



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering
Bachelor of Technology
Computer Science and Engineering
Spec. in Information Security

DATA STRUCTURES AND ALGORITHM

J-COMPONENT

Project Final Report

Title:

APARTMENT MANAGEMENT SYSTEM

Submitted by

17BCI0152- M. CHIRANJEEVI SHRIRAM KRISHNA

G1 SLOT (L41+42)

Under the guidance of

Prof. M.P GOPINATH

INTRODUCTION

The main aim of the project is to provide utility to maintain day-to-day operations of apartment or gated community. This will help us to maintain all the track record of sales, purchase of flats i.e., Bidding for unsold flats/new flats and complaint box etc. This software gives us the total information about the apartment, owners, tenants, working staff, management staff etc. It also provides different logins to users, owners, tenants and workers.

MOTIVATION:

- ❖ Since we are moving more towards digital world, it is very convenient for the users to access through internet and finish their work.
- ❖ Since we live in very large community sized Apartments it is very difficult for us to contact or openly talk about the complaints or the problems.
- ❖ To erase all these issues I have come upon with an idea called “APARTMENT MANAGEMENT SYSTEM” to develop the real estate and other financial growth.

APPLICATIONS

- ❖ Bidding can be made by the users, for the flats present in the apartment.
- ❖ Complaint Box is been introduced where users can register a problem.
- ❖ Provides information about the administrator and other apartment staff.
- ❖ It provides flexibility for the tenants to pay their rents through this online web application.

DATA STRUCTURES USED

- QUEUES
- STACKS
- LINKED LIST
- STRUCTURES

DETAILED EXPLANATION OF THE APPLICATION PROVIDED BY THIS PROJECT

❖ Details:-

- It is very essential for the whole community to know about each other and other working staff.
- So, I have added an additional feature where we can find the details of all the members present in the apartment, which makes it very easy for owners to make contacts to their tenants and other owners and working staff also.
- To achieve this I am using Linked List, where I am going to create a link to each and every detail of a particular member.
- Example: If we click on a Flat No., then we can find out the owner name, tenant name(if rented), house details, contact details etc.

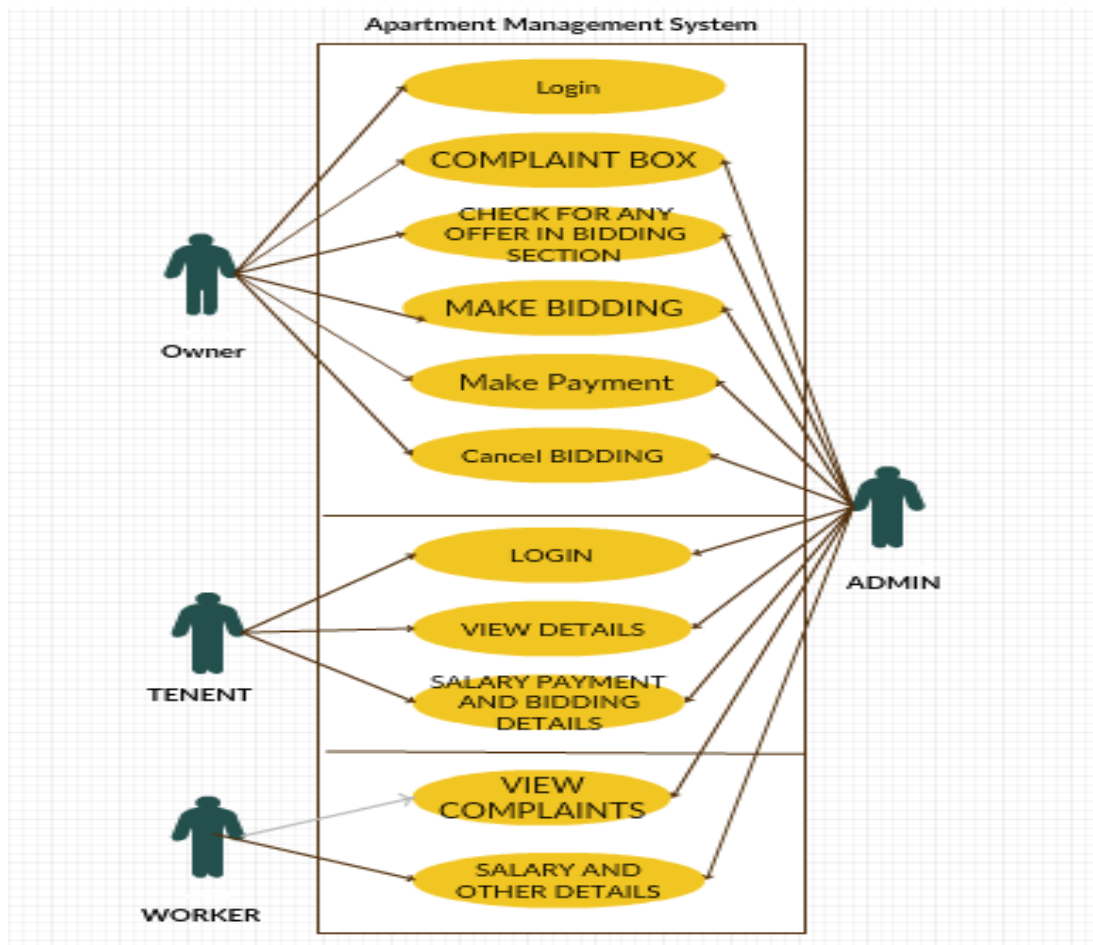
❖ BIDDING:

- Bidding is also useful for the users to buy their flats sitting in their flats itself.
- Since bidding manually takes a lot of money to establish a stage and to gather all the participants and to organise. It takes a lot of time, money, and labour.
- So, we implemented BIDDING in this project to make it convenient for the users.
- For maintaining this bidding I am using STACK.
- The reason I am using stack is initially a bidding amount is announced and the users can participate and enrol by paying higher amount than the initial and it gets added in the stack.
- After a certain time, the person who is above the stack will be announced as the winner.(since he has given the maximum amount)

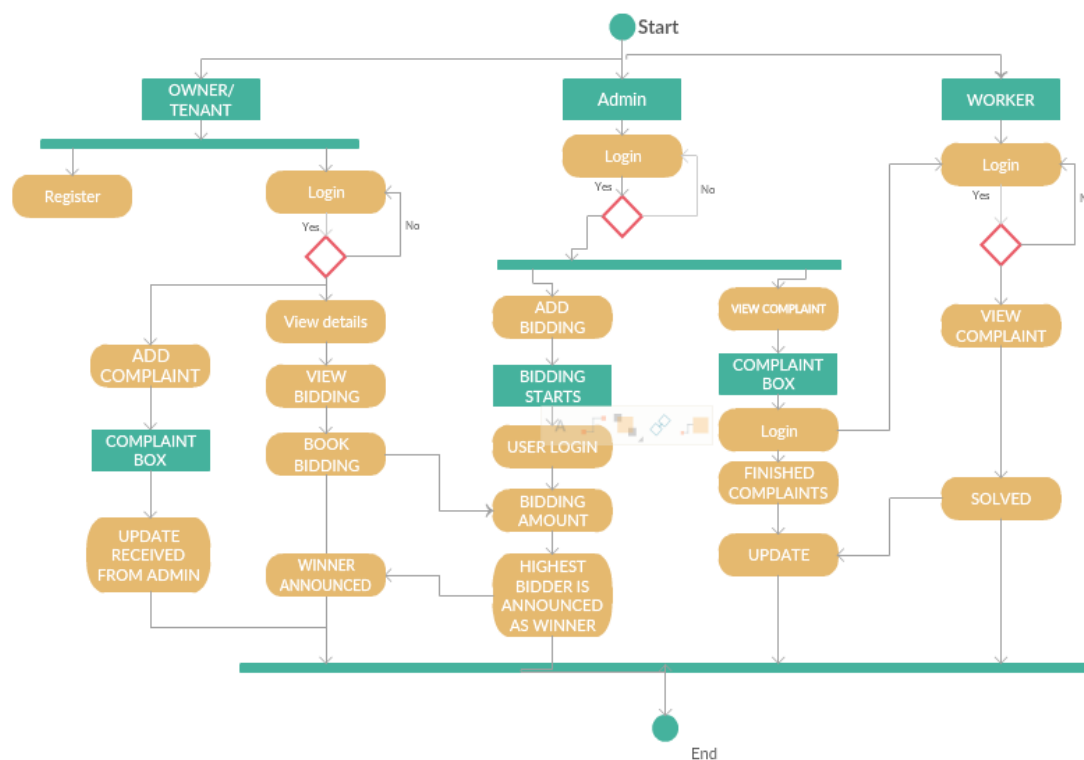
❖ COMPLAINT BOX:

- This is set up to make it comfortable for the user to file a complaint easily to the higher authority.
- This is done using Queue Data Structure.
- The complaint which enters 1st need to be solved 1st irrespective to anything.
- This provides easy and quick solution for the problems existing in the Apartment.

USE CASE DIAGRAM:



FLOWCHART:



SOURCE CODE:

```
#include<iostream>
#include<stdlib.h>
#include<fstream>
#include<math.h>
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<conio.h>
#include<iomanip>
#include<string>
#include<cctype>
#define int_value 5000
using namespace std;
class complaint
{
public:
    void customer_StoreComplaint();
    void admin_menu();
    void create_complaint();
    void cancel_complaint();
    void view_complaint();
    void exit();
    complaint();
};
complaint::complaint ()
{
}
struct node
{
    int complaint_number;
    string customerName;
```

```

        string date;

        string x;

        node *prev;

        node *next;

        node *link;

    }*q, *temp;

    node *start_ptr = NULL;

    node *head = NULL;

    node *last = NULL;

    fstream userfile("myuser.txt",ios::app);

    fstream pass("mypass.txt",ios::app);

    char username[15],password[20],full_name[30],phone_no[10];

    #define int_value 5000

    int top=0,stac[100];

    string name;

    void view_initial()

    {

    cout<<"\nthe initial value is "<<int_value;

    }

    void push()

    {

        stac[0]=int_value;

        cout<<"\nThe Base value is "<<stac[top]<<endl;

        int value;

        c:

        cout<<"\nEnter Name:";

        cin>>name;

        cout<<"\nEnter the value you want to bid(Greater than "<<stac[top]<<")"<<endl;

        cin>>value;

        if(top<100 && value>stac[top])

        {

            stac[++top]=value;

```

```

    }
    else if(value<=stac[top])
    {
        cout<<"\nThe bidding amount should be greater than base amount "<<endl;
        goto c;
    }
    else
    {
        cout<<"\nBidding Expired"<<endl;
    }
    cout<<"Your bidding amount is"<<value<<endl;
}

void bid()
{
    push();
}

void pop()
{
    cout<<"The winner of the BID is "<<name<<"\nValue is:"<<stac[top]<<endl;
}

void bidding()
{
    int c;
    do
    {
        system("cls");
        cout<<"\n";
        cout<<"\t\t===== \n";
        cout<<"\t\t  Apartment Management System - Bidding   \n";
        cout<<"\t\t===== \n\n\n";

        cout<<"\t\t----- \n";
    }
    while(c<10);
}

```

```

        cout<<"\t\t\t1. VIEW base value \t\t\t\n";

        cout<<"\t\t-----\n";

        cout<<"\t\t-----\n";

        cout<<"\t\t\t2. BID YOUR AMOUNT \t\t\t\n";

        cout<<"\t\t-----\n";

        cout<<"\t\t-----\n";

        cout<<"\t\t\t3. SHOW HIGGEST BIDDER\t\t\t\n";

        cout<<"\t\t-----\n\n";

        cout<<"\t\t\t4. EXIT\t\t\t\n";

        cout<<"\t\t-----\n\n";

        cout<<"Enter your choice:";

cin>>c;

switch(c)
{
case 1:
{
    view_initial();

    system("pause");

    break;
}

case 2:
{
    bid();

    system("pause");

    break;
}

case 3:
{
    pop();

    system("pause");

    break;
}
}

```



```

case 4:
{
    goto a;
    break;
}
}
}
while(c!=4);

    a://goto
    cout<<"thank you"<<endl;
    system ("PAUSE");
}
void details()
{
    FILE *fp, *ft;
    char another, choice;
    struct student
    {
        char first_name[50], last_name[50];
        long long int phoneno;
        int flat_no;
    };
    struct student e;
    char xfirst_name[50], xlast_name[50];
    int xphoneno[10],xflat_no[3];
    long int rectime;
    fp=fopen("details.txt","rb+");
    if (fp == NULL)
    {
        fp = fopen("users.txt","wb+");
        if (fp==NULL)
        {

```

```
puts("Cannot open file");  
return ;  
  
}  
  
}   
  
reccsize = sizeof(e);  
  
do  
{  
  
system("cls");  
  
cout << "\t\t===== APARTMENT MANAGEMENT SYSTEM =====";  
cout <<"\n\n                               ";  
cout << "\n\n";  
  
cout << "\n \t\t\t 1. Add   Details";  
cout << "\n \t\t\t 2. View  Details";  
cout << "\n \t\t\t 3. Modify Details";  
cout << "\n \t\t\t 4. Delete Details";  
cout << "\n \t\t\t 5. Exit   Details search!!!";  
cout << "\n\n";  
  
cout << "\t\t\t Enter Your Choice :";  
  
fflush(stdin);  
  
choice = getchar();  
  
switch(choice)  
{  
  
case '1':  
  
fseek(fp,0,SEEK_END);  
  
another ='Y';  
  
while(another == 'Y' || another == 'y')  
{  
  
system("cls");  
  
cout << "Enter the Firt Name : ";  
  
cin >> e.first_name;  
  
cout << "Enter the Last Name : ";  
  
cin >> e.last_name;
```

```

        cout << "Enter the phone no.  : ";
        cin >> e.phoneno;
        cout<<"Enter the flat no.  : ";
        cin >> e.flat_no;
        fwrite(&e,recsize,1,fp);
        cout << "\n Add Another Record (Y/N) ";
        fflush(stdin);
        another = getchar();
    }
    break;
case '2':
    system("cls");
    rewind(fp);
    cout << "==== View the Records in the Database ====";
    cout << "\n";
    while (fread(&e,recsize,1,fp) == 1)
    {
        cout << "\n";
        cout <<"\n" << e.first_name << setw(10) << e.last_name;
        cout << "\n";
        cout <<"\n" <<e.phoneno <<  setw(8) << e.flat_no;
    }
    cout << "\n\n";
    system("pause");
    break;

case '3' :
    system("cls");
    another = 'Y';
    while (another == 'Y' || another == 'y')
    {
        cout << "\n Enter the last name of the student : ";

```

```

    cin >> xlast_name;
    rewind(fp);
    while (fread(&e,recsize,1,fp) == 1)
    {
        if (strcmp(e.last_name,xlast_name) == 0)
        {
            cout << "Enter new the Firt Name : ";
            cin >> e.first_name;
            cout << "Enter new the Last Name : ";
            cin >> e.last_name;
            cout << "Enter new the phone number  : ";
            cin >> e.phoneno;
            cout << "Enter new the flat number  : ";
            cin >> e.flat_no;
            fseek(fp, - recsize, SEEK_CUR);
            fwrite(&e,recsize,1,fp);
            break;
        }
        else
            cout<<"record not found";
    }
    cout << "\n Modify Another Record (Y/N) ";
    fflush(stdin);
    another = getchar();
}
break;
case '4':
    system("cls");
    another = 'Y';
    while (another == 'Y' || another == 'y')
    {
        cout << "\n Enter the last name of the student to delete : ";

```

```

        cin >> xlast_name;

        ft = fopen("temp.dat", "wb");

        rewind(fp);

        while (fread (&e, recsize, 1, fp) == 1)

            if (strcmp(e.last_name, xlast_name) != 0)

                {

                    fwrite(&e, recsize, 1, ft);

                }

        fclose(fp);

        fclose(ft);

        remove("details.txt");

        rename("temp1.dat", "details.txt");

        fp = fopen("details.txt", "rb+");

        cout << "\n Delete Another Record (Y/N) ";

        fflush(stdin);

        another = getchar();

    }

    break;

case '5':

    fclose(fp);

    cout << "\n\n";

    cout << "\t\t\t\t\t THANK YOU FOR USING THIS SOFTWARE";

    cout << "\n\n";

    goto a;

    break;

}

}

while(choice != 5);

a: // goto

    cout << "thank you" << endl;

    system ("PAUSE");

```

```

    return ;
}
void complaint1()
{
    complaint admin;
    int option;
    do
    {
        cout<<"\n";
        cout<<"\t\t===== \n";
        cout<<"\t\t Apartment Management System - Complaint Box \n";
        cout<<"\t\t===== \n\n";
        cout<<"\t\t----- \n";
        cout<<"\t\t\t1. ADD complaint \t\t\t\n";
        cout<<"\t\t----- \n";
        cout<<"\t\t----- \n";
        cout<<"\t\t\t2. Admin Login \t\t\t\n";
        cout<<"\t\t----- \n";
        cout<<"\t\t----- \n";
        cout<<"\t\t\t3. EXIT \t\t\t\t\n";
        cout<<"\t\t----- \n\n";
        cout<<"\t\tEnter your choice: ";
        cin>>option;
        switch(option)
        {
            case 1:
                admin.customer_StoreComplaint();
                break;
            case 2:
                admin.admin_menu();
                break;
            case 3:

```

```

        admin.exit();

        goto a;

        break;

    }

}while(option!=3);

a://goto

cout<<"thank you"<<endl;

system ("PAUSE");

} //end main function

void complaint::customer_StoreComplaint()
{

    system("cls");

    cout<<"\n";

    cout<<"\t\t =====\n";

    cout<<"\t\t Apartment Management System - Complaint Box \n";

    cout<<"\t\t =====\n\n\n";

    cout <<"\nAdd Your Complaint Details\n";

    cout <<"_____ \n\n";

    node *temp;

    temp = new node;

    cout << "Type Complaint no: ";

    cin >> temp->complaint_number;

    cout<< "Enter Your Name: ";

    cin.ignore();

    getline(cin, temp->customerName);

    cout<<"Enter Date : ";

    cin>>temp->date;

    cout << "Complaint Description:( 1000 words maximum ) \n";

    cin.ignore();

    getline(cin, temp->x);

    cout<<"=====
===== "<<endl;

```

```

        cout << "Complaint added Successfully"<<endl;
        cout<<"=====
===== "<<endl;

        system ("PAUSE");

        temp->next=NULL;

        if(start_ptr!=NULL)
        {

            temp->next=start_ptr;

        }

        start_ptr=temp;

        system("cls");
    }

    void complaint::admin_menu()
    {

        complaint admin;

        int menu;

        string userName, userPassword;

        system("cls");

        cout<<"\n";

        cout<<"\t\t===== \n";

        cout<<"\t\t Apartment Management System - Complaint Box  \n";

        cout<<"\t\t===== \n\n\n";

        cout<<"\t\t----- \n";

        cout<<"\t\t\t Administrator / Staff Login  \n";

        cout<<"\t\t----- \n\n";

        cout << "\n\n\tPlease enter your user name: ";

        cin >> userName;

        cout << "\n\n\tPlease enter your user password: ";

        cin >> userPassword;


        if (userName == "user" && userPassword == "pass")
        {

            do

```



```

{
system("cls");
cout<<"\n";
cout<<"\t\t===== \n";
        cout<<"\t\t Apartment Management System - Complaint Box \t\t\n";
        cout<<"\t\t
===== \n\n\n";

        cout<<"\t\t----- \n";
        cout<<"\t\t||t1. Create a Complaint \t\t\t ||\n";
        cout<<"\t\t||t2. View Complaint\t\t\t ||\n";
        cout<<"\t\t||t3. Cancel a Complaint \t\t\t ||\n";
        cout<<"\t\t||t4. Logout\t\t\t\t ||\n";
        cout<<"\t\t||t5. Exit \t\t\t\t ||\n";
        cout<<"\t\t----- \n";
        cout<<"Enter choice: ";
        cin>>menu;
        switch (menu)
        {
                case 1:
                {
                        admin.create_complaint();
                        break;
                }
                case 2:
                {
                        admin.view_complaint();
                        system("PAUSE");
                        break;
                }
                case 3:
                {
                        admin.cancel_complaint();
                        system("PAUSE");

```

```

                break;
            }
        case 4:
        {
            cout<<"You are Logged Out...!\n\n\n";
            system ("PAUSE");
            admin.admin_menu();
            break;
        }
        case 5:
        {
            goto b;
            break;
        }
    }
    }while(menu!=5);

    b:
    cout<<"thank you"<<endl;
    system ("PAUSE");

}

else
{
    cout << "\n\n\tInvalid login attempt. Please try again.\n" << "\n";
    system("PAUSE");
    system ("cls");
    admin.admin_menu();
}

}

void complaint::create_complaint()
{
    system("cls");
    cout<<"\n";

```

```

cout<<"\t\t =====\n";
cout<<"\t\t Apartment Management System - Complaint Box \n";
cout<<"\t\t =====\n\n";
cout <<"\nAdd Complaint Details of Customer\n";
cout <<"_____ \n\n";
node *temp;
temp = new node;
cout << "Type Complaint no: ";
cin >> temp->complaint_number;
cout<< "Enter Customer Name: ";
cin.ignore();
getline(cin, temp->customerName);
cout<<"Enter Date : ";
cin>>temp->date;
cout << "Complaint Description:";
cout<<"( 1000 words maximum ) \n";
cin.ignore();
getline(cin, temp->x);

cout<<"=====
===== "<<endl;

cout << "Complaint added Successfully"<<endl;

cout<<"=====
===== "<<endl;

cout << "Go to Print Complaint to print the Complaint"<<endl;

cout<<"=====
===== "<<endl;

system ("PAUSE");
temp->next=NULL;
if(start_ptr!=NULL)
{
    temp->next=start_ptr;

```

```

    }

    start_ptr=temp;

    system("cls");
}

void complaint::view_complaint()
{
    int num;
    bool found;
    system("cls");
    node *temp;
    temp=start_ptr;
    found = false;
    cout<<"\n";
    cout<<"\t\t===== \n";
    cout<<"\t\t Apartment Management System - Complaint Box \n";
    cout<<"\t\t===== \n\n";
    cout<<" Enter the Complaint Number To Look into The Complaint Details\n";
    cin>>num;
    cout<<"\n";
    cout<<"-----" <<endl;
    cout <<"\t\tHere is the Complaint list\n";
    cout<<"-----" <<endl;
    if(temp == NULL)
    {
        cout <<"\t\tThere is no Complaint to show\n\t\t\tSo The List is Empty\n\n";
    }
    while(temp !=NULL && !found)
    {
        if (temp->complaint_number==num)
        {
            found = true;
        }
    }
}

```

```

        else
        {
            temp = temp -> next;
        }
    if (found)
    {
        cout << "Complaint Number : " << temp->complaint_number;
        cout << "\n";
        cout << "Customer Name: " << temp->customerName << endl;
        cout << "Complain Date : " << temp->date << endl;

        cout << "_____
        _____" << endl;
        cout << "-----" << endl;
        cout << "|Complaint description: |" << endl;
        cout << "-----" << endl;
        cout << temp->x;
        cout << "\n";

        cout << "_____
        _____" << endl;
    }
}

void complaint::cancel_complaint()
{
    system("cls");
    int num, count;
    cout << "\n";
    cout << "\t\t =====\n";
    cout << "\t\t Apartment Management System - Complaint Box \n";
    cout << "\t\t =====\n\n";
    cout << "Enter the Complaint number you want to delete \n";
    cin >> num;

```

```

node *q;
node *temp;
bool found;
if(start_ptr == NULL)
    cerr<<"Can not cancel from an empty list.\n";
else
{
    if(start_ptr->complaint_number == num)
    {
        q = start_ptr;
        start_ptr = start_ptr->next;
        count--;
        if(start_ptr == NULL)
            last = NULL;
        delete q;
        cout<<"The Complaint is Cancelled Successfully"<<endl;
    }
    else
    {
        found = false;
        temp = start_ptr;
        q = start_ptr->next;
        while((!found) && (q != NULL))
        {
            if(q->complaint_number != num)
            {
                temp = q;
                q = q-> next;
            }
            else
                found = true;
        }
    }
}

```

```

        if(found)
        {
            temp->next = q->next;
            count--;
            if(last == q)
                last = temp;
            delete q;
            cout<<"The Complaint is Cancelled Successfully"<<endl;
        }
        else
            cout<<"Complaint to be Cancelled is not in the list."<<endl;
    }
}

void complaint::exit()
{
    cout<<"\nYou choose to exit.\n"<<endl;
}

int main()
{
    while(1)
    {
        cout<<"\n\t\t*****";
        cout<<"\n\t\t* APARTMENT MANAGEMENT PROJECT *";
        cout<<"\n\t\t*****";
        int n;
        cout<<"\n";
        cout<<"\t\t===== \n";
        cout<<"\t\t\t Apartment Management System-WELCOME \t\t\t \n";
        cout<<"\t\t===== \n\n";
        cout<<"\t\t----- \n";
        cout<<"\t\t\t 1. Details \t\t\t \n";
    }
}

```

```

cout<<"\t\t-----\n";
cout<<"\t\t-----\n";
cout<<"\t\t\t2. Bidding \t\t\t\n";
cout<<"\t\t-----\n";
cout<<"\t\t-----\n";
cout<<"\t\t\t3. COMPLAINT BOX \t\t\t\n";
cout<<"\t\t-----\n\n";
cout<<"\t\tEnter your choice:";
cin>>n;
switch(n)
{
case 1:
{
    details();
    break;
}
case 2:
{
    bidding();
    break;
}
case 3:
{
    complaint1();
    break;
}
}
}
}

```


OUTPUT:

MAIN MENU (WELCOME):

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

*****
* APARTMENT MANAGEMENT PROJECT *
*****

=====
| Apartment Management System-WELCOME |
=====

-----
| 1. Details |
-----
| 2. Bidding |
-----
| 3. COMPLAINT BOX |
-----

Enter your choice:
```

DETAILS MENU (ENTER 1 IN THE MAIN WELCOME MENU)

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

===== APARTMENT MANAGEMENT SYSTEM =====

1. Add Details
2. View Details
3. Modify Details
4. Delete Details
5. Exit Details search!!!

Enter Your Choice :
```


VIEW AND ADD DETAILS:

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=== View the Records in the Database ===

chiru    shriram
8270802248    355
bhavesh    topper
8938372829    123
nikhil    raga
8298333233    90
Press any key to continue . . .
```

BIDDING: (ENTER 2 IN MAIN WELCOME MENU)


 C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

```
=====
|      Apartment Management System - Bidding      |
=====

-----
|      1. VIEW base value                          |
-----
|      2. BID YOUR AMOUNT                         |
-----
|      3. SHOW HIGGEST BIDDER                     |
-----
|      4. EXIT                                    |
-----

Enter your choice:1
the initial value is 5000Press any key to continue . . .
```

BID YOUR AMOUNT:

 C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

```
=====
|      Apartment Management System - Bidding      |
=====

-----
|      1. VIEW base value                          |
-----
|      2. BID YOUR AMOUNT                         |
-----
|      3. SHOW HIGGEST BIDDER                     |
-----
|      4. EXIT                                    |
-----

Enter your choice:2
The Base value is 5000
Enter Name:chiru
Enter the value you want to bid(Greater than 5000)
500000
Your bidding amount is500000
Press any key to continue . . .
```

C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

```
=====
|      Apartment Management System - Bidding      |
=====

-----
|      1. VIEW base value                          |
-----
|      2. BID YOUR AMOUNT                         |
-----
|      3. SHOW HIGGEST BIDDER                     |
-----
|      4. EXIT                                    |
-----
```

Enter your choice:2

The Base value is 500000

Enter Name:bhavesh

Enter the value you want to bid(Greater than 500000)
100000

The bidding amount should be greater than base amount

Enter Name:

VIEW THE WINNER:

C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

```
=====
|      Apartment Management System - Bidding      |
=====

-----
|      1. VIEW base value                          |
-----
|      2. BID YOUR AMOUNT                         |
-----
|      3. SHOW HIGGEST BIDDER                     |
-----
|      4. EXIT                                    |
-----
```

Enter your choice:3

The winner of the BID is chiranjeevi

Value is:1000000

Press any key to continue . . .

Complaint box: (Enter choice as 3 in the main WELCOME menu)

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=====
| Apartment Management System - Complaint Box |
=====

-----
| 1. ADD complaint                            |
-----
| 2. Admin Login                             |
-----
| 3. EXIT                                    |
-----

Enter your choice:
```

ADD COMPLAINT

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=====
| Apartment Management System - Complaint Box |
=====

Add Your Complaint Details
-----

Type Complaint no: 1234
Enter Your Name: chiru
Enter Date : 9-01-2019
Complaint Description:( 1000 words maximum )
The washroom maintainence is very poor.
=====
Complaint added Successfully
=====
Press any key to continue . . .
```

ADMIN LOGIN TO VIEW COMPLAINTS:

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=====
| Apartment Management System - Complaint Box |
=====

-----
| Administrator / Staff Login                 |
-----

Please enter your user name: user

Please enter your user password: pass
```

ADMIN DASHBOARD:

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=====
| Apartment Management System - Complaint Box |
=====

-----
||      1. Create a Complaint      ||
||      2. View Complaint          ||
||      3. Cancel a Complaint      ||
||      4. Logout                  ||
||      5. Exit                    ||
-----

Enter choice:
```

ADMIN VIEWS THE COMPLAINT:

```
C:\Users\chiru\Desktop\DATASTRUCTURE\ams.exe

=====
| Apartment Management System - Complaint Box |
=====

Enter the Complaint Number To Look into The Complaint Details
1234

-----
Here is the Complaint list
-----

Complaint Number : 1234
Customer Name: chiru
Complain Date : 9-01-2019

-----
|Complaint description: |
-----
The washroom maintainence is very poor.

-----
Press any key to continue . . .
```

CONCLUSION:

This software will make our day-to-day activity easy by making it digital. We can easily communicate in the apartment without any disturbances. This will reduce the cost by arranging a lot meeting, elections and bidding.

Here we are providing details search where we can view details of each owner. We have also provided the option of bidding where the users can bid for the available flats and other amenities present in the apartment which makes users as well as admins to conduct them easily and there will be no option of fraud. We also implemented a complaint box where the users can file a complaint on so and so things, which they are not satisfied. The complaint can be viewed by the admin, the complaint will be registered and the proper authorities take care of the problem.

REFERENCES:

<https://www.sanfoundry.com/c-program-stack-implementation/>

https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm

<https://codescracker.com/cpp/cpp-opening-closing-files.htm>

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-develop-cplusplus-simple>

<https://stackoverflow.com/questions/16424828/how-to-connect-mysql-database-using-c/16424893>