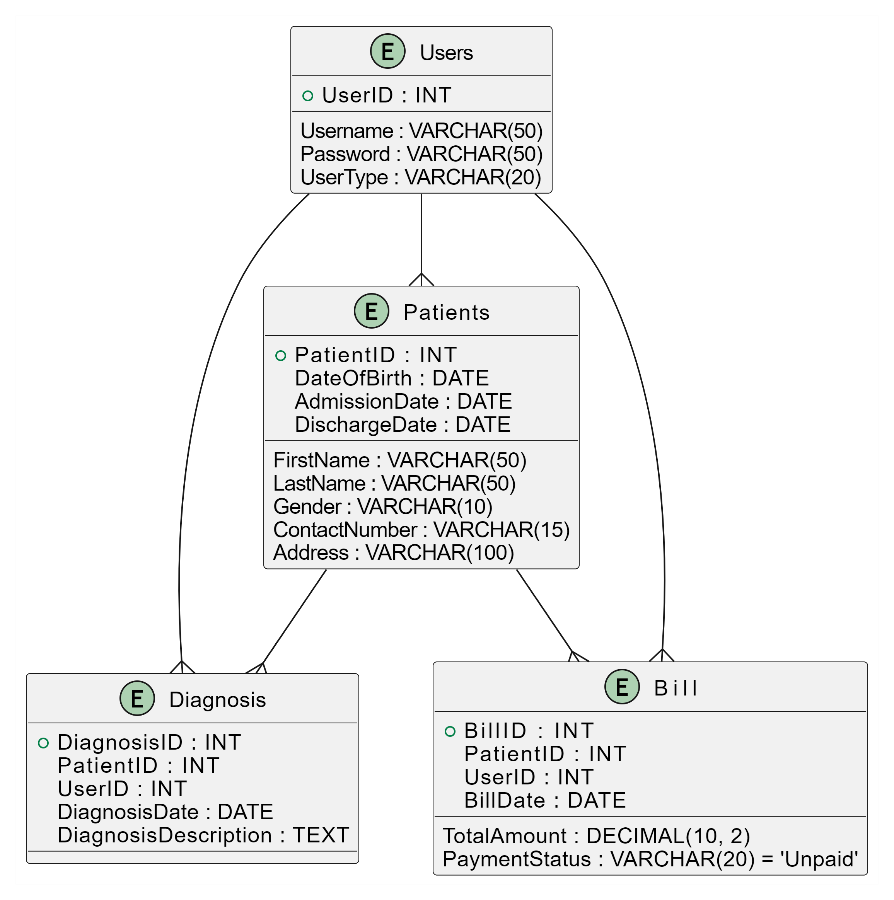
**Schema Visualization**

(tool used: draw.io | https://app.diagrams.net/)



*Fig.: Schema Design for Hospital Management System*

Users ( UserID, Username, Password, UserType )

Patients ( PatientID, FirstName, LastName, DateOfBirth, Gender, ContactNumber, Address, AdmissionDate, DischargeDate )

Diagnosis ( DiagnosisID, PatientID, UserID, DiagnosisDate, DiagnosisDescription )

Bill ( BillID, PatientID, UserID, BillDate, TotalAmount, PaymentStatus )

**Queries to create Tables**

* Creating a Table named ‘Users’

CREATE TABLE Users (

UserID INT PRIMARY KEY,

Username VARCHAR(50) NOT NULL,

Password VARCHAR(50) NOT NULL,

UserType VARCHAR(20) NOT NULL

);

* Creating a table named ‘Patients’

CREATE TABLE Patients (

PatientID INT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

DateOfBirth DATE NOT NULL,

Gender VARCHAR(10) NOT NULL,

ContactNumber VARCHAR(15),

Address VARCHAR(100),

AdmissionDate DATE,

DischargeDate DATE

);

* Creating a table named ‘Diagnosis’

CREATE TABLE Diagnosis (

DiagnosisID INT PRIMARY KEY,

PatientID INT,

UserID INT,

DiagnosisDate DATE,

DiagnosisDescription TEXT,

FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

* Creating a table named ‘Bill’

CREATE TABLE Bill (

BillID INT PRIMARY KEY,

PatientID INT,

UserID INT,

BillDate DATE,

TotalAmount DECIMAL(10, 2) NOT NULL,

PaymentStatus VARCHAR(20) DEFAULT 'Unpaid',

FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

**Indexing**

Indexing is a technique used to optimize the retrieval of records from database tables.

1. Primary Keys
   1. UserID
   2. PatientID
   3. DiagnosisID
   4. BillID
2. Foreign Keys
   1. Diagnosis Table:
      * UserID references Users(UserID)
      * PatientID references Patients(PatientID)
   2. Bill table:
      * PatientID references Patients(PatientID)
      * UserID references Users(UserID)

**Trigger**

Trigger is a set of instructions that are automatically executed in response to certain events on a particular table.

DELIMITER //

CREATE TRIGGER UpdatePaymentStatus

AFTER INSERT ON Bill

FOR EACH ROW

BEGIN

-- Check if the TotalAmount is greater than 0, and update PaymentStatus accordingly

IF NEW.TotalAmount > 0 THEN

UPDATE Bill

SET PaymentStatus = 'Paid'

WHERE BillID = NEW.BillID;

END IF;

END //

DELIMITER ;

set of instructions that are automatically executed in response to certain events on a particular ta