# Appointments & Scheduling Management

## Objective

Design and implement a robust appointment management system for a Pet Practice Management app where:

- Doctors configure their own available consultation schedules (days, times, diseases treated, expertise).
- Pet owners can browse/filter doctors based on their pet's disease and other criteria, view real-time available slots, and book appointments accordingly.

## Tasks

**1. Doctor Schedule Management**

- Allow doctors to set up their consultation schedule:
  - Select available days of the week.
  - Choose time slots for each day.
  - Tag the schedule with specializations (diseases/conditions treated, e.g., skin, dental, behavioral).
- Store doctor profiles and schedules in-app (emulated with local JSON, Array, or AsyncStorage).progress indicators.

**2. Doctor Directory & Filtering**

- Build a doctor listing/browsing screen for pet owners.
  - Filter doctors by disease/expertise, availability, and other attributes (location, ratings, etc., can be simulated).
  - Display only those doctors who currently have upcoming available slots for chosen disease/expertise.

**3. Appointment Booking Flow**

- Enable pet owners to:
  - Select disease or reason for visit.
  - View filtered list of available doctors matching that need.
  - Select a doctor and see a calendar with open time slots (from the actual doctor's saved schedule).
  - Book appointment for a chosen pet and time slot.
- Once booked, that slot becomes unavailable for others (simulate update to the schedule).

**4. My Appointments (Pet Owner & Doctor)**

- List upcoming and past appointments:
  - For pet owner: appointments grouped by pet, showing doctor, disease, date, and status.
  - For doctor: appointments grouped by slot, showing client/pet and status.

- Allow cancellation/rescheduling (with simulated updates to availability).

**5. State Management & Local Persistence**

- Store all schedules, bookings, and profiles locally (simulate using AsyncStorage or local arrays).
- On app reload, all data should persist and reflect latest changes.
- Gracefully handle booking conflicts or schedule updates.

**6. UI/UX & Navigation**

- Clean, modular navigation:
  - Separate flows: Doctor schedule setup; Owner filters and books.
- Clear, responsive feedback for successful bookings, errors (e.g., double-booking), or schedule conflicts.

**7. State & Persistence**

- Manage chat messages, chat open/closed state, appointment info with React Context or simple state manager.
- Persist messages and chat states (including closure state) locally for offline and reload persistence.

## Bonus

- Allow recurring availability patterns for doctors (e.g., alternate weeks).
- Integrate advanced filters: experience, “top-rated” status, language spoken.
- Add a notification/reminder for upcoming appointments.

## Deliverables

- Bare React Native project (TypeScript) implementing appointment and scheduling flows.
- Clear project README with instructions.
- Well-structured code with meaningful comments.