SRM UNIVERSITY ANDHRA PRADESH

Object Oriented Programming

Project on

"SRM Hostel management System"

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

Submitted by –

Name	**Branch and Year**	**Registration Number**
SRI HARSHA	CSE-E, 2nd Year	AP21110010007
B.V.N. HAREESH	CSE-E, 2nd Year	AP21110010839
NIMSHARI.R	CSE-E, 2nd Year	AP21110010009
DHRUV SHARMA	CSE-E, 2nd Year	AP21110010043
REETHU BHARGAVI.S	CSE-E, 2nd Year	AP21110010045

Under the Guidance of

Dr. Neeraj Kumar Sharma

ABSTRACT

- This project "Hostel Management System" target for all the hostel in the school and college and other this transition of hostel management system from the better control and timely response.
- This eliminates the time delay and paperwork, the warden is provided with the control on the system to add or deleting the details of the student and edit and see the details of the student details.
- Even the higher authorities of the school and college can easily access to see the data.

INTRODUCTION

- It is not recommended to use a manual system currently, when automated systems can be software or hardware.
- Hostels that do not have a management system typically do everything by hand.
- The verification of registration forms to other processes for saving data is done by hand, and most of the time, it is written on paper.
- As a result, an automated system can avoid a lot of repetitions.
- A computerized system that will help reduce the number of manuals inputs are being designed in response to the shortcomings of the current systems.
- We can overcome the shortcomings of the current manual system by increasing its efficiency with this system in place.
- The management of the hostel can save student records about their rooms and other things thanks to the system's design.
- It relieves them of manual labor, which makes it difficult to locate student records.
- The room allocation is better managed thanks to this system, which provides information about students as well as fee details.
- Special features like the number of students in a room, student ID, and free rooms or space will be included in the hostel management system.
- Each member of the administration and each student's information has a distinct identity in the administration.

OBJECTIVE

The project's main idea is to provide basic amenities that would suffice and quickly enhance the user's experience and requirements, which would help in maintaining transparency between members

Algorithm:

STEP1: -The code is a simple program that creates 10 nodes, each with an integer value and name.
STEP2: -The code then prints the names of all the nodes in order from 0 to 9.
STEP3: -The first line of the function create () declares a for loop that iterates through 10 times.
Each iteration has two statements: one is creating a new node and assigning it to header[i], which will
Be printed on-screen; the other statement assigns cn->next = NULL, which means that this node will
not have any children (children are nodes with parents).
STEP4: - If there is no child before this node, then header[i] becomes nn->num = 1; otherwise,
cn->next points to another parent node until it reaches NULL.
STEP5: -The code creates a new instance of the class "node" which is initialized with the value of 0.
STEP6: -The code then creates ten instances of "node" and initializes them with values 1, 0, and NULL respectively.
STEP7: -This will create ten nodes in total, one for each row in the matrix-like structure created by
these lines of code.
STEP8: -The variable is initialized to the header array, which contains all of the floor numbers in order
from 0-9.
STEP9: -The code then declares another variable called nn and initializes it to point to the first element
in the list (the one with index 0).
STEP10: -If so, then nn points back at itself and its previous value becomes cn again as shown below:
if (j == 5 || j == 6 || j == 7) { cn->num = 2; } If not for any of these conditions being true then nnn points
at itself and its previous value becomes null because nothing else has been assigned yet as shown below:
STEP11: - if
The code is used to generate a list of all the numbers in the range 1-100.
The code starts with a for loop that iterates through 100 times, which will display all the numbers from
1-100 on one line.
STEP12: -
The code then creates an array called header and initializes it with a 0.
It then uses this array to store the current number being displayed.
STEP13: -
The next part of the code is where it generates a list of every single possible number from 1-100 by
using nested if statements, which are explained below:

STEP14: - if (j
The code starts by printing the number of vacant cots in each room.
Then it prints a message for every room that is not empty.
The code starts with an if statement that checks whether there are any vacancies in the first three
rooms and then proceeds to check all other rooms.

STEP15: -
If there are no vacancies, it prints "Present" followed by a space and then moves on to the next line
of code.Otherwise, it prints "Vacant cots->" followed by the number of vacant cots in that particular
room followed by a space and then moves on to the next line of code.

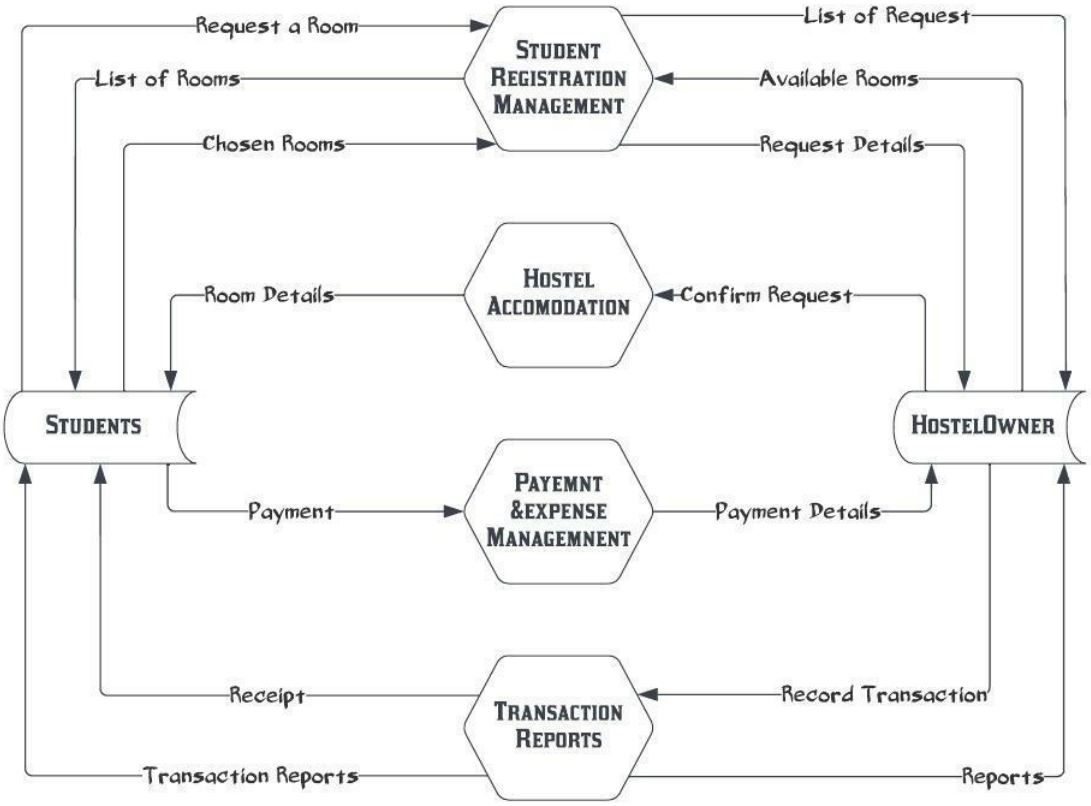
STEP16: -
This code also declares two more arrays: one for login and another for logout.
Next, there are three if statements inside mainmenu().

STEP17: -
If strcmp(name,l) does not equal 0 then it means that you have entered your own username into
this program and you need to register with your username before continuing with any other actions
in this program.

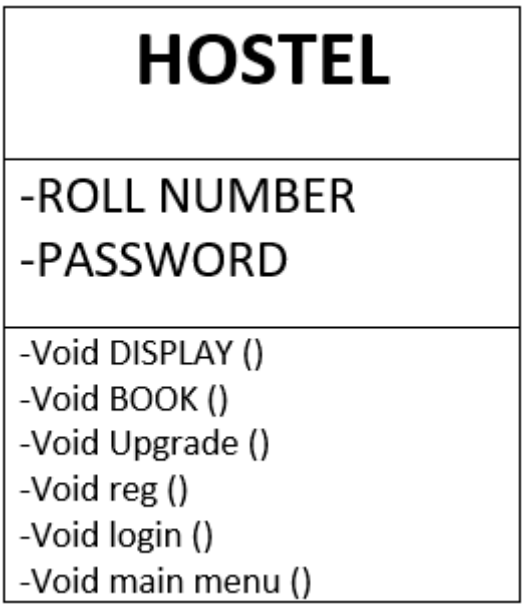
STEP18: -
If strcmp(name,l) equals 0 then it means that you are trying to log into someone else's account so they
can re-log out or change their password; therefore, they must exist as well as being logged into at least
once before proceeding further with anything else in mainmenu().

STEP19: - Finally if strcmp(name,"") equals 0 then it means that no one exists on this computer, and you need to create them by pressing enter when prompted during

Block Diagram



UML Diagram –


HOSTEL
-ROLL NUMBER
-PASSWORD
-Void DISPLAY ()
-Void BOOK ()
-Void Upgrade ()
-Void reg ()
-Void login ()
-Void main menu ()

SYSTEM IMPLEMENTATION

Modules or Functions

- Main Function of the program

```
int main()//MAIN FUCNTION
{

 hostel obj;
 obj.create();
 obj.admin();
 obj.mainmenu();
}
```

We have created a structure for storing the data of student and we have created a class hostel, containing functions like:

- Void reg ()
- Void admin ()
- Void upgrade ()
- Void book ()
- Void login ()
- Void main menu ()

```
struct node
{
int num{1};
 int fill = 0;
 char name[3][10];
 node *next;
 node *prev;
};
class hostel
{
 node *header[10];
 node *cn;


public:
 hostel()
 {
 for (int i = 0; i < 10; i++)
 header[i] = NULL;
 }
 void create()
{
 for (int i = 0; i < 10; i++)
 {
 for (int j = 0; j < 10; j++)
 {
 node *nn = new node;
 nn->next = NULL;
 nn->prev = NULL;
```

- In this registration function the use will register.

```
void reg()
{
FILE *fp;
struct login l;
fp=fopen("login.txt","w");
cout<<"\n========================";
cout<<"\n REGISTRATION";
cout<<"\n========================";
cout<<"\n\nUSER ID: ";
cin>>l.name;
fflush(stdin);
cout<<"\nPASSWORD: ";
cin>>l.pass;
cout<<"\n\n<<ENTER TO PROCEED>>";
getch();
fwrite(&l,sizeof(l),1,fp);
fclose(fp);
cout<<"\n\nACCOUNT SUCCESSFULLY CREATED";
cout<<"\n\nGO TO LOGIN PAGE\n\n";
cout<<">PRESS 1 FOR YES\n\n> PRESS 2 FOR NO\n\n";
cin>>n;
```

- In this function we display the main menu.

```
void mainmenu()//MAIN MENU
{
int key;
 char ch;
 int floorcheck;
 do
 {
 cout<<"\n\n\n========================";
 cout<<"\n MAINMENU";
 cout<<"\n=====================\n";
 cout << "1.BOOK A ROOM FOR 1 PERSON\n 2.DISPLAY THE CURRENT STATUS OF THE ROOMS\n 3.CANCEL BOOK\n4.UPDATE CURRENT ROOM"<< endl;
 cout<<"\n\nENTER YOUR CHOICE:";
 cout << " "  ;
 cin >> key;
```

- In this function the person will be login with the help of this function.

```
void login()//LOGIN
{
FILE *fp;
char c,name[30],pass[30];
fp=fopen("login.txt","r");
cout<<"\n================";
cout<<"\n LOGIN";
cout<<"\n================";
struct login l;
cout<<"\n\nUSER ID: ";
cin>>name;
fflush(stdin);
cout<<"\nPASSWORD: ";
cin>>pass;
```

- In this function we can upgrade the data.

```
void upgrade(int check)
{
char namecheck[10];
int room, i = 1;
try
{
if (check < 0 || check > 10)
```

- In this function we book the room for the user.

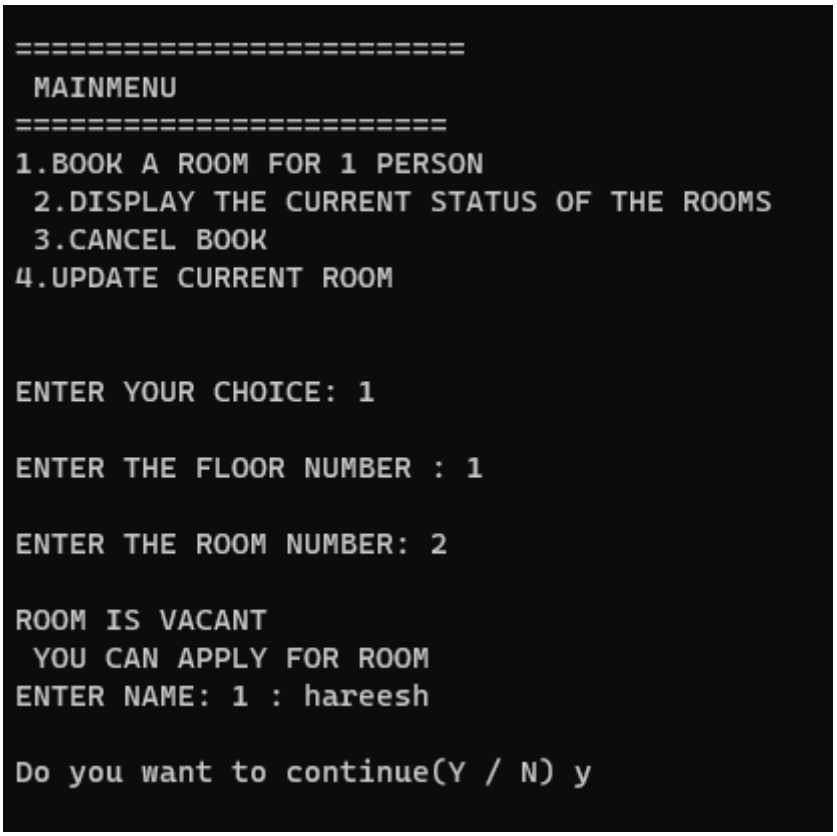
```
void book(int people)
{
int floor, room;
cout << "\nENTER THE FLOOR NUMBER : ";
cin >> floor;
try
{
if (floor < 0 || floor > 10)
```

RESULTS

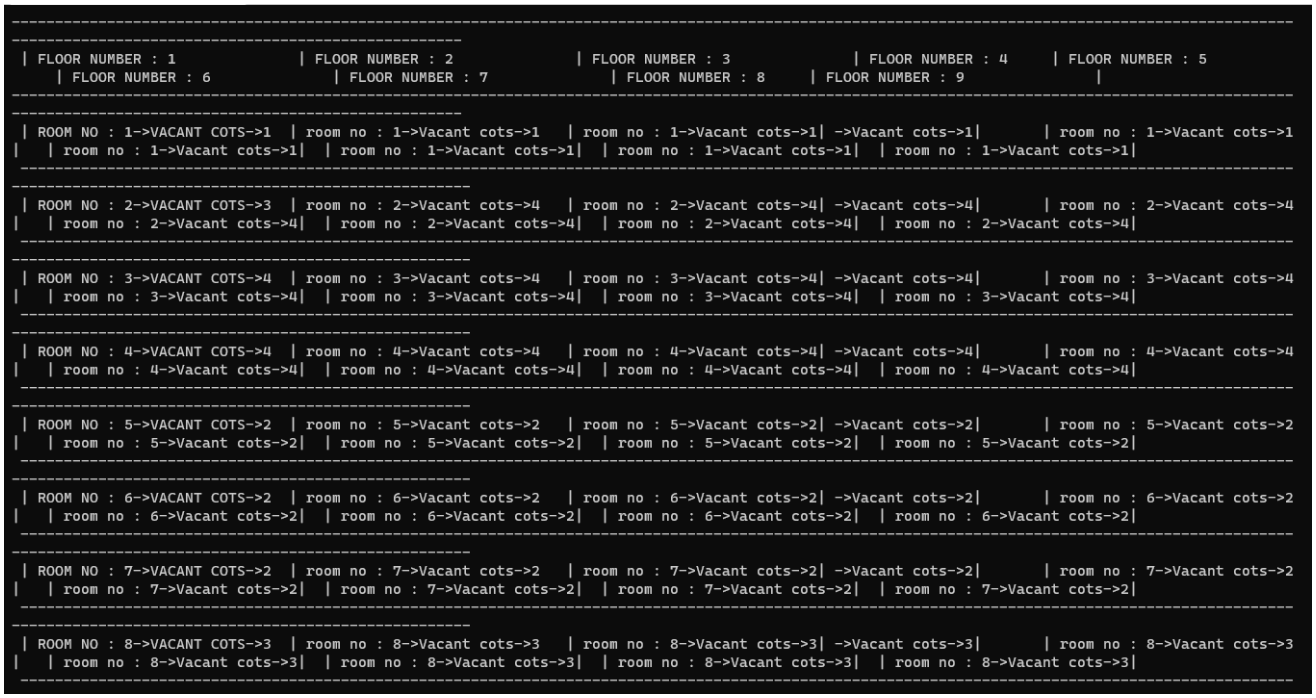
Sample input and output

```
====================================
 WELCOME TO HOSTEL MANAGEMENT SYSTEM
====================================

1.LOGIN         2.REGISTER


ENTER YOUR CHOICE: 1



================
 LOGIN
================

USER ID: AP21110010839

PASSWORD: hareesh



========================
 MAINMENU
========================
1.BOOK A ROOM FOR 1 PERSON
 2.DISPLAY THE CURRENT STATUS OF THE ROOMS
 3.CANCEL BOOK
4.UPDATE CURRENT ROOM
```

- Display Main Menu and book a room.



- Display The currency status of the rooms



```
 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 | ROOM NO : 9->VACANT COTS->3  | room no : 9->Vacant cots->3   | room no : 9->Vacant cots->3| ->Vacant cots->3|       | room no : 9->Vacant cots->3|   | room no : 9->Vacant cots->3|  | room no : 9->Vacant cots->3|  | room no : 9->Vacant cots->3|
 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 | ROOM NO : 10->VACANT COTS->1 | room no : 10->Vacant cots->1  | room no : 10->Vacant cots->1| ->Vacant cots->1|      | room no : 10->Vacant cots->1|  | room no : 10->Vacant cots->1|        | room no : 10->Vacant cots->1|         | room no : 10->Vacant cots->1|
 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

- Cancel book

```
=========================
 MAINMENU
=========================
1.BOOK A ROOM FOR 1 PERSON
 2.DISPLAY THE CURRENT STATUS OF THE ROOMS
 3.CANCEL BOOK
4.UPDATE CURRENT ROOM


ENTER YOUR CHOICE: 3
ENTER FLOOR NUMBER : 1
 ENTER THE ROOM NO : 2
 ENTER THE NAME TO BE DELETE:hareesh

RECORD DELETED : hareesh
```

- Update current room

```
=========================
 MAINMENU
=========================
1.BOOK A ROOM FOR 1 PERSON
 2.DISPLAY THE CURRENT STATUS OF THE ROOMS
 3.CANCEL BOOK
4.UPDATE CURRENT ROOM


ENTER YOUR CHOICE: 4
ENTER FLOOR NUMBER : 1
 Enter the room no : 2
Enter the name to be updated :hareesh
record not found
```

System Requirement Specification

S.No	Hardware Requirements	
I.	PROCESSOR:	PENTIUM(ANY) OR AMD ATHALON(3800+- 4200+ DUAL-CORE)
II.	MOTHERBOARD:	1.845 OR 915,995 FOR PENTIUM 0R MSI K9MM-V VIAK8M800+8237R PLUS CHIPSET FOR AMD ATHALON
III.	RAM:	512MB+
IV.	Hard disk:	SATA 40 GB OR ABOVE
V.	CD/DVD r/w multi-drive combo:	(If Backup required)
VI.	FLOPPY DRIVE 1.44 MB:	(If Backup required)
VII.	MONITOR:	14.1 or 15 -17 inch
VIII.	Keyboard and mouse:	(Of Any Company)
IX.	Printer:	(If a print is required – [Hard copy])

S.No	Software Requirements	
1.	Operating System:	Windows 7 and above
2.	IDE:	Dev C++
3.	Programming Language used:	C++