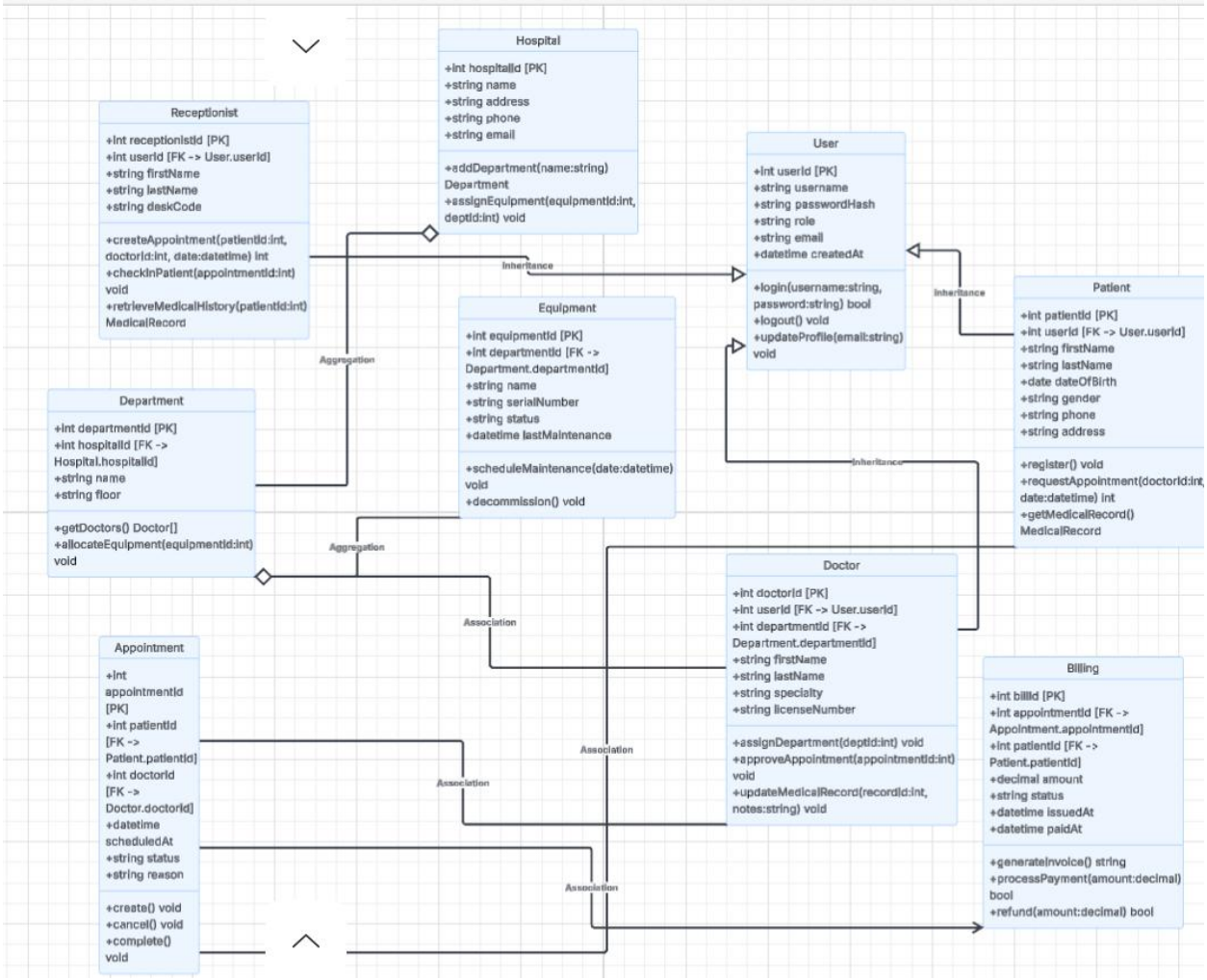

A **class diagram** represents the **static structure** of a system.
It shows:

- Classes (entities)
- Attributes and methods
- Relationships between classes
- Data ownership and dependencies

In a **Hospital Management System**, a class diagram models how hospital data such as patients, doctors, appointments, billing, and medical records are organized.

2. Key Classes Identified (11 Classes)

The system contains the following **11 core classes**:

1. **Person (Superclass)**

2. **Patient**
3. **Doctor**
4. **Receptionist**
5. **Department**
6. **Appointment**
7. **MedicalRecord**
8. **Prescription**
9. **Billing**
10. **Payment**
11. **Notification**

This satisfies the requirement of **8–12 key classes**.

3. Class Definitions (Attributes, Methods & Keys)

3.1 Person (Superclass)

Purpose: Base class for all people in the system.

Attributes

- personID : int **(PK)**
- name : String
- gender : String
- phone : String
- email : String

Methods

- login()
- logout()
- updateProfile()

Relationship

- Inherited by **Patient**, **Doctor**, and **Receptionist**

3.2 Patient (Subclass of Person)

Attributes

- patientID : int **(PK, FK → Person.personID)**
- dateOfBirth : Date
- bloodGroup : String
- address : String

Methods

- register()
- requestAppointment()
- viewMedicalHistory()

3.3 Doctor (Subclass of Person)

Attributes

- doctorID : int **(PK, FK → Person.personID)**
- specialization : String
- licenseNumber : String

Methods

- viewAppointments()
- diagnosePatient()
- writePrescription()

3.4 Receptionist (Subclass of Person)

Attributes

- receptionistID : int **(PK, FK → Person.personID)**
- shift : String

Methods

- registerPatient()
- scheduleAppointment()

- generateBill()

3.5 Department

Attributes

- departmentID : int **(PK)**
- departmentName : String
- location : String

Methods

- addDoctor()
- removeDoctor()

Relationships

- **Aggregation** with Doctor

3.6 Appointment

Attributes

- appointmentID : int **(PK)**
- appointmentDate : Date
- appointmentTime : Time
- status : String
- patientID : int **(FK → Patient.patientID)**
- doctorID : int **(FK → Doctor.doctorID)**

Methods

- createAppointment()
- cancelAppointment()
- updateStatus()

3.7 MedicalRecord

Attributes

- recordID : int **(PK)**

- diagnosis : String
- treatment : String
- recordDate : Date
- patientID : int **(FK → Patient.patientID)**

Methods

- addRecord()
- updateRecord()
- viewRecord()

Relationship

- **Composition** with Patient

3.8 Prescription

Attributes

- prescriptionID : int **(PK)**
- medicationDetails : String
- dosage : String
- recordID : int **(FK → MedicalRecord.recordID)**

Methods

- generatePrescription()
- updatePrescription()

3.9 Billing

Attributes

- billID : int **(PK)**
- billDate : Date
- totalAmount : double
- patientID : int **(FK → Patient.patientID)**

Methods

- calculateCharges()
- generateInvoice()

3.10 Payment

Attributes

- paymentID : int **(PK)**
- paymentDate : Date
- paymentMethod : String
- amountPaid : double
- billID : int **(FK → Billing.billID)**

Methods

- processPayment()
- verifyPayment()

Relationship

- **Association** with Billing

3.11 Notification

Attributes

- notificationID : int **(PK)**
- message : String
- notificationType : String
- sentDate : Date
- patientID : int **(FK → Patient.patientID)**

Methods

- sendSMS()
- sendEmail()

4.1 Inheritance (Generalization)

Person

├── Patient

├── Doctor

└── Receptionist

- Type: **Inheritance**
- Cardinality: 1..1

4.2 Patient – Appointment

- Relationship: **Association**
- Cardinality:
    - Patient 1..* Appointment
    - Appointment 1..1 Patient

4.3 Doctor – Appointment

- Relationship: **Association**
- Cardinality:
    - Doctor 1..* Appointment
    - Appointment 1..1 Doctor

4.4 Patient – MedicalRecord

- Relationship: **Composition**
- Cardinality:
    - Patient 1..1 — 1..* MedicalRecord
- Meaning: Medical records **cannot exist without a patient**

4.5 MedicalRecord – Prescription

- Relationship: **Composition**
- Cardinality:
    - MedicalRecord 1..1 — 0..* Prescription

4.6 Department – Doctor

- Relationship: **Aggregation**

- Cardinality:
    - Department 1..1 — 1..* Doctor
- Meaning: Doctors can exist independently of departments

4.7 Patient – Billing – Payment

- Patient 1..* Billing
- Billing 1..1 Payment