**2025-08-30 — Admin Booking System (Deep Audit & Current State)**

Living reference of how Admin Manual Booking currently works in code. This document captures the present state (components, data flow, RPCs/SQL, availability/override behavior), highlights gaps vs. our LOG, and lists open questions + next steps. Keep this updated as we iterate.

**0) Purpose & Scope**

* **Goal:** Precisely document the **current** Admin Manual Booking implementation (no proposed changes here).
* **Covers:** Admin Manual Booking UI flow, helper hooks/utils, invoked RPCs/SQL, tables touched, payload shapes, availability computation, override behavior, RLS touchpoints.
* **Excludes:** Client self‑service booking, payments processor specifics, public site flows.

**1) Quick Summary (from audit)**

* **Pages/Components analyzed:** 15
* **RPCs/SQL mapped:** 4
* **Tables touched:** 12 (7 read, 5 write)
* **Core status:** Booking works end‑to‑end (single + dual services), add‑ons persist, lifecycle updates visible in Admin.
* **Key gaps vs. LOG:** 5 (see §10)

**2) Route & Component Map**

**Primary entry points**

* AdminManualBooking (current)
* AdminBookingPage (legacy/compat if present)

**High‑level component groups**

* **Step 1 — Client & Pet**
  + Client selector (admin claims/unclaimed aware)
  + Pet selector (filters by selected client)
* **Step 2 — Services**
  + Primary service picker (duration + price by breed/size)
  + Secondary service picker (optional; dual‑service merge)
  + Service summary (combined duration, total price)
* **Step 3 — Staff**
  + Staff capability filter per service type
  + Multi‑select for required roles (e.g., bather + groomer)
* **Step 4 — Date & Time**
  + Booking calendar / time‑slot picker
  + Availability overlay (staff + shower) resolution
* **Create & After‑Create**
  + Booking create call
  + Add‑ons editor (e.g., tax/fees) and save
  + Status control (pending → confirmed → iniciado → …)

**Shared utilities/hooks (examples)**

* Pricing hook (breed/size rules)
* Availability hook(s) (staff + shower)
* Time‑slot helpers (minute granularity)
* Payload builders (client resolution, staff roles, services merge)

See Appendix A for a file/path list you can check off during code review.

**3) Data Model — Tables Touched**

*Reads/Writes will vary by step; this list represents the current audit view.*

* clients (R/W): admin markers, contact fields; link to pets
* pets (R): breed/size, client ownership
* services (R): names, base price, duration\_minutes
* service\_resources (R): required resources per service (e.g., bather, groomer, vet, shower)
* staff\_profiles (R): IDs, names, location\_id; capability linkage via service\_resources
* staff\_availability (R): immutable base staff availability (per minute slot)
* shower\_settings (R): day‑level defaults (capacity)
* shower\_availability (R/W): mutable per‑slot capacity counters
* appointments (W): core booking row (start, duration, price, status, client/pet)
* appointment\_staff (W): **links staff to appointment**; role **NOT NULL**
* appointment\_events (W): lifecycle log (created, confirmed, iniciado, canceled, etc.)
* appointment\_addons (W): add‑on lines linked to appointment (fees, surcharges)

Legacy/aliases seen in code: appointment\_providers, provider\_availability — treat as legacy names and prefer canonical names above.

**4) RPC/SQL Functions (Inventory)**

Parameters/returns summarized; mask PII when logging/debugging.

1. **create\_admin\_booking\_with\_dual\_services**
   * **Purpose:** Atomic create of a booking that may include a primary + secondary service; assigns staff; consumes availability.
   * **Inputs (key):**
     + Client resolution: \_client\_id **(admin‑created & unclaimed only)** or \_client\_user\_id (claimed)
     + Services: \_primary\_service\_id, optional \_secondary\_service\_id
     + Staff: \_provider\_ids/***staff\_profile\_ids*** (array); roles inferred per service\_resources
     + Time: \_date, \_time\_slot (minute index), computed duration (merged)
     + Flags: override (if any), location/clinic
   * **Returns:** appointment id + summary
   * **Side effects:** writes appointments, appointment\_staff, updates shower\_availability, logs appointment\_events.
2. **get\_staff\_availability\_summary / get\_available\_providers** (naming varies)
   * **Purpose:** Produce available time slots per staff given date/service duration; overlays existing appointments on base availability.
3. **get\_shower\_capacity**
   * **Purpose:** Resolve day‑specif