

**National Educational Society ®**



Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

**Synopsis**

**On**

***“Hostel Database Management System”***

**Submitted by,**

**Abhinav Mourya 4JN21CS002**

**Bhuvan S Maligi 4JN21CS033**

**Chandan S 4JN21CS035**

**Dhananjay V Hegde 4JN21CS040**

*Students of V semester BE under the supervision and guidance of*

**Mr. S Sathyanarayana BE, M.Sc (IT), M.Tech Dr. Poornima K M M.Tech, Ph.D**

Asst. Professor Dept. of CS&E Professor, CS&E

**2023-24**

**Introduction Problem Statement**

There are a lot of drawbacks in keeping and maintaining a hostel. Especially with a manual system. Since most hostels are being run by only one hostel manager, the number of students in a room are sometimes not known by the officer. He has to go room by room to ensure that a room is occupied or not. Sometimes people may be owing in the hostel and they are saved on papers or huge notebooks, and sometimes receipts. If the books should go missing or stolen, one would never be able to know if a student is owing or not. Room allocation also becomes a problem as the officer might not know which rooms are available or not. And some hostels have a lot of rooms or have mare storeys and it would be very tedious to go through all storeys in search of a free room for an applicant. Also the officer might not know the number of students in a room or know if a room is full or not.

**Objectives of the Project**

1. To make it easier for data collection, storage and referencing reliable.
2. To maintain the students as hostellers and waiting list students separately.
3. To process allotment list.

**Modules**

* Student (USN, Name, Mobile, Room no., Guardian, Guardian Name, Email, Address, Aadhar, Status)
* Room (R\_number, Room Status, Activate)
* Payment (Mobile\_No, Month, Amount)
* Fees (Amount, Due\_Month, USN)
* Employee (Status, Name, Father, Mobile\_No, Mother, Email, Address, Aadhar, Designation)

**Design and Implementation**

In this chapter there we specify the entities, the attributes selected and from that how the design has been achieved to provide the ER diagram and how the schema diagram is evolved.

**ER DIAGRAM**

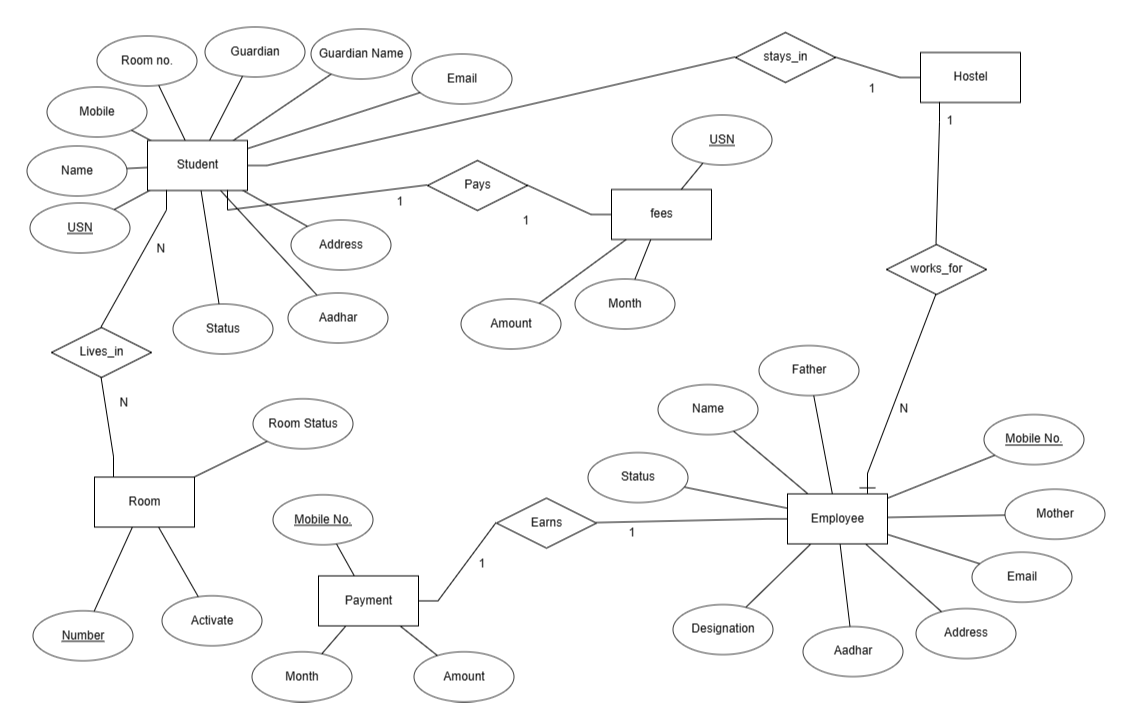
* ER Diagram is a graphical representation that depicts relationships among people, objects, places, concepts or events within the system.
* ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships. ERDs are generally depicted in one or more of the following models.

**Creation of an ER Diagram**

A conceptual data model, which lacks specific detail but provides an overview of the scope of the project and how data sets relate to one another.

A logical data model, which is more detailed than a conceptual data model, illustrating specific attributes and relationships among data points. While a conceptual data model does not need to be designed before a logical data model, a physical data model is based on a logical data model. A physical data model, which provides the blueprint for a physical manifestation – such as a relational database -- of the logical data model. One or more physical data models can be developed based on a logical data model.

**Hostel database management ER Diagram**

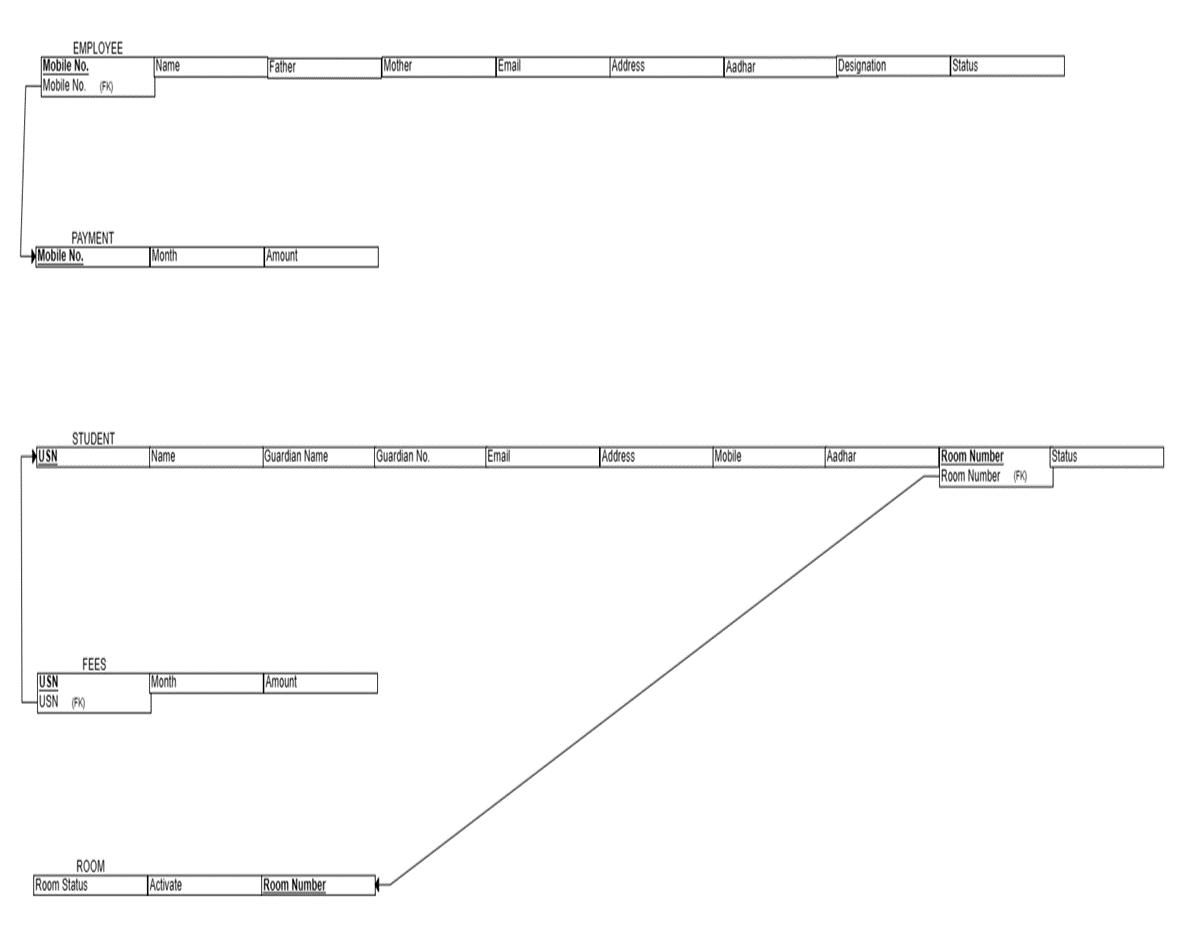


**ER Diagram of Hostel database**

**Schema Diagram**

A schema is an outline, diagram, or model. In computing, schemas are often used to describe the structure of different types of data. It also used to represent the primary key and the foreign key.

**Schema Description:**

* Employee: it contains attributes Mobile No., Father, Mother, Email, Address, Designation, Status.
* Payment: it contains attributes Mobile No. , Month, Amount.
* Student: it contains attributes USN, Name, Guardian Name , Guardian No., Email, Address, Mobile, Aadhar , Room Number, Status.
* Fess: it contains attributes USN, Month, Amount.
* Room: it contains attributes Room Number, Activate, Room Status.

**SCHEMA DIAGRAM**

**REQUIREMENT ANALYSIS AND DESIGN**

In this part we have provided the requirements for the development of the project, from the requirements you should give a high level system design, software requirements, etc.

**Basic Definition:**

Hostel management system is mainly used to maintain the Hostel, rooms, equipment and facilities provided in hostel and Students information record.

**Advantages:**

* + - * This system decreases the chance of error.
      * This system requires less time for completion of any work.
      * It is used to maintain the information such as Hostel and Student information.
      * Work load and man power required in maintenance of hostel and hostel rooms is very fast.

**Development Tools and Technologies**

**Front End**

**HTML,CSS,JAVASCRIPT(ANGULAR OR REACT)**

**Why HTML?**

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

**Features of HTML**

* It is easy to learn and easy to use.
* It is platform-independent.
* Images, videos, and audio can be added to a web page.
* Hypertext can be added to the text.
* Itis a markup language.

**Advantages of HTML**

* HTML is used to build websites.
* It is supported by all browsers.
* It can be integrated with other languages like CSS, JavaScript, etc.

**Disadvantages of HTML**

* HTML can only create static web pages. For dynamic web pages, other languages have to be used.
* A large amount of code has to be written to create a simple web page.
* The security feature is not good.

**Back End**

**MYSQL WITH NODEJS**

**Why MYSQL Access?**

MySQL is the world’s most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database.

MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com. Easy to use and easy to deploy.

Databases are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

**What Makes MySQL So Popular?**

* Open Source.
* Data Security.
* Scalability on Demand.
* Higher Efficiency.
* Integration with Windows OS
* Scalability
* Easy to Use. MySQL is an easy-to-use and flexible RDBMS.
* Secure. While choosing the right RDBMS software, the security of your data must be your priority.
* High Performance. A server cluster backs MySQL.
* Industry Standard.
* Import and Export of data in all major database system.
* Centralized Management
* Reliability
* Automating Tasks

**WHY NODEJS?**

Node.js is a server-side JavaScript runtime that enables developers to use JavaScript for both frontend and backend development. It is known for its speed, scalability, and efficiency due to its event-driven, non-blocking I/O model. Node.js is built on the V8 JavaScript engine, offering fast execution and making it suitable for applications that require handling many concurrent connections. Its extensive ecosystem (npm) and active community support contribute to its popularity. Node.js is particularly well-suited for real-time applications, scalable solutions, and microservices architectures. The use of a single programming language (JavaScript) throughout the development stack enhances code reusability and maintainability.

**Advantages of node.js**

* JavaScript Everywhere:

Node.js allows developers to use JavaScript on both the client and server sides. This enables a more consistent and seamless development experience, as developers can use the same language for both frontend and backend development.

* Single Programming Language:

Using JavaScript as the single programming language for both frontend and backend reduces the context-switching overhead for developers. They can work on the entire stack using a unified language, improving code reusability and maintainability.

* Event-Driven and Asynchronous:

Node.js is built on an event-driven, non-blocking I/O model, making it highly efficient for handling concurrent connections. This asynchronous nature allows for better scalability and responsiveness, making Node.js suitable for real-time applications.

* Fast Execution:

Node.js is built on the V8 JavaScript engine from Google, which is known for its fast execution. This translates to high-performance backend applications, particularly in scenarios where handling many concurrent connections is crucial.

* Large Ecosystem (npm):

Node.js has a vast and vibrant ecosystem of open-source packages and libraries available through npm (Node Package Manager). This makes it easy for developers to find and integrate third-party modules to extend the functionality of their applications.

* Community Support:

Node.js has a large and active community of developers. This community support is valuable for finding solutions to common problems, getting help, and staying updated on best practices and new features.

* Scalability:

Node.js is designed to be scalable, making it well-suited for applications that need to handle a large number of simultaneous connections. Its event-driven architecture allows developers to build scalable applications without the need for additional infrastructure.

* Real-time Applications:

The event-driven and asynchronous nature of Node.js makes it particularly well-suited for building real-time applications, such as chat applications, online gaming platforms, and collaborative tools.

* Cross-Platform Compatibility:

Node.js is compatible with multiple operating systems, making it easier to deploy applications across different platforms. This cross-platform compatibility contributes to the flexibility of development and deployment.

* Microservices Architecture:

Node.js is well-suited for a microservices architecture, where applications are built as a collection of small, independent services. Its lightweight nature and fast execution make it conducive to building and deploying microservices.