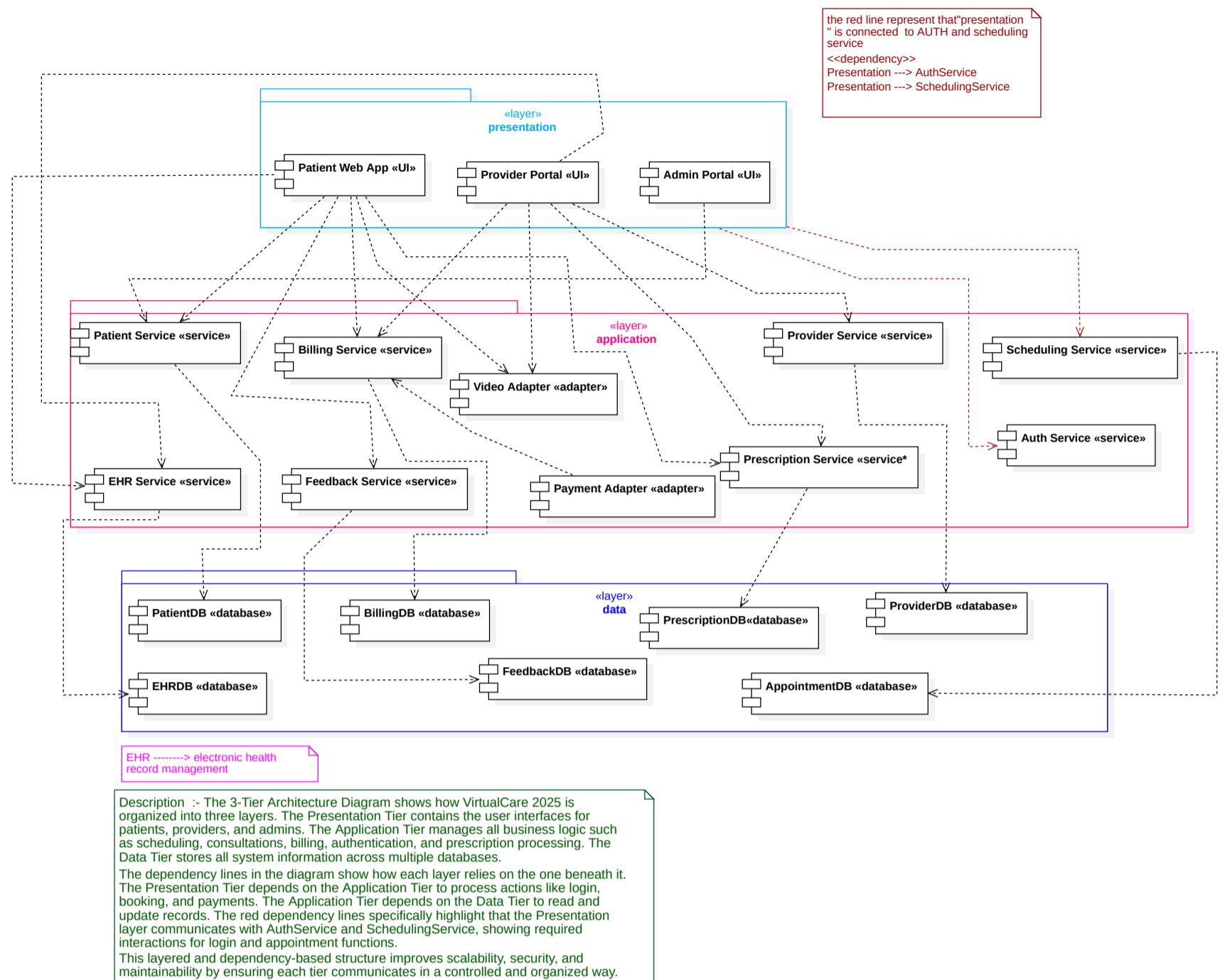
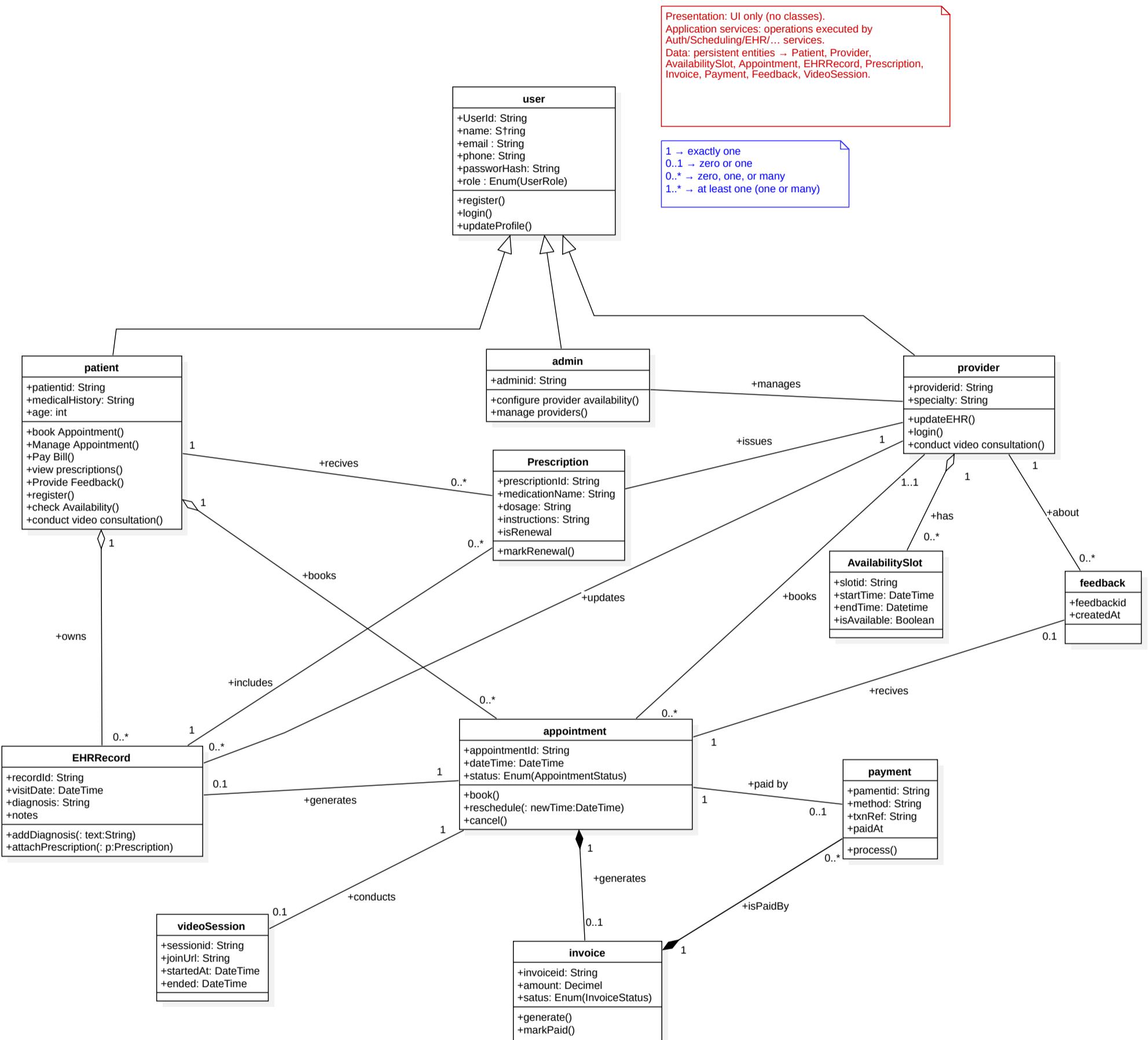




Description :- The Use Case Diagram for VirtualCare 2025 shows how patients, providers, admins, and external systems interact with the platform. It outlines key functions such as registration, appointment booking, video consultations, prescriptions, EHR updates, and billing. The diagram also uses include relationships to represent essential sub-tasks (e.g., Book Appointment includes Check Availability, Pay Bill includes Generate Invoice) and extend relationships to show optional or conditional actions (e.g., Manage Appointment extends to Reschedule or Cancel Appointment). These relationships help clarify the workflow, avoid repetition, and show how different use cases depend on each other within the system boundary. The user actor here comes from the class diagram . where admin , patient are generalization to user actor .

EHR -----> electronic health record management





The Class Diagram of VirtualCare 2025 shows the main system entities and how they interact. Users (patients, providers, admins) share common attributes through the User class, while Patient and Provider add their own details such as medical history and specialty. The diagram shows how a patient books appointments, receives prescriptions, owns EHR records, , while providers manage availability, issue prescriptions, conduct video sessions, and receive feedback.

The diagram also highlights aggregation and composition:

EHRRecord aggregates diagnoses and prescriptions, showing that these medical details belong to a patient's long-term health record.

Appointments are composed of invoices , meaning the object is created as part of the appointment process.

A provider aggregates multiple availability slots and so one with the rest.

Presentation: UI only (no classes).
Application services: operations executed by Auth/Scheduling/EHR/... services.
Data: persistent entities → Patient, Provider, AvailabilitySlot, Appointment, EHRRecord, Prescription, Invoice, Payment, Feedback, VideoSession.

1 → exactly one
0..1 → zero or one
0..* → zero, one, or many
1..* → at least one (one or many)



satruml — -zsh — 80x24

```
Last login: Sat Nov  8 16:52:27 on ttys000
maha@mahas-MacBook-Pro satruml % git init

Reinitialized existing Git repository in /Users/maha/Desktop/satruml/.git/
maha@mahas-MacBook-Pro satruml % git add .

maha@mahas-MacBook-Pro satruml % git commit -m "Final Part 1 submission"

On branch main
nothing to commit, working tree clean
maha@mahas-MacBook-Pro satruml % git log

commit a47c560e8a4ea87bbefdd1f6a498565321c3e0ec (HEAD -> main)
Author: mahamohamedelkhatib <maha.elkhatib@gaf.ac>
Date:   Sat Nov 8 17:11:06 2025 +0200

    Final Part 1 submission with diagrams and report
maha@mahas-MacBook-Pro satruml %
```