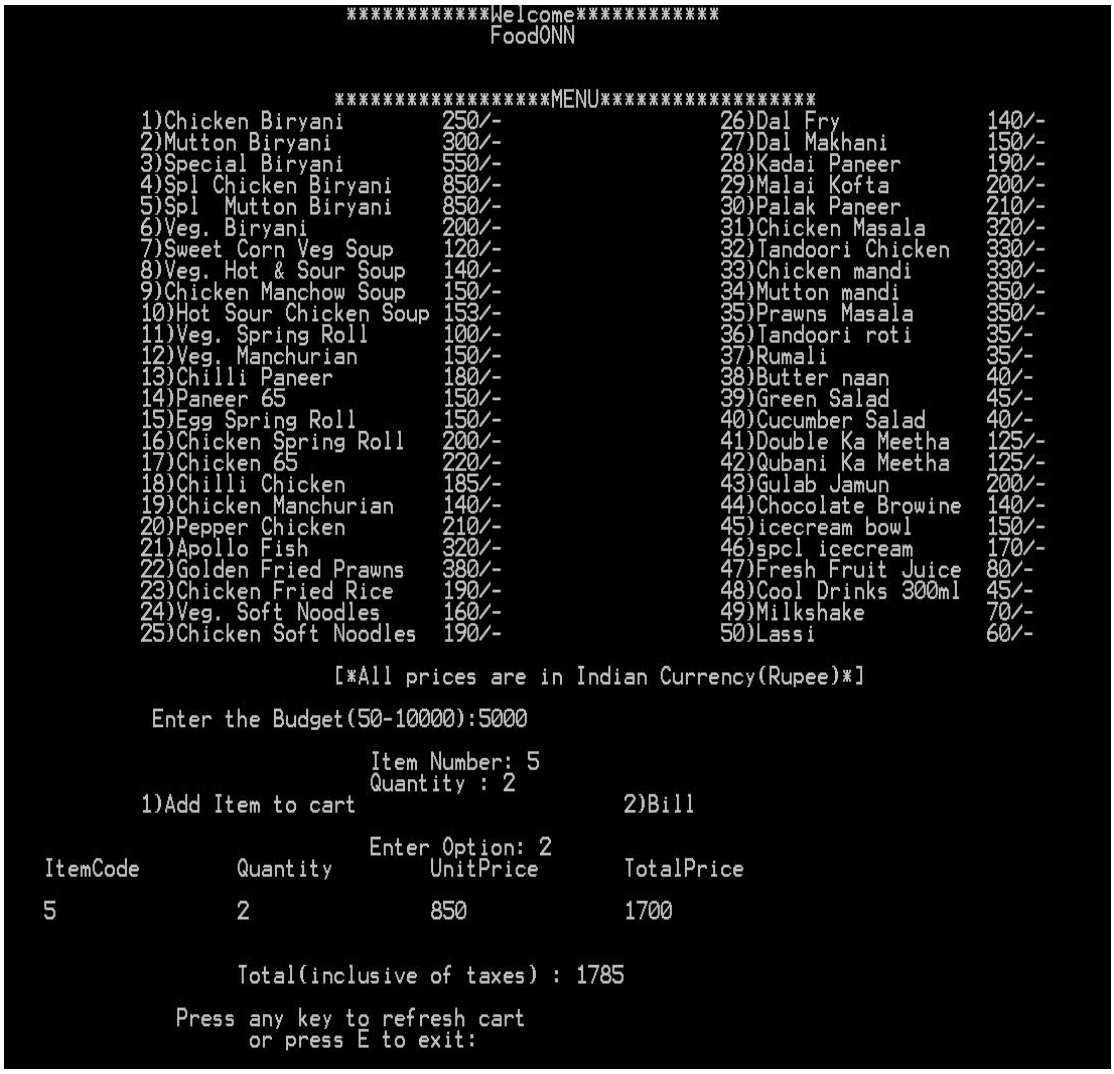


Fig. Billing Screen(within Budget)

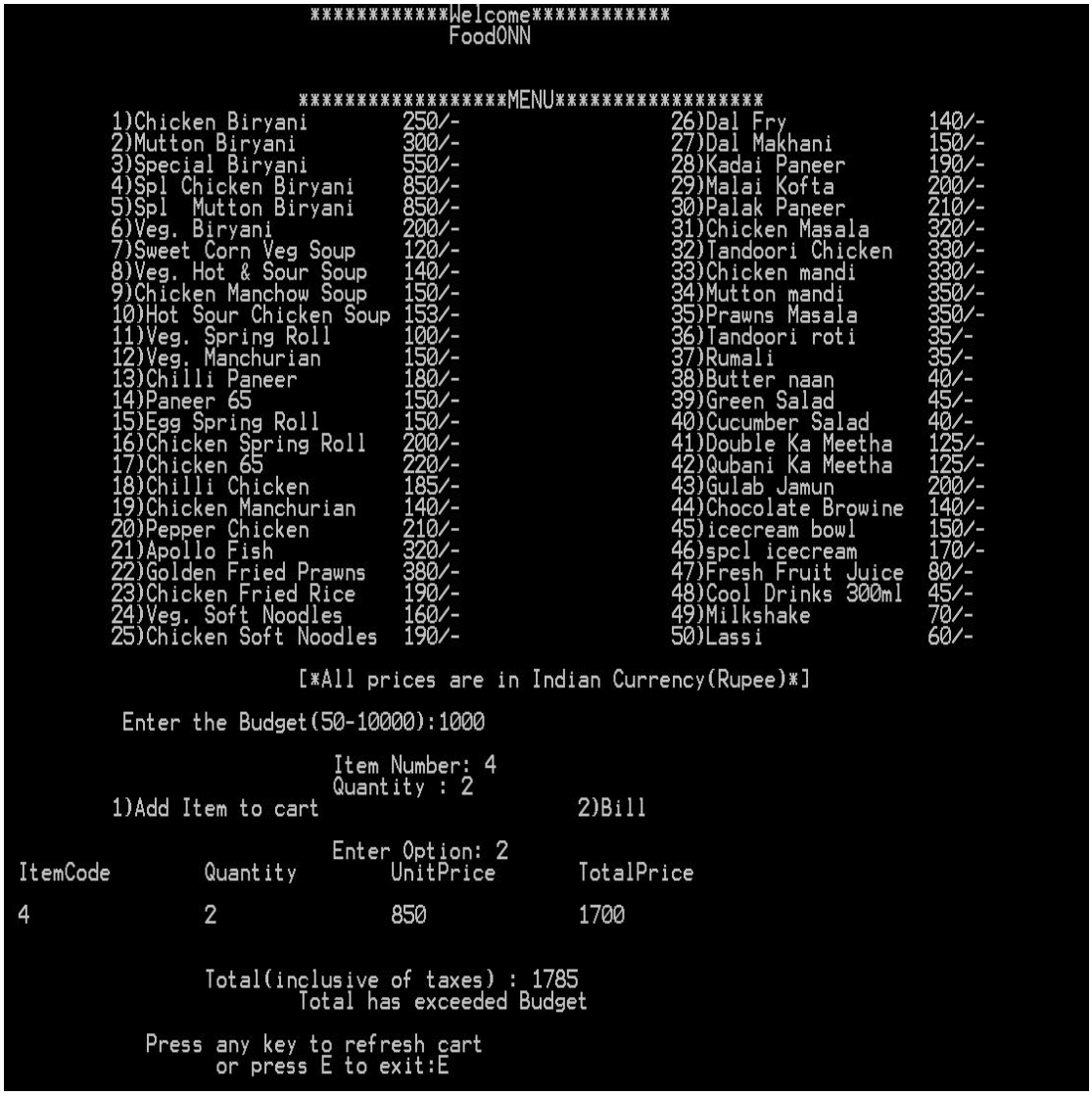


Fig. Billing Screen(Budget exceeded)

8.CONCLUSION

8.1 Conclusion:

We conclude that our application helps the user to search restaurant details. The details provided are the exact address,working hours,contact number and google reviews. The user has an option to view the menu of the desired restaurant.

This software features billing system. The user provides his/her budget initially. From the menu provided, the user selects the items corresponding to the item numbers. These items are added to the cart and a bill is generated which is inclusive of all taxes. The budget that is provided is compared with the bill. This helps the user to acknowledge the bill before consumption. If the budget is exceeded, the user needs to reduce the items in the cart.

8.2 Future Enhancements:

This c++ application can be further developed into a user friendly GUI based Java application. It can also be made into an Android based App. An additional feature that can be integrated is adding Google Maps for navigation. Future enhancements can be made to make this application work offline.

9.BIBLIOGRAPHY

Books:

1) An Introduction to Object-Oriented Programming and C++(ISRD group)
2) Object Oriented Programming with C++ 6th Edition(E. Balagurusamy)

3) Advanced Objected-Oriented Programming Using C++ 1 Edition (RS Salaria)

Websites:

1) https://en.wikipedia.org/wiki/Zomato

2) https://stackoverflow.com/questions/12146201/how-to-handle-or-avoid-a-stack-overflow-in-c

3) http://www.cplusplus.com/reference/string/string/

4) https://www.geeksforgeeks.org/loops-in-c/