# MINI PROJECT REPORT ON HOSTEL MANAGEMENT SYSTEM

## "Key Management Protocol in Adhoc Networks"

submitted by

PRAJAKTA CHAUDHARI :- PA18
SHWETA CHOUDHARY :- PA23
JANHAVI CHAVAN :- PA62
SHIVRAJ PATIL :- PA63

under the guidance of
SHAKTI KINGER MA'AM
at


School of Computer Engineering and Technology

# HOSTEL MANAGEMENT SYSTEM

***PROBLEM DEFINITION :-**

A system is implemented which deals with student registration in a hostel and that shows hostel specification on a interface in created python and is managed by the managers(employees) which is specified in the database created in MYSQL. The system also deals with all registration steps involved namely – If the student is new , if new then allow registration by accepting correct student info and payment requirements and if student already exists in the system allow login using otp and show student details and hostel information required for student to contact or report issues.

***TOOLS AND TECHNOLOGIES USED :-**

Implementation of hostel management system using MYSQL and python.

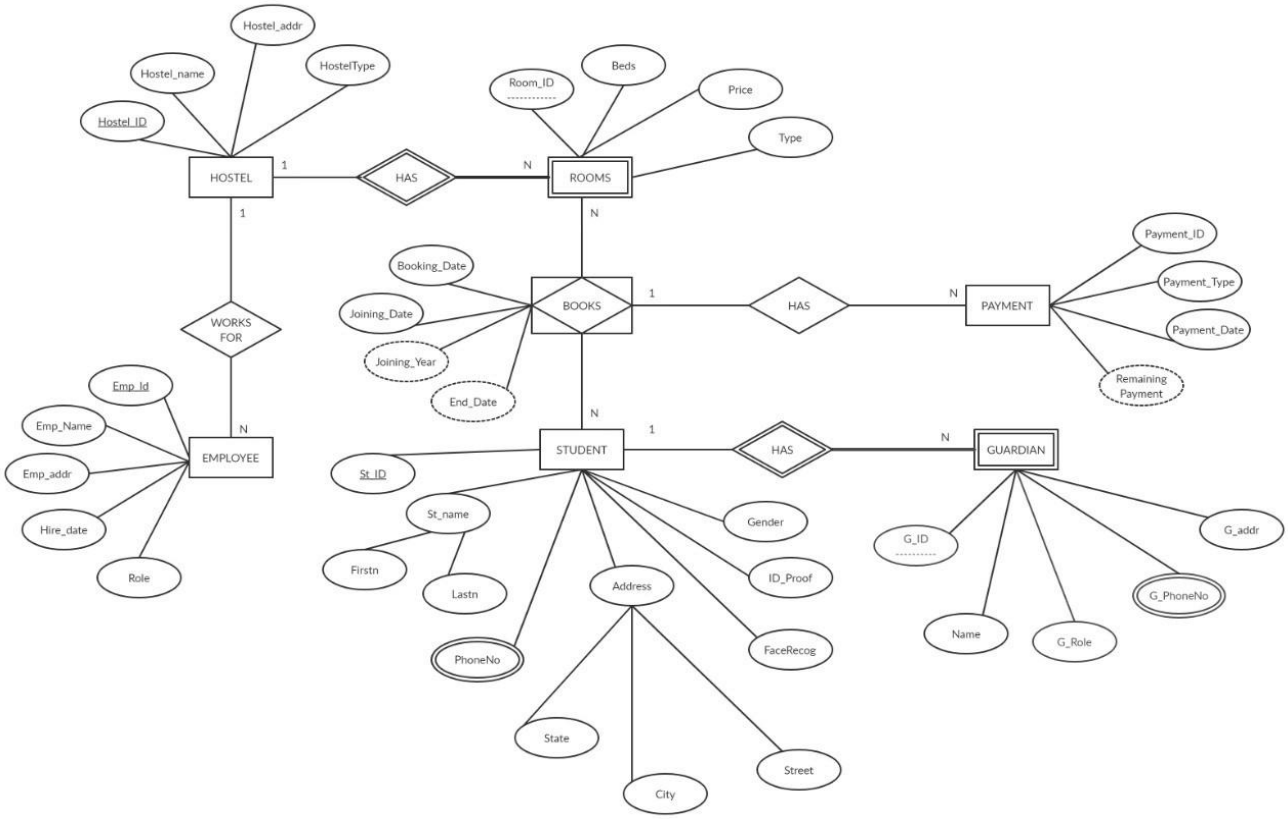
***MOTIVATION :-**

This system allows the student to register for a specific room from a particular hostel . System displays hostel details and hostel specification. Interaction with the student is done using graphical interface created in python . The system checks if student exists or not by otp login process and if does not exist it allows to register in our system by registration process .The system permits multiple students to share the room depending upon the number of beds available and student requirements. This system also uses face recognition algorithm in by image processing in python and adding those into our database for attendance purposes or any other unique identification functions.

***OBJECTIVE :-**

1. To reduce manual labor of insertion
2. To reduce keeping manual database student records
3. Insertion of new records is easy and fast in this
4. Easy to update student records by giving access to info through otp
5. Allows new students to register easily
6. Payment records / transaction maintenance is easy
7. This also allows to keep track of partial payments and due amounts
8. Access for student needs student_id and face recognition
9. Face recognition algorithm is used to improve/safe implementation

***ER – DIAGRAM :-**




*** SCHEMA :-**

HOSTEL(hostel_id,hostel_name,hostel_addr,hoteltype)

ROOM (hostel_id,room_id, beds, price,type)

EMPLOYEE (hostel_id,emp_id,emp_name,emp_addr,hiredate,role)

STUDENT (st_id,firstn,lastn,street,city,state,facerecog,id_proof,gender)

STD_PHONE (st_id,phoneno)

GUARDIAN (st_id,g_id,name,g_role,g_addr)

GD_PHONE (st_id,g_id,phoneno)

BOOKS (booking_id,hostel_id,room_id,st_id,booking_date,joining_date)

PAYEMENT (payment_id,booking_id,payment_type,amount,payment_date)

***NORMALIZATION :-**

| HOSTEL_ID | HOSTEL_NAME | ROOM_ID | PRICE | BEDS | TYPE | BOOKING_ID | BOOKING DATE |
|---|---|---|---|---|---|---|---|
| 101 | Vidya | 2001 | 105000 | 2 | Normal | 1 | 2018-08-12 |
| 101 | Sarthi | 3001 | 125000 | 3 | Balcony | 2 | 2019-08-10 |
| 102 | Shanti | 2001 | 75000 | 2 | Normal | 3 | 2019-08-12 |
| 102 | Avanti | 2001 | 95000 | 2 | Normal | 4 | 2019-08-13 |

| JOINING_DATE | ST_ID | ST_NAME | St_Address | Gender | PHONE_NO |
|---|---|---|---|---|---|
| 2018-09-12 | 1032170317 | Prajakta | Thane | Female | 912809183 |
| 2019-09-10 | 1032170317 | Prajakta | Thane | Female | 263173791 |
| 2019-09-12 | 1032180389 | Shreyas | Pune | Male | 737173917 |
| 2019-09-13 | 1032180546 | Savay | Solapur | Male | 958945509 |

We have different anomalies in the above table.
->Insertion anomaly: If we want to insert a room in this table we have to add the details of student as well.
->Deletion anomaly:If a student decides to leave the hostel we would have to delete the entire record the Room for that hostel also gets deleted.
->Updation anomaly:With redundant data, when we want to change the value of one columns of a particular Student, for example the ST_NAME, we must update all the Student records that assigned to the particular Hostel roomt otherwise the database will become inconsistent.

The above table is in 1NF.For it to be in 2NF we need to remove the partial dependencies and convert it into new tables.

Primery key:

HOSTEL_ID,ROOM_ID,BOOKING_ID,STUDENT_ID

Functional Dependencies:

HOSTEL_ID,ROOM_ID,BOOKING_ID,ST_ID->PRICE,BEDS,TYPES,BOOKING_DATE,JOINING_

DATE,ST_ID,ST_NAME,ST_ADDR,GENDER,PHONE_NO

HOSTEL_ID,ROOM_ID-->PRICE,BEDS,TYPE (Partial dependency)

ST_ID-->ST_NAME,ST_ADDRESS,GENDER,PHONE_NO. (Partial dependency)

2NF FORM:

Removing the partial dependency and creating a new table:

| HOSTEL_ID | ROOM_ID | PRICE | BEDS | TYPE |
|---|---|---|---|---|

| HOSTEL_ID | ROOM_ID | BOOKING_ID | ST_ID | BOOKING_DATE | JOINING_DATE |
|---|---|---|---|---|---|

| ST_ID | ST_NAME | ST_ADDR | GENDER | PHONE_NO |
|---|---|---|---|---|

3NF FORM:

We have to remove all the transitive dependencies if any.Since there are no transitive dependencies.The table is already in 3NF form.
Therefore,

BOOKING TABLE:

| HOSTEL_ID | ROOM_ID | BOOKING_ID | ST_ID | BOOKING_DATE | JOINING_DATE |
|---|---|---|---|---|---|

ROOM TABLE :-

| HOSTEL_ID | ROOM_ID | PRICE | BEDS | TYPE |
|---|---|---|---|---|

STUDENT TABLE:

| ST_ID | ST_NAME | ST_ADDR | GENDER | PHONE_NO |
|---|---|---|---|---|

***RELATIONAL SCHEMA DIAGRAM : -**