PROJECT REPORT

Submitted by

Vinay Vunnava
Regd no:RA2211026010544

Rufas kanikanti
Regd no: RA2211026010557

Keerthi pavan sugnanam
Regd no:RA2211026010561

Under the Guidance of
Dr.P.SARAVANAN

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in AIML

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

MAY 2023

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Project Report titled **"STUDENT HOSTEL MANAGEMENT"** is the bonafide work done by Vinay Vunnava regd no:RA2211026010544, Rufas kanikanti regd no:RA2211026010557 , keerthi pavan sugnanam regd no:RA2211026010561 who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE	**SIGNATURE**
Dr.P.Saravanan	**Dr.M.Pushpalatha**
OODP – Course Faculty	Professor & Head
Assistant Professor	Department of Computing Technologies
Department of Computing Technologies	School of Computing
SRMIST	SRMIST

TABLE OF CONTENTS

1.PROBLEM STATEMENT

Develop a program for student hostel management system through online
The Hostel Management System is developed for automating the activities of hostel. The software will be great relief to the employees. This software will help user in case of reporting, registration and searching the information about residents and rooms. The aim of the Hostel Management System is to carry out the activities of Hostel in an efficient way. It will take the operations of Hostel to an upper level by providing faster access to data and allowing addition, upgradation, modification, and deletion of data in a very systematic and reliable manner.

After make an observation on the current process of hostel management at school, it found that every single thing is done completely by manual.

Currently, the managements are preferring online system for easy process rather than offline mood system . It is because there are facing problem such as corrupted of data. The data about information of student hostel are store and keep not very well and systematically.

In the current process, the data are stored into the file but not in the database which is lead to data duplication, repetitive data, and isolation of data from one to another. It is also worried of something happen to the file, then all the data will lost.
In the current manual system, it will very difficult to find the hostel record and other information of student manually. Because it has been keep on the paper and it is easy to loss. It also consume time to search the paper of student hostel record one by one.
The manual system requires longer time for allocation the student to respective hostel, dorm, and bed.

Besides that, the manual application will lead toward a hassle data management for faster student allocation as well as managing the data for faster task such as student's activities, student outing record and managing visitors.

2.MODULES OF PROJECT

Users:-This help the administrator that he/she only crates new user by providing login details.

Change password:-Allows the user to change the password

Forget Password:-On entering the correct userid and hint answer, User will be able to regain his password.

Registration:-Allows to add the details of students regarding personal information, Academic Information, room details etc.Student details can also be modified and deleted.

Room Details:-Allows the user to add details of rooms. Also possible to view the status of rooms such as occupied,vacant. Room details can also be modified and deleted.

Fees:-Allows to enter the Fees details of the student. Student Fees details can also be modified and deleted.

Visitors:-Allows to add the visitors details. Also able to view the visitors depending on various search criteria.

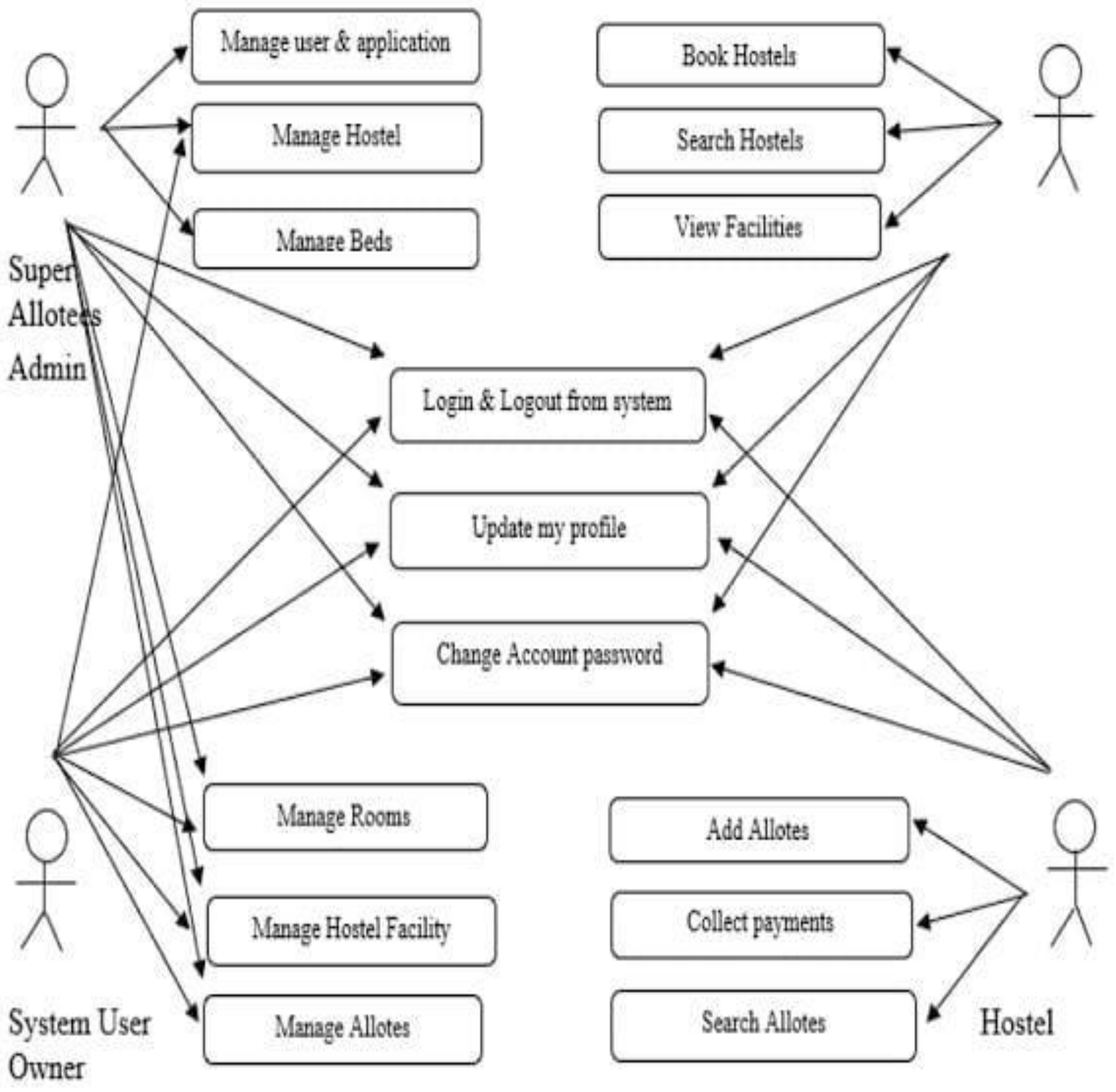
Employee :-Allows to add the details of Employees .Employee Details could be modified and deleted. Viewing the Employees as well as searching also could be done.

Status:-Displays the details of the students staying in rooms. Depending on the room number Student details will be displayed.

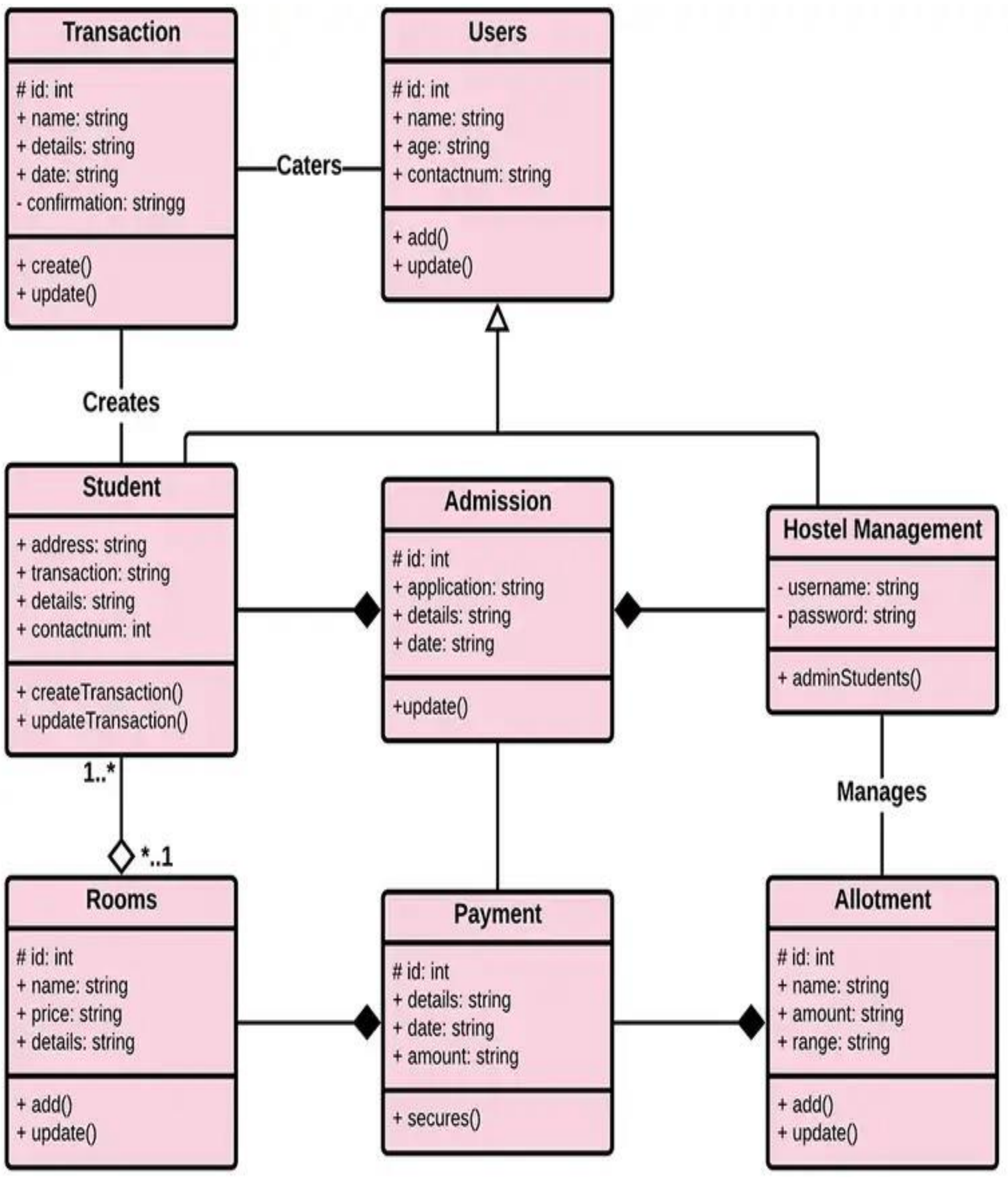
Reports:-Student Report and visitors report could be generated.

3.DIAGRAMS

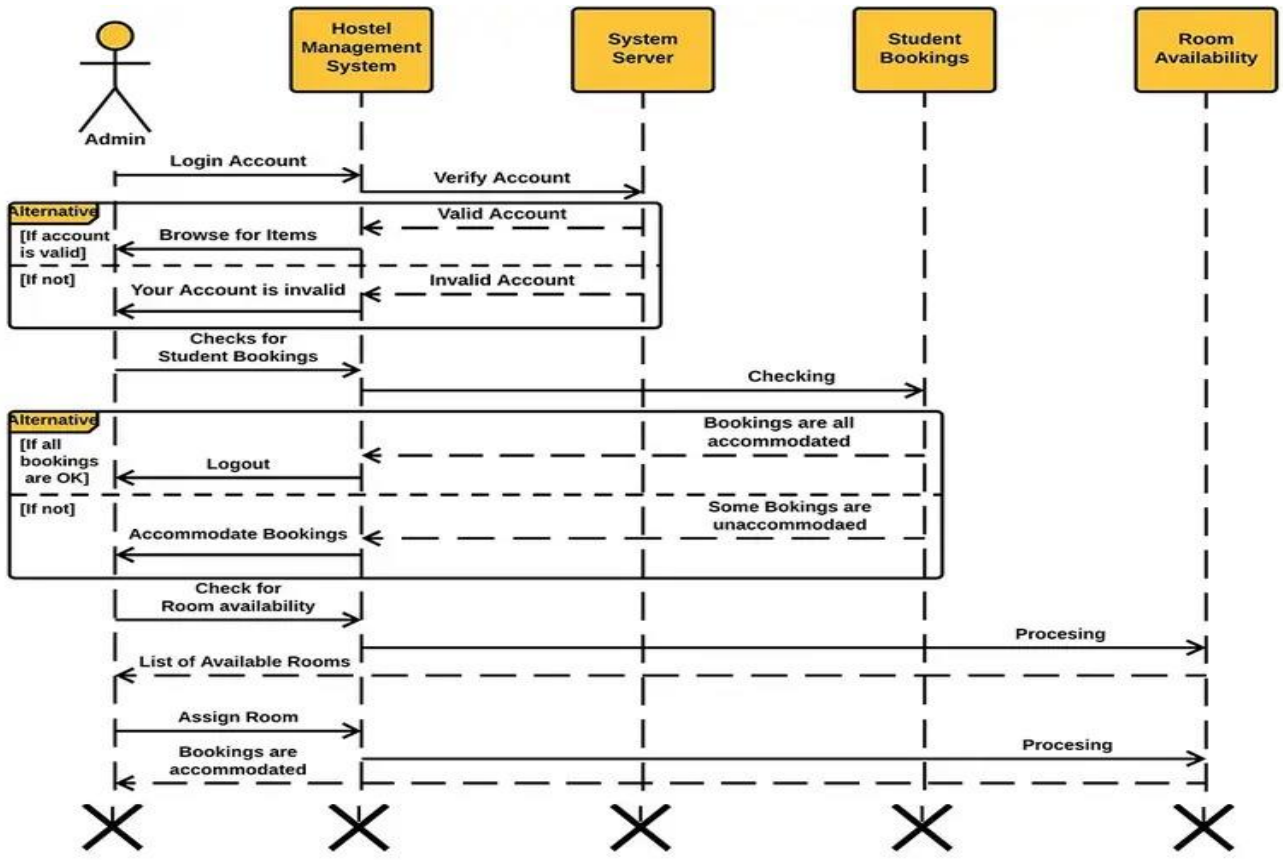
a.USE CASE DIAGRAM



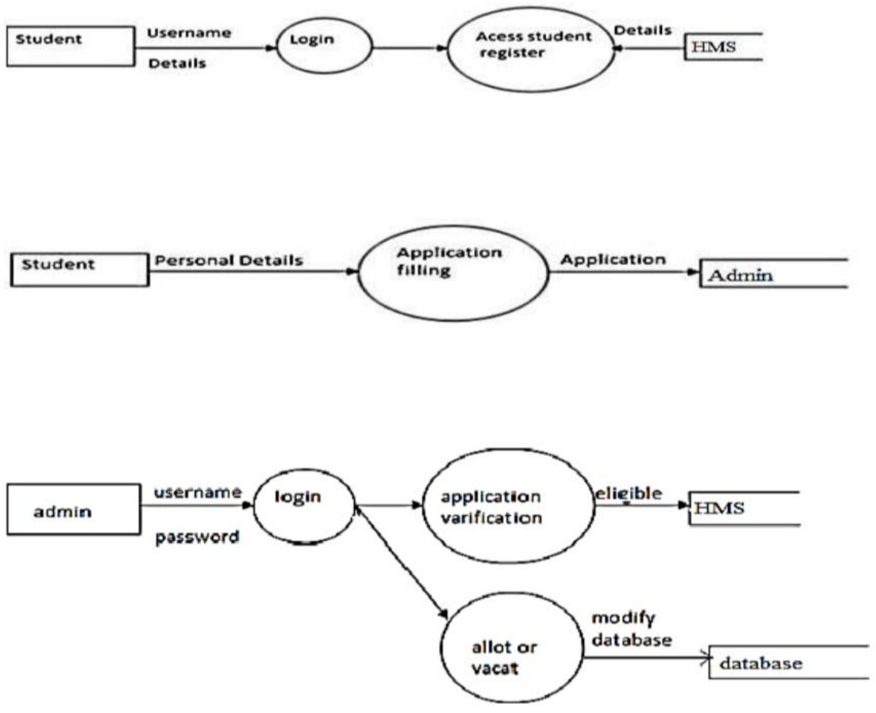
b.CLASS DIAGRAM



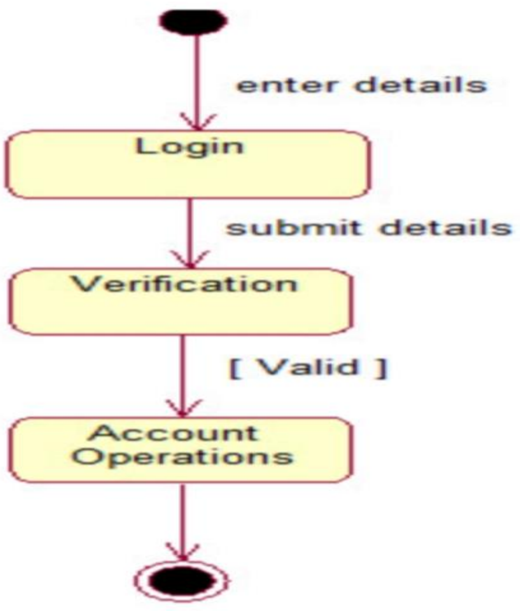
c.SEQUENCE DIAGRAM



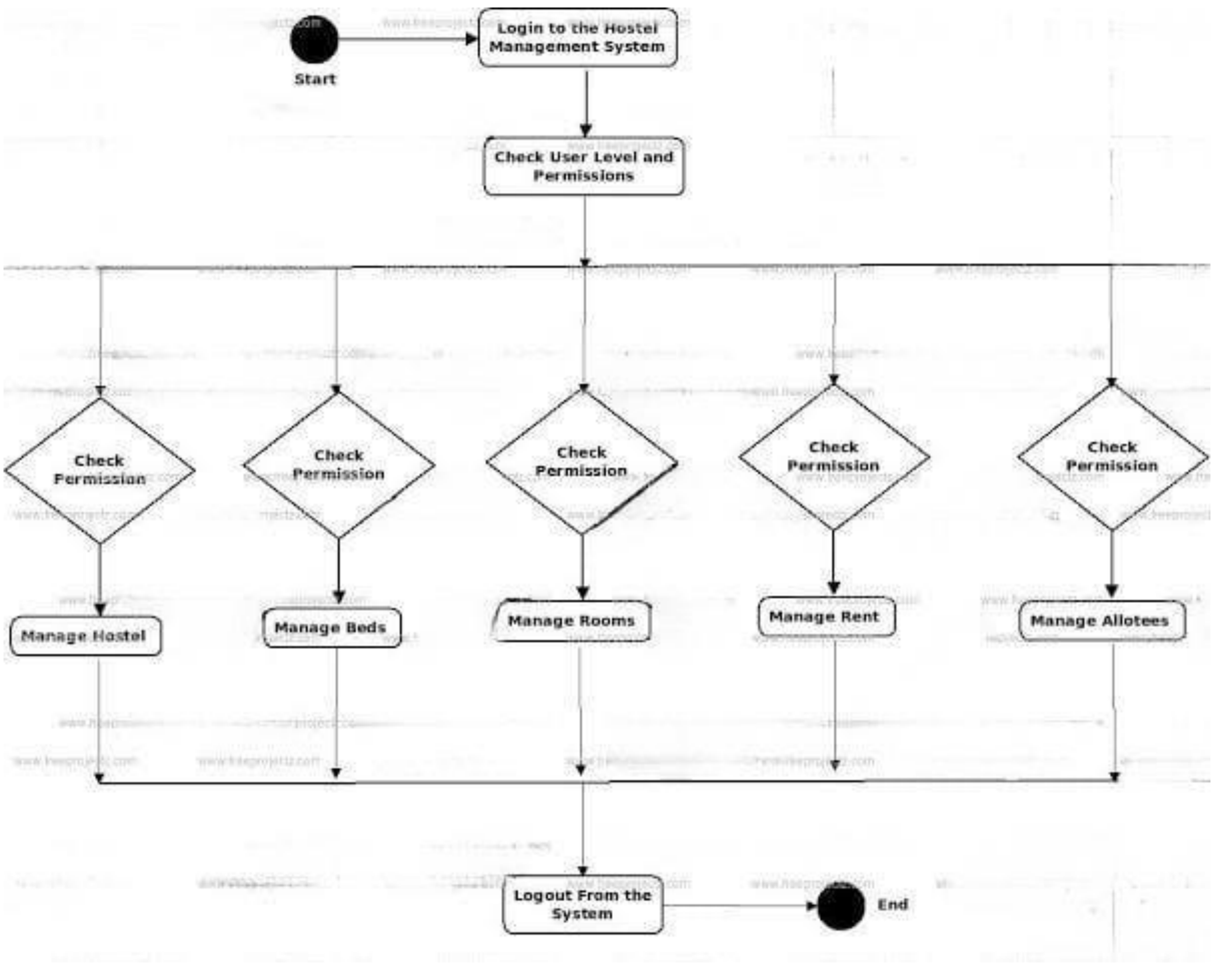
d.COLLABRATION DIAGRAM



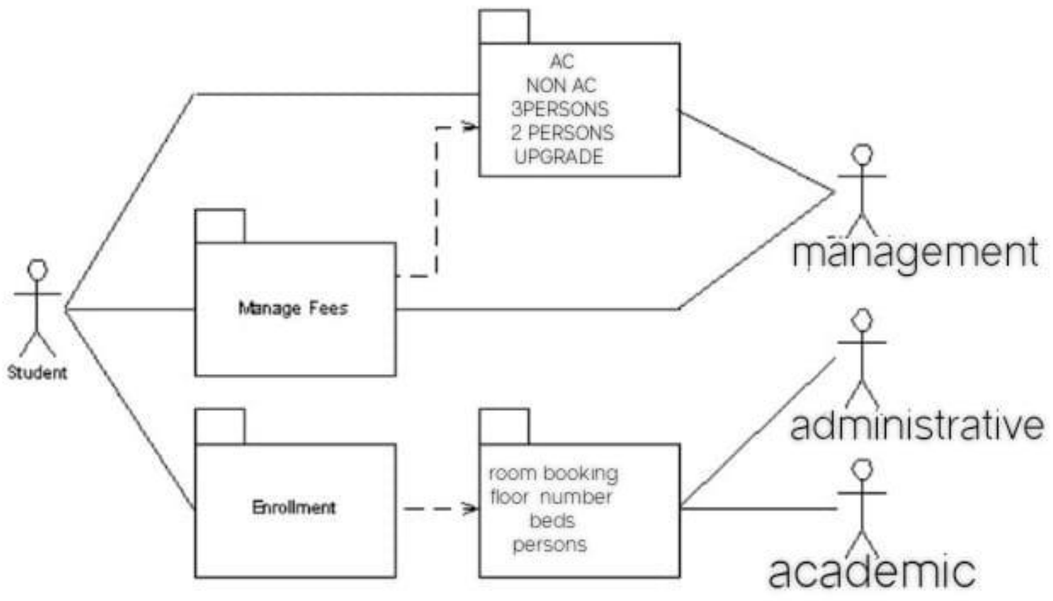
e.STATE CHART DIAGRAM



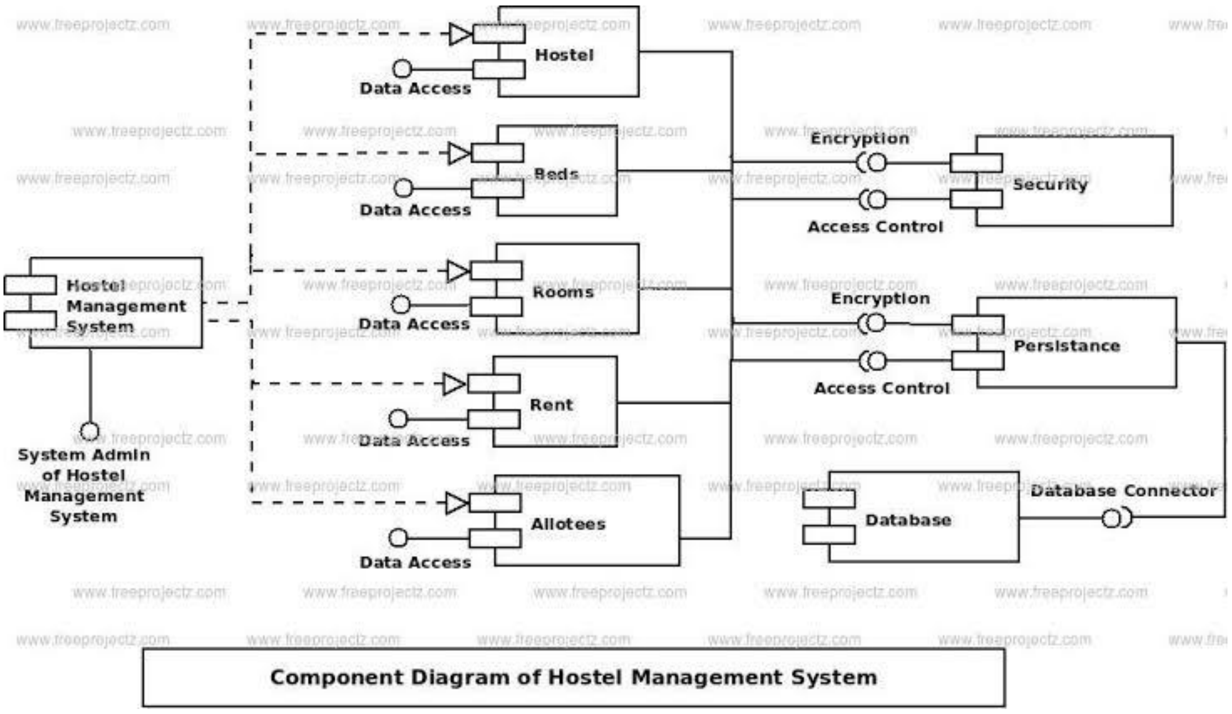
f.ACTIVITY DIAGRAM



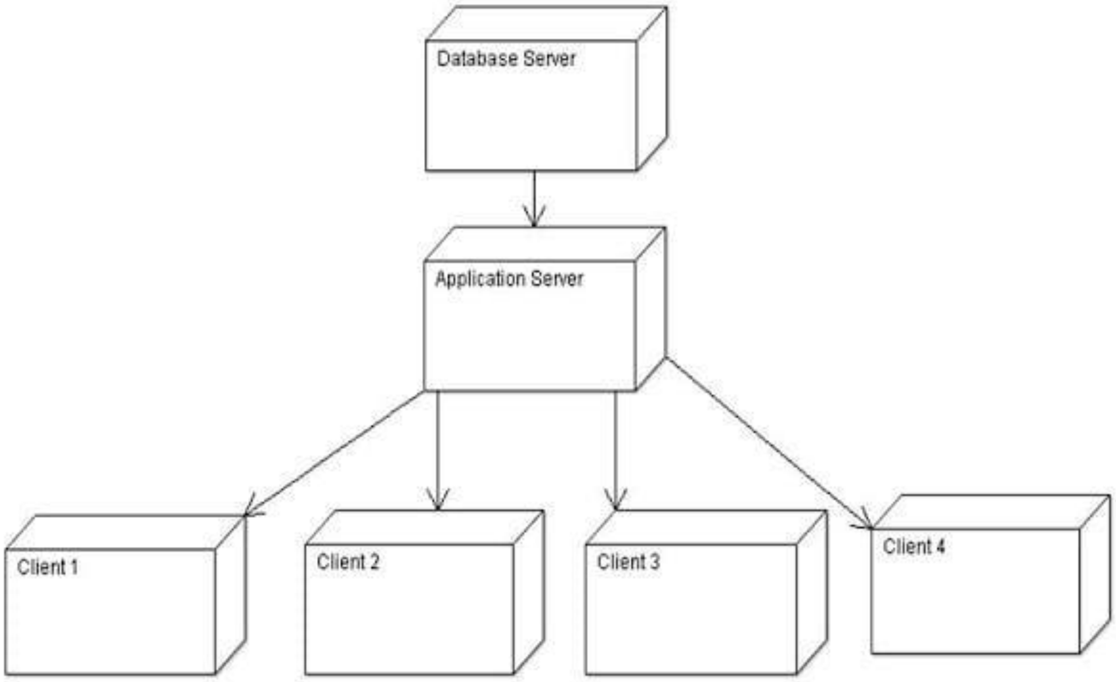
g.PACKAGE DIAGRAM



h.COMPONENT-DIAGRAM



i.DEPLOYMENT DIAGRAM



4.CODE

```
#include <iostream>
#include <string.h>
using namespace std;
struct node
{
    int num = 1;
    int fill = 0;
    char name[3][10];
    node *next;
    node *prev;
};
class hostel
{
    node *header[3];
    node *cn;

public:
    hostel()
    {
        for (int i = 0; i < 3; i++)
            header[i] = NULL;
    }
    void create()
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 9; j++)
            {
                node *nn = new node;
                nn->next = NULL;
                nn->prev = NULL;
                if (header[i] == NULL)
                {
```

```
                header[i] = nn;
                nn->num = 1;
            }
            else
            {
                node *cn = header[i];
                while (cn->next != NULL)
                {
                    cn = cn->next;
                }
                if (j == 3 || j == 5 || j == 7 || j == 8)
                {
                    cn->num = 3;
                }
                if (j == 2 || j == 4 || j == 6)
                {
                    cn->num = 2;
                }
                cn->next = nn;
                nn->prev = cn;
            }
        }
    }
}
void display()
{
    int j = 0, k = 0, l = 0;
    for (int i = 0; i < 48; i++)
    {
        cout << "--";
    }
    cout << "\n "
    ;
    for (int i = 1; i < 4; i++)
    {
```

```
    cout << "| Floor number : "<< i << " \t\t";
}
cout << "|\n" ;
for (int i = 0; i < 48; i++)
{
    cout << "--";
}
cn = header[j];
node *sn = header[j + 1];
node *tn = header[j + 2];
cout << "\n ";
while (cn != NULL)
{
    if (cn->fill != cn->num && cn->num != 0)
    {
        j++;
        cout << "| room no : "<< j;
        cout << "->Vacant cots->"<< cn->num;
    }
    else
    {
        j++;
        cout << "| room no : "<< j;
        cout << "->Present ";
    }
    if (sn->fill != sn->num && sn->num != 0)
    {
        k++;
        cout << "\t | room no : "<< j;
        cout << "->Vacant cots->"<< sn->num;
    }
    else
    {
        k++;
        cout << " \t | room no : "<< j;
```

```
            cout << "->Present ";
        }
        if (tn->fill != tn->num && tn->num != 0)
        {
            l++;
            cout << "\t | room no : "<< j;
            cout << "->Vacant cots->"<< tn->num << "| ";
        }
        else
        {
            l++;
            cout << "\t | room no : "<< j;
            cout << "->Present "<< " | ";
        }
        cout << " \n ";
        for (int i = 0; i < 48; i++)
        {
            cout << "--" ;
        }
        cout << "\n ";
        cn = cn->next;
        sn = sn->next;
        tn = tn->next;
    }
}
void book(int people)
{
    int floor, room;
    cout << "\nEnter the floor number : ";
    cin >> floor;
    try
    {
        if (floor < 0 || floor > 4)

        {
```

```
    throw(floor);
}
cn = header[floor - 1];

cout << "\nEnter the room number : ";
cin >> room;
try
{

    if (room < 0 || room > 10)
    {
        throw(room);
    }
    else
    {
        int i = 1;
        while (i < room)
        {
            cn = cn->next;
            i++;
        }
        if (cn->num >= people)
        {
            cout << "\nroom is vacant you can apply for room" ;

            int count = 0;
            while (cn->fill - 1 <= cn->num)
            {

                cout << "\nEnter name "<< cn -> fill + 1 << " : ";

                cin >> cn->name[cn->fill];
                count++;
                cn->fill++;
                if (count >= people)
```

```
                    {
                        break;
                    }
                }
                cn->num = cn->num - people;
            }

            else
            {
                cout << "\nroom is not vacant... SORRY !!!";
            }
        }
    }
    catch (int r)
    {
        cout << "\ninvalid room number : "<< r;
    }
}
catch (int r)
{
    cout << " \ninvalid floor number : " << r;
}
}
void cancle(int check)
{
    char namecheck[10];
    int flag = 0;
    int room, i = 1;
    try
    {
        if (check < 0 || check > 4)

        {
            throw(check);
        }
```

```
else
{
    cout << " Enter the room no : ";
    cin >> room;
    try
    {
        if (room < 0 || room > 10)
        {
            throw(room);
        }
        else
        {
            cout << " Enter the name to be delete :";

            cin >> namecheck;
            cn = header[check - 1];
            while (i < room)
            {
                cn = cn->next;
                i++;
            }
            i = 0;
            while (i < 3)
            {

                if (!strcmp(namecheck, cn -> name[i]))

                {
                    flag = 1;
                    break;
                    i = 0;
                }
                else
                    i++;
            }
```

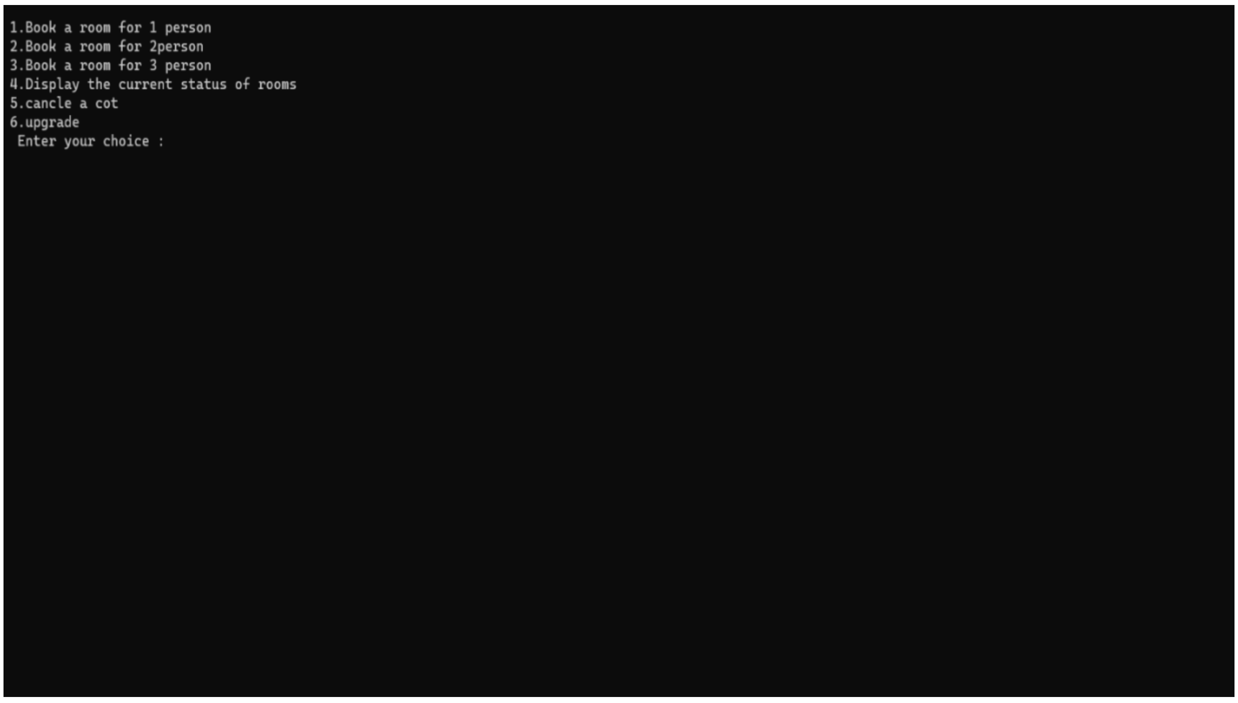
```
                    if (flag == 1 && cn->fill != 0)
                    {
                        cout << "\nrecord deleted : "<< cn -> name[i];

                        cn->name[i][0] ='A';
                        cn->name[i][1] ='\0';
                        cn->fill--;
                        cn->num++;
                    }
                    else

                        cout << "\nrecord not present ";
                }
            }
            catch (int r)
            {
                cout << "\ninvalid room number : " << r;
            }
        }
    }

    catch (int r)

    {
        cout << " \n floor dosn't exist : " << r;
    }
}
void upgrade(int check)
{
    char namecheck[10];
    int room, i = 1;
    try
    {
        if (check < 0 || check > 4)
```

```
{
    throw(check);
}
else
{
    cout << " Enter the room no : ";
    cin >> room;
    try
    {
        if (room < 0 || room > 10)

        {
            throw(room);
        }
        else
        {
            cout << "Enter the name to be updated :";

            cin >> namecheck;
            cn = header[check - 1];
            while (i < room)
            {
                cn = cn->next;
                i++;
            }
            i = 0;
            while (i < 3)
            {
                if (!strcmp(namecheck, cn -> name[i]))

                {
                    cout << "\nenter updated name : " ;

                    cin >> cn->name[i];
                    break;
```

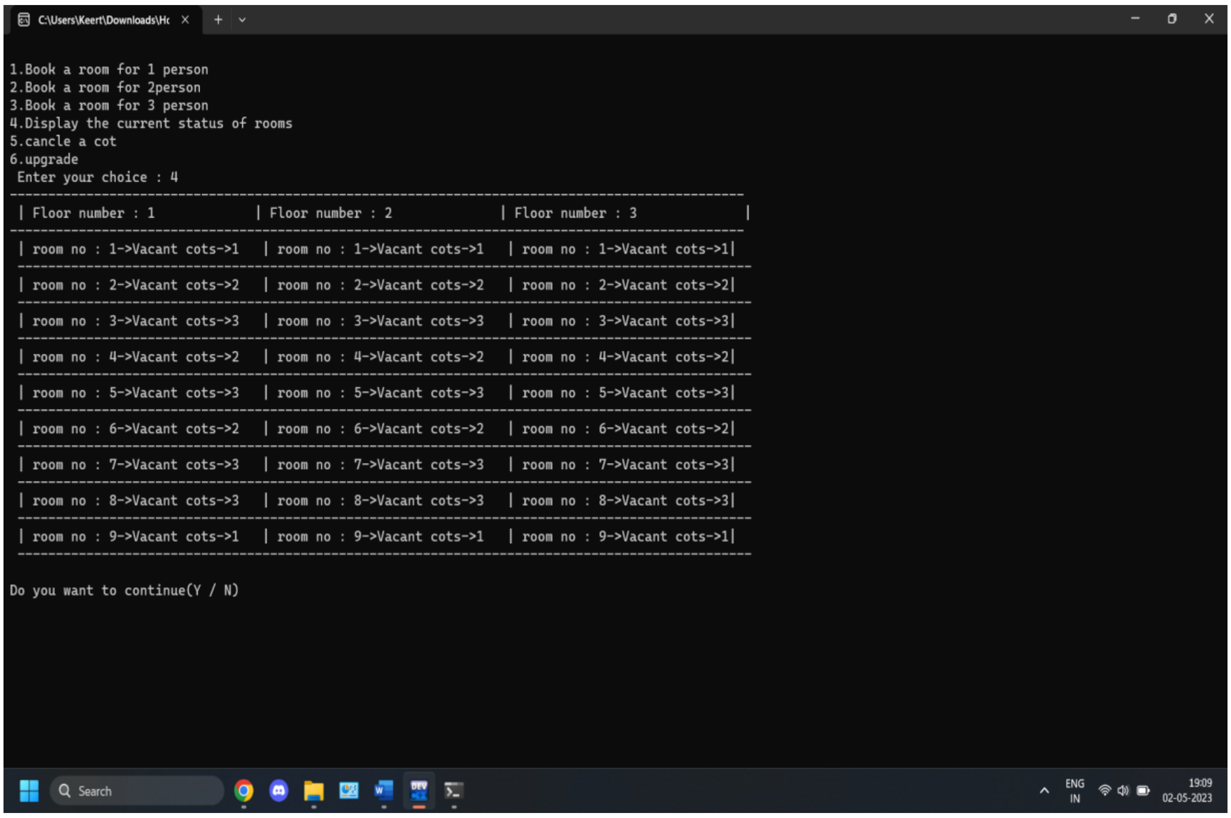
```
                        }
                        else
                            i++;
                    }
                    if (i >= 3)
                        cout << "record not found ";
                    else
                    {
                        cout << "\nrecord updated\nprevious name : "<< namecheck <<
"\nupdated name : "<< cn->name[i];
                    }
                }
            }
            catch (int r)
            {
                cout << "\ninvalid room number : "<< r;
            }
        }
    }

    catch (int r)

    {
        cout << "\n floor dosn't exist : "<< r;
    }
  }
};
int main()
{

  hostel obj;
  int key;
  char ch;
  int floorcheck;
  obj.create();
```

```
    do
    {
        cout << "\n1.Book a room for 1 person\n2.Book a room for 2person\n3.Book a room
for 3 person\n4.Display the current status of rooms\n5.cancle a cot\n6.upgrade"<< endl;
        cout << " Enter your choice : "   ;
        cin >> key;
        switch (key)
        {
        case 1:
        {
            obj.book(1);
            break;
        }
        case 2:
        {
            obj.book(2);
            break;
        }
        case 3:
        {
            obj.book(3);
            break;
        }
        case 4:
        {
            obj.display();
            break;
        }
        case 5:
        {
            cout << "Enter floor number : ";
            cin >> floorcheck;
            obj.cancle(floorcheck);
            break;
        }
        case 6:
        {
            cout << "Enter floor number : ";
            cin >> floorcheck;
            obj.upgrade(floorcheck);
            break;
        }

        default:
            cout << "\nInvalid choice ";
        }
        cout << "\nDo you want to continue(Y / N) ";
        cin >> ch;
    } while (ch =='Y'|| ch =='y');
}
```

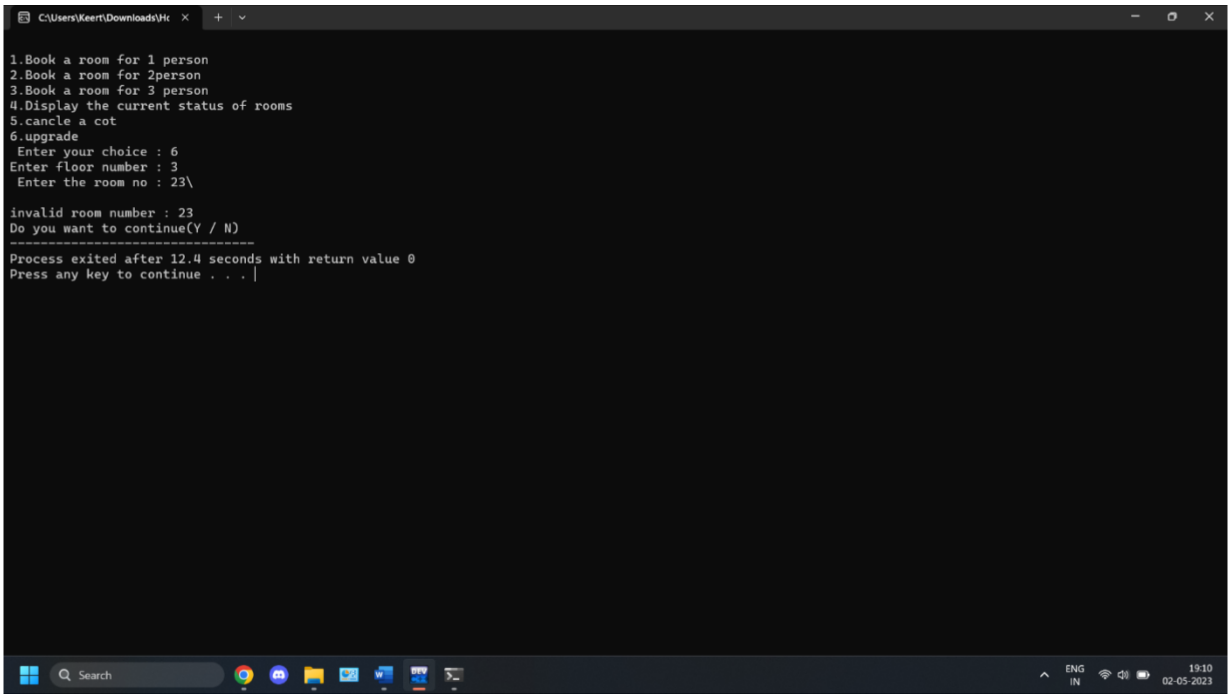
OUTPUT:



```
1.Book a room for 1 person
2.Book a room for 2person
3.Book a room for 3 person
4.Display the current status of rooms
5.cancle a cot
6.upgrade
 Enter your choice :
```



```
1.Book a room for 1 person
2.Book a room for 2person
3.Book a room for 3 person
4.Display the current status of rooms
5.cancle a cot
6.upgrade
 Enter your choice : 4
------------------------------------------------------------------------------------------------
 | Floor number : 1              | Floor number : 2              | Floor number : 3              |
------------------------------------------------------------------------------------------------
 | room no : 1->Vacant cots->1   | room no : 1->Vacant cots->1   | room no : 1->Vacant cots->1|
 -------------------------------------------------------------------------------------------------
 | room no : 2->Vacant cots->2   | room no : 2->Vacant cots->2   | room no : 2->Vacant cots->2|
 -------------------------------------------------------------------------------------------------
 | room no : 3->Vacant cots->3   | room no : 3->Vacant cots->3   | room no : 3->Vacant cots->3|
 -------------------------------------------------------------------------------------------------
 | room no : 4->Vacant cots->2   | room no : 4->Vacant cots->2   | room no : 4->Vacant cots->2|
 -------------------------------------------------------------------------------------------------
 | room no : 5->Vacant cots->3   | room no : 5->Vacant cots->3   | room no : 5->Vacant cots->3|
 -------------------------------------------------------------------------------------------------
 | room no : 6->Vacant cots->2   | room no : 6->Vacant cots->2   | room no : 6->Vacant cots->2|
 -------------------------------------------------------------------------------------------------
 | room no : 7->Vacant cots->3   | room no : 7->Vacant cots->3   | room no : 7->Vacant cots->3|
 -------------------------------------------------------------------------------------------------
 | room no : 8->Vacant cots->3   | room no : 8->Vacant cots->3   | room no : 8->Vacant cots->3|
 -------------------------------------------------------------------------------------------------
 | room no : 9->Vacant cots->1   | room no : 9->Vacant cots->1   | room no : 9->Vacant cots->1|
 -------------------------------------------------------------------------------------------------

Do you want to continue(Y / N)
```



```
1.Book a room for 1 person
2.Book a room for 2person
3.Book a room for 3 person
4.Display the current status of rooms
5.cancle a cot
6.upgrade
 Enter your choice : 6
Enter floor number : 3
 Enter the room no : 23\

invalid room number : 23
Do you want to continue(Y / N)
--------------------------------
Process exited after 12.4 seconds with return value 0
Press any key to continue . . . |
```

5.CONCLUSION:

Our project "HOSTEL MANAGEMENT SYSTEM" is a very helpful and important project thatwill manage various activities in the hostel like accommodation, rents, student records and manyother things that are very useful for a well-managed hostel. The scope of this project is to providethe facility of the living to the student of rural areas, who are unable to study further in their area.They will be able to have an affordable rates of living in hostel and they will be fully providedthe environment which they need for their studies. Another benefit of this project is that all thehostel managing works can be done easily through it by saving our time and also by saving thehuman efforts. This management system will be an errorless or bugs free management systemand will be able to show the records of even years without any confliction

6.REFRENCES:

1. LET US C++ , BPB PUBLICATION ,AUTHOR:YASHAVANT KANETKAR'S
2.THE PROGRAMMING LANGUAGE BOOK C++
3.W3SCHOOL, BLOGGER LEARNING AND PRATICESS WEBSITE