MAPÚA MALAYAN DIGITAL COLLEGE

A college under Mapúa Malayan Colleges Laguna

Database Plan Worksheet

Prepared by:

Jennifer B. Cerio

TABLE OF CONTENTS

1. Collections and Documents

Collection Name	Document Structure

Users	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k1')",
	"email": "admin@nailsalon.com",
	"password": "$2a$10$hashedPasswordHere",
	"name": "Admin User",
	"role": "ADMIN",
	"isActive": true,
	"lastLogin": "2026-02-05T14:30:00Z",
	"createdAt": "2026-01-01T10:00:00Z"
	}

NailTechs	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k2')",
	"name": "Sarah Cruz",
	"email": "sarah@nailsalon.com",
	"phone": "+639121112222",
	"commissionRate": 40,
	"discountRate": 10,
	"isActive": true,
	"createdAt": "2026-01-15T09:00:00Z"
	}

Clients	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k3')",
	"firstName": "Maria",
	"lastName": "Santos",
	"email": "maria@email.com",
	"phone": "+639123456789",
	"address": {
	"street": "123 Main Street",
	"city": "Quezon City",
	"province": "Metro Manila",
	"zipCode": "1100"
	},
	"clientType": "NEW",
	"totalBookings": 0,

	"completedBookings": 0,
	"totalSpent": 0,
	"lastVisit": null,
	"preferences": {
	"preferredLocation": "HOME",
	"preferredTime": "afternoon"
	},
	"healthInfo": {
	"allergies": ["latex"],
	"skinSensitivity": "normal"
	},
	"isActive": true,
	"createdAt": "2026-02-05T10:00:00Z"
	}

Services	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k4')",
	"name": "Manicure",
	"isActive": true,
	"createdAt": "2026-01-01T08:00:00Z"
	}

Bookings	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k5')",
	"bookingNumber": "GN-20260205001",
	"client": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k3')",
	"service": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k4')",
	"nailTech": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k2')",
	"appointmentDate": "2026-02-10",
	"appointmentTime": "14:00",
	"locationType": "HOME",
	"address": "123 Main Street, Quezon City",
	"serviceDescription": "French tips with floral nail art design",
	"status": "PENDING_PAYMENT",
	"pricing": {
	"totalPrice": 1000,
	"reservationFee": 200,
	"balanceDue": 800
	},
	"payment": {
	"reservationPaid": false,

	"reservationPaidAt": null, "reservationPaymentMethod": "GCASH", "reservationReference": "", "paymentProofUrl": "", "fullPaymentReceived": false }, "specialRequests": "Please bring light pink polish", "clientNotes": "", "adminNotes": "", "createdBy": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k1')", "confirmedBy": null, "expiresAt": "2026-02-06T14:00:00Z", "createdAt": "2026-02-05T14:00:00Z", "updatedAt": "2026-02-05T14:00:00Z" }

QuotationInvoices	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k6')",
	"invoiceNumber": "INV20260205001",
	"booking": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k5')",
	"client": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k3')",
	"nailTech": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k2')",
	"items": [
	{
	"description": "Gel Manicure",
	"quantity": 1,
	"unitPrice": 500,
	"totalPrice": 500
	},
	{
	"description": "Nail Art (Floral Design)",
	"quantity": 1,
	"unitPrice": 300,
	"totalPrice": 300
	},
	{
	"description": "Chrome Finish",
	"quantity": 1,
	"unitPrice": 200,
	"totalPrice": 200
	}
	],

	"subtotal": 1000, "discountPercentage": 10, "discountAmount": 100, "totalAmount": 900, "commissionRate": 40, "nailTechCommission": 360, "salonRevenue": 540, "notes": "Client requested matte finish. 10% VIP discount applied.", "createdBy": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k1')", "createdAt": "2026-02-05T15:00:00Z" }

Notifications	{
	"_id": "ObjectId('65a1b2c3d4e5f6g7h8i9j0k7')",
	"recipient": "maria@email.com",
	"type": "EMAIL",
	"category": "BOOKING_CONFIRMATION",
	"subject": "Booking Confirmation - GN-20260205001",
	"content": "Your booking has been received...",
	"status": "SENT",
	"sentAt": "2026-02-05T14:05:00Z",
	"createdAt": "2026-02-05T14:00:00Z"
	}

Additional Details (Optional):

Users Collection

Required Fields:

- email (String, unique) - Admin login email
- password (String) - Hashed password for security
- name (String) - Full name of admin user
- role (String) - User role: "SUPER_ADMIN", "ADMIN", "MANAGER", or "STAFF"
- isActive (Boolean) - Account status

Optional Fields:

- lastLogin (Date) - Last login timestamp
- createdAt (Date) - Account creation date

NailTechs Collection

Required Fields:

- name (String) - Nail technician's full name
- email (String, unique) - Contact email
- phone (String) - Contact phone number
- commissionRate (Number) - Percentage of service price the nail tech earns (e.g., 40 = 40%)
- discountRate (Number) - Percentage discount nail tech can offer to clients (e.g., 10 = 10%)
- isActive (Boolean) - Employment status (true = active, false = inactive)

Optional Fields:

- createdAt (Date) - Record creation date

Additional Details:

- commissionRate: Default is 40%, meaning nail tech earns 40% of the total booking price
- discountRate: Maximum discount the nail tech is authorized to give clients (e.g., 10% for regular clients, VIPs, etc.)
- Admin can adjust these rates per nail tech based on experience, performance, or special arrangements

Clients Collection

Required Fields:

- firstName (String) - Client's first name
- lastName (String) - Client's last name
- email (String, unique) - Contact email
- phone (String, unique) - Contact phone number
- clientType (String) - "NEW" or "REPEAT"

- isActive (Boolean) - Account status

Optional Fields:

- address (Object) - Full address for home service
- totalBookings (Number) - Count of all bookings
- completedBookings (Number) - Count of completed appointments
- totalSpent (Number) - Total amount spent
- lastVisit (Date) - Last appointment date
- preferences (Object) - Client preferences
- healthInfo (Object) - Health-related information
- createdAt (Date) - Registration date

Services Collection

Required Fields:

- name (String) - Service type: "Manicure", "Pedicure", or "Mani+Pedi"
- isActive (Boolean) - Service availability status

Optional Fields:

- createdAt (Date) - Service creation date

Note:

- Only 3 service types exist in the system: Manicure, Pedicure, and Mani+Pedi
- Pricing is not stored here as each client has different designs/requirements
- Detailed pricing is determined per booking through QuotationInvoices using Google Sheets

Bookings Collection

Required Fields:

- bookingNumber (String, unique) - Unique booking identifier (e.g., "GN-20260205001")
- client (ObjectId, ref: Clients) - Reference to client document
- service (ObjectId, ref: Services) - Reference to service type (Manicure, Pedicure, or Mani+Pedi)
- nailTech (ObjectId, ref: NailTechs) - Reference to nail technician
- appointmentDate (Date) - Appointment date
- appointmentTime (String) - Start time in HH:MM format (e.g., "14:00")
- locationType (String) - "HOME" or "STUDIO"
- serviceDescription (String) - Detailed description of what client wants (designs, colors, etc.)
- status (String) - Booking status: "PENDING_PAYMENT", "CONFIRMED", "COMPLETED", "CANCELLED", "NO_SHOW"
- pricing (Object) - Price breakdown with totalPrice, reservationFee, balanceDue
- payment (Object) - Payment tracking information

Optional Fields:

- address (String) - Required if locationType is "HOME"
- specialRequests (String) - Client special requests
- clientNotes (String) - Notes from client
- adminNotes (String) - Internal admin notes
- createdBy (ObjectId, ref: Users) - Admin who created booking
- confirmedBy (ObjectId, ref: Users) - Admin who confirmed payment
- expiresAt (Date) - 24-hour expiration time for payment
- createdAt (Date) - Booking creation timestamp
- updatedAt (Date) - Last update timestamp

Note:

- service field stores which base service type (Manicure/Pedicure/Mani+Pedi)

- serviceDescription field stores the detailed customization (e.g., "gel polish with chrome finish and floral design")
- Final pricing is itemized in QuotationInvoice after service completion

Quotation Invoices Collection

Required Fields:

- invoiceNumber (String, unique) - Unique invoice identifier (e.g., "INV20260205001")
- booking (ObjectId, ref: Bookings) - Reference to the booking this invoice is for
- client (ObjectId, ref: Clients) - Reference to client
- nailTech (ObjectId, ref: NailTechs) - Reference to nail tech who performed service
- items (Array of Objects) - List of service items with description, quantity, unitPrice, totalPrice
- subtotal (Number) - Sum of all item prices before discount
- totalAmount (Number) - Final amount after discount (what client pays)
- commissionRate (Number) - Nail tech's commission percentage (copied from NailTech document)
- nailTechCommission (Number) - Amount the nail tech earns (totalAmount × commissionRate / 100)
- salonRevenue (Number) - Amount the salon keeps (totalAmount - nailTechCommission)

Optional Fields:

- discountPercentage (Number) - Discount percentage applied (0-100)
- discountAmount (Number) - Discount amount in pesos (subtotal × discountPercentage / 100)
- notes (String) - Additional notes or special terms
- createdBy (ObjectId, ref: Users) - Admin who created the invoice

- createdAt (Date) - Invoice creation date

Calculation Examples:

Subtotal: ₱1,000

Discount (10%): -₱100

Total Amount: ₱900 (what client pays)

Nail Tech Commission (40% of ₱900): ₱360

Salon Revenue (60% of ₱900): ₱540

Additional Details:

- Pricing for items is pulled from Google Sheets named "pricing" which contains a list of services and their prices
- Admin manually adds line items from the pricing sheet when creating an invoice
- Each item can be customized (e.g., "Gel Manicure + Chrome Finish" as separate line items)
- Only created for bookings that have been confirmed (status = "CONFIRMED")
- Commission is calculated on the discounted price (totalAmount), not the subtotal
- Discount cannot exceed the nail tech's authorized discountRate from their profile
- Used to generate detailed invoice for client after service completion and track nail tech earnings

Notifications Collection

Required Fields:

- recipient (String) - Email or phone number
- type (String) - "EMAIL" or "SMS"
- category (String) - Notification type

- content (String) - Message content
- status (String) - "PENDING", "SENT", "FAILED"

Optional Fields:

- subject (String) - Email subject line
- sentAt (Date) - Timestamp when sent
- error (String) - Error message if failed
- createdAt (Date) - Creation timestamp

2. CRUD Operations

Feature Name	Operation	Description
Client Booking System	Create	Save new messages submitted by users.
	Read	Display submitted messages in the admin panel.
	Update	Allow admins to mark messages as resolved.
	Delete	Allow users to delete their own messages.

Feature Name	Operation	Description
Client Management	Create	Register new client when they submit their first booking with complete information (name, email, phone, address, health info).
	Read	Display client list in admin panel with search and filter capabilities. Show client profile with booking history and preferences.
	Update	Update client information when they modify their profile. Automatically update clientType from "NEW" to "REPEAT" after first completed booking. Update statistics (totalBookings, totalSpent, lastVisit) after

		each booking.
	Delete	Soft delete by setting isActive to false. Retain booking history for records.

Feature Name	Operation	Description
Service Management	Create	Admin can add new service types (though typically only 3 exist: Manicure, Pedicure, Mani+Pedi).
	Read	Display active service types to clients during booking process as radio buttons or dropdown selection. Show all services in admin panel.
	Update	Toggle service availability (isActive status) to temporarily disable a service type.
	Delete	Soft delete by setting isActive to false. Maintains historical booking records.

Feature Name	Operation	Description
Nail Tech Management	Create	Add new nail technician with basic information (name, email, phone) and set their commission rate (percentage of earnings) and maximum discount rate they can offer to clients.
	Read	Display list of active nail techs to clients during booking. Show all nail techs with their commission rates and discount rates in admin management page. Display nail tech earnings reports based on commission from completed bookings.
	Update	Modify nail tech contact information, commission rate, or discount rate through

		admin panel. Admin can adjust rates based on performance, experience, or special arrangements.
	Delete	Soft delete by setting isActive to false. Maintain historical booking records and commission calculations.

Feature Name	Operation	Description
Payment Verification (Admin)	Create	Create transaction record when client uploads payment proof.
	Read	Display pending payment bookings in admin dashboard. View uploaded payment proof images.
	Update	Update booking payment status when admin confirms payment receipt. Update booking status from "PENDING_PAYMENT" to "CONFIRMED". Record admin who verified payment and timestamp.
	Delete	Not applicable - payment records are kept for audit trail.

Feature Name	Operation	Description
Quotation Invoices	Create	Create detailed invoice for confirmed bookings by pulling service prices from Google Sheets named "pricing". Admin manually adds line items (e.g., Gel Manicure, Nail Art, Chrome Finish) with quantities and prices. Apply discount if applicable (cannot exceed nail tech's authorized discountRate). System automatically calculates: subtotal, discount amount, total amount, nail tech commission (based on their commissionRate), and salon revenue.

	Read	Display invoice for specific booking showing itemized services, discount applied, total amount, commission breakdown (nail tech earnings vs salon revenue). View all invoices in admin panel. Show invoice to client after service completion. Generate nail tech earnings reports by aggregating commission amounts from all their completed invoices.
	Update	Modify invoice items, quantities, prices, or discount percentage before finalizing. System recalculates commission automatically when invoice is updated. Update notes or special terms.
	Delete	Allow deletion of draft invoices only. Finalized invoices are kept for accounting records and commission tracking.

Feature Name	Operation	Description
Calendar & Availability Management	Create	Create new booking which automatically blocks that time slot.
	Read	Query available time slots by checking existing bookings for specific date and nail tech. Display calendar view with all bookings color-coded by status.
	Update	Reschedule bookings by updating appointmentDate and appointmentTime. System checks for conflicts before allowing update.
	Delete	Cancel bookings which releases the time slot back to available.

Feature Name	Operation	Description

Notification System	Create	Generate notification records for booking confirmations, payment reminders, and appointment reminders based on booking events.
	Read	View notification history and delivery status in admin panel. Track which notifications were successfully sent.
	Update	Update notification status from "PENDING" to "SENT" or "FAILED" after delivery attempt.
	Delete	Auto-delete successfully sent notifications older than 90 days to maintain database performance.

Feature Name	Operation	Description
User/Admin Management	Create	Create new admin user accounts with assigned roles and permissions.
	Read	Display list of admin users with their roles and last login information.
	Update	Modify admin user details, roles, and permissions. Update lastLogin timestamp on each login.
	Delete	Soft delete by setting isActive to false. Maintain audit trail of who created/modified bookings.

3. UI and Data Flow

UI Elements and Their Functions:

UI Element	Functionality

Booking Form (Client)	Multi-step form where clients: 1) Select service type (Home/Studio), 2) Choose base service (Manicure, Pedicure, or Mani+Pedi), 3) Select nail tech, 4) Pick date/time, 5) Describe specific design/customization they want, 6) Enter personal information. Submits data to create new booking.
Service Type Radio Buttons	Simple selection of 3 service types: Manicure, Pedicure, or Mani+Pedi. Client must select one before proceeding to next step.
Service Description Text Area	Large text field where client describes detailed customization (e.g., "French tips with floral nail art and chrome finish"). This will be used by admin to create itemized invoice later.
Nail Tech Selection Cards	Display available nail technicians with name and contact info. Clicking a card selects the nail tech and advances to next step.
Calendar/Time Slot Picker	Interactive calendar showing available dates. Time slots display as buttons with real-time availability (green=available, gray=booked).
Payment Proof Upload	File upload field for clients to submit payment screenshot. Uploads image to cloud storage and saves URL to booking document.
Admin Dashboard	Displays key metrics (today's bookings, pending payments, revenue) retrieved from database using aggregation queries. Shows real-time counts and charts.
Bookings Table (Admin)	Searchable, filterable data table displaying all bookings. Each row has action buttons (View, Confirm Payment, Create Invoice, Cancel) that trigger update operations.
Payment Verification Modal	Admin form to verify payment by viewing uploaded proof, entering reference number, and confirming amount received. Updates booking payment status.
Invoice Creation Form (Admin)	Form to create detailed invoice for confirmed bookings. Admin adds line items by selecting from Google Sheets pricing data. Each item has description, quantity, and unit price. Admin can apply discount (validated against nail

	tech's authorized discountRate). System automatically calculates and displays: subtotal, discount amount, total amount (what client pays), nail tech commission (based on their commissionRate %), salon revenue, and commission breakdown. Shows warning if discount exceeds nail tech's authorized rate.

Data Flow Description:

1. CREATE - Client Booking Submission

Flow:

1. Client clicks "Book Now" button on website
2. Client completes multi-step booking form:
    - Selects location type (Home/Studio)
    - Selects base service type (Manicure, Pedicure, or Mani+Pedi)
    - Describes specific design/customization in text area (e.g., "French tips with floral nail art and chrome finish")
    - Selects preferred nail technician
    - Picks appointment date and time from available slots
    - Enters/verifies personal information (name, email, phone, address if home service)
    - Adds any special requests
3. Admin quotes the estimated price based on service type and description
4. Client clicks "Submit Booking"
5. Frontend validates form data (required fields, email format, phone format)
6. Frontend sends POST request to /api/bookings with booking data
7. Backend server receives request and performs validation
8. Server generates unique bookingNumber (e.g., "GN-20260205001")
9. Server calculates expiresAt (current time + 24 hours)
10. Server creates new Booking document in MongoDB Atlas with service reference and serviceDescription
11. Server creates Client document (if new client) or updates existing client
12. Server creates Notification document for booking confirmation email
13. MongoDB returns created booking document with generated _id
14. Server sends success response with booking details and payment instructions
15. Frontend displays confirmation modal with:

- Booking reference number
- Service type and description
- Payment QR code and instructions
- 24-hour countdown timer
- Upload payment proof button

16. Email notification is sent to client with booking confirmation

MongoDB Operation:

```
javascript
db.bookings.insertOne({
  bookingNumber: "GN-20260205001",
  client: ObjectId("..."),
  service: ObjectId("..."), // Manicure, Pedicure, or Mani+Pedi
  serviceDescription: "French tips with floral nail art and chrome finish",
  nailTech: ObjectId("..."),
  appointmentDate: ISODate("2026-02-10"),
  appointmentTime: "14:00",
  locationType: "HOME",
  status: "PENDING_PAYMENT",
  pricing: {
    totalPrice: 1000,
    reservationFee: 200,
    balanceDue: 800
  },
  expiresAt: ISODate("2026-02-06T14:00:00Z"),
  // ... other fields
})
```

---

2. READ - Admin Views Pending Payments

Flow:

1. Admin logs into admin dashboard
2. Admin navigates to "Bookings" page
3. Admin clicks "Pending Payment" filter tab
4. Frontend sends GET request to /api/bookings?status=PENDING_PAYMENT

5. Backend receives request and queries MongoDB
6. MongoDB executes find() query filtering by status
7. MongoDB also populates references (client, service, nailTech data)
8. Database returns array of pending booking documents
9. Backend formats data and sends response to frontend
10. Frontend receives booking data and renders table
11. Each row displays:
    - Booking number
    - Client name (from populated reference)
    - Service type (Manicure/Pedicure/Mani+Pedi from populated reference)
    - Service description (custom details)
    - Appointment date/time
    - Time remaining until expiration
    - Payment proof thumbnail (if uploaded)
    - Action buttons (View Proof, Confirm, Cancel)

MongoDB Operation:

```
javascript
db.bookings.find({
  status: "PENDING_PAYMENT"
}).populate('client', 'firstName lastName email')
  .populate('service', 'name')
  .populate('nailTech', 'name')
  .sort({ createdAt: -1 })
```

---

3. UPDATE - Admin Confirms Payment

Flow:

1. Admin clicks "View" button on pending payment booking
2. Payment verification modal opens showing:
    - Client details
    - Booking information
    - Uploaded payment proof image
3. Admin verifies payment screenshot matches booking amount
4. Admin enters payment reference number from screenshot

5. Admin selects payment method (GCash, Bank Transfer, Cash)
6. Admin clicks "Confirm Payment" button
7. Frontend shows confirmation dialog
8. Admin confirms action
9. Frontend sends PATCH request to /api/bookings/:id/confirm-payment with:
    - Payment reference number
    - Payment method
    - Confirmed amount
10. Backend validates admin session and permissions
11. Backend retrieves booking from database
12. Backend updates booking document:
    - Sets payment.reservationPaid = true
    - Sets payment.reservationPaidAt = current timestamp
    - Sets payment.reservationReference = reference number
    - Sets status = "CONFIRMED"
    - Sets confirmedBy = admin user ID
13. Backend updates Client document statistics
14. Backend creates Notification document for confirmation email
15. MongoDB executes update operation
16. Server sends success response
17. Frontend updates UI:
    - Removes booking from pending list
    - Shows success message
    - Displays in confirmed bookings
18. Email/SMS confirmation sent to client

MongoDB Operation:

```
javascript
db.bookings.updateOne(
  { _id: ObjectId("65a1b2c3d4e5f6g7h8i9j0k5") },
  {
    $set: {
      "payment.reservationPaid": true,
      "payment.reservationPaidAt": new Date(),
      "payment.reservationReference": "GC123456789",
      "payment.reservationPaymentMethod": "GCASH",
      "status": "CONFIRMED",
```

```
    "confirmedBy": ObjectId("admin-id"),
    "updatedAt": new Date()
   }
 }
)
```

---

4. DELETE - Auto-Cancel Expired Bookings

Flow:

1. System cron job runs every hour (scheduled background task)
2. Cron job queries database for expired unpaid bookings
3. Query finds bookings where:
    - status = "PENDING_PAYMENT"
    - expiresAt < current time
    - payment.reservationPaid = false
4. For each expired booking:
    - Update status to "CANCELLED"
    - Set cancellationReason = "Payment not received within 24 hours"
    - Release the time slot (no longer blocks availability)
5. Create notification records for cancellation emails
6. Send cancellation notification to clients
7. Log cancellation events for admin review

MongoDB Operation:

```
javascript
db.bookings.updateMany(
 {
  status: "PENDING_PAYMENT",
  "payment.reservationPaid": false,
  expiresAt: { $lt: new Date() }
 },
 {
  $set: {
   status: "CANCELLED",
   cancellationReason: "Payment not received within 24 hours",
```

```
    updatedAt: new Date()
   }
 }
)
```

---

5. READ - Check Available Time Slots

Flow:

1. Client selects nail technician in booking form
2. Client clicks on calendar to select date
3. Frontend sends GET request to /api/availability with:
    - Selected date
    - Nail tech ID
4. Backend receives request
5. Backend queries Bookings collection for existing bookings on that date for that nail tech
6. Backend calculates available time slots:
    - Defines standard operating hours (e.g., 9 AM - 6 PM)
    - Removes slots with existing confirmed bookings
    - Removes standard lunch break (12 PM - 1 PM)
    - Adds 15-minute buffer between appointments
    - Generates hourly time slots (e.g., 9:00, 10:00, 11:00, 2:00, 3:00, 4:00, 5:00)
7. Backend returns array of available time slots
8. Frontend displays time slots as clickable buttons
9. Available slots shown in green
10. Booked slots grayed out and disabled

MongoDB Operations:

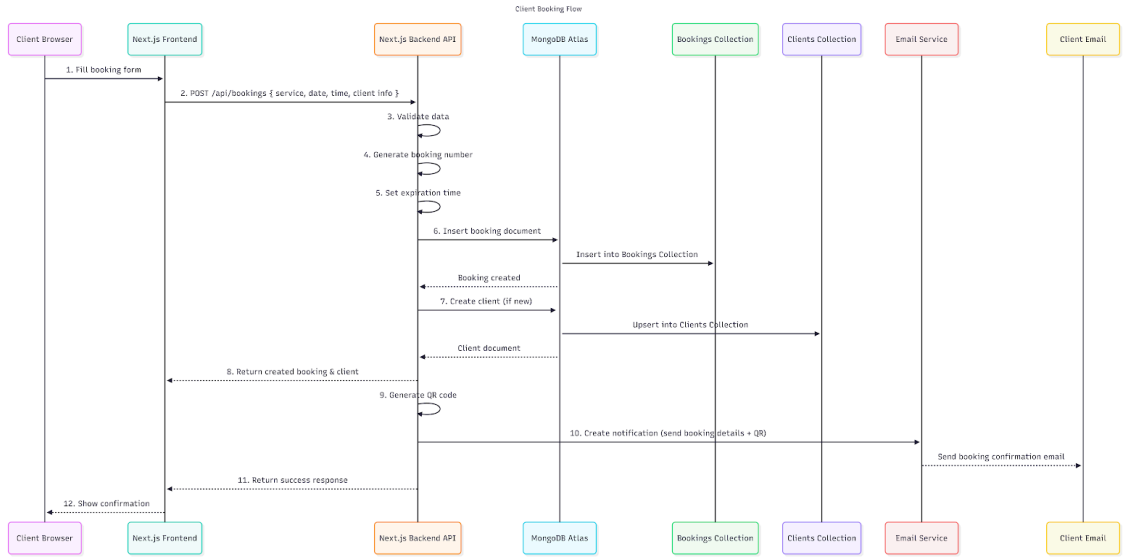
```
javascript
// Get existing bookings for the date and nail tech
db.bookings.find({
  nailTech: ObjectId("tech-id"),
  appointmentDate: ISODate("2026-02-10"),
  status: { $in: ["PENDING_PAYMENT", "CONFIRMED"] }
```

})

4. Diagrams

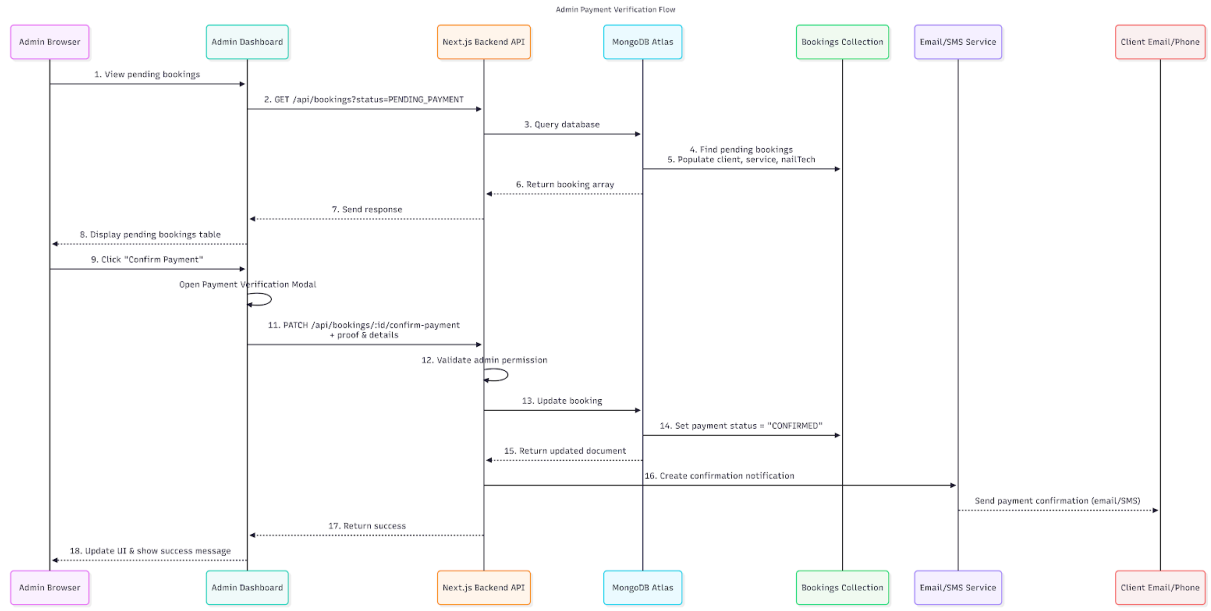
CLIENT BOOKING FLOW

Client Booking Flow.png



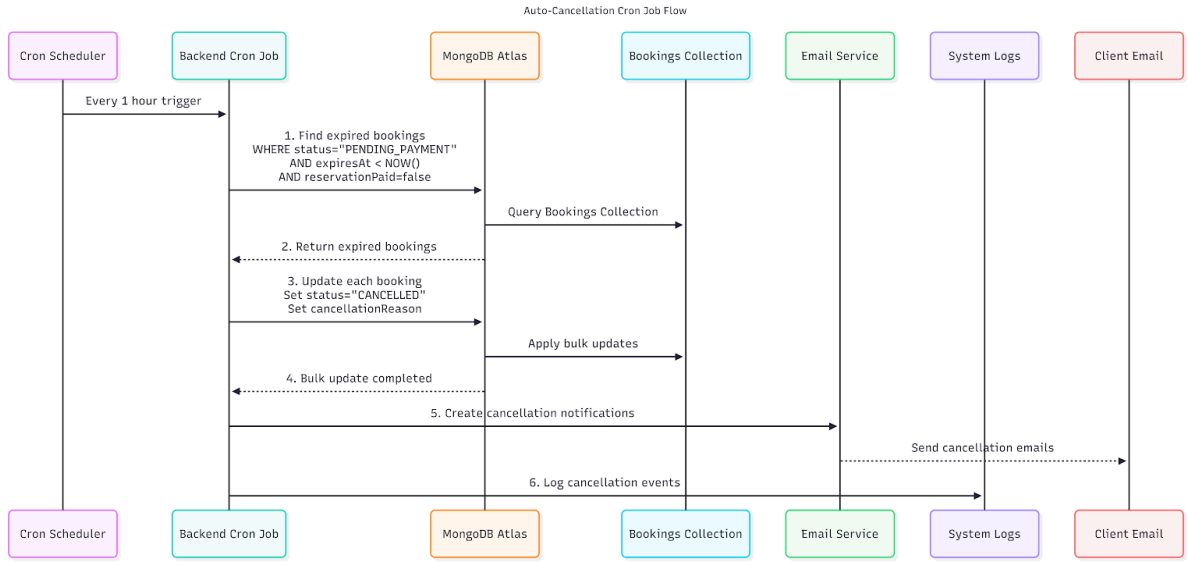
ADMIN PAYMENT VERIFICATION FLOW

Admin Payment Verification Flow.png



Auto Cancelation Cron Job Flow

Auto Cancelation Cron Job Flow.png



4. Additional Notes

Data Validation Rules:

- **Email**: Must follow standard email format (validated using regex)
- **Phone**: Philippine mobile format (+639XXXXXXXXX or 09XXXXXXXXX)
- **Appointment Date**: Cannot be in the past, must be within 60 days advance booking window
- **Appointment Time**: Must be in HH:MM format (e.g., "14:00"), must not conflict with existing bookings
- **Service Description**: Required field, minimum 10 characters, maximum 500 characters
- **Booking Expiration**: Automatically set to 24 hours from creation time
- **Client Type**: Automatically changes from "NEW" to "REPEAT" after first completed booking
- **Location Type**: Must be either "HOME" or "STUDIO"
- **Address**: Required if location type is "HOME"

Database Indexing Strategy:

To ensure fast queries and good performance, the following indexes will be created:

Bookings Collection:

- bookingNumber (unique index) - for fast lookup by booking reference
- appointmentDate + nailTech + status (compound index) - for availability checking
- client (index) - for retrieving client booking history
- service (index) - for filtering by service type
- status (index) - for filtering by booking status
- expiresAt (TTL index) - for automatic cleanup of old cancelled bookings

Clients Collection:

- email (unique index) - for login and duplicate prevention
- phone (unique index) - for client lookup and verification
- clientType (index) - for filtering new vs repeat clients

Services Collection:

- name (unique index) - ensures only one of each service type exists
- isActive (index) - for filtering active services

NailTechs Collection:

- email (unique index)
- isActive (index)

QuotationInvoices Collection:

- invoiceNumber (unique index) - for fast invoice lookup
- booking (index) - for finding invoice by booking
- client (index) - for client invoice history

Security Considerations:

- **Password Hashing**: All admin passwords hashed using bcrypt with 10 salt rounds
- **Authentication**: JWT tokens with httpOnly cookies for admin sessions

- **Authorization**: Role-based access control (RBAC) for different admin levels
- **Input Sanitization**: All user inputs sanitized to prevent XSS and injection attacks
- **File Upload**: Payment proof images validated for file type and size, stored in secure cloud storage
- **Rate Limiting**: API endpoints rate-limited to prevent abuse (100 requests per 15 minutes per IP)

Backup and Data Retention:

- **Automatic Backups**: MongoDB Atlas performs automatic daily backups
- **Data Retention**:
    - Active bookings: Indefinitely
    - Completed bookings: 2 years for accounting/tax purposes
    - Cancelled bookings: 90 days then soft deleted
    - Notifications: 90 days then automatically deleted
    - Quotations: 30 days after declined/expired

Google Sheets Backup Integration:

- Every confirmed booking automatically backed up to Google Sheets
- Backup includes: booking number, client info, service, date/time, payment status
- Used as secondary backup and for easy sharing with non-technical staff
- Updated via Google Sheets API whenever booking status changes

Scalability Considerations:

- **Connection Pooling**: Reuse database connections for better performance
- **Caching**: Frequently accessed data (services list, nail tech info) cached in Redis
- **Pagination**: All list views paginated (20 items per page) to reduce load
- **Lazy Loading**: Calendar loads one month at a time on demand
- **Image Optimization**: Payment proofs compressed and resized before storage

Notification Timing:

Automated Emails/SMS sent at:

- T+0 minutes: Booking confirmation (immediate)
- T+1 hour: Payment reminder (if unpaid)
- T+12 hours: Payment urgent reminder (if unpaid)

- T+23 hours: Final payment warning (if unpaid)
- T+24 hours: Cancellation notice (if unpaid)
- T-24 hours: Appointment reminder (1 day before)
- T-2 hours: Final appointment reminder