**PLANNING**

**🔧 Phase 1: Database & Backend Setup**

**✅ 1. Define Database Schema (You may have done this already)**

* clients: Unique by name or other identifiers
* cases: Linked to clients
* consults: Linked to cases

Ensure:

* Timestamps: created\_at, updated\_at
* is\_deleted flags for soft deletes
* Foreign key constraints

**✅ 2. Core Logic (Backend/API Layer)**

* Implement endpoints:
  + GET /clients – return clients ordered by last activity
  + GET /clients/:id/cases – get cases for client
  + GET /cases/:id/consults – get consults for case
  + POST/PUT/DELETE – for all tables
* Update updated\_at for:
  + clients when a related case or consult is updated
  + cases when a related consult is updated

**🎨 Phase 2: Godot Frontend UI**

**✅ 1. UI Design**

* Views:
  + **Client List** (sorted by updated\_at)
  + **Case List** (when a client is selected)
  + **Consults List** (when a case is selected)
* Use ItemList or Tree nodes for hierarchy views

**✅ 2. View Modes and Navigation**

* Add navigation state management (current\_mode: dashboard, client, case, consult)
* Load corresponding data on selection

**✅ 3. Sorting & Filtering**

* Sort consults by created\_at DESC
* When a consult or case is added/edited, re-fetch affected views and sort accordingly
* Push client to top when they’re updated indirectly via a case or consult

**🔁 Phase 3: Interactivity & Data Sync**

**✅ 1. Connect Godot to API**

* Use HTTPRequest or your HttpManager
* Async load on UI interactions

**✅ 2. Editing & Adding**

* Add dialogs or scenes for:
  + Creating/editing clients
  + Creating/editing cases
  + Adding consult notes

**✅ 3. Update Propagation**

* On successful edit:
  + Refresh related entities
  + Update timestamps up the chain

**✅ Phase 4: Polish and Optimize**

**🔹 UI Polish**

* Better layout, fonts, margins
* Icons or color for status/importance

**🔹 Performance**

* Lazy-load consults for long histories
* Cache frequently loaded data

**🔹 Error Handling**

* Show messages for failed network/database calls

**🧪 Phase 5: Testing & Deployment**

**✅ Testing**

* Manual test all flows
* Test client reordering logic

**✅ Deployment**

* Deploy backend (e.g., on shared host, VPS, or cloud service)
* Export Godot frontend (e.g., to desktop or web)