**Project**

**On**

**Hospital Management System**

**Prepaired By :**

**Name : Himanshu Dodrajka**

**Roll No : U18CO056**

**HOSPITAL**

A hospital is a health care institution providing patient treatment with specialized staff and equipment. Hospitals are usually funded by public sector by health organisations (for profit or non-profit), by health insurance companies, or by charities, including direct charitable donations. Hospitals have a range of departments (e.g., surgery, and urgent care etc).

**HOSPITAL MANAGEMENT SYSTEM**

A hospital management system is an information system that manages the aspects of a hospital. This may include the administrative, financial, and medical processing. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow. This program can look after Inpatients, OPD patients, records, database treatments, status illness, billings etc. it also maintains their hospital info such as ward id, Doctor in Charge, Department administering etc. Now with a laboratory module to handle all lab operations...!!! Not only has this it also looked after doctor and staff records and payments. Now with advanced features like LAN connectivity, ICD10 disease database, Webcam support.

**NEED OF HMS**

1. Minimized documentation and no duplication of records.
2. Reduced paper work.
3. Improved Patient Care
4. Better Administration Control
5. Faster information flow between various departments
6. Smart Revenue Management
7. Effective billing of various services

8. Exact stock information

**Product Function**

The data represented in hospital management application will perform the following major function:- 

* Patient Details: - It includes inpatient and outpatient details. 
* Tasks:- It includes the various tasks which are used in hospital 
* Billing Details:-This software will help to calculate the bill much quicker and simpler way. This enables the organization to keep the information in efficient and systematic way.
* Doctor Details : - It includes detail of doctor and detail of doctor attending a patient.
* Deleting and Updating Details : - It delete the patient Details which are discharged from hospital or Doctor who left the job.

It updates the condition of patient, their age , Doctor details etc.

Which changes due time.

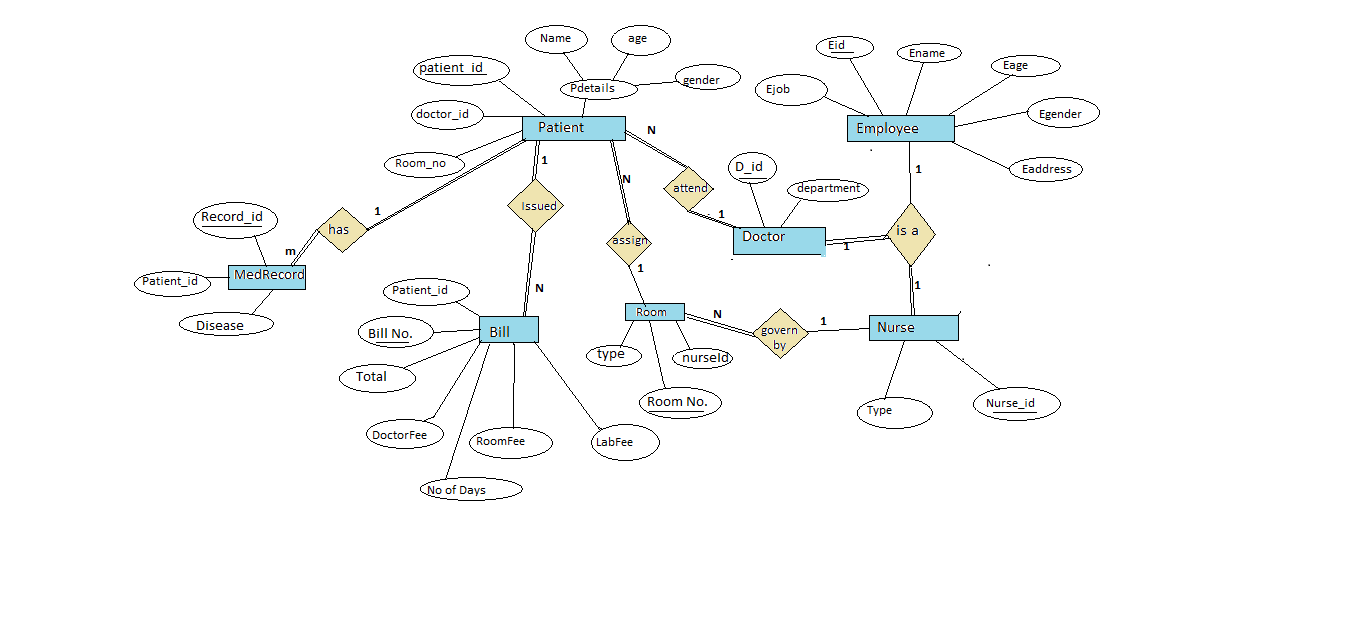
**SPECIFIC REQUIREMENTS**

It describes all the details that the software developer need to know for designing and developing the system. This section Describe the software or language used by developer.

--NO Front-End.

* Oracle live online .
* Pl SQL for Back-end programming.

**ER Diagram of Hospital Management System :**

** For normalied form**

**Note :- Address here is taken as only as city name . so it is not a composite attribute.**

**Relational Model:**

Conversion of Entity into Relational Model:

Tables:

Denormalized :

Doctor

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Doctor\_id | Id of Doctor | varchar |
| Doctor\_name | Name | varchar |
| Doctor\_age | Age | number |
| Doctor\_gender | Gender | varchar |
| Doctor\_address | Address | varchar |
| department | Department | varchar |

Nurse

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Nurse \_id | Id of Nurse | varchar |
| Nurse \_name | Name | varchar |
| Nurse \_age | Age | number |
| Nurse \_gender | Gender | varchar |
| Nurse \_address | Address | varchar |
| Nurse\_type | Type of work | varchar |

Patient

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Patient\_Id | Id of patient | varchar |
| name | Name | varchar |
| age | Age | number |
| gender | Gender | varchar |
| Room No. | Room no. Of patient | number |
| Disease | Disease to patient | varchar |
| Doctor\_id | Id of Doctor | varchar |

Room

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Room No. | Number of room | number |
| R\_type | Type of Room | varchar |
| Nurse\_id | Id of nurse | varchar |

Bill

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Patient\_Id | Id of patient | varchar |
| Bill\_no | No. On bill recipt | number |
| No\_of\_days | Number of days  Patient admitted | Number |
| Doctor\_charge | Doctor’s Fee | number |
| Lab\_charge | Lab’s Fee | number |
| Room\_charge | Room Fee | number |
| Total | Total amount | number |

As seen from above table some data can be redundant

So Normalization is require.

**Normalization of Relational Model**

**All the FD’s of table is given below:**

**For Doctor table:**

(doctor\_id,department) --> (doctor\_name,doctor\_age,doctor\_gender,doctor\_address)

**For Nurse table:**

(nurse\_id,nurse\_type) --> (nurse\_name,nurse\_age,nurse\_gender,nurse\_address)

**For Patient table:**

(patient\_id) --> (name,age,gender,room\_no,disease,doctor\_id)

**For Bill table:**

(Bill\_no) --> (no\_of\_days,patient\_id,doctor\_charge,room\_charge,lab\_charge,total)

**For Room table :**

(room\_no) --> (nurse\_id,r\_type)

**1NF:**

**Patient** table is not in 1NF state as **disease** attribute may have more than one value as a person may have more than one disease at same time.

So we decompose patient table as follows:

Patient --> Patient,MedRecord.

Patient table

(patient\_id) --> (name,age,gender,room\_no,disease,doctor\_id)

MedRecord table

(record\_id) --> (patient\_id,disease)

Patient table divide in two table Patient and MedRecord

As follows:

Patient

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Patient\_Id | Id of patient | varchar |
| name | Name | varchar |
| age | Age | number |
| gender | Gender | varchar |
| Room No. | Room no. Of patient | number |
| Doctor\_id | Id of Doctor | varchar |

MedRecord

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Patient\_Id | Id of patient | varchar |
| Disease | Disease | varchar |
| Record\_id | Id of record | varchar |

All Other Tables **Bill , Doctor , Nurse , Room** will remain same.

So 1NF contains table :

**Bill , Doctor , Nurse , Room , Patient , MedRecord.**

**2NF:**

for Doctor table candidate key is {doctor\_id,department} and for Nurse table candidate key {nurse\_id,nurse\_type}

And thier is also dependency from doctor\_id and nurse\_id too.

As Doctor and Nurse table contain partial dependecies hence it these two table are not in 2NF state .

**Converting Doctor and Nurse into 2NF:**

Doctor table can be decomposed into two table Doctor\_info , Doctor

Similarly Nurse table can be decompose into Nurse\_info , Nurse.

But both Nurse\_info and Doctor\_info contain similar data and attribute so Doctor\_info and Nurse\_info table becomes a single table Employee.

Doctor , Nurse ---> Doctor,Nurse,Employee.

Employee

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Eid | Id of em[ployee | varchar |
| Ename | Name | varchar |
| Eage | Age | number |
| Egender | Gender | varchar |
| Eaddress | Address | varchar |
| E\_type | Type Employee | varchar |

Doctor

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| doctor \_id | Id of doctor | varchar |
| Department | department | varchar |

Nurse

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Data type |
| Nurse \_id | Id of Nurse | varchar |
| Nurse\_type | Type of work | varchar |

Other Tables **Bill , Room , Patient , MedRecord** will remain same.

So 2NF contain Tables :

**Bill , Room , Patient , MedRecord , Employee , Doctor , Nurse.**

**3NF:**

**None of the above table contains transitive dependency so it is already in 3NF state.**

So 3NF contain Tables :

**Bill , Room , Patient , MedRecord , Employee , Doctor , Nurse.**

**BCNF:**

**In All of the above table the dependencies have only candidate key in left side of dependency so it is already in BCNF.**

So BCNF contain Tables :

**Bill , Room , Patient , MedRecord , Employee , Doctor , Nurse.**

**After Normalization final tables are :**

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Eid | Id of employee | varchar | Primary key |
| Ename | Name | varchar | Not NULL |
| Eage | Age | number | Not Null |
| Egender | Gender | varchar | Not Null |
| Eaddress | Address | varchar |  |
| Ejob | Type of Employee | varchar | Not NULL |

**Doctor**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| doctor \_id | Id of doctor | varchar | Primary key |
| Department | department | varchar | Not Null |

**Nurse**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Nurse \_id | Id of Nurse | varchar | Primary key |
| Nurse\_type | Type of work | varchar | Not Null |

**Patient**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Patient\_Id | Id of patient | varchar | Primary key |
| name | Name | varchar | Not NULL |
| age | Age | number | Not NULL |
| gender | Gender | varchar | Not NULL |
| Room No. | Room no. Of patient | number | Foreign key  (Room) |
| Doctor\_id | Id of Doctor | varchar | Foreign key |

**MedRecord**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Patient\_Id | Id of patient | varchar | foreign key |
| Disease | Disease to patient | varchar | Not Null |
| Record\_id | Id of record | varchar | Primary key |

**Room**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Room No. | Number of room | number | Primary key |
| R\_type | Type of Room | varchar | Not NULL |
| Nurse\_id | Id of nurse | varchar | Foreign key |

**Bill**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data type | Condition |
| Patient\_Id | Id of patient | varchar | Foreign key |
| Bill\_no | No. On bill recipt | number | Primary key |
| No\_of\_days | Number of days  Patient admitted | Number | Not NULL |
| Doctor\_charge | Doctor’s Fee | number | Not NULL |
| Lab\_charge | Lab’s Fee | number | Not NULL |
| Room\_charge | Room Fee | number | Not NULL |
| Total | Total amount | number | Not NULL |

**Key Constraints** :

**primary Keys , foreign Key , null , Unique constraints are already included in the above tables.**