HOSPITAL MANAGEMENT SYSTEM

* Each entity is described along with its attributes, and primary/foreign key where applicable.

**Patient**:

* + PatientID (Primary Key)
  + Name
  + Gender
  + DOB (Date of Birth)
  + Medical History

**Doctor:**

* + DoctorID (Primary Key)
  + Name
  + Specialization
  + Shift Timing
  + DepartmentID (Foreign Key)

**Appointment:**

* + AppointmentID (Primary Key)
  + PatientID (Foreign Key)
  + DoctorID (Foreign Key)
  + Appointment Date
  + Appointment Time
  + Status

**Prescription:**

* + PrescriptionID (Primary Key)
  + DoctorID (Foreign Key)
  + PatientID (Foreign Key)
  + Date
  + Medication Name
  + Dosage

**Ward:**

* + WardID (Primary Key)
  + Ward Type
  + Ward Name
  + Availability

**Treatment:**

* + TreatmentID (Primary Key)
  + PatientID (Foreign Key)
  + DoctorID (Foreign Key)
  + Start Date
  + End Date

**Test:**

* + TestID (Primary Key)
  + PatientID (Foreign Key)
  + DoctorID (Foreign Key)
  + Test Name
  + Test Result

**Billing:**

* + BillID (Primary Key)
  + PatientID (Foreign Key)
  + Bill Date
  + Total Amount

**Inventory:**

* + ItemID (Primary Key)
  + Item Name
  + Unit Price
  + Supplier

**Department:**

* + DepartmentID (Primary Key)
  + Department Name
  + Location.
* These relationships describe how the attributes of each entity are interconnected within the Hospital Management System.

1. **Patient**:
   * One-to-Many with Appointment: Each patient can have multiple appointments.
   * One-to-Many with Prescription: Each patient can have multiple prescriptions.
   * One-to-Many with Treatment: Each patient can undergo multiple treatments.
   * One-to-Many with Test: Each patient can undergo multiple tests.
   * One-to-Many with Billing: Each patient can have multiple billing records.
2. **Doctor**:
   * One-to-Many with Appointment: Each doctor can have multiple appointments.
   * One-to-Many with Prescription: Each doctor can create multiple prescriptions.
   * One-to-Many with Treatment: Each doctor can oversee multiple treatments.
   * One-to-Many with Test: Each doctor can order multiple tests.
   * Many-to-One with Department: Each doctor works in only one department.
3. **Appointment**:
   * Many-to-One with Patient: Each appointment is associated with one patient.
   * Many-to-One with Doctor: Each appointment is associated with one doctor.
4. **Prescription**:
   * Many-to-One with Patient: Each prescription is associated with one patient.
   * Many-to-One with Doctor: Each prescription is created by one doctor.
5. **Ward**:
   * One-to-Many with Patient: Each ward can accommodate multiple patients.
6. **Treatment**:
   * Many-to-One with Patient: Each treatment is administered to one patient.
   * Many-to-One with Doctor: Each treatment is overseen by one doctor.
7. **Test**:
   * Many-to-One with Patient: Each test is conducted for one patient.
   * Many-to-One with Doctor: Each test is ordered by one doctor.
8. **Billing**:
   * Many-to-One with Patient: Each billing record is associated with one patient.
9. **Inventory**:
   * Many-to-One with Department: Each inventory item belongs to one department.
10. **Department**:
    * One-to-Many with Doctor: Each department can have multiple doctors.
    * One-to-Many with Inventory: Each department can have multiple inventory items.

**UML Diagram**

* UML class diagram representing the entities and their relationships based on the provided information:

