

HARDIK JAIN(2020A7PS0102P)

UTSAV GOEL(2020A7PS0984P)

HOSTEL ALLOCATION PORTAL

1. System requirement specification (SRS)

1. Introduction

1.1 Objective

The main objective of this will be to deal with the Hostel Management Portal in an efficient way.

1.2 Scope of the project

- This portal is designed for allocation of hostels to students according to their preference in universities and colleges..
- The student database contains the required information for the students that have to be allocated a room.
- After inserting the hostel preferences and four wingmates , a wing is randomly allocated based upon their choices.

2. Overall Description

2.1 System Requirement

We need latest version of Chrome on the system to run sql files, a software to run javascript and jquerys and mysql to run sql queries.

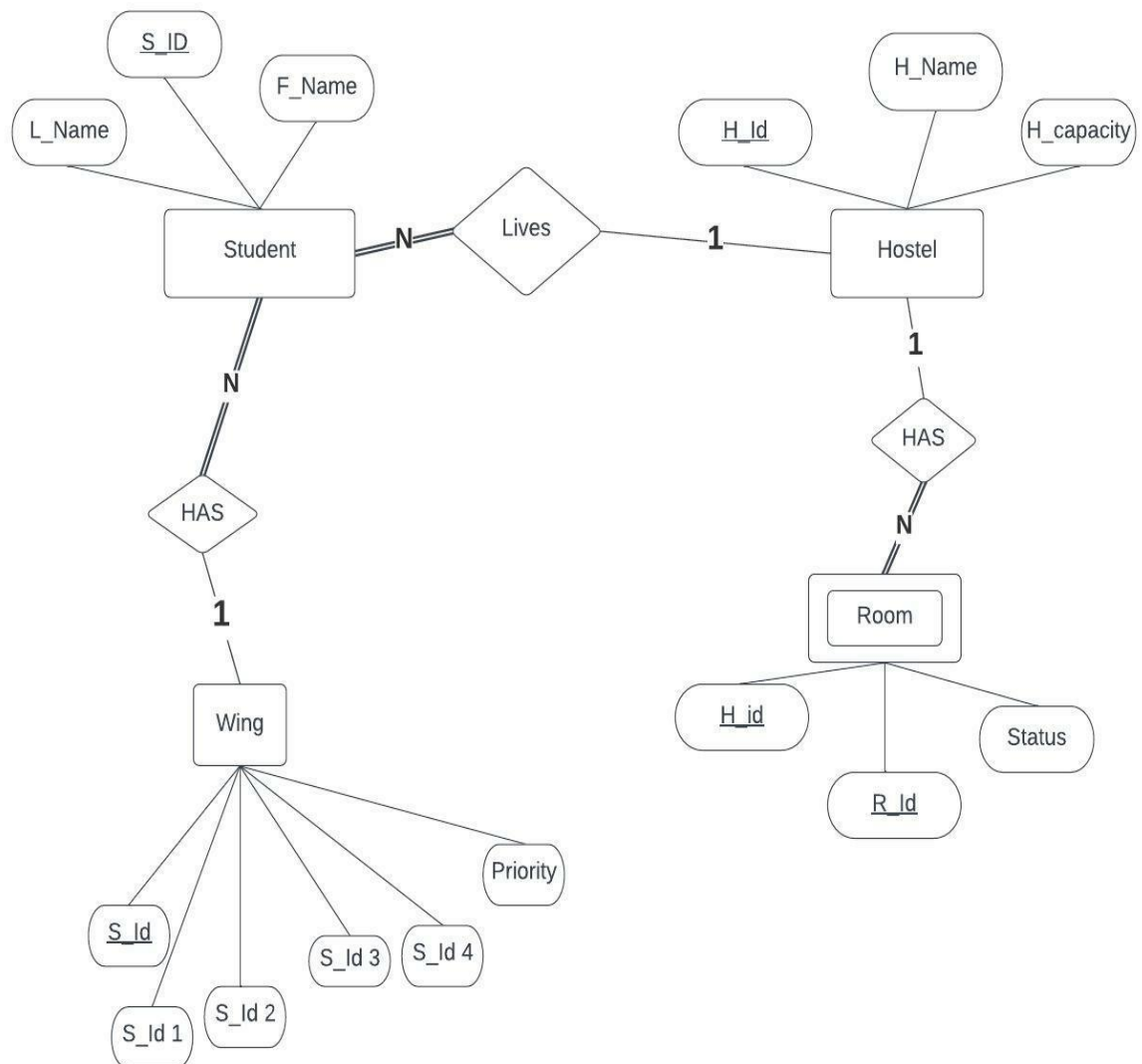
3. User Requirements Definition

The major User Requirements from the Hostel Management Portal are as follows

1. Less human error
2. Strength and strain of manual labor can be reduced
3. Data consistency
4. Data redundancy can be avoided to some extent

2. SYSTEM MODELLING

a)ER Model



b) Schema Design

1)

Student	<u>S_Id</u>	F_Name	L_Name
---------	-------------	--------	--------

The Relation Student consists of:-

Student Id(INT) , First name(Char) , last name of student(Char).

Here primary key of the table is S_Id.

First name is compulsory for each student whereas last name is the choice of student.

2)

Hostel	<u>H_Id</u>	H_name	H_cap
--------	-------------	--------	-------

The Relation Hostel consists of :-

Hostel Id(Int) , Hostel name(Char) and Hostel capacity(Int).

Here primary key is H_Id.

3)

Rooms	<u>H_Id</u>	<u>R_Id</u>	Status
-------	-------------	-------------	--------

The Relation rooms consists of:-

Hostel Id(INT) , room Id(INT) and status of the room(INT) i.e it is allotted or empty.

Here Primary key is (H_Id + R_Id).

And foreign key is a H_Id.

4)

Wing	<u>S_Id</u>	<u>S_Id1</u>	<u>S_Id2</u>	<u>S_Id3</u>	<u>S_Id4</u>	Priority
------	-------------	--------------	--------------	--------------	--------------	----------

The relation Wing consists of :-

S_Id(Id of Leader of wing) ,

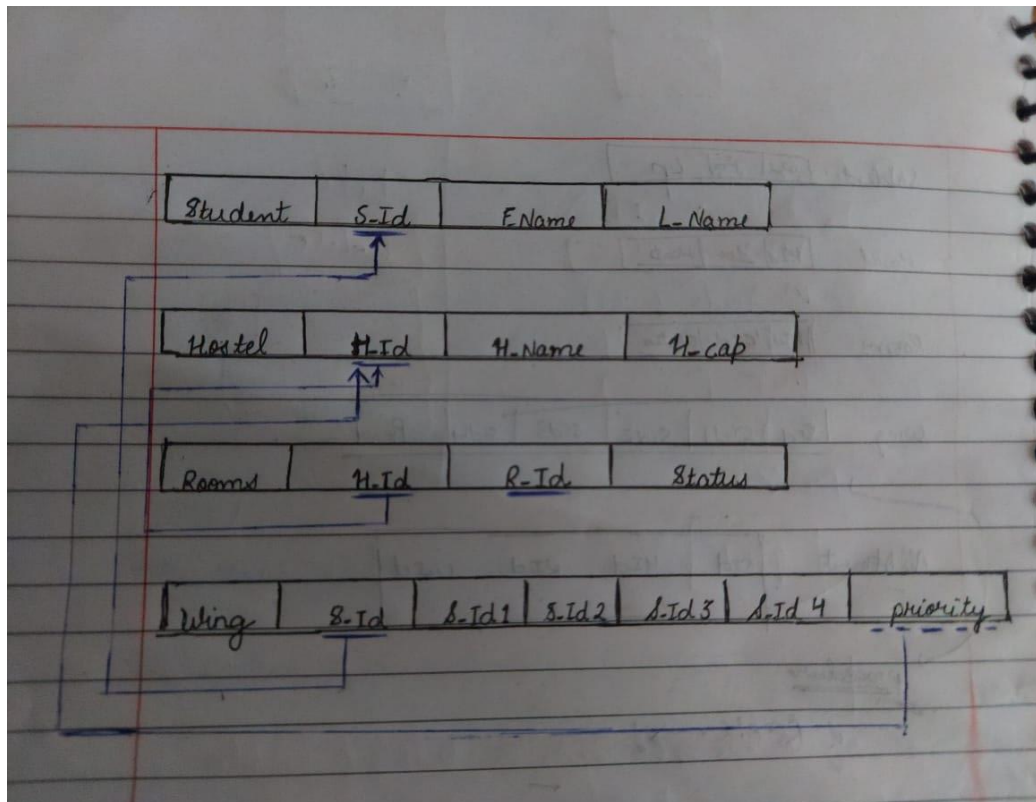
S_id1 , S_id2 , S_id3, S_id4 (Id's of members of wing)

Priority(Which hostel the want , they need to fill in the Id of that hostel).

Here S_id is the primary key and Priority is the foreign key.

Above is the description of 4 tables used.

The diagram shown below in the Schema diagram.....



c) Data Normalisation

The Er diagram and Schema design and diagram I have made are after normalisation.

Some of the major redundancies I removed are as follows:-

1. I am using Student Id in every other relation , to avoid saving student name again and again in every other table.
2. I am using Hostel Id in every other relation , to avoid saving Hostel name and capacity again and again in every other table.
3. To avoid writing wing id again and again in front of every student of the respective wing , I added all students with the same wing id and same hostel preference in the same row.
4. In rooms table , it was necessary to repeat Hostel id again and again because room number 1 is present in all the hostels , if I used the row representation of rooms i.e all rooms of the same hostel in the same row then it would be very chaotic to update the Status of room for all three hostels.

d) Tables Required :-

- Student (S_id , F_name, L_name)
- Hostel (H_id,H_name,H_cap)
- Rooms (H_id , R_id, Status)
- Wing(S_id, S_id1, S_id2, S_id3, S_id4, priority)
- One additional allocation table will be created to represent the allocated hostels to each respective student.

Allotment(S_id, H_id , R_id , W_id)

e) Additional components

For simplicity , I have assumed that wing leader will himself enter the name of all it's wingmates and enter his preference of the hostel rather than giving other students to join the wing.

Also, I have kept size of wing constant i.e. 5. If u enter lesser number of students, system won't accept them as a wing.