Odoo OCA module Legal Case Management Requirements (v18)

> **Audience:** Students building a first, working OCA-style module. Keep it tiny, clean, and testable. **License:** AGPL-3 • **Edition:** Odoo 18 Community • **Goal:** A minimal app to create clients/lawyers, register cases, plan hearings, attach documents, and (optionally) issue a fixed-fee invoice.

---

1) Scope (What is expected)

- Create **Lawyers** and **Clients** (as `res.partner` flags).
- Register **Cases** with a short lifecycle and unique reference.
- Plan **Hearings/Sittings** on a calendar.
- Attach **Documents** to a case (simple metadata only).
- **Fixed-fee invoicing only** (one invoice per case). Payments use standard Accounting flow.
- **Simple reports** using list views + export; one printable "Case Summary".

Out of scope for MVP: timesheets/T&M, retainers, milestones, portals, advanced DMS, complex security.

---

2) Roles (Simple)

- **Legal User**: create & edit cases they own; create hearings & documents; create draft invoices.
- **Legal Manager**: see all cases; close cases; approve invoices.

Minimal record rule: Legal Users see cases where they are **responsible_lawyer** or explicitly added to **member_ids**. Managers see all.

---

3) Data Model (Minimal)

Model	Purpose	Key Fields
`legal.case`	Legal matter	name (sequence), client_id (res.partner), responsible_lawyer_id (res.partner), member_ids (m2m res.users), case_type (selection), stage (selection: intake, active, closed), open_date, close_date, description
`legal.hearing`	Hearing/ sitting	case_id, name, date_start, date_end, location, status (planned/held/adjourned/cancelled), notes
`res.partner`	People/	is_lawyer (bool), is_client (bool), bar_number (char, optional)

(extend) firms

Files: use `ir.attachment` linked to `legal.case` via chatter (no custom model). Optional tag via a simple `selection` on attachment using context (nice-to-have; can be skipped in MVP).

---

4) Core Features & Acceptance

4.1 Lawyers & Clients

- Add two boolean fields on partner: **is_lawyer**, **is_client**.
- Quick filters in Contacts for Lawyers and Clients. **Accept:** A partner flagged as Lawyer/Client can be selected on a case.

4.2 Case Registration

- Sequence `CASE/${year}/${seq}`.
- Minimal fields: client, responsible_lawyer, case_type, stage, description.
- Stages: **Intake** → **Active** → **Closed** (selection).
- Smart buttons: **Hearings (count)**, **Invoices (count)**. **Accept:** Creating a case gives a unique reference; changing stage to Closed sets `close_date`.

4.3 Hearings/Sittings

- Create hearings from the Case (one2many) and from a standalone menu.
- Calendar & List views. **Accept:** Hearing appears on Calendar with start/end and links back to its case.

4.4 Documents

- Upload attachments from the Case chatter.
- “Documents” smart button opens attachments domain-filtered to this case. **Accept:** After upload, file is visible when opening the case’s attachments.

4.5 Invoicing (Fixed Fee)

- On case, field **fixed_fee_amount** (monetary). Button **Create Invoice** → creates a draft customer invoice with one line (product = "Legal Services", price = fixed_fee_amount, analytic = case if available).
- Show invoice count; payment status relies on standard Accounting. **Accept:** Clicking Create Invoice generates a draft invoice linked to the case; posting & registering payment updates the invoice status.

4.6 Reports (Very Simple)

- **List exports**: Cases (CSV/XLSX) via standard Export.

- **Case Summary (QWeb PDF)**: shows case header, parties, hearings list, and attachment names. **Accept:** Case Summary prints from a case action and downloads as PDF.

---

5) UI/UX (Minimal Menus & Views)

- **Menu: Legal**
  - **Cases** (tree, form, kanban optional)
  - **Hearings** (calendar, tree, form)
  - **Reporting** → *Case Summary* action on record (no separate menu needed)
  - **Configuration** (optional) → Case Types (selection can be hard-coded in MVP)

Case Form Tabs: *Overview* (all fields), *Hearings* (one2many), *Chatter* (attachments & log).

---

6) Security & Access

- `ir.model.access.csv` for `legal.case` and `legal.hearing` (read/write/create for user group; full for manager).
- Record rule for Legal User: case responsible or member.
- Keep it simple: attachments inherit case access via chatter.

---

7) Technical Checklist

- `__manifest__.py`: name, version, depends: `["base", "mail", "account", "calendar"]` (calendar optional, but recommended).
- Data: security (groups, access, rules), sequences, menus, actions, views, demo data.
- Python: models for `legal.case`, `legal.hearing`; compute counts; button to create invoice.
- Views: tree/form/calendar; smart buttons; printing action.
- Tests: create partner/case/hearing; create invoice; print report.
- Lint: OCA pre-commit, pylint-odoo; i18n ready.

---

8) Demo Data (Small)

- 2 lawyers, 3 clients.
- 2 open cases, 1 closed case.
- 3 hearings across the cases.

- 1 fixed-fee invoice (draft).

---

9) Stretch (Optional if time permits)

- Simple `case_type` model instead of selection.
- “Next hearing date” computed on case.
- Color badges by stage in tree/kanban.
- Minimal dashboard (kanban metrics) using stat buttons.

---

10) Acceptance Criteria (Summary)

- **Partners:** Boolean flags work; filters show Lawyers/Clients.
- **Cases:** Unique sequence; stage flow works; close_date set on Close.
- **Hearings:** Calendar entries linked to cases.
- **Documents:** Attach & view via case; count visible.
- **Billing:** One-click fixed-fee invoice created and linked to case.
- **Reports:** Case Summary prints; list exports work.

End of MVP Requirements