Project Report
On

Mess Management System

Submitted By:

Nitesh Meena & Raju
Roll No. 61 & 75

Under the guidance of: -

Chitra Bhaskar Mam
(Project Coordinator)

ACKNOWLEDGEMENT

I would like to convey my heartful thanks to Chitra Bhaskar Mam who always gave valuable suggestions & guidance for completion of my project. She helped me to understand & remember important details of the project. My project has been a success only because of his guidance.

I am especially indented & I am also beholden to my friends. And finally, I thank to the members of my family for their support & encouragement.

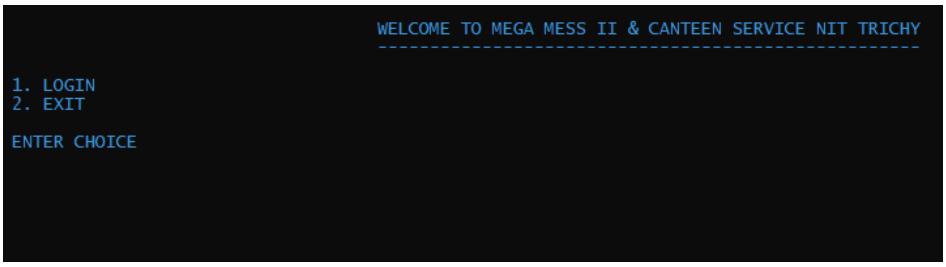
Table of Contents

Abstract

The Mess Management System is designed & developed for a canteen to provide food at hostel and colleges and offices. This system make the work of the caterers easy as it keep all the records of the customers. This system first check the availability of the of the food which is asked by the customer and the take order. It also print the final bill of the customer.

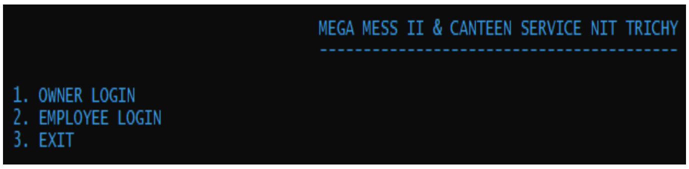
It automates the Systems records, their Selling and Maintenance, Balance evaluation, due to calculation other functions. In other words, you can say it a complete Mess management System. In this project we can easily maintain food sales details.

List of Figures/ tables/ screens

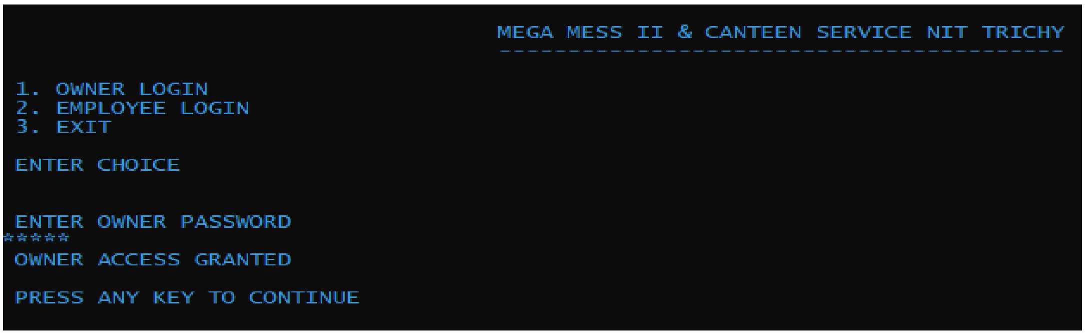


```
                                        WELCOME TO MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        ----------------------------------------------------

1. LOGIN
2. EXIT

ENTER CHOICE
```

LOGIN FORM

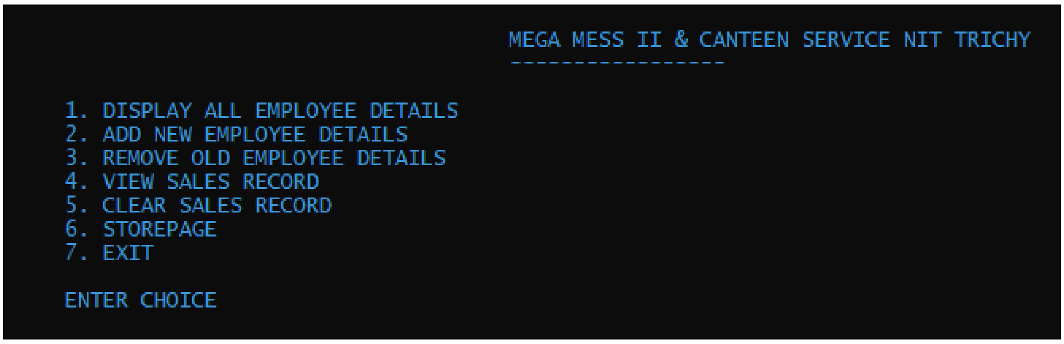


```
                              MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                              -----------------------------------------

1. OWNER LOGIN
2. EMPLOYEE LOGIN
3. EXIT
```



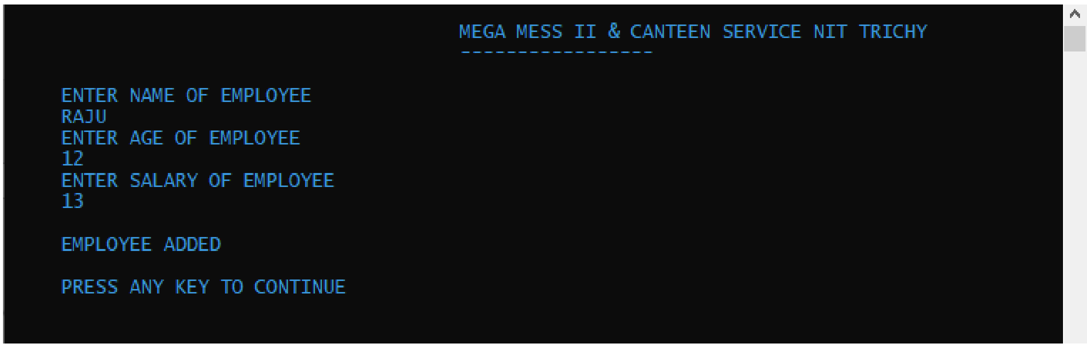
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------------------------------

 1. OWNER LOGIN
 2. EMPLOYEE LOGIN
 3. EXIT

 ENTER CHOICE


 ENTER OWNER PASSWORD
*****
 OWNER ACCESS GRANTED

 PRESS ANY KEY TO CONTINUE
```

MAIN PAGE



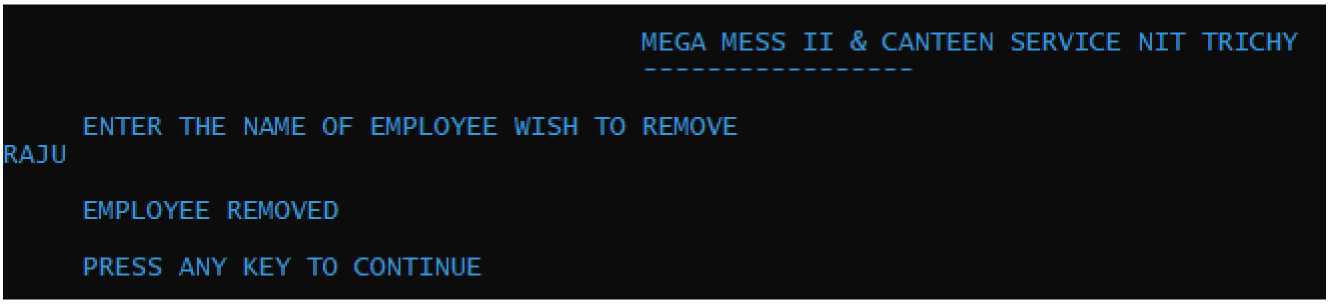
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------


    1. DISPLAY ALL EMPLOYEE DETAILS
    2. ADD NEW EMPLOYEE DETAILS
    3. REMOVE OLD EMPLOYEE DETAILS
    4. VIEW SALES RECORD
    5. CLEAR SALES RECORD
    6. STOREPAGE
    7. EXIT

    ENTER CHOICE
```

EMPLOYEE ADD



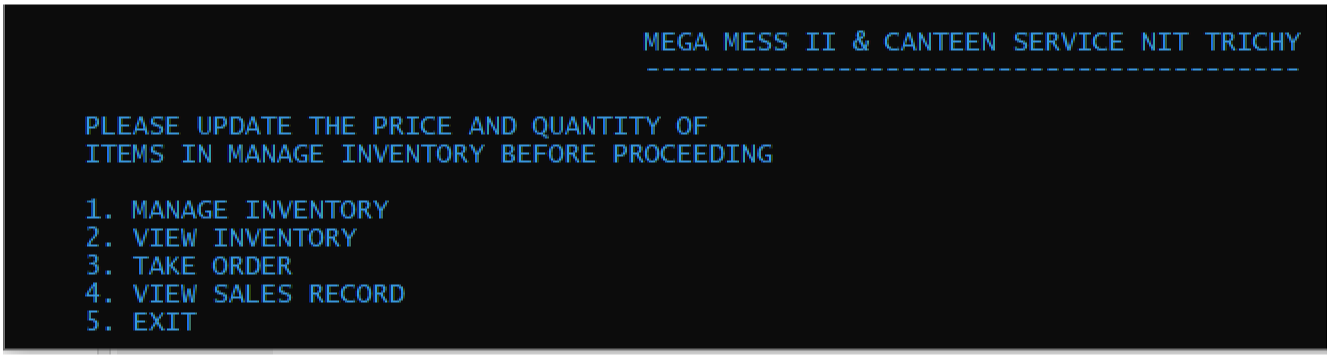
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------

     ENTER NAME OF EMPLOYEE
     RAJU
     ENTER AGE OF EMPLOYEE
     12
     ENTER SALARY OF EMPLOYEE
     13

     EMPLOYEE ADDED

     PRESS ANY KEY TO CONTINUE
```

EMPLOYEE REMOVE



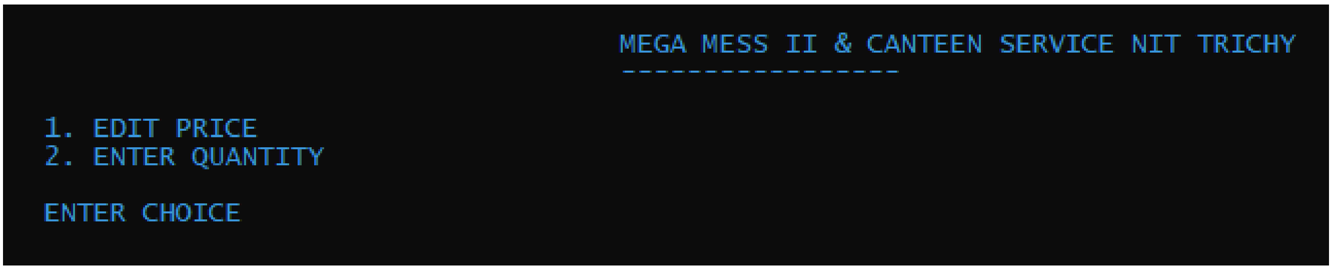
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------

     ENTER THE NAME OF EMPLOYEE WISH TO REMOVE
RAJU

     EMPLOYEE REMOVED

     PRESS ANY KEY TO CONTINUE
```

SALES AND MANAGE ITEMS



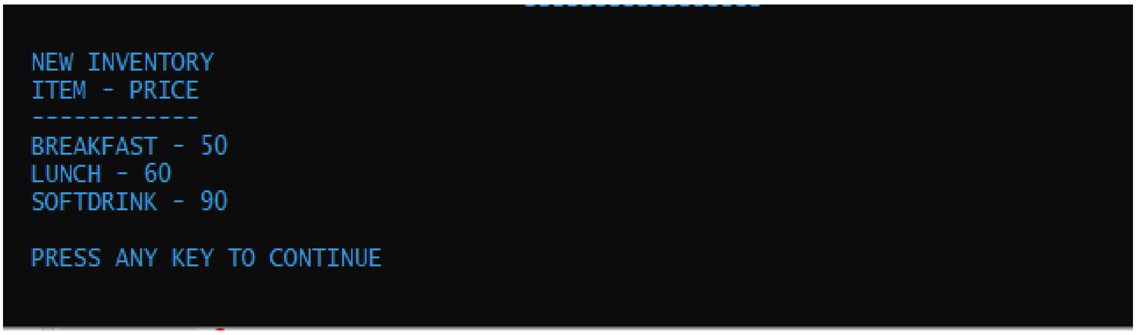
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------------------------------

     PLEASE UPDATE THE PRICE AND QUANTITY OF
     ITEMS IN MANAGE INVENTORY BEFORE PROCEEDING

     1. MANAGE INVENTORY
     2. VIEW INVENTORY
     3. TAKE ORDER
     4. VIEW SALES RECORD
     5. EXIT
```

PRICE AND QUANTITY MANAGEMENT CHOICE



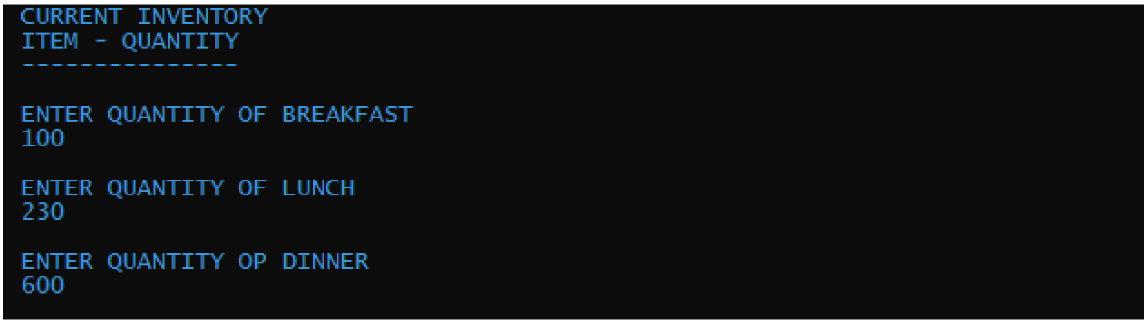
```
                                        MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                        -----------------

  1. EDIT PRICE
  2. ENTER QUANTITY

  ENTER CHOICE
```

PRICE MODIFICATION



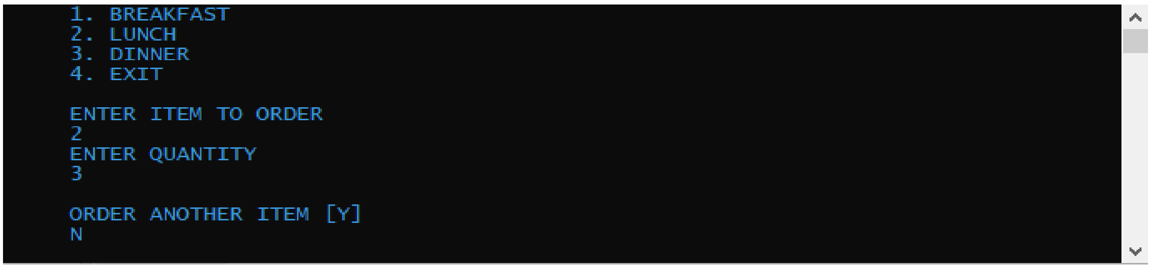
```
NEW INVENTORY
ITEM - PRICE
------------
BREAKFAST - 50
LUNCH - 60
SOFTDRINK - 90

PRESS ANY KEY TO CONTINUE
```

QUANTITY MODIFICATION



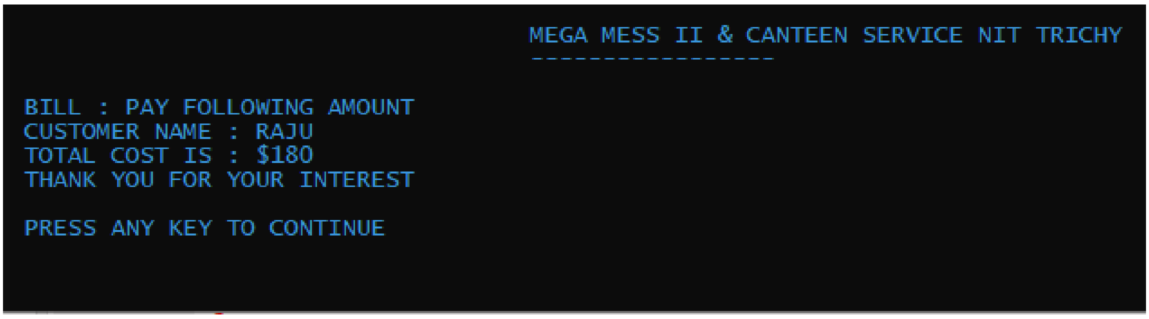
```
CURRENT INVENTORY
ITEM - QUANTITY
---------------

ENTER QUANTITY OF BREAKFAST
100

ENTER QUANTITY OF LUNCH
230

ENTER QUANTITY OP DINNER
600
```

TAKE ORDER



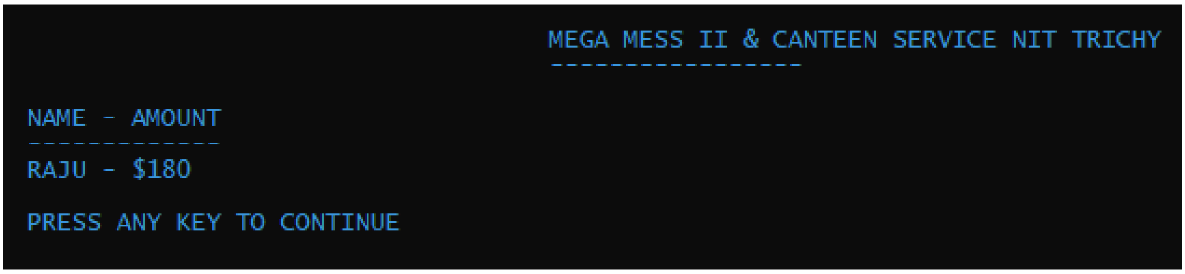
```
    1. BREAKFAST
    2. LUNCH
    3. DINNER
    4. EXIT

    ENTER ITEM TO ORDER
    2
    ENTER QUANTITY
    3

    ORDER ANOTHER ITEM [Y]
    N
```

BILL



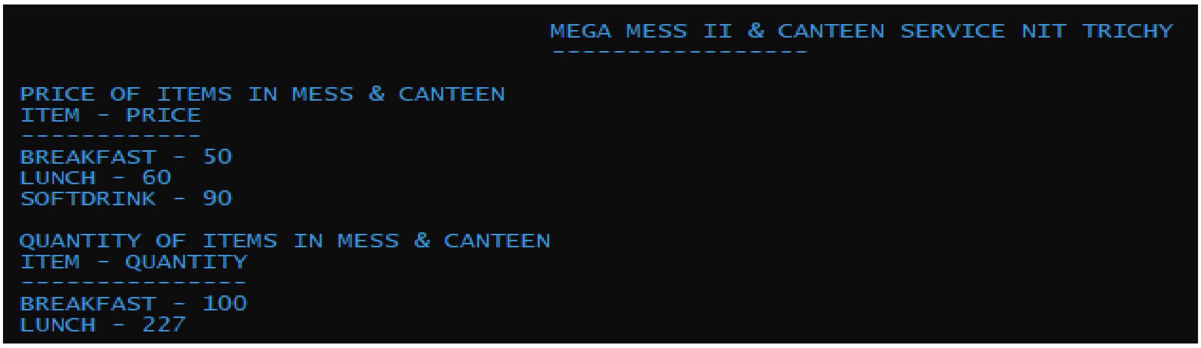
```
                                         MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                         -----------------

BILL  : PAY FOLLOWING AMOUNT
CUSTOMER NAME : RAJU
TOTAL COST IS : $180
THANK YOU FOR YOUR INTEREST

PRESS ANY KEY TO CONTINUE
```

SALES RECORD



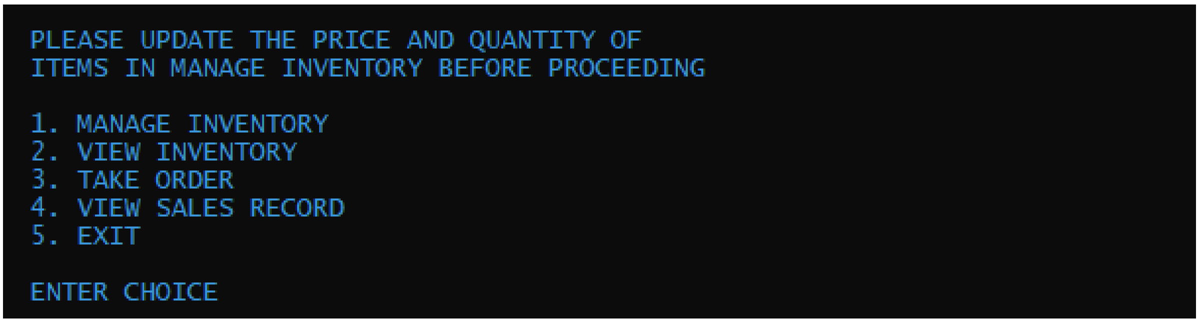
```
                                         MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                         -----------------

NAME - AMOUNT
-------------
RAJU - $180

PRESS ANY KEY TO CONTINUE
```

STOCKS



```
                                         MEGA MESS II & CANTEEN SERVICE NIT TRICHY
                                         -----------------

PRICE OF ITEMS IN MESS & CANTEEN
ITEM - PRICE
------------
BREAKFAST - 50
LUNCH - 60
SOFTDRINK - 90

QUANTITY OF ITEMS IN MESS & CANTEEN
ITEM - QUANTITY
---------------
BREAKFAST - 100
LUNCH - 227
```

EMPLOYEE LOGIN INTERFACE



```
PLEASE UPDATE THE PRICE AND QUANTITY OF
ITEMS IN MANAGE INVENTORY BEFORE PROCEEDING

1. MANAGE INVENTORY
2. VIEW INVENTORY
3. TAKE ORDER
4. VIEW SALES RECORD
5. EXIT

ENTER CHOICE
```

Database Design

Database Design in most important in any project. We are using the following table to store the information.

Price table

Name	Data type
Breakfast	integer
Lunch	integer
Dinner	integer
Total_amt	float

Quantity

Breakfast	Data type
Lunch	integer
Dinner	integer
Total	integer

Sales Record

Customer Name	Data type
Name 1	integer
Name 2	integer
Name 3	Integer

Employee Table

Customer Name	Age	Salary
Name 1	integer	integer
Name 2	integer	integer
Name 3	Integer	integer

Symbols & Abbreviations

Objective

The objective of my project to provides management facility to Mess Management. We have given many facilities for users in this project.

1) Login screen
2) Owner/employee login
3) Add /remove employee
4) Modify stock
5) Update price
6) Delete/show sales record
7) Order food
8) Billing
9) Availability of food

Purpose

It is the project about Mess management System. It automates the Systems records, their Selling and Maintenance, Balance evaluation, due to calculation other functions. In other words you can say it a complete mess management System.

In this project we can easily maintain food sales details. We can see the available food details before selling the particular items .

Project Scope

The scope of project **‘Mess management system’** is to Develop C++ based software to support for daily sale, receipt & Balance of food items and maintain the all information of sales.

This software will be very useful for the small as well as big mess.

It will also save lots of time, as system will perform all tasks in quick time profiting customer and shop owner as well, so it will prove very economical in every respect.

ANALYSIS

Introduction

- The existing system is referred to as manual system.
- In this system each record of the purchase and sale in needed to be entered in a record book and accounting book should be kept carefully.
- This system involves a lot of paper work.
- Records are maintained in registers.

Software Requirement Specification

User requirement

- Technology is used to handle data so that data can be stored efficiently and retrieved when needed.
- Less time is consumed in this process.
- User friendly system.

Software requirement

Front End Tool: C++

Back End Tool: File handling

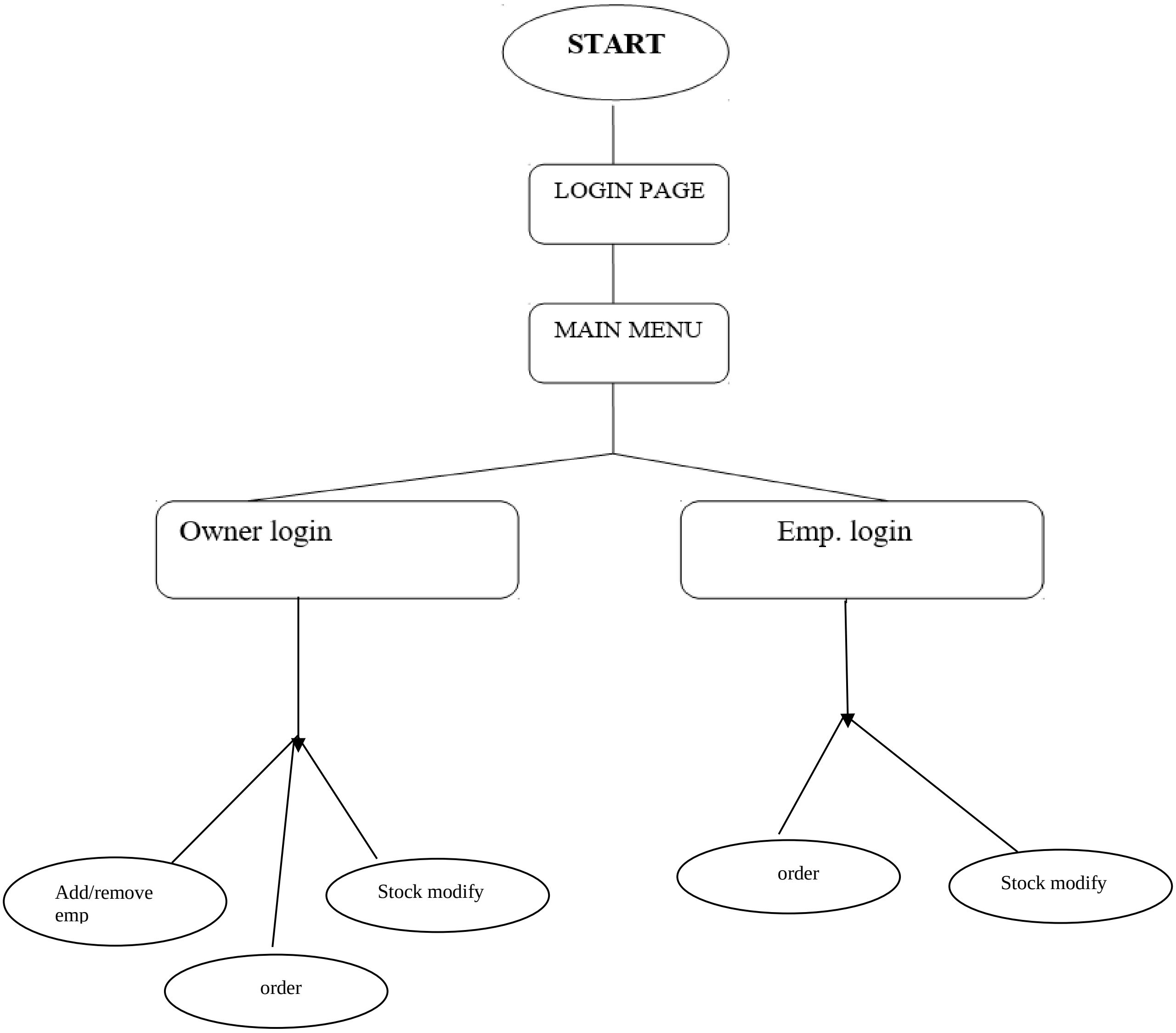
PLATFORM USED: Windows XP, Windows 10.

Hardware requirement

1) 20 GB HDD Free Space
2) 128 MB RAM

3) P IV or above Processor
4) Monitor
5) Keyboard: Standard
6) Mouse: Optional

Content diagram of Project



DESIGN

Introduction

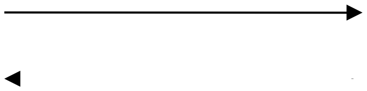
DATA FLOW DIAGRAMS (DFD’S)

The DFD was first developed by Larry Constance as a way of expressing system in a graphical form. A DFD, also known as Bubble Chart, has a purpose of clarifying system requirement and identifying major transformation that will become the programs in the system design.

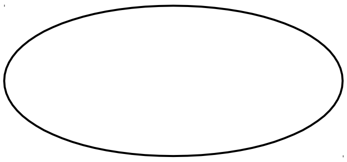
DFD SYMBOLS

1. A **SQUARE** defines a source or destination of system data

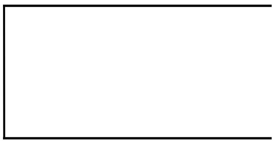
2. An **ARROW** identifies data flow or data in motion. It is a pipeline through which information flow.



3. A **CIRCLE** or a **BUBBLE** (Some people use an over bubble) represents a process transforms in coming data flow into outgoing data flow.



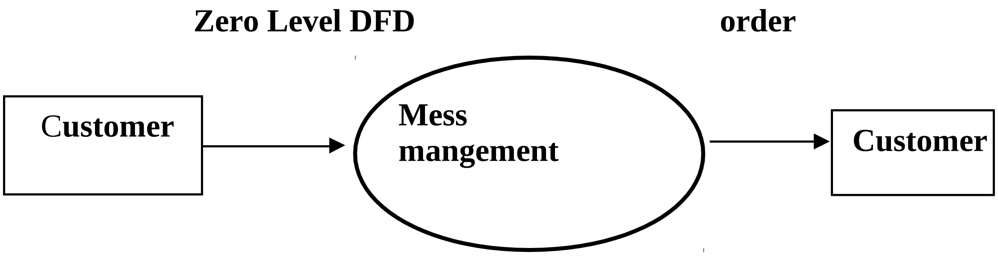
4. An **OPEN RECTANGLE** is a data store or data at rest or a temporary rest repository of data.



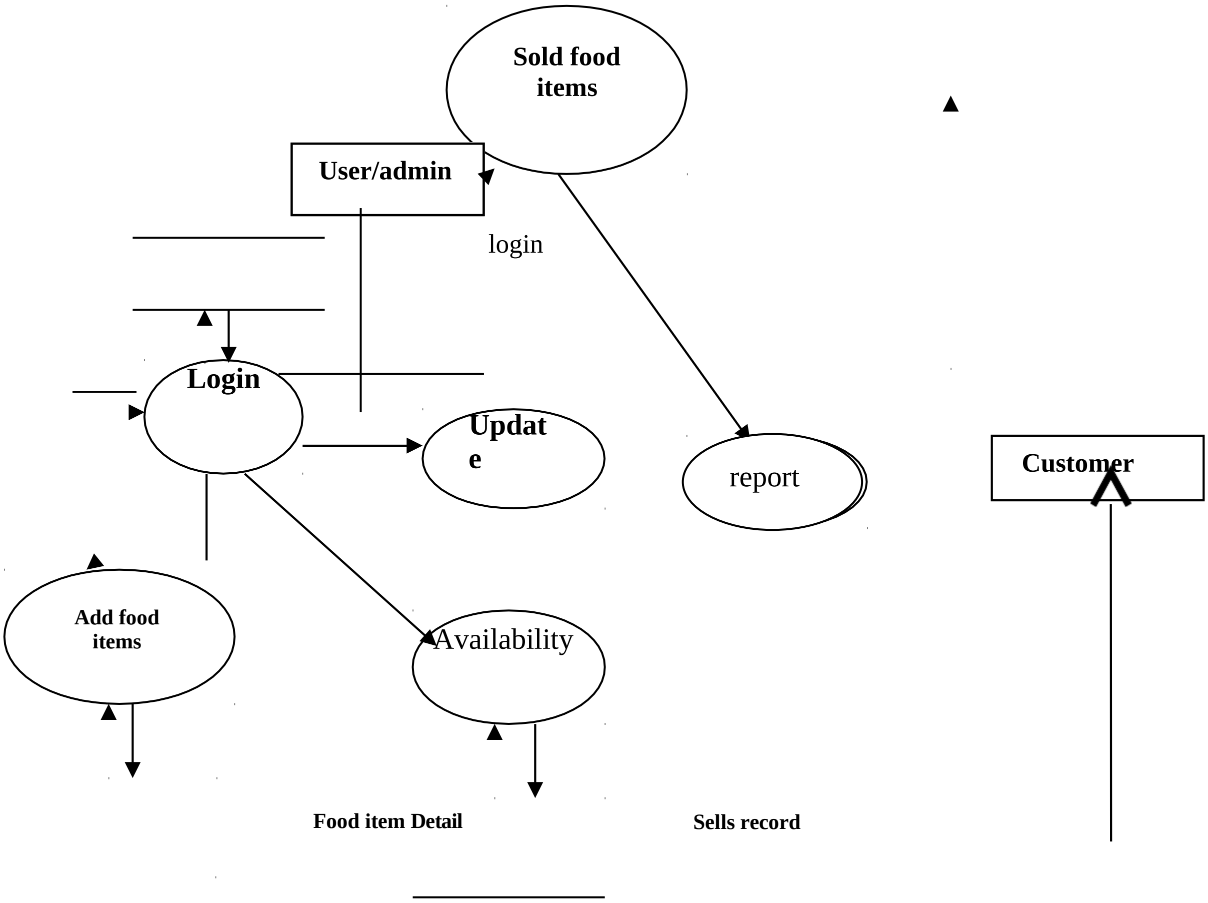
Note that a DFD describe what data flow (logical) rather than they are processed, so it does not depend on hardware, software and data structure or file organization.

Data Flow Diagram:

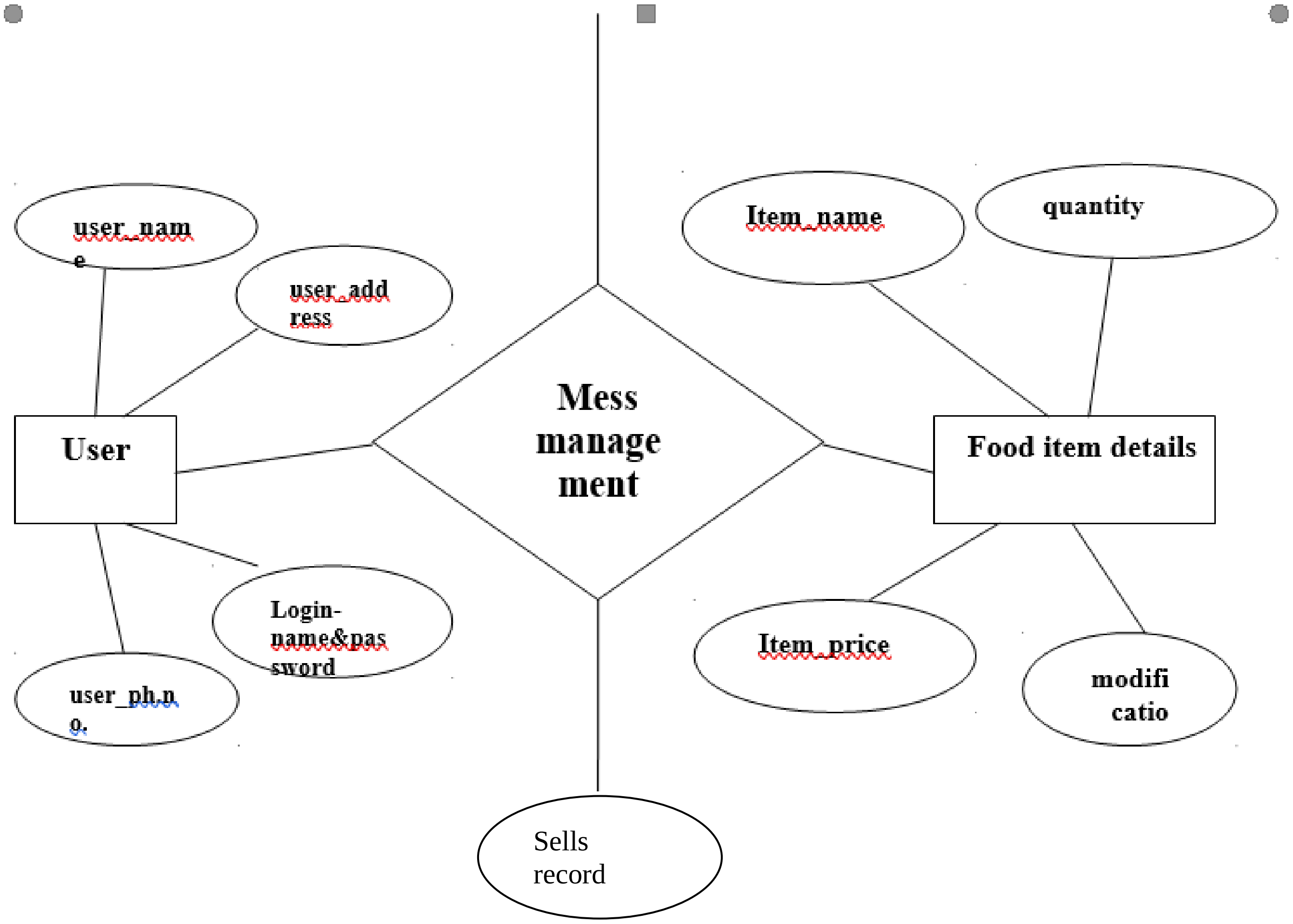
Zero Level DFD



First Level DFD



ER Diagram:-


user_nam
e
user_add
ress
User
Login-
name&pas
sword
user_ph.n
o.
Mess
manage
ment
Item_name
quantity
Food item details
Item_price
modifi
catio
Sells
record

Coding

```
//*************************************************************
*******
#include<iostream>
#include<fstream>
#include<windows.h>
#include<dos.h>
#include<stdio.h>
#include<cstdlib>
#include<string>
#include<conio.h>
using namespace std;
COORD coord={0,0};
void gotoxy(int x,int y)
{
    coord.X=x;
    coord.Y=y;

SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord)
;
}
int breakfastpr,lunchpr,dinnerpr;
int breakfastqty,lunchqty,dinnerqty;
void billpage();
void empjump();
void billjump();
void handlecust();
class login;
class store;

class order;
class billing;
class store
{
    public:
        string item;
        string name;
        int amt;
        int ch;
        int rate;
        int qty=0;
        void mnginvt();
        void storepageswitch();
        void price();
        void viewinvt();
        void quantity();
        void storepage()
        {
            storepageswitch();
        }
};
void store::storepageswitch()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"------------------------------------------"<<endl<<endl;
   cout<<"     PLEASE UPDATE THE PRICE AND QUANTITY
OF"<<endl;
   cout<<"     ITEMS IN MANAGE INVENTORY BEFORE
PROCEEDING"<<endl<<endl;
   cout<<"     1. MANAGE INVENTORY"<<endl;
   cout<<"     2. VIEW INVENTORY"<<endl;
   cout<<"     3. TAKE ORDER"<<endl;
   cout<<"     4. VIEW SALES RECORD"<<endl;
   cout<<"     5. EXIT"<<endl<<endl;
   cout<<"     ENTER CHOICE"<<endl;
   cout<<"     ";cin>>ch;
   while(ch!=1||ch!=2||ch!=3)
   {
      switch(ch)
      {
         case 1:
            mnginvt();
            break;
         case 2:
            viewinvt();
            break;
         case 3:
            handlecust();
            break;
         case 4:
            billjump();
            break;
         case 5:
            exit(0);
         default:
            cout<<endl<<"     INVALID CHOICE"<<endl<<endl;
            cout<<"     ENTER CHOICE"<<endl<<endl;
            cout<<"     ";cin>>ch;
      }
   }
}
void store::mnginvt()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     1. EDIT PRICE"<<endl;
   cout<<"     2. ENTER QUANTITY"<<endl<<endl;
   cout<<"     ENTER CHOICE"<<endl;
   cout<<"     ";cin>>ch;
   while(ch!=1||ch!=2)
   {
      switch(ch)
      {
         case 1:
            price();
            break;
         case 2:
            quantity();
            break;
         default:
            cout<<endl<<"     INVALID CHOICE"<<endl<<endl;
            cout<<"     ENTER CHOICE"<<endl<<endl;
            cout<<"     ";cin>>ch;
      }
   }
}
void store::price()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     CURRENT INVENTORY"<<endl;
   ifstream currprice("PRICE.txt");
   cout<<"     "<<"ITEM"<<" - "<<"PRICE"<<endl;
   cout<<"     ------------"<<endl;
   while(currprice>>item>>rate)
   {
      cout<<"     "<<item<<" - "<<rate<<endl;
   }
   currprice.close();
   remove("PRICE.txt");
   fstream editprice("PRICE.txt",ios::app);
   cout<<endl<<"     ENTER PRICE OF BREAKFAST"<<endl;
   cout<<"     ";cin>>breakfastpr;cout<<endl;
   editprice<<"BREAKFAST"<<' '<<breakfastpr<<endl;
   cout<<"     ENTER PRICE OF LUNCH"<<endl;
   cout<<"     ";cin>>lunchpr;cout<<endl;
   editprice<<"LUNCH"<<' '<<lunchpr<<endl;
   cout<<"     ENTER PRICE OF DINNER"<<endl;
   cout<<"     ";cin>>dinnerpr;cout<<endl;
   editprice<<"SOFTDRINK"<<' '<<dinnerpr<<endl;;
   editprice.close();

   system("CLS");cout<<endl;
   gotoxy(40,0);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     NEW INVENTORY"<<endl;
   ifstream viewprice("PRICE.txt");
   cout<<"     "<<"ITEM"<<" - "<<"PRICE"<<endl;
   cout<<"     ------------"<<endl;
   while(viewprice>>item>>rate)
   {
      cout<<"     "<<item<<" - "<<rate<<endl;
   }
   viewprice.close();
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
   cout<<"     ";getch();
   storepage();
}
```

```
void store::quantity()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     CURRENT INVENTORY"<<endl;
   ifstream currqty("QUANTITY.txt");
   cout<<"     "<<"ITEM"<<" - "<<"QUANTITY"<<endl;
   cout<<"     ---------------"<<endl;
   while(currqty>>item>>qty)
   {
      cout<<"     "<<item<<" - "<<qty<<endl;
   }
   currqty.close();
   remove("QUANTITY.txt");
   fstream editqty("QUANTITY.txt",ios::app);
   cout<<endl<<"     ENTER QUANTITY OF BREAKFAST"<<endl;
   cout<<"     ";cin>>breakfastqty;cout<<endl;
   editqty<<"BREAKFAST"<<' '<<breakfastqty<<endl;
   cout<<"     ENTER QUANTITY OF LUNCH"<<endl;
   cout<<"     ";cin>>lunchqty;cout<<endl;
   editqty<<"LUNCH"<<' '<<lunchqty<<endl;
   cout<<"     ENTER QUANTITY OP DINNER"<<endl;
   cout<<"     ";cin>>dinnerqty;cout<<endl;
   editqty<<"SOFTDRINK"<<' '<<dinnerqty<<endl;
   editqty.close();
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     NEW INVENTORY"<<endl;
   cout<<"     "<<"ITEM"<<" - "<<"QUANTITY"<<endl;
   cout<<"     --------------------"<<endl;
   ifstream viewqty("QUANTITY.txt");
   while(viewqty>>item>>qty)
   {
      cout<<"     ";cout<<item<<" - "<<qty<<endl;
   }
   viewqty.close();
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
   cout<<"     ";getch();
   storepage();
}
void store::viewinvt()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   cout<<"     PRICE OF ITEMS IN MESS & CANTEEN"<<endl;
   cout<<"     ITEM - PRICE"<<endl;
   cout<<"     ------------"<<endl;
   ifstream vwpr("PRICE.txt");
   while(vwpr>>item>>rate)
   {
      cout<<"     ";cout<<item<<" - "<<rate<<endl;
   }
   vwpr.close();
   cout<<endl<<"     QUANTITY OF ITEMS IN MESS &
CANTEEN"<<endl;
   cout<<"     ITEM - QUANTITY"<<endl;
   cout<<"     ---------------"<<endl;
   ifstream vwqt("QUANTITY.txt");
   while(vwqt>>item>>rate)
   {
      cout<<"     ";cout<<item<<" - "<<rate<<endl;
   }
   vwqt.close();
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
   cout<<"     ";getch();
   storepage();
}
class billing
{
   public:
      string name;
      int cost1,cost2,cost3,qty,totalcost;
      store s;
      int amt;
      char dec;
      void viewstat();
      void clearstat();
      billing()
      {
          cost1=0;
          cost2=0;
          cost3=0;
      }
      void bill1(int qty)
      {
          cost1=0;
          cost1=breakfastpr*qty;
      }
      void bill2(int qty)
      {
          cost2=0;
          cost2=lunchpr*qty;
      }
      void bill3(int qty)
      {
          cost3=0;
          cost3=dinnerpr*qty;
      }
      void bill()
      {
          system("CLS");
          gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
          gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
          totalcost=cost1+cost2+cost3;
          cout<<"     ENTER NAME OF CUSTOMER"<<endl;
```

```
      cout<<"     ";cin>>name;cout<<endl;
      system("CLS");
      gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
      gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
      cout<<"     BILL : PAY FOLLOWING AMOUNT "<<endl;
      cout<<"     CUSTOMER NAME : "<<name<<endl;
      cout<<"     TOTAL COST IS : "<<"$"<<totalcost<<endl;;
      cout<<"     THANK YOU FOR YOUR INTEREST"<<endl;
      fstream billmod("SALESRECORD.txt",ios::app);
      billmod<<name<<' '<<totalcost<<endl;
      billmod.close();
      cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
      cout<<"     ";getch();
      billpage();
    }
};
void billing::viewstat()
{
  system("CLS");
  gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
  gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
  ifstream viewsr("SALESRECORD.txt");
  cout<<"     NAME - AMOUNT"<<endl;
  cout<<"     -------------"<<endl;
  while(viewsr>>name>>amt)
  {
    cout<<"     "<<name<<" - "<<"$"<<amt<<endl;
  }
  viewsr.close();
  cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
  cout<<"     ";getch();
  s.storepage();
}
void billing::clearstat()
{
  system("CLS");
  gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
  gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
  cout<<"     CLEAR SALES RECORD [Y]"<<endl;
  cout<<"     ";cin>>dec;
  if(dec=='Y')
  {
    remove("SALESRECORD.txt");
    cout<<endl<<"     SALES RECORD CLEARED"<<endl;
    cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
    cout<<"     ";getch();
    remove("EMPLOYEE.txt");
    ifstream newsr("EMPLOYEE.txt");
    newsr.close();
    empjump();
  }
  else
  {
    cout<<endl<<"     SALES RECORD ARE NOT CLEARED"<<endl;
    cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
    cout<<"     ";getch();
    empjump();

  }
}
class order
{
  public:
    int tqty;
    string titem;
    billing b;
    int invtqty;
    string invtitem;
    char dec;
    int ch;
    int qty;
    void orderitem1();
    void orderitem2();
    void orderitem3();
    void orderpageswitch();
    void orderpage()
    {
      system("CLS");
      gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
      gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
      cout<<"     1. BREAKFAST"<<endl;
      cout<<"     2. LUNCH"<<endl;
      cout<<"     3. DINNER"<<endl;
      cout<<"     4. EXIT"<<endl<<endl;
      cout<<"     ENTER ITEM TO ORDER"<<endl;
      cout<<"     ";cin>>ch;
      orderpageswitch();
    }
};
void order::orderpageswitch()
{
  while(ch!=1||ch!=2||ch!=3||ch!=4)
  {
    switch(ch)
    {
      case 1:
          orderitem1();
          break;
      case 2:
          orderitem2();
          break;
      case 3:
          orderitem3();
          break;
      case 4:
          exit(0);
          break;
      default:
```

```
            cout<<endl<<"     INVALID CHOICE"<<endl;
            cout<<"     ENTER CHOICE"<<endl;
            cout<<"     ";cin>>ch;
      }
   }
}
void order::orderitem1()
{
   cout<<"     ENTER QUANTITY"<<endl;
   cout<<"     ";cin>>qty;cout<<endl;
   if(breakfastqty>=qty)
   {
      breakfastqty=breakfastqty-qty;
      ofstream temp1("temp1.txt");
      ifstream snackorder("QUANTITY.txt");
      while(snackorder>>titem>>tqty)
      {
         if(titem!="BREAKFAST")
         {
            temp1<<titem<<' '<<tqty<<endl;
         }
         else
         {
            temp1<<"BREAKFAST"<<' '<<breakfastqty<<endl;
         }
      }
      temp1.close();
      snackorder.close();
      remove("QUANTITY.TXT");
      rename("temp1.txt","QUANTITY.txt");
      b.bill1(qty);
      cout<<"     ORDER ANOTHER ITEM [Y]"<<endl;
      cout<<"     ";cin>>dec;
      if(dec=='Y')
      {
         orderpage();
      }
      else
      {
         b.bill();
      }
   }
   else
   {
      cout<<endl<<"     NOT AVAILABLE"<<endl;
      cout<<"     SELECT ANOTHER ITEM"<<endl;
      cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
      cout<<"     ";getch();
      system("CLS");
      gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
      gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
      ifstream orderout("QUANTITY.txt");
      cout<<"     ITEM - QUANTITY"<<endl;
      cout<<"     ---------------"<<endl;
      while(orderout>>invtitem>>invtqty)
```

```
        {
            cout<<"     "<<invtitem<<" - "<<invtqty<<endl;
        }
        orderout.close();
        cout<<endl<<"     PRESS ANY KEY FOR NEW ORDER"<<endl;
        cout<<"     ";getch();
        orderpage();
    }
}
void order::orderitem2()
{
    cout<<"     ENTER QUANTITY"<<endl;
    cout<<"     ";cin>>qty;cout<<endl;
    if(lunchqty>=qty)
    {
        lunchqty=lunchqty-qty;
        ofstream temp2("temp2.txt");
        ifstream lunchorder("QUANTITY.txt");
        while(lunchorder>>titem>>tqty)
        {
            if(titem!="LUNCH")
            {
                temp2<<titem<<' '<<tqty<<endl;
            }
            else
            {
                temp2<<"LUNCH"<<' '<<lunchqty<<endl;
            }
        }
        temp2.close();
        lunchorder.close();
        remove("QUANTITY.TXT");
        rename("temp2.txt","QUANTITY.txt");
        b.bill2(qty);
        cout<<"     ORDER ANOTHER ITEM [Y]"<<endl;
        cout<<"     ";cin>>dec;
        if(dec=='Y')
        {
            orderpage();
        }
        else
        {
            b.bill();
        }
    }
    else
    {
        cout<<endl<<"     NOT AVAILABLE"<<endl;
        cout<<"     SELECT ANOTHER ITEM"<<endl;
        cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
        cout<<"     ";getch();
        system("CLS");
        gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
        gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
        ifstream orderout("QUANTITY.txt");
        cout<<"     ITEM - QUANTITY"<<endl;
        cout<<"     ---------------"<<endl;
        while(orderout>>invtitem>>invtqty)
        {
            cout<<"     "<<invtitem<<" - "<<invtqty<<endl;
        }
        orderout.close();
        cout<<endl<<"     PRESS ANY KEY FOR NEW ORDER"<<endl;
        cout<<"     ";getch();
        orderpage();
    }
}
void order::orderitem3()
{
    cout<<"     ENTER QUANTITY"<<endl;
    cout<<"     ";cin>>qty;cout<<endl;
    if(dinnerqty>=qty)
    {
        dinnerqty=dinnerqty-qty;
        ofstream temp3("temp3.txt");
        ifstream sdorder("QUANTITY.txt");
        while(sdorder>>titem>>tqty)
        {
            if(titem!="SOFTDRINK")
            {
                temp3<<titem<<' '<<tqty<<endl;
            }
            else
            {
                temp3<<"SOFTDRINK"<<' '<<dinnerqty<<endl;
            }
        }
        temp3.close();
        sdorder.close();
        remove("QUANTITY.TXT");
        rename("temp3.txt","QUANTITY.txt");
        b.bill3(qty);
        cout<<"     ORDER ANOTHER ITEM [Y]"<<endl;
        cout<<"     ";cin>>dec;
        if(dec=='Y')
        {
            orderpage();
        }
        else
        {
            b.bill();
        }
    }
    else
    {
        cout<<endl<<"     NOT AVIALABLE"<<endl;
        cout<<"     SELECT ANOTHER ITEM"<<endl;
        cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
        cout<<"     ";getch();
        system("CLS");
        gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
```

```
TRICHY";
    gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
    ifstream orderout("QUANTITY.txt");
    cout<<"     ITEM - QUANTITY"<<endl;
    cout<<"     ---------------"<<endl;
    while(orderout>>invtitem>>invtqty)
    {
      cout<<"     "<<invtitem<<" - "<<invtqty<<endl;
    }
    orderout.close();
    cout<<endl<<"     PRESS ANY KEY FOR NEW ORDER"<<endl;
    cout<<"     ";getch();
    orderpage();
  }
}
class employee
{
  public:
    int ch,age;
    char name[50];
    long int sal;
    void addemp();
    void displayemp();
    void removeemp();
    void editemp();
    void emppageswitch();
    void emppage()
    {
      system("CLS");
      gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
      gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
      cout<<"     1. DISPLAY ALL EMPLOYEE DETAILS"<<endl;
      cout<<"     2. ADD NEW EMPLOYEE DETAILS"<<endl;
      cout<<"     3. REMOVE OLD EMPLOYEE DETAILS"<<endl;
      cout<<"     4. VIEW SALES RECORD"<<endl;
      cout<<"     5. CLEAR SALES RECORD"<<endl;
      cout<<"     6. STOREPAGE"<<endl;
      cout<<"     7. EXIT"<<endl<<endl;
      cout<<"     ENTER CHOICE"<<endl;
      cout<<"     ";cin>>ch;
      emppageswitch();
    }
};
void employee::emppageswitch()
{
  while(ch!=1||ch!=2||ch!=3||ch!=4||ch!=5)
  {
    switch(ch)
    {
      case 1:
        displayemp();
        break;
      case 2:
        addemp();
        break;
```

```
         case 3:
            removeemp();
            break;
         case 4:
            {
               billing b1;
               b1.viewstat();
            }
            break;
         case 5:
            {
               billing b2;
               b2.clearstat();
            }
         case 6:
            {
               store s;
               s.storepage();
            }
            break;
         case 7:
            exit(0);
            break;
         default:
            cout<<endl<<"     INVALID CHOICE"<<endl;
            cout<<"     ENTER CHOICE"<<endl;
            cout<<"     ";cin>>ch;
      }
   }
}
void employee::addemp()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"------------------"<<endl<<endl;
   ofstream newemployee("EMPLOYEE.txt",ios::app);
   cout<<"     ENTER NAME OF EMPLOYEE"<<endl;
   cout<<"     ";cin>>name;
   cin.sync();
   cout<<"     ENTER AGE OF EMPLOYEE"<<endl;
   cout<<"     ";cin>>age;
   cout<<"     ENTER SALARY OF EMPLOYEE"<<endl;
   cout<<"     ";cin>>sal;
   newemployee<<name<<' '<<age<<' '<<sal<<endl;
   newemployee.close();
   cout<<endl<<"     EMPLOYEE ADDED"<<endl;
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE";
   cout<<"     ";getch();
   emppage();
}
void employee::displayemp()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
```

```
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   ifstream employee("EMPLOYEE.txt");
   cout<<"     EMPLOYEE - AGE - SALARY"<<endl;
   cout<<"     -----------------------"<<endl;
   while (employee>>name>>age>>sal)
   {
      cout<<"     "<<name<<" - "<<age<<" - "<<sal<<endl ;
   }
   employee.close();
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
   cout<<"     ";getch();
   emppage();
}
void employee::removeemp()
{
   system("CLS");
   gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
   gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
   char tname[50];
   ifstream emp1("EMPLOYEE.txt");
   ofstream emp2("temp.txt");
   cout<<"     ENTER THE NAME OF EMPLOYEE WISH TO
REMOVE"<<endl;
   cout<<"     ";cin>>tname;
   while(emp1>>name>>age>>sal)
   {
      if(strcmp(name,tname)!=0)
      {
         emp2<<name<<' '<<age<<' '<<sal<<endl;
      }
   }
   emp1.close();
   emp2.close();
   remove("EMPLOYEE.txt");
   rename("temp.txt","EMPLOYEE.txt");
   cout<<endl<<"     EMPLOYEE REMOVED"<<endl;
   cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
   cout<<"     ";getch();
   emppage();
}
class login
{
  public:
     string pass="";
     int ch;
     char c;
     void loginpageswitch();
     void homepageswitch();
     void employeelogin();
     void ownerlogin();
     void emp();
     void own();
     void homepage()
     {
        system("CLS");
        gotoxy(40,1);cout<<"WELCOME TO MEGA MESS II & CANTEEN
SERVICE NIT TRICHY";
        gotoxy(40,2);cout<<"---------------------------------------------------
"<<endl<<endl;
        cout<<"     1. LOGIN"<<endl;
        cout<<"     2. EXIT"<<endl<<endl;
        cout<<"     ENTER CHOICE"<<endl;
        cout<<"     ";cin>>ch;
        homepageswitch();
     }
     void loginpage()
     {
        system("CLS");
        gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
        gotoxy(40,2);cout<<"----------------------------------------"<<endl<<endl;
        cout<<"     1. OWNER LOGIN"<<endl;
        cout<<"     2. EMPLOYEE LOGIN"<<endl;
        cout<<"     3. EXIT"<<endl<<endl;
        cout<<"     ENTER CHOICE"<<endl;
        cout<<"     ";cin>>ch;
        loginpageswitch();
     }
};
void login::homepageswitch()
{
   while(ch!=1||ch!=2)
   {
      switch(ch)
      {
      case 1:
         loginpage();
         break;
      case 2:
         exit(0);
         break;
      default:
         cout<<endl<<"     INVALID CHOICE"<<endl;
         cout<<"     ENTER CHOICE"<<endl;
         cout<<"     ";cin>>ch;
      }
   }
}
void login::loginpageswitch()
{
   while(ch!=1||ch!=2||ch!=3)
   {
      switch(ch)
      {
         case 1:
           ownerlogin();
           break;
         case 2:
           employeelogin();
           break;
         case 3:
            exit(0);
            break;
         default:
            cout<<endl<<"     INVALID CHOICE"<<endl;
            cout<<"     ENTER CHOICE"<<endl;
            cout<<"     ";cin>>ch;
      }
   }
}
void login::ownerlogin()
{
   while(pass!="MADJ@21")
   {
      pass="";
      cout<<endl<<"     ENTER OWNER PASSWORD"<<endl;
      cout<<"     ";c=_getch();
      while(c!=13)
      {
         pass.push_back(c);
         cout<<"*";
         c=getch();
      }
      if(pass=="NITTOWNER")
       {
          cout<<endl<<"     OWNER ACCESS GRANTED"<<endl;
          cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
          cout<<"     ";getch();
          own();
       }
      else
      {
         cout<<endl<<"     INVALID PASSWORD"<<endl;
      }
   }
}
void login::employeelogin()
{
   while(pass!="employee")
   {
      pass="";
      cout<<"     ENTER EMPLOYEE PASSWORD"<<endl;
      cout<<"     ";c=_getch();
      while(c!=13)
      {
         pass.push_back(c);
         cout<<"*";
         c=getch();
      }
      if(pass=="NITTEMP")
      {
         cout<<endl<<"     EMPLOYEE ACCESS GRANTED"<<endl;
         cout<<endl<<"     PRESS ANY KEY TO CONTINUE"<<endl;
         cout<<"     ";getch();
         emp();
      }
      else
```

```
        {
            cout<<endl<<"     INVALID PASSSWORD"<<endl;
        }
    }
}
void login::emp()
{
    store s;
    s.storepage();
}
void login::own()
{
    employee e;
    e.emppage();
}
void billpage()
{
    int ch;
    system("CLS");
    gotoxy(40,1);cout<<"MEGA MESS II & CANTEEN SERVICE NIT
TRICHY";
    gotoxy(40,2);cout<<"-----------------"<<endl<<endl;
    cout<<"     1. NEW ORDER"<<endl;
    cout<<"     2. EXIT"<<endl<<endl;
    cout<<"     ENTER CHOICE"<<endl;
    cout<<"     ";cin>>ch;
    while(ch!=1||ch!=2)
    {
        switch(ch)
        {
            case 1:
                {
                    order o;
                    o.orderpage();
                }
                break;
            case 2:
                exit(0);
                break;
            default:
                cout<<endl<<"     INVALID CHOICE"<<endl;
                cout<<"     ENTER CHOICE"<<endl;
                cout<<"     ";cin>>ch;
        }
    }
}
void handlecust()
{
    order o;
    o.orderpage();
}
void billjump()
{
    billing b;
    b.viewstat();
}
```

```
void empjump()
{
    employee e;
    e.emppage();
}
int main()
{
    system("MEGA MESS II & CANTEEN SERVICE NIT TRICHY
MANAGEMENT SYSYTEM");
    system("color 03");
    login l;
    l.homepage();
    return 0;
}

//**************************************************************
**
//                              END OF PROJECT
//**************************************************************
**
```

`

IMPLEMENTATION & RESULTS

Explanation of Key functions

1. Login

In this, existing user/administrator enters the user-name and password and access the main form. This contains user name and password as module.

2. Main Form

If administrator accesses this form using authentic login-name and password then the administrator has administrative power as add the product, create user account, add new branded computer system, add company, change the shop record . Administrator have full power But user have some restricted power only which is provided by administrator.

In this user can sell computer part/assemble computer system/branded computer system, update customer record. Using this the user/administrator can see the report of sold computer part, branded computer system in any time duration. In this there is option to exit.

3. Sell food items

In sell assembled mess management part module, user/administrator can sell available food items and modify stocks. In this ,the user/administrator enters the customer name, a and chooses if another food item add to sell according to request of customer. The user / administrator can chose any item and quantity, After selling, the user/administrator can print bill receipt.

4. Add items

In this, the administrator can add new food items by entering its name, price range. Here administrator can also see the available food items and its information. And the administrator can also delete any product.

5. Items Price Report:

This report shows the price of all available food items and its detail as item-name.

6. Sells Records

This section contains all records of previous sell with price also contain all previous billing report

Details of functions that are used in this project

1. billing();
2. employee();
3. gotoxy(int x,inty);
4. handlecust();
5. login();
6. order();
7. store();

1. Billing();

This function is created to calculate the final bill of the employee or customer the food he dined. The attributes of this function are-

- bill1(); this bill is generated for breakfast.
- bill2(); this bill is generated for lunch.
- bill3(); this bill is generated for dinner.
- clearstat(): this finction clear all previous calculation for another order.
- Billjump(): jump on the particular order bill.
- Billpage(): generate to final bill.

2. Employee();

Add/remove new/existing employees in the database.

The attributes of this function are-

- Addemp();adds the new employee.
- Removeemp();remove the existing employee.
- Editemp():edit emp details.
- Dispemp();displays employee details.

3. gotoxy(int x,inty);

- The gotoxy() funtion is used to simply move the cursor on your monitor screen wherever desired.
- It is declared in the "conio.h" header file.
- It takes 2 parameters, gotoxy(int x, int y). The 1st parameter specifies the x-co-ordinate on the screen, i.e., horizontal position(from left to right) on the screen. The 2nd parameter specifies the y-co-ordinate on the screen, i.e., Vertical position(from top to bottom) on the screen. Thus, gotoxy(1,1) tells the compiler to move the cursor to the left-top-most corner of the screen.

4. **Handlecust();**

This function is used to handle all the functionalities of the customer. This also manages the order, bill and other details of customers.

5. **Login();** This function is used to handle login requirements. The following are the attributes of this login function.

   - Employeelogin(); this attribute handle the validation of employee login.
   - Homepage(); this handles the home page functionalities .
   - Own();this attribute handle the validation of owner login.

6. **Handlecust();**

This function is used to handle all the functionalities of the customer. This also manages the order, bill and other details of customers.

7. **Login();**

This function is used in this source code for security purpose. For opening software either you should have either employee password or owner password

Attributes of this function are

1.void login::(emp);choose loging option
2.void login::employeelogin();emp login option
3.void login::homepage() go to login homepage
4.void login::homepageswitch() two switch option for emp or owner
5.void login::own() select owner login option
6.void login::ownerlogin() after entering pwd login to owner account

8. **Order()**

This function is used to take order from customer. Also used for billing process

Attributes of this function

1.void order::order1() take order of that store in fist attribute
2.void order::order2() take order of second
3.void order::order3() take order of third
4.void order::orderpage() this will take you order page
5.void order::orderpageswitch() this will switch order pages

9. **Store()**

This function is used for storing data of available food item quantity
Also price of given items and availability of items

Attributes of this function are

1.void store::mnginvt() this is used open store page option
2.void store::price() this is used to modify the price of items

3.void store::quantity() this is used to modify quantity of items
4.void store::storepage() this is used backward to store page
5.void store::storepageswitch() this is used to switching between storage page
6.void store::viewinvt() this is used to view all items and price

10.

TESTING & VALIDATION

Software testing is a criterion of software quality assurance and represents the ultimate review of specified designing and coding.

System testing reveals the presence of errors in the software developed.

Testing is the process of executing a program with the intent of finding an error.

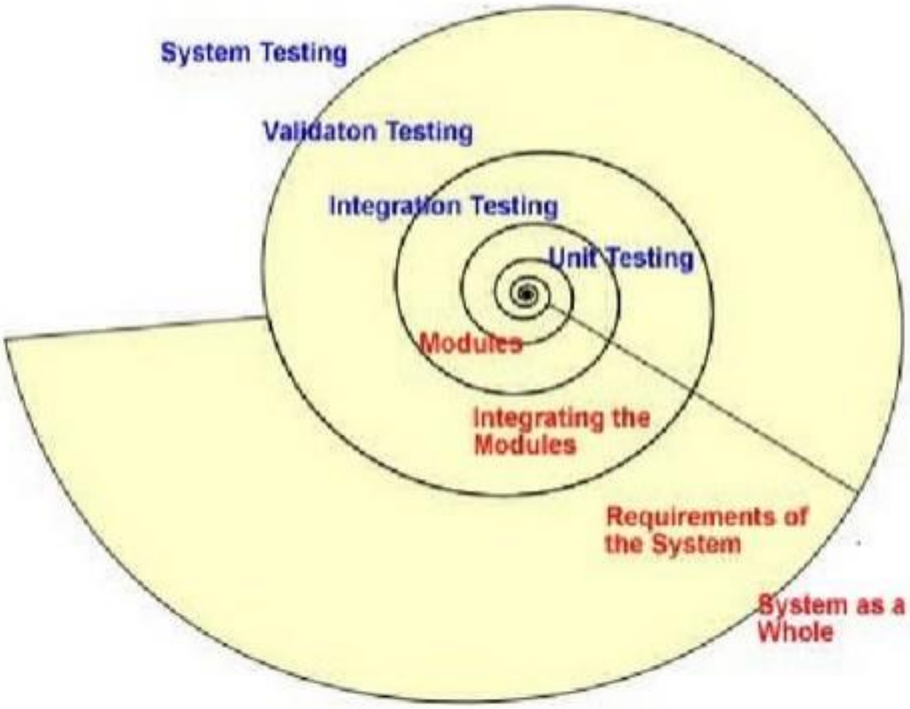
A good test is one that has a high probability for finding a yet undiscovered error.

A successful test is one that uncovers the hidden errors.

Characteristics of Testing:

Testing begins at the module level and works outward and towards the integration of the entire computer based system.

Different testing techniques are appropriate at different level of time.



CONCLUSION

Project Conclusion

Nothing is perfect in this world. So, we are also no exception. Although, we have tried our best to present the information effectively, yet, there can be further enhancement in the Application. We have taken care of all the critical aspects, which need to take care of during the development of the Project. Like the things this project also has some limitations and can further be enhances by someone, because there are certain drawbacks that do not permit the system to be 100% accurate.

Future enhancement

The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Improvements can be appended by changing the existing modules.
We think that not a single project is ever considered as complete forever because our mind is always thinking new and our necessities also are growing.

Our application Also, if you see at the first glance that you find it to be complete but we want to make it still mature and fully automatic.

As system is flexible you can generate more report and screen as and when required.

The system is modified in future as per the owner requirement.

In this system we can add more reports about users so more and more Information about computer parts.

Bibliography

The C++ Programming Language , 3rd Edition / Stroustrup Addison-Wesley.

C++ Primer, 3rd Edition / Lippman and Lajoie Addison-Wesley.
The Scott Meyers series, Addison-Wesley:

Effective C++ : 50 Specific Ways to Improve Your Programs and Designs, 2nd Edition

More Effective C++ : 35 New Ways to Improve Your Programs and Designs
and *Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library*

Beginners' C++ Course, 3rd Edition   Ron Wein