

# CS 5200

## Database Management Systems

### Hospital Management System

By Anuj Anand

[GitHub Link](#)

#### Problem Statement

When the COVID pandemic hit the United States in 2020, the majority of hospitals were saturated or nearly saturated with patients and could not handle such a huge increase in such a short amount of time. One of the reasons for this is the administrative inefficiency in the billing systems.

#### Solution Statement

Billing systems are generally in charge of monitoring and managing the hospital's finances. The goal of this project is to implement an efficient database management system with a focus on billing systems. The front end user interface can be used by the admins of hospitals to quickly create, access, and maintain records (stored in the database) of patients visiting a doctor and generate bills for the respective visit. This database will contain records on the doctors, patients, medicines, bills generated for each patient, insurances, the appointments fixed between patients and doctors, prescribed medicines, etc.

#### User Data Models

The database would typically be accessed by the staff members who have administrative access to the database records. It could also be used by the doctors to look up various patients, and medicines present in the hospital. These are the user models present in the database:

##### 1) Patient

The main user data model in this project is the Patient model which stores information of all patients who have visited the hospital. This table contains information such as the name, gender, date of birth, phone number, and address.

Fields: first name, last name, gender, dob, phone number, address

Primary Key: Patient ID  
Foreign Keys: None

## 2) Doctor

This is the second user model in the database that would contain basic information of all the doctors present in the hospital. The fields include name, age and designation (which is a portable enumeration).

Fields: first name, last name, age, designation  
Primary Key: Doctor ID  
Foreign Keys: Designation -> Designation.designation  
(Designation that refers to the designation field in the Designation model)

# Domain Object Models

## 1) Appointment

This domain object model is used to keep records of the appointments fixed by each patient for each doctor. This model also serves as an associate class to map the many-to-many relationship between the patients and doctors.

Fields: Created, date, time  
Primary Key: Appt ID  
Foreign Keys: Patient ID, Doctor ID

## 2) Department

Model used to store information of the different departments in the hospital. It shares a many-to-many relationship with the user model, Doctor.

Fields: Department name, building name  
Primary Key: Department ID  
Foreign Keys: None

## 3) Medicine

Model used to store all the medicines currently available in the hospital's pharmacy. This shares a One-to-Many relationship with the domain object model, Prescribed Medicine, where this model is on the One side.

Fields: Medicine name, type, cost  
Primary Key: Medicine ID

Foreign Keys: None

#### 4) Prescribed Medicine

Model used to store all medicines prescribed to each patient in the database. This shares a One-to-Many relationship with two models - domain object model, Medicine, and user model, Patient, where this model is on the Many side.

Fields: Number of days, time

Primary Key: ID

Foreign Keys: Patient ID, Medicine ID

#### 5) Insurance

Model used to store information of all insurance providers belonging to each patient in the hospital. This shares a One-to-Many relationship with the user object model, Patient, where this model is on the Many side.

Fields: Provider name, expiry date

Primary Key: Insurance ID

Foreign Keys: Patient ID

#### 6) Bill

Model used to record every bill generated by the hospital for each patient. This shares a One-to-Many relationship with two models - the user object model, Patient, and the domain object model, Insurance, where this model is on the Many side.

Fields: Doctor charge, medicine charge

Primary Key: Bill ID

Foreign Keys: Insurance ID, Patient ID

#### 7) Lab Fee

Model used to record charges incurred by the patient for lab tests conducted in the hospital. This model inherits information from the model, Bill and so shares a One-to-One relationship with the same.

Fields: Test name, date, type, charge

Primary Key: Lab fee ID (or Bill ID)

Foreign Keys: Lab fee ID

## 8) Room Fee

Model used to record charges incurred by the patient for staying in the hospital. This model inherits information from the model, Bill and so shares a One-to-One relationship with the same.

Fields: room\_number, type, number of beds, number of days, charge

Primary Key: room Fee ID (or Bill ID)

Foreign Keys: Room fee ID

## Portable Enumeration

### 1) Designation

This is an enumeration for the designation field in the user model, Doctor.

Fields: Designation

Primary Key: Designation

Foreign Keys: None

## User Interface

The user interface was wholly developed using Python. The other two main components used were [SQLAlchemy](#) as the ORM tool and [Streamlit](#) for the front-end development. (The complete method to run the code is provided in the README file in the GitHub repo.)

### SQLAlchemy

SQLAlchemy is a library that provides a method to communicate between Python programs and databases. Generally, it's used as a Object Relational Mapper (ORM) tool that converts Python classes to tables present in relational databases and also converts function calls to SQL statements.