

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY**

BHAGALPUR,Bihar

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COURSE NAME:-SOFTWARE ENGINEERING

**SUBMITTED BY:**

**MANDEEP JAGLAN (180101028)**

**VERCHASWA SHARMA (180101047)**

**DHRUV SRIVASTAVA (180101015)**

**GOKUL SANKAR (180101017)**

**ANKUSH (180101007)**

Table of contents

[PROJECT TITLE 3](#_heading=h.1fob9te)

[ACKNOWLEDGEMENT 4](#_heading=h.3znysh7)

[ABSTRACT 5](#_heading=h.2et92p0)

[PROBLEM STATEMENT 6](#_heading=h.tyjcwt)

[OVERVIEW 7](#_heading=h.3dy6vkm)

[Program 10](#_heading=h.1t3h5sf)

[Output 20](#_heading=h.4d34og8)

[Designing 23](#_heading=h.2s8eyo1)

[∙](#_heading=h.17dp8vu) ENTITY RELATIONSHIP MODEL 23

[∙](#_heading=h.3rdcrjn) DATA FLOW DIAGRAM 25

[1. LEVEL 0 DFD 25](#_heading=h.26in1rg)

[2. LEVEL 1 DFD 26](#_heading=h.lnxbz9)

[∙](#_heading=h.35nkun2) USER CASE DIAGRAM 27

[∙](#_heading=h.1ksv4uv) DECISION TREE 28

# PROJECT TITLE

**HOSTEL**

**MANAGEMENT SYSTEM**

**To record student details**



# ACKNOWLEDGEMENT

**This satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.**

**We are grateful to our project guide Dr. Babul Prasad Tewari for guidance, inspiration and constructive suggestions that helps us in the preparation of the project.**

# 

# 

# ABSTRACT

**“Hostel Management System” is a software developed for managing various activities in the hostel. For the past few years the numbers of educational institutions are increasing rapidly. Thereby the numbers of studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.**

**Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly. We implemented a complete C++ program for “Hostel Management System” to record student’s detail.**

# 

# PROBLEM STATEMENT

**This system is designed in favour of the hostel management which helps them to save the records of the students about their rooms and the other things. It helps them from the manual work from which it is very difficult to find the record of the students and the information of about the those ones who are willing to change the hostel and rooms. All the hostels are present and are managed manually by the administration. Thus there are a lot of repetitions which can be easily avoided. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly. We improve the efficiency the system thus overcome the drawbacks of the existing system.**

# OVERVIEW

* Introduction
* Hostel management system gives the correct idea of student details, building information, rooms allotment to each student and capacity of each hostel.
* It allows admin to keep record of the student’s personal detail and also related to their hostel details.
* Prepare a file that store the record of existing students with their roll id, mobile no and their room details i.e. their respected room no and the current hostel name.
* Prepare a file to store the details about the hostels i.e. capacity, name and currently students boarding.
* Implemented a method which allow facility for transfer of rooms, hostel change.
* Rooms allotted to new students on the basis of current seat availability.
* Implementation
* Implement the hostel management system using C++ programming language.
* Uses the concept of object oriented programming and file management system to manage the details of hostel and students.
* We have designed two class namely hostel and student to record hostels and students detail respectively.

* Hostel class
* Variable

Name Data type Function

| Name | String | Name of hostel |
| --- | --- | --- |
| Capacity | Int | Total number of students the hostel can board |
| Curr\_num | Int | Number of students currently boarding in hostel |

* Methods

Name Function

| create() | To take input |
| --- | --- |
| print() | To print the values to output |
| retname() | Returns the name |
| retcap() | Returns the capacity |
| retcurr() | Returns number of students boarding |
| inc\_curr() | To increment the value of curr\_num |
| dec\_curr() | To decrement the value of curr\_num |

* Student class
* Variable

Name Data type function

| Fname | char | First name of student |
| --- | --- | --- |
| lname | char | Surname of student |
| hname | char | Name of hostel the student is currently residing at |
| roll\_no | int | Name of hostel the student is currently residing at |
| room\_no | int | Room Number of student |
| phno | int | Phone Number of student |

* Methods

Name Function

| create() | To take input |
| --- | --- |
| print() | To print the values to output |
| retroll() | Returns the roll number |
| rethname() | Returns the hostel name |
| retfname() | Returns the first name |
| change\_room(int) | Changes the value of roomno variable |
| change\_hostel(string) | Changes the value of hostel variable |

# Program

#include<iostream>

#include<fstream>

#include<limits>

#include<cstring>

**using** **namespace** std**;**

/\*--------------------------------------------------------------------------------------------------------------------------------------

Hostel : class

Variable type use

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

name string Name of hostel

capacity int Total number of students the hostel can board

curr\_num int Number of students currently boarding in hostel

Methods type use

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

create() void To take input

print() void To print the values to output

retname() string Returns the name

retcap() int Returns the capacity

retcurr() int Returns number of students boarding

inc\_curr() void To increment the value of curr\_num

dec\_curr() void To decrement the value of curr\_num

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

class Hostel**{**

char name**[**15**];**

int capacity**;**

int curr\_num**;**

public**:**

void create**(){**

cout **<<** "Enter name of hostel: "**;**

cin**.**getline**(**name**,** 15**);**

cout **<<** "Enter capacity of hostel: "**;**

cin **>>** capacity**;**

curr\_num **=** 0**;**

cin**.**ignore**(**numeric\_limits**<**streamsize**>::**max**(),**'\n'**);**

**}**

void print**(){**

cout **<<** name **<<** "\n"**;**

cout **<<** "Total capacity: " **<<** capacity **<<** "\n"**;**

cout **<<** "Number of students currently boarding: " **<<** curr\_num **<<** "\n"**;**

**}**

char**\*** retname**(){**

**return** name**;**

**}**

int retcap**(){**

**return** capacity**;**

**}**

int retcurr**(){**

**return** curr\_num**;**

**}**

void inc\_curr**(){**

curr\_num**++;**

**}**

void dec\_curr**(){**

curr\_num**--;**

**}**

**};**

/\* Declaration of find\_hostel and inc\_hostel functions

for use in Student class\*/

bool find\_hostel**(**char hostel\_name**[]);**

void inc\_hostel**(**char hname**[]);**

/\*--------------------------------------------------------------------------------------------------------------------------------------

Student : class

Variable type use

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fname string First name of student

lname string Surname of student

hname string Name of hostel the student is currently residing at

rollno int Roll number of student

roomno int Room Number of student

phno string Phone Number of student

Methods type use

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

create() void To take input

print() void To print the values to output

retroll() int Returns the roll number

rethname() string Returns the hostel name

retfname() string Returns the first name

change\_room(int) void Changes the value of roomno variable

change\_hostel(string) void Changes the value of hostel variable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

class Student**{**

char fname**[**10**];**

char lname**[**15**];**

char hname**[**15**];**

int rollno**;**

int roomno**;**

char phno**[**11**];**

public**:**

void create**(){**

cout **<<** "Enter First Name: "**;**

cin**.**getline**(**fname**,**10**);**

cout **<<** "Enter Last Name: "**;**

cin**.**getline**(**lname**,**15**);**

cout **<<** "Enter Roll Number: "**;**

cin **>>** rollno**;**

cin**.**ignore**(**numeric\_limits**<**streamsize**>::**max**(),**'\n'**);**

cout **<<** "Enter Hostel Name: "**;**

cin**.**getline**(**hname**,**15**);**

**while(!**find\_hostel**(**hname**)){**

cout **<<** "Hostel not found, please try again:\n"**;**

cin**.**getline**(**hname**,**15**);**

**}**

inc\_hostel**(**hname**);**

cout **<<** "Enter Room Number: "**;**

cin **>>** roomno**;**

cout **<<** "Enter Phone Number: "**;**

cin**.**ignore**(**numeric\_limits**<**streamsize**>::**max**(),**'\n'**);**

cin**.**getline**(**phno**,**11**);**

**}**

void print**(){**

cout **<<** fname **<<** " " **<<** lname **<<** "\n"**;**

cout **<<** rollno **<<** "\n"**;**

cout **<<** "Hostel name: " **<<** hname **<<** "\n"**;**

cout **<<** "Room Number: " **<<** roomno **<<** "\n"**;**

cout **<<** "Phone Number: " **<<** phno **<<** "\n"**;**

**}**

int retroll**(){**

**return** rollno**;**

**}**

char**\*** rethname**(){**

**return** hname**;**

**}**

char**\*** retfname**(){**

**return** fname**;**

**}**

void change\_room**(**int room**){**

cout **<<** "Old Room: " **<<** roomno **<<** "\n"**;**

roomno **=** room**;**

cout **<<** "New Room: " **<<** roomno **<<** "\n"**;**

**}**

void change\_hostel**(**char new\_hname**[**15**]){**

strcpy**(**hname**,**new\_hname**);**

**}**

**};**

/\* Function used to find a hostel from file and use

inc\_curr() function on it \*/

void inc\_hostel**(**char hname**[]){**

Hostel ht**;**

fstream f**;**

f**.**open**(**"hostel.dat"**,**ios**::**out**|**ios**::**in**);**

**while(**f**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**ht**.**retname**(),**hname**)** **==** 0**){**

ht**.**inc\_curr**();**

int pos **=** **-**1 **\*** **sizeof(**ht**);**

f**.**seekp**(**pos**,**ios**::**cur**);**

f**.**write**((**char**\*)&**ht**,sizeof(**ht**));**

**}**

**}**

f**.**close**();**

cout **<<** "..........................\n"**;**

**}**

/\* Function used to find a hostel from file and use

dec\_curr() function on it \*/

void dec\_hostel**(**char hname**[]){**

Hostel ht**;**

fstream f**;**

f**.**open**(**"hostel.dat"**,**ios**::**out**|**ios**::**in**);**

**while(**f**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**ht**.**retname**(),**hname**)** **==** 0**){**

ht**.**dec\_curr**();**

int pos **=** **-**1 **\*** **sizeof(**ht**);**

f**.**seekp**(**pos**,**ios**::**cur**);**

f**.**write**((**char**\*)&**ht**,sizeof(**ht**));**

**}**

**}**

f**.**close**();**

cout **<<** "..........................\n"**;**

**}**

/\* Function to be called if a student wishes to

transfer rooms\*/

void transfer\_rooms**(**int roll**){**

bool flag **=** **false;**

int new\_room**;**

Student st**;**

fstream f**;**

f**.**open**(**"student.dat"**,**ios**::**out**|**ios**::**in**);**

**while(**f**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**roll **==** st**.**retroll**()){**

cout **<<** "Enter the new room number: "**;**

cin **>>** new\_room**;**

st**.**change\_room**(**new\_room**);**

int pos **=** **-**1 **\*** **sizeof(**st**);**

f**.**seekp**(**pos**,**ios**::**cur**);**

f**.**write**((**char**\*)&**st**,sizeof(**st**));**

flag **=** **true;**

**}**

**}**

**if(!**flag**){**

cout **<<** "No student with given roll number\n"**;**

**}**

f**.**close**();**

cout **<<** "..........................\n"**;**

**}**

/\* Function to be called if a student wishes to

transfer hostels\*/

void transfer\_hostel**(**int roll**){**

char hname**[**15**];**

bool flag **=** **false;**

fstream f**;**

Student st**;**

f**.**open**(**"student.dat"**,**ios**::**in**);**

**while(**f**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**roll **==** st**.**retroll**()){**

flag **=** **true;**

**}**

**}**

f**.**close**();**

**if(!**flag**){**

cout **<<** "No student with given roll number\n"**;**

**return;**

**}**

flag **=** **false;**

cout **<<** "Enter new hostel name: "**;**

cin**.**getline**(**hname**,**15**);**

**while(!**find\_hostel**(**hname**)){**

cout **<<** "Hostel not found, please try again:\n"**;**

cin**.**getline**(**hname**,**15**);**

**}**

f**.**open**(**"student.dat"**,**ios**::**out**|**ios**::**in**);**

**while(**f**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**roll **==** st**.**retroll**()){**

dec\_hostel**(**st**.**rethname**());**

inc\_hostel**(**hname**);**

st**.**change\_hostel**(**hname**);**

int room**;**

cout **<<** "Enter new room number: "**;**

cin **>>** room**;**

st**.**change\_room**(**room**);**

int pos **=** **-**1 **\*** **sizeof(**st**);**

f**.**seekp**(**pos**,**ios**::**cur**);**

f**.**write**((**char**\*)&**st**,sizeof(**st**));**

**}**

**}**

f**.**close**();**

cout **<<** "..........................\n"**;**

**}**

/\* Function used to create Student object

and save it to file\*/

void write\_student**(){**

Student st**;**

st**.**create**();**

ofstream out**;**

out**.**open**(**"student.dat"**,** ios**::**app**);**

out**.**write**((**char**\*)&**st**,** **sizeof(**st**));**

out**.**close**();**

**}**

/\* Function used to create Hostel object

and save it to file\*/

void write\_hostel**(){**

Hostel ht**;**

ht**.**create**();**

ofstream out**;**

out**.**open**(**"hostel.dat"**,** ios**::**app**);**

out**.**write**((**char**\*)&**ht**,sizeof(**ht**));**

out**.**close**();**

**}**

/\* Prints all Student objects in file to output\*/

void print\_all\_students**(){**

Student st**;**

ifstream in**;**

in**.**open**(**"student.dat"**);**

**while(**in**.**read**((**char**\*)&**st**,sizeof(**st**))){**

st**.**print**();**

cout **<<** "..........................\n"**;**

**}**

in**.**close**();**

**}**

/\* Prints all Hostel objects in file to output\*/

void print\_all\_hostel**(){**

Hostel ht**;**

ifstream in**;**

in**.**open**(**"hostel.dat"**);**

**while(**in**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

ht**.**print**();**

cout **<<** "..........................\n"**;**

**}**

in**.**close**();**

**}**

/\* Finds and prints all students who live in given hostel\*/

void find\_print\_students**(**char hostel\_name**[]){**

bool flag **=** **false;**

Student st**;**

ifstream in**;**

in**.**open**(**"student.dat"**);**

**while(**in**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**strcmp**(**st**.**rethname**(),**hostel\_name**)** **==** 0**){**

st**.**print**();**

cout **<<** "..........................\n"**;**

flag **=** **true;**

**}**

**}**

in**.**close**();**

**if(!**flag**){**

cout **<<** "No student or hostel doesn't exist\n"**;**

**}**

**}**

/\* Finds and prints the student with

given roll number \*/

void find\_print\_student**(**int roll**){**

bool flag **=** **false;**

Student st**;**

ifstream in**;**

in**.**open**(**"student.dat"**);**

**while(**in**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**st**.**retroll**()** **==** roll**){**

st**.**print**();**

cout **<<** "..........................\n"**;**

flag **=** **true;**

**}**

**}**

in**.**close**();**

**if(!**flag**){**

cout **<<** "No student with given roll number\n"**;**

**}**

**}**

/\* Finds and prints all students with given first name\*/

void find\_print\_student**(**char first\_name**[]){**

bool flag **=** **false;**

Student st**;**

ifstream in**;**

in**.**open**(**"student.dat"**);**

**while(**in**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**strcmp**(**st**.**retfname**(),**first\_name**)** **==** 0**){**

st**.**print**();**

cout **<<** "..........................\n"**;**

flag **=** **true;**

**}**

**}**

in**.**close**();**

**if(!**flag**){**

cout **<<** "No student with given first name\n"**;**

**}**

**}**

/\* Function used to search for a Hostel object and

returns true if that object exists in the file\*/

bool find\_hostel**(**char hostel\_name**[]){**

Hostel ht**;**

ifstream in**;**

in**.**open**(**"hostel.dat"**);**

**while(**in**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**ht**.**retname**(),**hostel\_name**)** **==** 0**){**

**return** **true;**

**}**

**}**

in**.**close**();**

**return** **false;**

**}**

/\* Finds and prints the hostel with given name\*/

void find\_print\_hostel**(**char hostel\_name**[]){**

bool flag **=** **false;**

Hostel ht**;**

ifstream in**;**

in**.**open**(**"hostel.dat"**);**

**while(**in**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**ht**.**retname**(),**hostel\_name**)** **==** 0**){**

ht**.**print**();**

cout **<<** "..........................\n"**;**

flag **=** **true;**

**}**

**}**

in**.**close**();**

**if(!**flag**){**

cout **<<** "Hostel doesn't exist\n"**;**

**}**

**}**

/\* Removes a student from file if it exists\*/

void rem\_student**(**int roll**){**

bool flag **=** **false;**

Student st**;**

fstream f1**,**f2**;**

f1**.**open**(**"student.dat"**,**ios**::**in**|**ios**::**out**);**

f2**.**open**(**"temp.dat"**,**ios**::**out**);**

**while(**f1**.**read**((**char**\*)&**st**,sizeof(**st**))){**

**if(**st**.**retroll**()** **==** roll**){**

dec\_hostel**(**st**.**rethname**());**

flag **=** **true;**

**}**

**else{**

f2**.**write**((**char**\*)&**st**,sizeof(**st**));**

**}**

**}**

f1**.**close**();**

f2**.**close**();**

remove**(**"student.dat"**);**

rename**(**"temp.dat"**,**"student.dat"**);**

**if(!**flag**){**

cout **<<** "Student with given roll number doesn't exist\n"**;**

**}**

**else{**

cout **<<** "Student Removed\n"**;**

**}**

cout **<<** "..........................\n"**;**

**}**

/\* Removes a hostel from file if it exists

and has no students living in it currently\*/

void rem\_hostel**(**char hname**[]){**

bool flag **=** **false;**

Student st**;**

Hostel ht**;**

fstream f**,** f1**,**fp**;**

fp**.**open**(**"hostel.dat"**,**ios**::**in**);**

**while(**fp**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**hname**,**ht**.**retname**())** **==** 0**){**

**if(**ht**.**retcurr**()** **>** 0**){**

cout **<<** "Student(s) still present in hostel\n"**;**

**return;**

**}**

**}**

**}**

f**.**open**(**"hostel.dat"**,**ios**::**in**|**ios**::**out**);**

f1**.**open**(**"temp.dat"**,**ios**::**out**);**

**while(**f**.**read**((**char**\*)&**ht**,sizeof(**ht**))){**

**if(**strcmp**(**hname**,**ht**.**retname**())** **==** 0**){**

flag **=** **true;**

**}**

**else{**

f1**.**write**((**char**\*)&**ht**,sizeof(**ht**));**

**}**

**}**

f**.**close**();**

f1**.**close**();**

remove**(**"hostel.dat"**);**

rename**(**"temp.dat"**,**"hostel.dat"**);**

**if(!**flag**){**

cout **<<** "Given hostel doesn't exist\n"**;**

**}**

**else{**

cout **<<** "Hostel Removed\n"**;**

**}**

cout **<<** "..........................\n"**;**

**}**

/\* Creates the menu of options to be displayed\*/

void create\_menu**(){**

cout **<<** "\n\t\t\tSelect an option"**;**

cout **<<** "\n\t\t\t1) Create new student entry"**;**

cout **<<** "\n\t\t\t2) Create new hostel entry"**;**

cout **<<** "\n\t\t\t3) Print all student entries"**;**

cout **<<** "\n\t\t\t4) Print all hostel entries"**;**

cout **<<** "\n\t\t\t5) Print all student entries residing in a given hostel"**;**

cout **<<** "\n\t\t\t6) Search for student entry with given roll number"**;**

cout **<<** "\n\t\t\t7) Print all student entries with the same first name"**;**

cout **<<** "\n\t\t\t8) Search for hostel entry with given name"**;**

cout **<<** "\n\t\t\t9) Change the room of student"**;**

cout **<<** "\n\t\t\t10) Change the hostel of student"**;**

cout **<<** "\n\t\t\t11) Remove a student entry"**;**

cout **<<** "\n\t\t\t12) Remove a hostel entry"**;**

cout **<<** "\n\t\t\t13) Quit"**;**

cout **<<** "\n\t\t\t Any other button to print this menu again"**;**

cout **<<** "\n"**;**

**}**

/\* Checks if the hostel.dat file is empty or not created\*/

bool isempty**(){**

bool res**;**

ifstream f**(**"hostel.dat"**);**

res **=** f**.**is\_open**();**

**if(**f**){**

f**.**seekg**(**0**,**ios**::**end**);**

size\_t size **=** f**.**tellg**();**

**if(**size **==** 0**){**

res **=** **false;**

**}**

**}**

f**.**close**();**

**return** **(!**res**);**

**}**

int main**(){**

cout **<<** "\n\t\t\tWelcome.....\n"**;**

**if(**isempty**()){**

cout **<<** "The hostel file is empty\n"**;**

cout **<<** "Please enter hostel data...\n"**;**

write\_hostel**();**

**}**

create\_menu**();**

int op**;**

bool quit **=** **false;**

**while(!**quit**){**

cin **>>** op**;**

**switch(**op**){**

**case** 1**:** write\_student**();** **break;**

**case** 2**:** write\_hostel**();** **break;**

**case** 3**:** print\_all\_students**();** **break;**

**case** 4**:** print\_all\_hostel**();** **break;**

**case** 5**:** **{**

cout **<<** "Enter name of hostel: "**;**

char hname**[**15**];**

cin**.**getline**(**hname**,**15**);**

find\_print\_students**(**hname**);**

**break;**

**}**

**case** 6**:{**

int roll**;**

cout **<<** "Enter roll number: "**;**

cin **>>** roll**;**

find\_print\_student**(**roll**);**

**break;**

**}**

**case** 7**:{**

char fname**[**10**];**

cout **<<** "Enter first name: "**;**

cin**.**getline**(**fname**,**10**);**

find\_print\_student**(**fname**);**

**break;**

**}**

**case** 8**:{**

char hname**[**15**];**

cout **<<** "Enter name of hostel: "**;**

cin**.**getline**(**hname**,**15**);**

find\_print\_hostel**(**hname**);**

**break;**

**}**

**case** 9**:{**

int roll**;**

cout **<<** "Enter roll number of student: "**;**

cin **>>** roll**;**

transfer\_rooms**(**roll**);**

**break;**

**}**

**case** 10**:{**

int roll**;**

cout **<<** "Enter roll number of student: "**;**

cin **>>** roll**;**

transfer\_hostel**(**roll**);**

**break;**

**}**

**case** 11**:{**

int roll**;**

cout **<<** "Enter roll number of student: "**;**

cin **>>** roll**;**

rem\_student**(**roll**);**

**break;**

**}**

**case** 12**:{**

char hname**[**15**];**

cout **<<** "Enter name of hostel: "**;**

cin**.**getline**(**hname**,**15**);**

rem\_hostel**(**hname**);**

**break;**

**}**

**case** 13**:{**

quit **=** **true;**

**break;**

**}**

**default:** create\_menu**();**

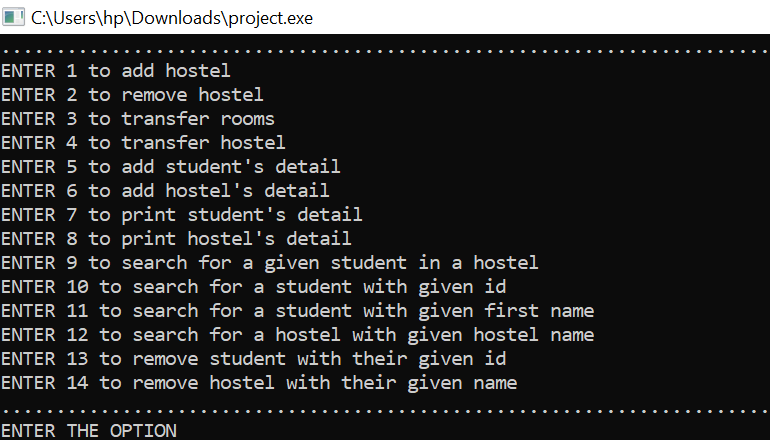
**}**

**}**

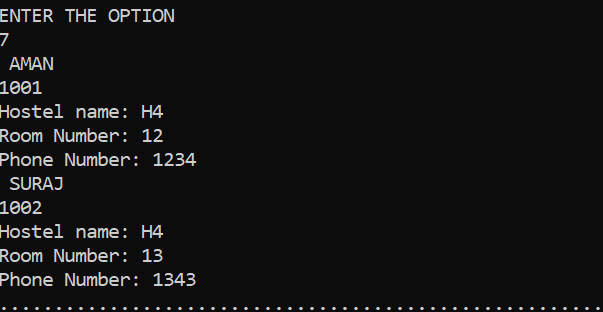
**}**

## Output

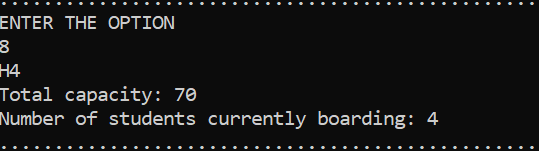
* **Menu which enlist all the options.**



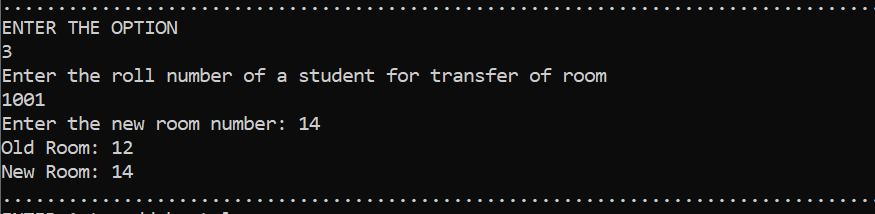
* Created a student.dat file which record the student’s detail.



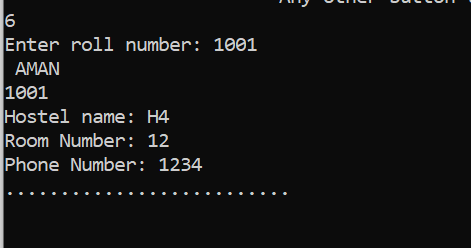
* **Created hostel.dat file which record the details of Hostels.**



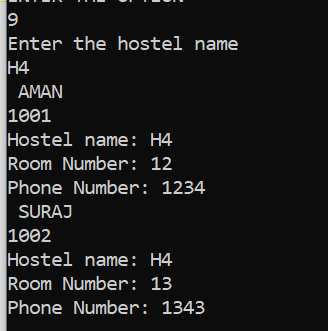
* **For room transfer**



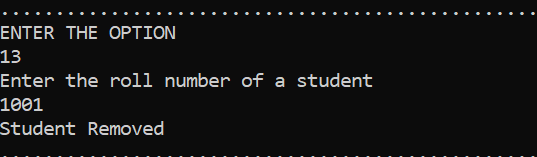
* Show the details of allotee belongs with hostel details through their id.



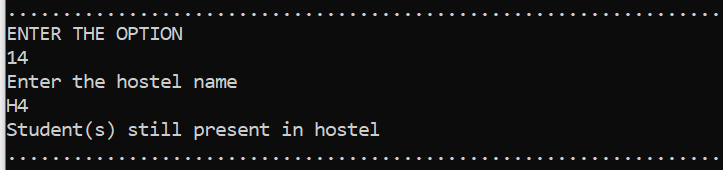
* Show the details of student with given hostel name



* Remove student with given id.



* Remove details hostel if no one is present in hostel.



# Designing

# ENTITY RELATIONSHIP MODEL

* Shows the ER model for Hostel Management System.
* Describes the relationship between entities Hostel, Student and Administration with their related attributes.



















* It is a high-level data **model**
* **model** is used to define the data elements and relationship for a specified system
* It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
* Basic components











# DATA FLOW DIAGRAM

## 1. LEVEL 0 DFD

* Shows the context diagram which give the overview of Hostel Management System .
* Single Bubble represent the entire system with its relationship to external entities.
* Input/output details are represented by single arrows.





Students information





Manage hostel’s details Record of Room details



## 2. LEVEL 1 DFD

* The context diagram is decomposed into multiple bubbles/process .

Retrieve 



hostel details

Student details 

















Available rooms 

Admin details 

* COMPONENTS

| Data flow  Storage details  Process |
| --- |

# USER CASE DIAGRAM

* **Representation of a user’s interaction with the system that shows the relationship between the user and different use cases in which the user is involved.**
* **It mainly have given components :-**
* **Actors which is the user .**
* **System boundary**
* **Relationships**
* **Given diagram shows the interaction between two users i.e. student and administration.**
* **Here actors are the users their relation with different use cases.**

|  |
| --- |











Student Administration

# DECISION TREE

* **It is flowchart in which each internal node represent the test on an attribute and branch corresponds to the decision taken on the basis of outcome of the test.**
* **Each leaf denotes the final action after the decision taken.**
* **Following decision tree shows the Allotment system for a hostel which allow the student to change the room/hostel on account of final decision taken by the administration.**





No Yes







No Yes No Yes

