

## Bing

Certainly! Let's include the **Admin class** in our comprehensive **UML class diagram** for the proposed **Lab Appointment System** at **ABC Laboratories**. The Admin class will manage administrative tasks related to the system. Here's the updated diagram:

```
@startuml
class Patient {
    - patientID: int
    - name: string
    - email: string
    - phone: string
    + register(): void
    + getAppointmentDetails(): void
    + downloadReports(): void
}

class Test {
    - testID: int
    - testName: string
    - doctorID: int
    + createTestRecord(): void
}

class Technician {
    - technicianID: int
    - name: string
    + performTest(): void
}

class Doctor {
    - doctorID: int
    - name: string
    + recommendTest(): void
}

class Billing {
    - billID: int
    - amount: float
    - paymentMethod: string
    + generateBill(): void
    + emailReceipt(): void
}

class Appointment {
    - appointmentID: int
    - patientID: int
    - doctorID: int
    - appointmentTime: datetime
    - status: string
    + scheduleAppointment(): void
    + confirmAppointment(): void
    + cancelAppointment(): void
    + updateAppointmentTime(newTime: datetime): void
}

class Admin {
    - adminID: int
    - name: string
    + manageAppointments(): void
    + manageTests(): void
    + generateReports(): void
}
```

Patient --> Test

Test --> Technician  
Test --> Doctor  
Patient --> Billing  
Billing --> Appointment  
Admin --> Appointment  
Admin --> Test

@endum1

Explanation:

- The **Admin class** is now included, representing administrative functions within the system.
- Relationships:
  - Admin manages both Appointment and Test classes.
  - Patient, Test, Technician, Doctor, and Billing remain unchanged.

Feel free to explore further or ask for any additional details! 🌟