# MINI PROJECT

## 1 Objective

The objective of this project is to provide a interface where a hotels can expand their business, people can order food easily, delivery man can work efficiently with the help of this single software. This project helps people to efficiently order food within simple and minimal steps. The project provide an interface to customer and the hotels to interact in a productive manner and result in user friendly outcomes by improving the services regularly.

## 2 Function Description

### 2.1 random_number_generator

This function is used to generate random numbers which are used further. Other functions having purpose of login call this function to assign registration ID to the user. This function takes upper limit as a parameter to generate numbers within a certain limit and get desired result.

### 2.2 delay

This function is used to produce a delay of output during runtime of the program. This function uses the standard time.h library of c and runs a empty loop until the desired time duration is reached. The amount of time the function needs to delay is taken as a parameter in the function and used accordingly by other functions when they call it.

### 2.3 hotel_login

This function sets up the user to use the services of hotel in the program. This function takes the necessary information from the user about their hotel and store them in the user defined structure of hotel. This function also call the random number generator function to assign the registration ID to the hotel.

### 2.4 display_hotel

This function displays the entered information by the hotel manager about their hotel and their unique ID. This function also confirms the user that their registration is successful and they can further proceed with other services.

### 2.5 hotel_food

This function is used to let the hotel manager make their menu and provide their customer with the food they are offering. This function also lets the hotel to set price of their food.

## 2.6 display_menu

This function is used while the customer is ordering food to show him the menu. It assigns a code to each food and lets the user order their food by selecting the specific code of the respective food item.

## 2.7 customer_login

This function is another side of the project that is the customer side and it is used to register the customer with the services. It asks for necessary information like name, address and date of birth. This function also calls random number generator function to assign registration ID to the user.

## 2.8 display_customer

This function displays the entered information by the customer and their unique ID. This function also confirms the customer that their registration is successful and they can further proceed with other services like ordering food.

## 2.9 order

This function lets the customer create his order. Firstly the display menu function is called so that the user can decide what he needs to order and then the order is generated and the bill is shown to the user.

## 2.10 delivery_man_login

This is yet another section of the service where delivery man can register itself and provide delivery services. This function asks the delivery man to enter necessary information like name, vehicle number etc.

## 2.11 display_delivery_info

This function is used once the customer has orders his food and the food is on its way. This function also shows the delivery man information who is handling the order. This function calls the delay function and delays the runtime for 4 seconds to make the program more realistic and show that the the order is being processed.

## 2.12 ask_feedback

This function asks the customer for his feedback once the order is delivered. This covers the last aspect of the project that is the management side of the services. The feedback is used to improve the services. This function also calls the delay function to reflect the delivery time of the order and hence make the project more realistic.

# 3 Profiler Report

Profiler report provides the dynamic analysis of the program. It is useful for program optimisation and increase performance as it contains various aspects of program like space or time complexity of the program frequency and duration of function calls etc.

## 3.1 Flat Profile

The flat profile the amount of time program spent executing each function.

This section shows the flat profile report which was obtained using gprof profiler.

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ms/call  ms/call  name
100.00      0.22     0.22        2   110.00   110.00  delay
  0.00      0.22     0.00        2     0.00     0.00  random_number_generator
  0.00      0.22     0.00        1     0.00   110.00  ask_feedback
  0.00      0.22     0.00        1     0.00     0.00  customer_login
  0.00      0.22     0.00        1     0.00     0.00  delivery_man_login
  0.00      0.22     0.00        1     0.00     0.00  display_customer
  0.00      0.22     0.00        1     0.00   110.00  display_delivery_info
  0.00      0.22     0.00        1     0.00     0.00  display_hotel
  0.00      0.22     0.00        1     0.00     0.00  display_menu
  0.00      0.22     0.00        1     0.00     0.00  hotel_food
  0.00      0.22     0.00        1     0.00     0.00  hotel_login
  0.00      0.22     0.00        1     0.00     0.00  order

```

```
 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
    else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
    function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
    the function in the gprof listing. If the index is
    in parenthesis it shows where it would appear in
    the gprof listing if it were to be printed.
FF
Copyright (C) 2012-2017 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.
```

## 3.2 Call Graph

The call graph shows the call tree of the program. It shows which function are called by respective function and the other functions that calls the respective function.

The functions present above the indexed function(the function which is being analysed) in a section of the call graph are the functions that called the indexed function and the functions present below it are the functions called by the indexed function.

```
Call graph (explanation follows)


granularity: each sample hit covers 4 byte(s) for 4.55% of 0.22 seconds

index % time    self  children    called     name
                0.11    0.00       1/2          display_delivery_info [4]
                0.11    0.00       1/2          ask_feedback [3]
[1]    100.0    0.22    0.00       2          delay [1]
-----------------------------------------------
                                                <spontaneous>
[2]    100.0    0.00    0.22                  main [2]
                0.00    0.11       1/1          display_delivery_info [4]
                0.00    0.11       1/1          ask_feedback [3]
                0.00    0.00       1/1          hotel_login [13]
                0.00    0.00       1/1          display_hotel [10]
                0.00    0.00       1/1          hotel_food [12]
                0.00    0.00       1/1          customer_login [7]
                0.00    0.00       1/1          display_customer [9]
                0.00    0.00       1/1          order [14]
                0.00    0.00       1/1          delivery_man_login [8]
-----------------------------------------------
                0.00    0.11       1/1          main [2]
[3]     50.0    0.00    0.11       1          ask_feedback [3]
                0.11    0.00       1/2          delay [1]
-----------------------------------------------
                0.00    0.11       1/1          main [2]
[4]     50.0    0.00    0.11       1          display_delivery_info [4]
                0.11    0.00       1/2          delay [1]
-----------------------------------------------
                0.00    0.00       1/2          hotel_login [13]
                0.00    0.00       1/2          customer_login [7]
[6]      0.0    0.00    0.00       2          random_number_generator [6]
```

```
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[7]      0.0    0.00    0.00       1         customer_login [7]
                0.00    0.00       1/2           random_number_generator [6]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[8]      0.0    0.00    0.00       1         delivery_man_login [8]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[9]      0.0    0.00    0.00       1         display_customer [9]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[10]     0.0    0.00    0.00       1         display_hotel [10]
-----------------------------------------------
                0.00    0.00       1/1           order [14]
[11]     0.0    0.00    0.00       1         display_menu [11]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[12]     0.0    0.00    0.00       1         hotel_food [12]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[13]     0.0    0.00    0.00       1         hotel_login [13]
                0.00    0.00       1/2           random_number_generator [6]
-----------------------------------------------
                0.00    0.00       1/1           main [2]
[14]     0.0    0.00    0.00       1         order [14]
                0.00    0.00       1/1           display_menu [11]
```

```
This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.

Each entry in this table consists of several lines.  The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.
This line lists:
    index  A unique number given to each element of the table.
   Index numbers are sorted numerically.
   The index number is printed next to every function name so
   it is easier to look up where the function is in the table.

    % time This is the percentage of the `total' time that was spent
   in this function and its children.  Note that due to
   different viewpoints, functions excluded by options, etc,
   these numbers will NOT add up to 100%.

    self This is the total amount of time spent in this function.

    children This is the total amount of time propagated into this
   function by its children.

    called This is the number of times the function was called.
   If the function called itself recursively, the number
   only includes non-recursive calls, and is followed by
   a `+' and the number of recursive calls.

    name The name of the current function.  The index number is
   printed after it.  If the function is a member of a
   cycle, the cycle number is printed between the
   function's name and the index number.
```

```
For the function's parents, the fields have the following meanings:

     self This is the amount of time that was propagated directly
    from the function into this parent.

     children This is the amount of time that was propagated from
    the function's children into this parent.

     called This is the number of times this parent called the
    function `/' the total number of times the function
    was called.  Recursive calls to the function are not
    included in the number after the `/'.

     name This is the name of the parent.  The parent's index
    number is printed after it.  If the parent is a
    member of a cycle, the cycle number is printed between
    the name and the index number.

If the parents of the function cannot be determined, the word
`<spontaneous>' is printed in the `name' field, and all the other
fields are blank.
```

```
For the function's children, the fields have the following meanings:

 self This is the amount of time that was propagated directly
from the child into the function.

 children This is the amount of time that was propagated from the
child's children to the function.

 called This is the number of times the function called
this child `/' the total number of times the child
was called.  Recursive calls by the child are not
listed in the number after the `/'.

 name This is the name of the child.  The child's index
number is printed after it.  If the child is a
member of a cycle, the cycle number is printed
between the name and the index number.

If there are any cycles (circles) in the call graph, there is an
entry for the cycle-as-a-whole.  This entry shows who called the
cycle (as parents) and the members of the cycle (as children.)
The `+' recursive calls entry shows the number of function calls that
were internal to the cycle, and the calls entry for each member shows,
for that member, how many times it was called from other members of
the cycle.
FF
Copyright (C) 2012-2017 Free Software Foundation, Inc.
Index by function name

 [3] ask_feedback            [9] display_customer       [12] hotel_food
 [7] customer_login          [4] display_delivery_info  [13] hotel_login
 [1] delay                  [10] display_hotel          [14] order
 [8] delivery_man_login     [11] display_menu            [6] random_number_generator
```

# 4 Debugging Process

Debugging is an important aspect in project development as it is very useful process to identify errors and make the program work as we would like it.

The problem in the project was that while the customer was ordering food the first time he the command to read the respective food code didn't work normally.

```
PS C:\Users\Ansh\Desktop\3rd Sem\Programming Practices> ./a.exe


Welcome to ABC company


Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
1

Please register before you start using our services

Your details are
Name: asdf
Address: 12qwer
Your registration id is: 687

The menu is:
Code    Name
1       pizza

2       cola

Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): Enter the quantity you would like to buy:
```

As you can see that the program didn't ask for input for the food code.

After performing following debugging steps it was identified that the input buffer had
n in it which was preventing input to be taken, so to solve it a getchar was needed to be introduced, so that it could first remove the
n from input buffer and then the program would run normally.

These are the gdb steps

```
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "mingw32".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from C:\Users\Ansh\Desktop\3rd Sem\Programming Practices\a.exe...done.
(gdb) break 167
Breakpoint 1 at 0x4017e7: file project.c, line 167.
(gdb) run
Starting program: C:\Users\Ansh\Desktop\3rd Sem\Programming Practices/a.exe
[New Thread 8088.0x5194]
[New Thread 8088.0x5d8c]


Welcome to ABC company


Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
1

Please register before you start using our services

Enter your hotel details
Name: xyz hotel
Password(only in numerical digits): 1234
Confirm Password: 1234

You have successfully registered your hotel.
        xyz hotel
Here is your registration ID: 41

Enter the food items and its rate you would be serving (Enter X to stop entering itmes):
Food: pizza
Price: 320
Food: X
```

```
Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
2

Please register before you start using our services

Enter your details
Name: asdf
Address: 12qwerty
Date of birth: 15/34/5678


You registration is successfull! You can now use our services
Your details are
Name: asdf
Address: 12qwerty
Your registration id is: 687

The menu is:
Code    Name
1       pizza


Breakpoint 1, order () at project.c:167
167             printf("Enter the code of the item you would like to buy (enter -1 to stop orderi
(gdb) erff
Undefined command: "erff".  Try "help".
(gdb) next
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill
(gdb) next
170             if(food_code==-1){
(gdb) next
174             printf("Enter the quantity you would like to buy: ");
(gdb) next
Enter the quantity you would like to buy: 175           scanf("%d", &quantity_food);
(gdb) next
2
177             bill = bill + quantity_food*h1.price[food_code-1];
(gdb)
```

# 5 Project Source Code

```
/*
 *Mini Project
 *Food Management System
 */
```

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

struct customer
{
    char name[50];
    char address[50];
    char dob[10];
    int  id;
}c1;

struct hotel
{
    char name[100];
    int registeration_id;
    int password;
    char food[5][20];
    int price[5];
    int count_food;
}h1;

struct delivery
{
    char name[50];
    char vehicle_name[20];
    int vehicle_number;
}d1;

struct management
{
    int delivered_on_time;
    int packaged_properly;
    int complete_order_recieved;
    int bill_recieved;
}m1;

/*
 * This function generates random numbers within a limit using long
 int as a parameter for the the upper limit.
 */
int random_number_generator(long int limit){

    int temporary = (rand()%(limit+1));
    return temporary;
}
```

```
/*
 * This function is used to produce a delay of output during runtime
 of the program.
 * The amount of time the function needs to delay is taken as a
 parameter in the function.
 */
void delay(int number_of_seconds)
{

    // seconds are converted into milli seconds because clock
    measures time in milli seconds
    int milli_seconds = 1000 * number_of_seconds;

    clock_t start_time = clock();

    // empty loop until corresponding time is elapsed
    while (clock() < start_time + milli_seconds){}
}


/*
 * This function lets the hotel register itself and start using
 other services
 */
void hotel_login(){

    printf("\nEnter your hotel details\n");
    printf("Name: ");
    // getchar is used to empty the input stream containing \n from
    previous integer input.
    getchar();
    // fgets is used to take input of the string using standard
    input
    fgets(h1.name,100,stdin);

    printf("Password(only in numerical digits): ");
    scanf("%d", &h1.password);
    // this loop is used to confirm the entered password
    while(1){
        int temp;
        printf("Confirm Password: ");
        scanf("%d", &temp);
        if (temp==h1.password)
        {
            break;
```

```
        }

    }

    // assigning ID to hotel using random number generator.
    h1.registeration_id = random_number_generator(999999999);

}

/*
 * This function displays the entered information by user
 */
void display_hotel(){

    printf("\nYou have successfully registered your hotel.\n");
    printf("\t%s", h1.name);
    printf("Here is your registration ID: %d\n", h1.registeration_id);

}


/*
 * This function helps the hotel to create thier menu and list the
 prices of food.
 */
void hotel_food(){

    printf("\nEnter the food items and its rate you would be serving
    (Enter X to stop entering itmes):\n");
    int i=0;

    // This loop is used to input the food and its price for the
    menu
    while(i<5){
        char current_food[20];
        printf("Food: ");
        // getchar is used to empty the input stream containing \n
        from previous integer input.
        getchar();
        fgets(current_food, 20, stdin);

        if(current_food[0]=='X'){
            break;
        }
        else{
            strcpy(h1.food[i], current_food);
```

```
            printf("Price: ");
            scanf("%d", &h1.price[i]);
            i++;
        }

    }
    h1.count_food = i;
    printf("\n");

}


/*
 * This function is used to display the menu to customer while he is
 ordering food
 */
void display_menu(){

    printf("\nThe menu is:\n");
    printf("Code\tName\n");
    for(int index=0; index<h1.count_food; index++){
        printf("%d\t%s\n", index+1, h1.food[index]);
    }


}


/*
 * This function lets the customer register and use the services.
 */
void customer_login(){

    printf("\nEnter your details\n");
    printf("Name: ");
    getchar();
    fgets(c1.name,50,stdin);

    printf("Address: ");
    fgets(c1.address, 50, stdin);

    printf("Date of birth: ");
    fgets(c1.dob, 10, stdin);

    c1.id = random_number_generator(4444);
    printf("\n");
```

```
}

/*
 * This function displays the information entered by the customer.
 */
void display_customer(){

    printf("\nYou registration is successfull! You can now use our
    services\n");
    printf("Your details are\n");
    printf("Name: %s", c1.name);
    printf("Address: %s", c1.address);
    printf("Your registration id is: %d\n", c1.id);

}

/*
 * This function lets the user order food
 */
void order(){

    display_menu();

    // the bill variable starts with zero and increases as the
    customer orders
    int bill = 0;

    // loop to take the order from customer
    while(1){

        int food_code, quantity_food;
        printf("Enter the code of the item you would like to buy
        (enter -1 to stop ordering and display your bill): ");
        // getchar is used to empty the input stream containing \n
        from previous integer input.
        getchar();
        scanf("%d", &food_code);

        if(food_code==-1){
            break;
        }

        printf("Enter the quantity you would like to buy: ");
        scanf("%d", &quantity_food);

        bill = bill + quantity_food*h1.price[food_code-1];
```

```
    }

    printf("\nYour bill is %d rupees.\n", bill);


}


/*
 * This function lets the delivery man register.
 */
void delivery_man_login(){

    printf("\nEnter your information\n");
    printf("Name: ");
    // getchar is used to empty the input stream containing \n from
    previous integer input.
    getchar();
    fgets(d1.name, 50, stdin);
    printf("Vehicle name: ");
    fgets(d1.vehicle_name, 20, stdin);
    printf("Vehicle number: ");
    scanf("%d", &d1.vehicle_number);

}


/*
 * This function displays the delivery information to the customer
 */
void display_delivery_info(){

    // delay is used to show that the order is being prepared
    delay(4);
    printf("\n\nYour order is on its way\n");
    printf("Delivery Man information is \n");
    printf("Name: %s", d1.name);
    printf("Vehicle name: %s", d1.vehicle_name);
    printf("Vehicle mnumber: %d", d1.vehicle_number);
}


/*
 * this function asks the user for feedback once the order is
 complete.
 */
void ask_feedback(){
```

```
    // delay is being used to show that the order is on its way
    delay(6);
    printf("\n\nYour order is delivered\n");
    printf("\nPlease provide your feedback (enter 1 for satisfied
    and 0 for not satisfied): \n");
    printf("1. Did you recieved your order on time: ");
    scanf("%d", &m1.delivered_on_time);
    printf("2. Did you recieved your order packaged properly: ");
    scanf("%d", &m1.packaged_properly);
    printf("3. Was the order you recived correct and complte: ");
    scanf("%d", &m1.complete_order_recieved);
    printf("4. Did you recieved bill with your order: ");
    scanf("%d", &m1.bill_recieved);

    printf("\nThank you for providing feedback! Our management team
    will positively work to improve our services based on your
    feedback.\n");
}



int main(){


    printf("\n\nWelcome to ABC company\n\n");

    // the loop that controls the flow of whole program
    while(1){

        printf("\nAre you a\n\tHotel Manager (enter 1)\n\tCustomer
        (enter 2)\n\tDelivery Man (enter 3)\n\tEnter -1 to exit\n");
        int field_selector;
        scanf("%d", &field_selector);

        if(field_selector==1){

            printf("\nPlease register before you start using our
            services\n");


            hotel_login();
            display_hotel();
            hotel_food();

        }
        else if(field_selector==2){
```

```
            printf("\nPlease register before you start using our
            services\n");

            customer_login();
            display_customer();
            order();
            printf("Your order will be delivered soon...");
            display_delivery_info();
            ask_feedback();

        }
        else if(field_selector==3){

            delivery_man_login();

        }
        else if(field_selector==-1){
            printf("Thank you for using our services.");
            break;
        }

    }


    return 0;
}
```

# 6 Code Output

The output of the program is as follows:

```
Welcome to ABC company


Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
1

Please register before you start using our services

Enter your hotel details
Name: XYZ Hotel
Password(only in numerical digits): 1234
Confirm Password: 1235
Confirm Password: 1234

You have successfully registered your hotel.
        XYZ Hotel
Here is your registration ID: 41
```

```
Enter the food items and its rate you would be serving (Enter X to stop entering itmes):
Food: Pizza
Price: 140
Food: Burger
Price: 60
Food: Pasta
Price: 100
Food: Coca-Cola
Price: 50
Food: X
```

```
Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
3

Enter your information
Name: Delivery Man 1
Vehicle name: Jupiter ZX
Vehicle number: 3549
```

```
Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
2

Please register before you start using our services

Enter your details
Name: Ansh Jain
Address: 23-B abc Nagar
Date of birth: 16/02/2004


You registration is successfull! You can now use our services
Your details are
Name: Ansh Jain
Address: 23-B abc Nagar
Your registration id is: 687
```

```
The menu is:
Code    Name
1       Pizza

2       Burger

3       Pasta

4       Coca-Cola

Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): 1
Enter the quantity you would like to buy: 1
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): 2
Enter the quantity you would like to buy: 2
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): 1
Enter the quantity you would like to buy: 0
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): 3
Enter the quantity you would like to buy: 1
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): 4
Enter the quantity you would like to buy: 2
Enter the code of the item you would like to buy (enter -1 to stop ordering and display your bill): -1

Your bill is 460 rupees.
Your order will be delivered soon...

Your order is on its way
Delivery Man information is
Name: Delivery Man 1
Vehicle name: Jupiter ZX
Vehicle mnumber: 3549
```

```
Please provide your feedback (enter 1 for satisfied and 0 for not satisfied):
1. Did you recieved your order on time: 1
2. Did you recieved your order packaged properly: 1
3. Was the order you recived correct and complte: 1
4. Did you recieved bill with your order: 1

Thank you for providing feedback! Our management team will positively work to improve our services based on your feedback.

Are you a
        Hotel Manager (enter 1)
        Customer (enter 2)
        Delivery Man (enter 3)
        Enter -1 to exit
-1
Thank you for using our services.
PS C:\Users\Ansh\Desktop\3rd Sem\Programming Practices>
```