# Salon & Academy Management System: Architectural Strategy

## Part 1: The Case for a Dedicated Web Dashboard

*Why the Owner/Admin should move to the Web, while Customers stay on Mobile.*

### 1. Real-World Usage Patterns

* **The "Desk" Factor:** Managing a business (approving 50 course applications, reconciling monthly finances, adjusting staff rosters) is "deep work." It is tedious on a 6-inch screen.
* **Data Density:** Admin tables often have 10+ columns (Name, Phone, Service, Date, Time, Status, Payment Ref, Staff, etc.). On a phone, you have to scroll horizontally endlessly. On a 15-inch laptop screen, you see the whole picture instantly.
* **Multitasking:** Admins often need to have the calendar open in one tab and a payment gateway in another. This workflow is native to Web/Desktop but cumbersome on Mobile.

### 2. UX Separation (The "Two-Brain" Approach)

* **Customer App (Flutter):**
  + **Goal:** Instant gratification, speed, visuals.
  + **Vibe:** Immersive, simple navigation, push notifications.
* **Admin Dashboard (Web - React/Next.js):**
  + **Goal:** Efficiency, data manipulation, control.
  + **Vibe:** Dashboard widgets, dense data tables, export buttons, bulk actions.

### 3. Security & Scalability

* **Role Security:** You can lock the Admin Web URL so it is only accessible from the Salon's IP address (Wi-Fi), preventing staff from checking data from home.
* **Future Proofing:** If you open a second branch, the Web Dashboard can easily add a dropdown to switch between "Branch A" and "Branch B."

## Part 2: Technical Structure

### The "Single Brain" Architecture

Do not create two separate backends. Use one API that serves both interfaces.

1. **Backend (Node.js / Python / Go):**
   * Holds the database (Users, Appointments, Courses, Payments).
   * Handles logic (e.g., "If appointment booked, send SMS").
2. **Frontend A: Mobile App (Flutter):**
   * Used by: **Customers** (Role: user).
   * Features: Booking, Portfolio, Course Application, Profile, Support.
3. **Frontend B: Admin Web Panel (React.js recommended):**
   * Used by: **Owner, Manager, Receptionist** (Roles: owner, admin, staff).
   * Features: Heavy lifting, CRUD operations, Analytics, Support Management.

## Part 3: Appointment Booking Lifecycle (New Logic)

*Handling the "Pay Now" vs. "Pay Later" expiration rules.*

### The 4-Hour Reservation Rule

To prevent no-shows while offering flexibility, we implement a strict time-window for unpaid bookings.

1. **User Options:**
   * **Option A: Pay Full Online:** Appointment is confirmed immediately (status: confirmed).
   * **Option B: Pay at Salon:** User books the slot without immediate payment.
2. **The Logic (Backend):**
   * If Option B is chosen, the appointment is created with status: reserved and payment\_status: unpaid.
   * **The Timer:** The system records created\_at timestamp.
   * **The Window:** The user has exactly **4 Hours** to visit the salon and pay.
3. **Auto-Cancellation (Cron Job):**
   * A background job runs every 10 minutes.
   * **Check:** Find all appointments where status == 'reserved' AND created\_at < (now - 4 hours).
   * **Action:** Automatically change status to cancelled and release the slot for other customers.
   * **Notification:** Send user push notification: *"Your reservation has expired due to non-payment."*
4. **In-Person Finalization:**
   * If the user arrives within 4 hours, the Receptionist/Owner opens the Admin Panel (Web or Mobile).
   * Locates the reserved appointment.
   * Clicks **"Mark as Paid"**.
   * Status changes to confirmed. Timer stops.

## Part 4: The "Courses Portal" & Payment Logic

*Updated flow for Sales Team Alerts and Deposit Rules.*

### The User Flow: "Apply, Alert, Secure"

#### Step 1: Browse & Apply (Lead Generation)

* **User Action:** Clicks "Apply Now" on a course.
* **Data:** User details (Name, Phone, City) are saved.
* **System Action (Immediate):**
  + **Alert Sales Team:** Send notification to Admin Web & Owner App. *"New Lead: Sarah applied for Makeup Course."*
  + **Status:** status: pending\_booking.

#### Step 2: Booking the Seat (The 10% Rule)

* **User UI:** After applying, the app shows: *"Your application is received. To secure your seat, please pay the booking fee."*
* **Dynamic Rule:** The Admin sets the booking percentage (e.g., 10%) in the Web Dashboard.
* **Calculation:** If Course = PKR 100k, and Rule = 10%, User must pay PKR 10k.
* **Payment:** User pays via App (Online) or uploads Bank Receipt.

#### Step 3: Confirmation

* **System Action:**
  + Updates paid\_amount: 10,000.
  + Updates status: secured.
  + **Sales Team:** Sees the lead convert to a "Student" in the dashboard.

## Part 5: Admin (Owner) Functionalities - Detailed Specification

*A comprehensive breakdown of the Web Dashboard capabilities.*

### 1. Dashboard & Overview

* **KPIs:** Today’s appointments, upcoming appointments, revenue (salon + courses), popular services, active courses, pending applications.
* **Quick Actions:** Approve/cancel latest appointments, approve/reject course applications, add offer, upload gallery image.
* **Calendar View:** Day/week/month schedule of all appointments by staff.

### 2. Appointment Management

* **Appointments List:** Filter by date, staff, service, status.
* **Reservation Monitor:** Specific filter for "Expiring Soon" (appointments within the 4-hour window).
* **Actions:**
  + **Approve/Reject/Cancel:** Change status manually.
  + **Mark Paid:** Convert reserved -> confirmed (stops the 4-hour timer).
  + **Reschedule:** Change time/staff.

### 3. Services & Pricing

* **CRUD Services:** Name, Category, Duration, Price, Image.
* **Service Monitoring (List View):** Table displaying active services with columns:
  + **Name:** Customer Name.
  + **Service:** Service Type (e.g., Haircut).
  + **Payment:** Status (Paid/Unpaid/Partial).
  + **Status:** Operational Status (Pending/Active/Completed).
  + **Location:** Branch or specific salon section.
  + **Calendar:** Date & Time slot.
* **Configuration:** Set default display rules.

### 4. Staff / Experts Management (Updated)

* **Staff Profiles (Complete):**
  + **Personal:** Name, Profile Photo, Bio.
  + **Professional:** Role, Skills, **Experience** (Years/Certifications).
  + **Contact:** Phone Number, Email, **Address** (Home/Residential).
  + **Operational:** Work Schedule (Roster).
* **Service Mapping:** Which services each expert can perform.
* **Performance View:** Appointments per staff, revenue, ratings.

### 5. Offers & Promotions

* **CRUD Offers:** Create discount banners.

### 6. Gallery Management

* **Upload:** High-res images for portfolio.

### 7. Course Management

* **Course Catalog:** CRUD courses.
* **Settings:**
  + **Set Booking %:** Input field to define the deposit percentage (e.g., 10%, 20%) required to book a seat.
* **Sales Pipeline:** View Leads vs. Secured Students.

### 8. User Roles & Permissions Engine

* **Super Admin (Owner):** Full access to everything. Can create/delete other admin accounts.
* **Role Management Interface:** The Owner can create custom roles or assign predefined ones:
  + **Receptionist:**
    - *Can:* View Calendar, Book Appointments, Mark Payments (Stop Timer), View User List.
    - *Cannot:* View total monthly revenue, Change course prices, Delete staff.
  + **Sales Team / Course Manager:**
    - *Can:* View Course Applications, Call Leads, Update Course Status.
    - *Cannot:* Manage Salon Appointments, Edit Gallery.
  + **Staff (Stylist):**
    - *Can:* View only *their own* schedule.
    - *Cannot:* See customer phone numbers (optional privacy rule).

### 9. Business Rules Engine

*The "Brain" of the operation where the Owner defines how the system behaves.*

* **Booking Rules:**
  + **Deposit Percentage:** Slider to set course seat reservation fee (e.g., 5% to 50%).
  + **Reservation Window:** Input field to change the "4-Hour Rule" (e.g., extend to 6 hours or reduce to 2 hours).
* **Operation Modes:**
  + **Strict Mode:** No bookings allowed without payment.
  + **Flexible Mode:** Allow "Pay at Salon" (enables the 4-hour timer).

### 10. Notifications & Communication

* **Send Broadcast:** Push/email campaigns to all or filtered users.
* **Triggered Notifications:** View log of automated messages.

### 11. Payments & Finance

* **Transactions List:** All appointment & course payments (amount, method, status, user, reference).
* **Reports:** Revenue by day/month, by service, by course, by staff.

### 12. Settings & Configuration

* **Salon Profile:** Name, address, contact, logo, branding colors.
* **Working Hours:** Global open/close times, holidays, blackout dates.
* **Integrations:** Payment gateway keys, FCM keys, email/SMS providers.

### 13. Support Module (New - Admin Managed)

* **Scope:** Dedicated section solely managed by Admin/Owner.
* **Ticket Management:** View user inquiries submitted from the app (e.g., payment issues, booking errors).
* **Status Tracking:** Mark queries as "Open," "In Progress," or "Resolved."
* **Direct Communication:** Reply mechanism (updates appear in user app).
* **FAQ Manager:** Create and update a help center for the mobile app.

## Part 6: Migration Strategy & The "20/100" Control Split

*Defining exactly what the Owner can do on the Phone (20%) vs. the Web (100%).*

### 1. The 100% Web Control (Command Center)

* **Platform:** React/Next.js Web Dashboard.
* **Scope:** **Comprehensive & Absolute.** Every single aspect of the business, from the smallest database field to the largest financial report, is accessible here.
* **Exclusive Features (Web Only):**
  + **Full Staff Management:** Editing address, experience, and contracts.
  + **Support Module:** Handling complex user tickets.
  + **Bulk Actions:** Uploading images, exporting reports.
  + **Rule Configuration:** Changing the "4-hour timer" duration or the "10% deposit" rule.
  + **Role Assignment:** Creating new admin accounts and assigning permissions.

### 2. The 20% Mobile Control (On-the-Go)

* **Platform:** Flutter App (via "Hidden Admin Mode").
* **Scope:** Critical, time-sensitive actions only. This is "enough" for daily floor operations.
* **Features:**
  1. **View Schedule:** See today's appointments.
  2. **Payment Collection:** Quick button to "Mark Paid" (stops the 4-hour expiry).
  3. **Alerts:** Push notifications for New Course Leads or New Bookings.
  4. **Emergency:** Button to "Close Shop" (mark all staff unavailable for the rest of the day).

## Part 7: MERN Tech Stack Implementation Guide

### Frontend (Admin Panel – React)

* **Core:** React + TypeScript.
* **UI:** MUI / Ant Design.
* **State:** React Query (for real-time appointment lists).

### Backend (Node.js + Express)

* **Cron Jobs:** Use node-cron or agenda.js.
  + *Task:* Run every 10 mins. Find status: reserved where created\_at < now - 4h. Update to cancelled.
* **Events:** Use EventEmitter for "New Course Application" -> Trigger Notification Service.

### Database (MongoDB)

* **Models:**
  + Appointment: Added fields reservationExpiresAt (Date), paymentStatus (enum: 'unpaid', 'partial', 'full').
  + Course: Added field bookingDepositPercentage (Number).
  + Staff: Updated with experience, contact, address.
  + SupportTicket: New collection for the Support Module.