AI NEXUS

Hospital Patient Management System

Patient Module — Features & UI Test Flow Guide

Version 1.2.0 · February 2026 · CONFIDENTIAL
Prepared by Ai Nexus Engineering Team
Updated: Added 7 new features — Vitals, Insurance, Family Relationships, Photo Upload, ID Card, Duplicate Detection, CSV Export

Table of Contents

1. Overview & System Architecture

The Patient Module is the foundational data layer of the Ai Nexus Hospital Management System. It provides HIPAA-compliant patient record management including registration, search, profile view, demographic updates, status management, vitals recording, insurance tracking, family relationships, photo management, and a full HIPAA audit trail. All PHI is encrypted at rest (AES-256-GCM) and every access is immutably audit-logged.

Layer	Technology	Purpose
Frontend	React 18.3.1 + TypeScript + Ant Design 5	SPA at http://localhost
API Gateway	Nginx (port 80)	Reverse-proxy; routes `/api/*` to backend
Backend	Spring Boot 3.4.1 / Java 17	REST API at `hospital-api:8080`
Security	Spring Security 6 + JWT (HMAC-SHA256)	Stateless auth + RBAC
Database	PostgreSQL 15	Patient records + immutable audit logs
Audit	AOP `@AfterReturning` + `REQUIRES_NEW` TX	All read/write actions logged
Encryption	AES-256-GCM (`AttributeConverter`)	PHI fields encrypted in DB
Search Index	Plaintext index cols + HMAC hashes	Enables search over encrypted data
Charts	Recharts	Vitals trend line charts
QR Code	qrcode.react	Patient ID card QR generation

> **Base URL:** http://localhost | **API Base:** http://localhost/api/v1 | **Health:** http://localhost/actuator/health

API Endpoints Summary

Resource	Endpoint	Methods
Auth	`/api/v1/auth/dev-login`	POST
Patients	`/api/v1/patients`	GET, POST
Patient	`/api/v1/patients/{patientId}`	GET, PUT
Status	`/api/v1/patients/{patientId}/status`	PATCH
Photo	`/api/v1/patients/{patientId}/photo`	GET, POST, DELETE
Duplicates	`/api/v1/patients/{patientId}/potential-duplicates`	GET
Vitals	`/api/v1/patients/{patientId}/vitals`	GET, POST
Insurance	`/api/v1/patients/{patientId}/insurance`	GET, POST
Insurance item	`/api/v1/patients/{patientId}/insurance/{id}`	PUT, DELETE
Relationships	`/api/v1/patients/{patientId}/relationships`	GET, POST
Relationship item	`/api/v1/patients/{patientId}/relationships/{relatedId}`	DELETE
Audit Trail	`/api/v1/patients/{patientId}/audit-trail`	GET
CSV Export	`/api/v1/patients/export/csv`	GET

2. User Roles & Permissions

The module enforces Role-Based Access Control (RBAC) at both the API layer (Spring Security) and the UI layer (conditional rendering). Four roles are supported:

Role	Badge	Register	View/Search	Edit	Status	Vitals	Audit Trail	Photo	Insurance	Relationships	CSV Export
RECEPTIONIST	Blue	✔	✔	✔	✘	✘	✘	✔	✔	✔	✔
DOCTOR	Green	✘	✔	✘	✘	✔	✘	✘	✘	✘	✘
NURSE	Cyan	✘	✔	✘	✘	✔	✘	✘	✘	✘	✘
ADMIN	Red	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

> **How to login as any role:** Go to http://localhost/login → enter any username → select role from dropdown → Sign In.

FEATURE 1

3. Authentication — Dev Login

A development-only login page issues a real HMAC-SHA256 JWT token so the UI can be tested without a production Auth Module. The token is stored in `localStorage` and sent as `Authorization: Bearer <token>` on every API call.

UI Test Steps

#	Step	Action / Expected Result
1	Open browser	Navigate to http://localhost — redirects to `/login` automatically
2	Enter username	Type any username, e.g. `receptionist1`, `dr.smith`, `nurse.jane`, `admin`
3	Select role	Choose from dropdown: RECEPTIONIST | DOCTOR | NURSE | ADMIN
4	Read role description	Description below dropdown explains permissions for selected role
5	Click Sign In	`POST /api/v1/auth/dev-login` → JWT token issued and stored
6	Observe header	Username + colour-coded role badge + Sign Out button appear in navbar
7	Confirm redirect	Automatically navigated to `/patients` (Patient List page)

Expected Results

- JWT token stored in `localStorage` under key `'token'`
- Header shows: username | role badge (RECEPTIONIST=blue, DOCTOR=green, NURSE=cyan, ADMIN=red)
- Redirected to Patient List page
- Dev mode warning banner visible on login page

Test All 4 Roles

Username	Role	Expected Header Badge
receptionist1	RECEPTIONIST	Blue badge
dr.smith	DOCTOR	Green badge
nurse.jane	NURSE	Cyan badge
admin	ADMIN	Red badge

FEATURE 2

4. Patient Registration

Allows RECEPTIONIST and ADMIN roles to register new patients. The system generates a unique Patient ID (format: `P` + year + 3-digit sequence, e.g. **P2026001**) and optionally a MRN (`MRN2026001`), validates all fields, detects potential duplicate phone numbers, and records the registration user and timestamp.

4.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN** before testing.

4.2 Happy Path — Register a New Patient

#	Step	Action
1	Navigate	From Patient List, click **Register New Patient** button (top-right)
2	Personal Info	First Name = `Sarah`, Last Name = `Connor`, DOB = `1985-07-04`, Gender = `Female`
3	Contact Info	Phone = `+1-555-200-0001`, Email = `sarah.connor@example.com`, Address = `789 Elm St`, City = `Chicago`, State = `IL`, ZIP = `60601`
4	Emergency Contact	Name = `John Connor`, Phone = `+1-555-200-0002`, Relationship = `Son`
5	Medical Info	Blood Group = `B+`, Known Allergies = `Sulfa drugs`, Chronic Conditions = `Asthma`
6	Submit	Click **Register Patient** button
7	Verify success	Toast notification: `'Patient registered — ID: P2026XXX'`
8	Verify redirect	Automatically navigated to patient detail page for the new patient

4.3 Form Sections

Section	Required Fields	Optional Fields
Personal Information	First Name, Last Name, Date of Birth, Gender	—
Contact Information	Phone Number	Email, Address, City, State, ZIP
Emergency Contact	—	Name, Phone, Relationship
Medical Information	—	Blood Group, Allergies, Conditions

4.4 Validation Testing

#	Test Case	Expected Result
1	Blank required field	Click into First Name, tab away → red error: `'First name is required'`
2	Blank phone	Click into Phone, tab away → red error: `'Phone number is required'`
3	Invalid phone format	Type `'12345'` → red error: `'Phone: +1-XXX-XXX-XXXX, (XXX) XXX-XXXX, or XXX-XXX-XXXX'`
4	Invalid email	Type `'notanemail'` → red error: `'Invalid email address'`
5	Future date of birth	Open date picker → future dates are greyed out and unselectable

6	Submit empty form	Click Register Patient with no data → all required fields highlight red

4.5 Duplicate Phone Warning

#	Step	Expected Result
1	Register patient 1	Register with phone `+1-555-200-0001` → success (P2026XXX)
2	Register patient 2	Register another patient with same phone `+1-555-200-0001`
3	Observe warning	Yellow warning notification (10s): `'Another patient with this phone number may already exist'`
4	Confirm registration	Second patient is registered successfully despite warning (both records saved)

4.6 RBAC — Register Button Hidden for DOCTOR/NURSE

- Login as `dr.smith / DOCTOR` → Register New Patient button is **NOT** visible on Patient List
- Login as `nurse.jane / NURSE` → Register New Patient button is **NOT** visible on Patient List
- Direct URL `/patients/new` as DOCTOR → `POST /api/v1/patients` returns `403 Forbidden`

FEATURE 3 — UPDATED

5. Patient Search & Filtering

The Patient List page provides live search with 300ms debounce across multiple fields, combined with dropdown filters for status, gender, and blood group. An **Advanced Filters** collapsible panel adds city, state, age range, and clinical condition toggles. All authenticated roles can search and view the patient list.

5.1 Default State

- Page loads at http://localhost/patients
- Active patients shown by default (Status filter pre-set to `'Active'`)
- Table columns: **Patient ID | MRN | Full Name | Age | Gender | Phone | Status | Action**
- Pagination shows total count (e.g. `'7 patients'`) with configurable page size

5.2 Search Test Cases

#	Test Case	Expected Result
1	Search by name	Type `'Connor'` → only patients with Connor in name appear
2	Search by Patient ID	Type `'P2026001'` → exact patient appears
3	Search by MRN	Type `'MRN2026001'` → matching patient appears
4	Search by phone	Type `'555-200'` → patients with that phone prefix appear
5	Search by email	Type `'sarah.connor'` → matching patient appears
6	Clear search	Click ✕ on search box → full list restored
7	No results	Type `'ZZZNOMATCH'` → table shows `'No patients found'`
8	Live debounce	Type quickly — results update 300ms after last keystroke, not per character

5.3 Basic Filter Test Cases

#	Filter	Expected Result
1	Status: Inactive	Only deactivated patients shown
2	Status: clear	All statuses shown
3	Gender: Female	Only female patients
4	Blood Group: B+	Only B+ patients
5	Combine filters	Search `'Connor'` + Gender `'Female'` → only female Connors
6	Clear all filters	Click ✕ on each filter → full active list restored

5.4 Advanced Filters (collapsible panel)

Click ▶ **Advanced Filters** to expand. The following additional filters are available:

#	Filter	Behaviour
1	City	Text input — filters by city (case-insensitive)
2	State	Text input — filters by state

3	Age From	Number input (0–150) — minimum age
4	Age To	Number input (0–150) — maximum age
5	Has Allergies toggle	Switch — when ON shows only patients with known allergies recorded
6	Has Chronic Conditions toggle	Switch — when ON shows only patients with chronic conditions recorded
7	Clear button	Resets all 6 advanced filters at once

> **Note:** Age range is converted to birth year range internally (e.g. Age 30–40 → birth years 1986–1996 for year 2026). All filters combine with AND logic.

5.5 Navigation from List

- Click any patient row → navigates to `/patients/{patientId}` (patient detail page)
- RECEPTIONIST/ADMIN rows show **Edit** button in Action column
- DOCTOR/NURSE rows show no Edit button in Action column

FEATURE 4 — UPDATED

6. Patient Profile View

The patient detail page displays all patient information organized in **nine sections**: Personal Info (with photo), Contact Info, Emergency Contact, Medical Info, Family Relationships, Insurance, Vitals History, and Record Information. A **Duplicate Detection alert** appears at the top if potential duplicates exist. Every access generates an immutable HIPAA READ audit log entry.

6.1 Page Header Elements

Element	Description
Patient full name	Large heading
Patient ID	Subtitle — e.g. `P2026001`
MRN tag	Blue tag next to ID — e.g. `MRN: MRN2026001` (if assigned)
Back to List	Returns to `/patients`
Audit Trail	Button — ADMIN only; opens audit trail modal
Print ID Card	Button — all roles; opens ID card print modal
Edit Patient	Button — RECEPTIONIST and ADMIN only
Deactivate/Activate	Button — ADMIN only

6.2 Test Steps — Information Cards

#	Card	What to Verify
1	Personal Info	Status badge, patient photo (or initials avatar), DOB (formatted), Age (auto-calculated), Gender, Blood Group
2	Contact Info	Phone, Email, Address, City, State, ZIP (shows `'—'` if empty)
3	Emergency Contact	Name, Phone, Relationship (shows `'—'` if empty)
4	Medical Info	Known Allergies, Chronic Conditions (shows `'—'` if empty)
5	Family Relationships	Linked patient names with relationship type colour-coded tags
6	Insurance	Insurance records with provider, policy, coverage type, primary badge
7	Vitals History	Latest readings stats, trend chart (≥2 entries), history table
8	Record Info	Registered By, Registered At, Last Updated By, Last Updated At

6.3 RBAC — Action Buttons by Role

Role	Edit Patient	Deactivate/Activate	Audit Trail	Record Vitals	Manage Insurance	Manage Relationships	Upload Photo
RECEPTIONIST	✔ Visible	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible	✔ Visible	✔ Visible
DOCTOR	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible	✘ Hidden	✘ Hidden	✘ Hidden
NURSE	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible	✘ Hidden	✘ Hidden	✘ Hidden
ADMIN	✔ Visible	✔ Visible	✔ Visible	✔ Visible	✔ Visible	✔ Visible	✔ Visible

6.4 Status Badge Colour Coding

- ACTIVE patient → green dot + `'Active'` text (never colour alone — WCAG 2.1 AA compliant)
- INACTIVE patient → red dot + `'Inactive'` text

6.5 404 Page

#	Step	Expected Result
1	Navigate to invalid ID	Visit http://localhost/patients/INVALID999
2	Observe error	Error alert: patient not found / 404 response from API

FEATURE 5

7. Edit Patient

RECEPTIONIST and ADMIN roles can update any patient's demographic information. The edit form is pre-populated with current data, applies the same validation rules as registration, and records who made the update and when.

7.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN**.

7.2 Happy Path — Edit Patient

#	Step	Action / Expected Result
1	Open patient detail	Navigate to any patient's detail page
2	Click Edit Patient	Navigates to `/patients/{id}/edit`
3	Verify pre-fill	All fields pre-populated with current patient data
4	Modify fields	Change City to `'Springfield'`, add `', Diabetes'` to Chronic Conditions
5	Submit	Click **Save Changes** button
6	Verify success	Toast: `'Patient updated'`
7	Verify redirect	Returns to patient detail page
8	Verify changes saved	Updated City and Conditions visible in detail page
9	Verify audit fields	Record Info shows: Last Updated By = your username, Last Updated At = now

7.3 Read-Only Fields

- **Patient ID** — shown in page subtitle only, not in form
- **Registration date** — not present in edit form, only in detail view

7.4 Validation on Edit Form

#	Test Case	Expected Result
1	Clear First Name	Delete first name, tab away → red error: `'First name is required'`
2	Invalid email	Change email to `'bad-email'` → red error: `'Invalid email address'`
3	Cancel	Click Cancel → returns to detail page with no changes saved

FEATURE 6

8. Deactivate / Activate Patient

ADMIN-only feature. Patient records are never permanently deleted — they are deactivated (soft delete). Deactivation requires a confirmation modal. Activation is immediate. Both actions are fully audit-logged.

8.1 Prerequisites

> Login as **admin / ADMIN**.

8.2 Deactivate a Patient

#	Step	Expected Result
1	Open an ACTIVE patient	Navigate to any patient with green `'Active'` status badge
2	Click Deactivate	Click the red **Deactivate Patient** button
3	Confirmation modal	Modal appears: `'Are you sure you want to deactivate this patient?'`
4	Click Cancel	Modal closes, patient remains ACTIVE
5	Click Deactivate again	Modal appears again
6	Confirm	Click Deactivate in modal → action executes
7	Verify status	Status badge: green `'Active'` → red `'Inactive'`
8	Verify button change	Button label changes to `'Activate Patient'`
9	Toast message	Success toast: `'Patient deactivated'`
10	Verify list filter	Return to list (Active filter) → deactivated patient no longer appears

8.3 Activate a Patient

#	Step	Expected Result
1	Open an INACTIVE patient	From list: clear Status filter → find patient with red `'Inactive'` badge → click row
2	Click Activate	Click **Activate Patient** button — no confirmation needed
3	Verify instant action	No modal — activates immediately
4	Verify status	Status badge: red `'Inactive'` → green `'Active'`
5	Toast message	Success toast: `'Patient activated'`
6	Verify in list	Return to list (Active filter) → patient appears again

8.4 RBAC — Deactivate Button Hidden for non-ADMIN

- RECEPTIONIST — Deactivate/Activate button not visible on detail page
- DOCTOR — Deactivate/Activate button not visible on detail page
- NURSE — Deactivate/Activate button not visible on detail page
- Direct API call `PATCH /api/v1/patients/{id}/status` as RECEPTIONIST → `403 Forbidden`

FEATURE 7 — UPDATED

9. Role-Based UI Visibility

The UI adapts dynamically based on the logged-in user's role. Buttons and controls are **hidden** — not just disabled — for unauthorized roles. This enforces the minimum necessary principle at the presentation layer (API layer enforces it independently).

UI Element / Page	RECEPTIONIST	DOCTOR	NURSE	ADMIN
Register New Patient (List)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Export CSV (List)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Edit button per row (List)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Edit Patient button (Detail)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Deactivate Patient (Detail)	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible
Activate Patient (Detail)	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible
Audit Trail button (Detail)	✘ Hidden	✘ Hidden	✘ Hidden	✔ Visible
Print ID Card (Detail)	✔ Visible	✔ Visible	✔ Visible	✔ Visible
Record Vitals button	✘ Hidden	✔ Visible	✔ Visible	✔ Visible
Add Insurance (Detail)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Edit/Delete Insurance (Detail)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Add Family Link (Detail)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Upload/Delete Photo (Detail)	✔ Visible	✘ Hidden	✘ Hidden	✔ Visible
Patient list — view/search	✔	✔	✔	✔
Patient detail — view	✔	✔	✔	✔

How to Verify

#	Step	What to Check
1	Login as RECEPTIONIST	Register + Edit + CSV + Photo + Insurance + Relationships visible. No Deactivate, no Vitals record, no Audit Trail.
2	Logout → Login DOCTOR	Record Vitals visible. No Register, no Edit, no Deactivate, no Audit Trail.
3	Logout → Login NURSE	Same as DOCTOR — can record vitals, read-only otherwise.
4	Logout → Login ADMIN	All buttons visible including Deactivate/Activate and Audit Trail.

FEATURE 8

10. Sign Out

#	Step	Expected Result
1	Locate Sign Out button	Top-right corner of the app header — visible when logged in
2	Click Sign Out	Click the Sign Out button (door icon)
3	Token cleared	`localStorage 'token'` key is removed
4	Verify redirect	Navigated to `/login` page
5	Verify session ended	Header no longer shows username or role badge
6	Test protected URL	Visit http://localhost/patients — API returns 401, error displayed
7	Verify login required	Must log in again to access any patient data

- JWT token removed from `localStorage` on sign out
- Subsequent API calls return `401 Unauthorized`
- Login page shown — user must re-authenticate

FEATURE 9 — HIPAA — UPDATED

11. HIPAA Audit Trail

Every access to patient data generates an immutable audit log entry stored in PostgreSQL. The audit trail is enforced at the AOP layer (Spring `@AfterReturning` aspect) using a `REQUIRES_NEW` transaction. Database-level triggers prevent any `UPDATE` or `DELETE` of audit records. **New in v1.2.0:** An in-app Audit Trail modal is available to ADMIN users directly on the patient detail page.

11.1 Audit Log Schema

Field	Type	Content
`user_id`	VARCHAR(100)	UUID of the logged-in user
`username`	VARCHAR(100)	Display username (e.g. `receptionist1`)
`user_role`	VARCHAR(50)	Role at time of action
`action`	VARCHAR(30)	Action type (see 11.2)
`patient_id`	VARCHAR(10)	Patient ID (e.g. P2026001) — no PHI stored
`ip_address`	VARCHAR(45)	Request IP address
`occurred_at`	TIMESTAMPTZ	UTC timestamp of the action

11.2 All Audit Action Types

User Action in UI	API Endpoint	Audit Action
Register new patient	`POST /api/v1/patients`	`PATIENT_CREATE`
View patient detail page	`GET /api/v1/patients/{id}`	`PATIENT_READ`
Submit edit form	`PUT /api/v1/patients/{id}`	`PATIENT_UPDATE`
Confirm deactivation	`PATCH /api/v1/patients/{id}/status`	`PATIENT_DEACTIVATE`
Click Activate Patient	`PATCH /api/v1/patients/{id}/status`	`PATIENT_ACTIVATE`
Record vitals	`POST /api/v1/patients/{id}/vitals`	`VITALS_CREATE`
View vitals list	`GET /api/v1/patients/{id}/vitals`	`VITALS_READ`
Add insurance	`POST /api/v1/patients/{id}/insurance`	`INSURANCE_CREATE`
Update insurance	`PUT /api/v1/patients/{id}/insurance/{id}`	`INSURANCE_UPDATE`
Delete insurance	`DELETE /api/v1/patients/{id}/insurance/{id}`	`INSURANCE_DELETE`
Add family relationship	`POST /api/v1/patients/{id}/relationships`	`RELATIONSHIP_ADD`
Remove family relationship	`DELETE /api/v1/patients/{id}/relationships/{id}`	`RELATIONSHIP_REMOVE`
Upload patient photo	`POST /api/v1/patients/{id}/photo`	`PHOTO_UPLOAD`
Delete patient photo	`DELETE /api/v1/patients/{id}/photo`	`PHOTO_DELETE`

11.3 Audit Trail UI Modal (ADMIN only — NEW)

#	Step	Expected Result

1	Login as ADMIN	Navigate to any patient detail page
2	Click Audit Trail	Button in page header → modal opens (900px wide)
3	View entries	Table shows: Time, Action (colour-coded tag), Username, Role, IP Address
4	Action colours	CREATE=green, READ=blue, UPDATE=orange, DEACTIVATE=red, ACTIVATE=green, VITALS/INSURANCE/RELATIONSHIP actions in various colours
5	Pagination	20 entries per page; older entries on subsequent pages
6	Always fresh	Data always re-fetched on open (`staleTime: 0`) — no cached data
7	Close modal	Click ✕ or press Escape → modal closes
8	Non-ADMIN	Audit Trail button is not visible; `GET /api/v1/patients/{id}/audit-trail` returns `403`

11.4 Database Verification

Run after performing UI actions:

```
docker exec hospital-postgres psql -U hospital_user -d hospital_db \
  -c "SELECT to_char(occurred_at,'HH24:MI:SS') AS time, action, patient_id, username, user_role \
      FROM audit_logs ORDER BY occurred_at DESC LIMIT 15;"
```

11.5 Immutability Test

#	Step	Expected Result
1	Attempt UPDATE	`UPDATE audit_logs SET action='TAMPERED' WHERE id=1;`
2	Verify rejection	PostgreSQL raises: `'audit_logs are immutable (HIPAA requirement)'`
3	Attempt DELETE	`DELETE FROM audit_logs WHERE id=1;`
4	Verify rejection	Same immutability error — row cannot be deleted

HIPAA Note: Audit logs are retained for a minimum of 6 years per HIPAA §164.530(j). Records older than 6 years are moved to `audit_logs_archive` (also immutable). No PHI is stored in the audit log — only patient IDs.

FEATURE 10 — NEW

12. Vitals History

Allows DOCTOR, NURSE, and ADMIN roles to record patient vital signs. All authenticated roles can view vitals history. The `VitalsHistoryCard` on the patient detail page shows the latest readings, a trend chart, and a paginated history table.

12.1 Prerequisites

> Login as **dr.smith / DOCTOR**, **nurse.jane / NURSE**, or **admin / ADMIN** to record. Any role can view.

12.2 Record Vitals — Happy Path

#	Step	Action / Expected Result
1	Open patient detail	Navigate to any patient's detail page
2	Scroll to Vitals History card	Card is at the bottom of the detail page
3	Click Record Vitals	Button in card header (visible for DOCTOR, NURSE, ADMIN) → modal opens
4	Fill temperature	Enter `37.2` (°C)
5	Fill blood pressure	Systolic = `120`, Diastolic = `80`
6	Fill pulse	Enter `72` (BPM)
7	Fill respiratory rate	Enter `16`
8	Fill SpO_2	Enter `98.5` (%)
9	Fill weight	Enter `72.5` (kg) — BMI auto-calculated if height available
10	Submit	Click **Save Vitals** → `POST /api/v1/patients/{id}/vitals`
11	Verify success	Modal closes; new row appears at top of vitals table
12	Verify latest readings	Latest readings row updates with new values

12.3 Vitals Fields

Field	Unit	Required	Notes
Temperature	°C	No	Decimal precision (e.g. 37.5)
Blood Pressure Systolic	mmHg	No	Integer
Blood Pressure Diastolic	mmHg	No	Integer; both systolic and diastolic together
Pulse Rate	BPM	No	Integer
Respiratory Rate	breaths/min	No	Integer
Oxygen Saturation (SpO_2)	%	No	Decimal precision
Weight	kg	No	Decimal precision
BMI	—	No	Auto-calculated: weight(kg) / height(m)²

12.4 Vitals Display — Three Sections

Section	Description	Condition
Latest Readings Row	Stat cards: BP, Pulse, Temp, SpO_2, Weight, BMI	Shown when ≥1 entry exists
Trend Chart	Recharts `LineChart` showing Pulse, BP Systolic, BP Diastolic, SpO_2 over time	Shown when ≥2 entries; up to last 10 readings; chronological order
History Table	Full paginated table (10/page) with all vitals and who recorded them	Always shown when entries exist

12.5 RBAC Test

#	Test	Expected Result
1	Login as DOCTOR	Record Vitals button **visible** in Vitals History card
2	Login as NURSE	Record Vitals button **visible**
3	Login as RECEPTIONIST	Record Vitals button **NOT visible**
4	Direct API as RECEPTIONIST	`POST /api/v1/patients/{id}/vitals` → `403 Forbidden`

FEATURE 11 — NEW

13. Patient Insurance

Allows RECEPTIONIST and ADMIN roles to add, edit, and delete patient insurance records. Multiple insurance records are supported per patient. The primary insurance is visually marked.

13.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN**.

13.2 Add Insurance — Happy Path

#	Step	Action / Expected Result
1	Open patient detail	Navigate to any patient's detail page
2	Scroll to Insurance card	Card shows existing insurance records (or empty state)
3	Click Add Insurance	Button in card header → modal opens
4	Fill provider	Enter `'BlueCross BlueShield'`
5	Fill policy number	Enter `'BCB-2026-001234'`
6	Fill group number	Enter `'GRP-5678'` (optional)
7	Coverage type	Select `'MEDICAL'` from dropdown
8	Subscriber name	Enter `'Sarah Connor'` (optional)
9	Subscriber DOB	Enter subscriber date of birth (optional)
10	Valid From	Select start date
11	Valid To	Select end date (optional)
12	Mark as Primary	Check **Is Primary** toggle
13	Submit	Click **Save** → `POST /api/v1/patients/{id}/insurance`
14	Verify success	Modal closes; new insurance record appears in card
15	Verify primary badge	**PRIMARY** badge shown on the primary insurance record

13.3 Insurance Fields

Field	Required	Notes
Provider Name	Yes	Insurance company name
Policy Number	Yes	Policy identifier
Group Number	No	Group/employer code
Coverage Type	No	e.g. MEDICAL, DENTAL, VISION
Subscriber Name	No	Name of the policy holder
Subscriber DOB	No	Date of birth of subscriber

Valid From	No	Coverage start date
Valid To	No	Coverage end date
Is Primary	No	Boolean toggle; marks this as the primary insurance

13.4 Edit Insurance

#	Step	Expected Result
1	Click Edit on an insurance record	Modal opens pre-populated with all current values
2	Update policy number	Change value and save
3	Verify success	Updated record shown in insurance card

13.5 Delete Insurance

#	Step	Expected Result
1	Click Delete on an insurance record	Confirmation dialog appears
2	Confirm delete	`DELETE /api/v1/patients/{id}/insurance/{insuranceId}` → record removed
3	Verify removal	Insurance record no longer appears in card

13.6 RBAC Test

Role	Add	Edit	Delete
RECEPTIONIST	✔	✔	✔
DOCTOR	✘ (button hidden)	✘	✘
NURSE	✘ (button hidden)	✘	✘
ADMIN	✔	✔	✔

FEATURE 12 — NEW

14. Family Relationships

Allows RECEPTIONIST and ADMIN roles to link patients as family members with typed bidirectional relationships. Relationship types are colour-coded. Clicking a related patient's name navigates to their profile.

14.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN**. At least 2 patients must be registered.

14.2 Add Family Relationship — Happy Path

#	Step	Action / Expected Result
1	Open patient detail	Navigate to Patient A's detail page
2	Scroll to Family Relationships card	Card shows existing relationships (or empty state)
3	Click Add Relationship	Button in card header → modal opens
4	Search for related patient	Type Patient B's name or ID in the search field
5	Select patient	Click Patient B from the search results
6	Select relationship type	Choose from: SPOUSE, PARENT, CHILD, SIBLING, GUARDIAN, WARD
7	Submit	Click **Add** → `POST /api/v1/patients/{id}/relationships`
8	Verify success	Patient B appears in Patient A's Family Relationships card
9	Verify inverse	Open Patient B's detail → Patient A appears with inverse relationship type

14.3 Relationship Types & Colours

Type	Colour	Inverse
SPOUSE	Red	SPOUSE
PARENT	Blue	CHILD
CHILD	Cyan	PARENT
SIBLING	Green	SIBLING
GUARDIAN	Purple	WARD
WARD	Gold	GUARDIAN

14.4 Remove Relationship

#	Step	Expected Result
1	Click Remove on a relationship	`DELETE /api/v1/patients/{id}/relationships/{relatedId}`
2	Verify removal	Both directions removed — relationship gone from both patients' profiles

14.5 Navigate to Related Patient

#	Step	Expected Result
1	Click related patient name	Navigates to that patient's detail page
2	Verify context switch	All sections (vitals, insurance, etc.) load for the related patient

14.6 RBAC Test

- DOCTOR / NURSE: **Add Relationship** button is not visible; `POST /api/v1/patients/{id}/relationships` → `403`
- RECEPTIONIST / ADMIN: Full add and remove access

FEATURE 13 — NEW

15. Patient Photo Upload

RECEPTIONIST and ADMIN roles can upload, view, and delete a patient profile photo. Photos are stored securely and served via authenticated API (JWT required). The photo appears in the Personal Information card on the patient detail page. If no photo exists, an avatar with the patient's initials is shown.

15.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN**.

15.2 Upload Photo — Happy Path

#	Step	Action / Expected Result
1	Open patient detail	Navigate to any patient's detail page
2	Locate photo section	Top-left of Personal Information card — shows initials avatar if no photo
3	Click Upload Photo	Camera icon button → file picker opens
4	Select a JPEG or PNG	Choose a valid image file ≤2MB
5	Verify upload	`POST /api/v1/patients/{id}/photo` → photo replaces avatar
6	Verify display	Patient photo shown in card; blob URL used for secure display

15.3 Validation

#	Test Case	Expected Result
1	File type validation	Select a `.gif` or `.pdf` → error: `'Only JPEG and PNG files are allowed'`
2	File size validation	Select a file > 2MB → error: `'File must be smaller than 2MB'`
3	Cancel selection	Close file picker without selecting → no change

15.4 Delete Photo

#	Step	Expected Result
1	Click Delete Photo	Trash icon button on the photo → confirmation prompt
2	Confirm deletion	`DELETE /api/v1/patients/{id}/photo` → photo removed
3	Verify fallback	Initials avatar shown again

15.5 Security — Authenticated Photo Serving

Security Note: Photos are NOT served as public static files. Every photo request goes through `GET /api/v1/patients/{id}/photo` with the JWT token in the Authorization header. The UI fetches the photo as a blob via axios, creates a temporary blob URL (`URL.createObjectURL`), and revokes it on component unmount to prevent memory leaks.

15.6 RBAC Test

Role	Upload	View	Delete

RECEPTIONIST	✔	✔	✔
DOCTOR	✘	✔	✘
NURSE	✘	✔	✘
ADMIN	✔	✔	✔

FEATURE 14 — NEW

16. Patient ID Card (Print)

All authenticated roles can generate and print a physical patient ID card directly from the patient detail page. The card includes patient name, ID, date of birth, blood group, emergency contact, and a QR code encoding the Patient ID. Print is handled via browser `@media print` CSS — only the card is printed, not the full page UI.

16.1 Test Steps

#	Step	Action / Expected Result
1	Open patient detail	Navigate to any patient's detail page
2	Click Print ID Card	Button in page header (visible to all roles) → modal opens
3	View card preview	Card displays in modal with all patient information
4	Verify QR code	QR code is rendered; scanning should return the Patient ID (e.g. `P2026001`)
5	Click Print	Browser print dialog opens with only the ID card visible
6	Cancel print	Close print dialog → modal remains open
7	Close modal	Click ✕ → modal closes

16.2 ID Card Contents

Field	Source
Hospital logo / name	Static (Ai Nexus)
Patient full name	From patient record
Patient ID	Business ID (e.g. P2026001)
Date of birth	Formatted date
Blood group	From patient record (or `'—'` if not set)
Emergency contact	Name and phone
QR code	Encodes Patient ID only (no PHI)

16.3 Print Behaviour

- `@media print` CSS hides all page chrome (navbar, header, sidebar, modal overlay)
- Only the `.patient-id-card` div is printed
- No API call made — uses data already loaded on the page
- Works in all modern browsers (Chrome, Safari, Firefox)

FEATURE 15 — NEW

17. Duplicate Detection

When a patient detail page is opened, the system automatically checks for potential duplicate patient records that share the same phone number (by HMAC hash) or the same name and date of birth combination. If duplicates are found, a yellow warning alert is shown at the top of the page.

17.1 How It Works

The backend compares:

Phone number HMAC hash — exact match on normalized phone

Name + DOB — same first name, last name, and date of birth

The current patient is excluded from the results. The check happens automatically when the detail page loads.

17.2 Test Steps — Trigger Duplicate Alert

#	Step	Action / Expected Result
1	Register Patient A	Register `John Smith`, DOB `1980-01-15`, phone `+1-555-300-0001`
2	Register Patient B	Register `John Smith`, DOB `1980-01-15`, phone `+1-555-300-0002` (same name+DOB)
3	Open Patient A detail	Navigate to Patient A's detail page
4	Observe alert	Yellow warning alert at top: `'1 possible duplicate patient detected'`
5	Read description	Alert shows: `'Another patient record shares the same phone number or name and date of birth. Please review to avoid duplicate records.'`
6	Click View Duplicates	Button in alert → `DuplicatesModal` opens
7	View duplicate list	Modal lists Patient B with name, DOB, phone, and a link to their profile
8	Navigate to duplicate	Click on Patient B's name → navigates to Patient B's detail page
9	No duplicates case	Open a patient with unique name+DOB and phone → no alert shown

17.3 Alert States

State	Display
0 duplicates	No alert (component renders nothing)
1 duplicate	`'1 possible duplicate patient detected'`
2+ duplicates	`'N possible duplicate patients detected'`

17.4 View Duplicates Modal

Column	Content
Patient ID	Business ID (e.g. P2026001) — clickable link
Full Name	First + Last name

Date of Birth	Formatted date
Phone	Phone number
Status	Active / Inactive badge

FEATURE 16 — NEW

18. CSV Export

RECEPTIONIST and ADMIN roles can export the current filtered patient list as a CSV file. The export respects all active search and filter parameters — the exported data matches exactly what is shown on screen.

18.1 Prerequisites

> Login as **receptionist1 / RECEPTIONIST** or **admin / ADMIN**.

18.2 Export All Active Patients

#	Step	Action / Expected Result
1	Navigate to Patient List	http://localhost/patients
2	Ensure default filters	Status = Active, no search
3	Click Export CSV	Button with download icon in page header
4	Verify download	Browser downloads `patients_export.csv` automatically
5	Open CSV	Verify it contains all active patients matching the current view

18.3 Export with Filters Applied

#	Filter Applied	Expected CSV Content
1	Search = `'Connor'`	Only patients with `'Connor'` in name/ID/phone/email
2	Status = Inactive	Only inactive patients
3	Gender = Female	Only female patients
4	City (advanced) = `'Chicago'`	Only Chicago patients
5	Age 30–50	Only patients aged 30–50
6	Has Allergies ON	Only patients with allergies recorded
7	Combined filters	CSV respects all active filters simultaneously

18.4 CSV Contents

The exported CSV includes:

Column	Notes
Patient ID	e.g. `P2026001`
MRN	Medical Record Number
First Name	
Last Name	
Date of Birth	ISO format (YYYY-MM-DD)
Age	

Gender	
Blood Group	
Phone Number	
Email	
Address	
City	
State	
ZIP	
Status	ACTIVE or INACTIVE
Registered By	
Registered At	ISO 8601 UTC

18.5 Error Handling

#	Scenario	Expected Result
1	Export as DOCTOR	Export CSV button **not visible**
2	Direct API as DOCTOR	`GET /api/v1/patients/export/csv` → `403 Forbidden`
3	Network error during export	Error notification: `'Export failed'`

19. Non-Functional Requirements Coverage

NFR Category	Requirement	Implementation	Status
Performance	Search results < 2s	Indexed search columns + debounce	✔
Performance	Registration < 3s	Async API + optimistic UI	✔
Performance	Vitals recording < 2s	Lightweight payload, REQUIRES_NEW TX	✔
Security	AES-256 encryption at rest for PHI	`AttributeConverter` on all PHI fields	✔
Security	JWT-based stateless auth	HMAC-SHA256 JWT, 8hr expiry	✔
Security	RBAC enforced at API level	Spring Security URL-level rules	✔
Security	Audit logs immutable	DB trigger blocks `UPDATE` / `DELETE`	✔
Security	Audit log 6-year retention	Archive table + Flyway migration	✔
Security	Photo access requires JWT	Blob served through authenticated API	✔
Security	No PHI in audit logs	Only `patient_id` , `username` , `action` logged	✔
Usability	Inline validation on blur	Zod + react-hook-form `mode: onBlur`	✔
Usability	300ms search debounce	`useDebounce` hook	✔
Usability	Success notifications auto-dismiss	Ant Design notification `duration=5`	✔
Usability	Confirmation for destructive actions	`ConfirmModal` component	✔
Usability	WCAG 2.1 AA — colour + text labels	`StatusBadge` : colour + text always	✔
Usability	Vitals trend chart	Recharts `LineChart` (≥2 readings)	✔
Data Integrity	Unique Patient ID	DB sequence + `PatientIdGeneratorService`	✔
Data Integrity	Future DOB blocked	`DatePicker disabledDate` + Zod refine	✔
Data Integrity	Records never permanently deleted	Soft delete via status only	✔
Data Integrity	Bidirectional relationships	Both directions stored; inverse auto-computed	✔
Data Integrity	BMI auto-calculated	`weightKg / (heightM²)` on backend	✔
Responsive	Mobile 320px to desktop 2560px	Ant Design grid system (Col/Row)	✔

20. Complete Test Execution Checklist

Work through this checklist in order. Check off each item as you complete it.

Login

- ☐ Login as `receptionist1 / RECEPTIONIST` → blue badge in header
- ☐ Login as `dr.smith / DOCTOR` → green badge
- ☐ Login as `nurse.jane / NURSE` → cyan badge
- ☐ Login as `admin / ADMIN` → red badge

Registration

- ☐ Register Sarah Connor (all fields filled) → success toast + redirect to detail
- ☐ Blur empty First Name field → inline error shown
- ☐ Enter invalid email → inline error shown
- ☐ Try future date of birth → date picker blocks it
- ☐ Register second patient with same phone → yellow duplicate warning toast (10s)
- ☐ Login as DOCTOR → Register button **not visible**; CSV Export button **not visible**

Patient List & Search

- ☐ Active patients shown by default
- ☐ Search by name → results filter live (300ms debounce)
- ☐ Search by Patient ID → exact match found
- ☐ Search by MRN → exact match found
- ☐ Search by phone number → results filter
- ☐ Apply Status = Inactive filter
- ☐ Apply Gender filter
- ☐ Apply Blood Group filter
- ☐ Expand Advanced Filters → City, State, Age From/To, Has Allergies, Has Chronic toggles visible
- ☐ Apply City advanced filter → results filter
- ☐ Apply Age Range advanced filter → results filter
- ☐ Toggle Has Allergies → only allergy patients shown
- ☐ Combine search + basic + advanced filter
- ☐ Search with no match → `'No patients found'`
- ☐ Click patient row → navigates to detail page

Profile View

- ☐ All sections visible: Personal Info (with photo), Contact, Emergency, Medical, Family, Insurance, Vitals, Record Info
- ☐ MRN tag shown in page header (if assigned)
- ☐ Active patient shows green status badge with `'Active'` text
- ☐ Inactive patient shows red status badge with `'Inactive'` text
- ☐ As DOCTOR/NURSE: no Edit, no Deactivate, no Audit Trail, no CSV, no Insurance buttons; Record Vitals visible
- ☐ As RECEPTIONIST: Edit visible, no Deactivate, no Audit Trail; Insurance + Photo + Relationships visible; no Record Vitals
- ☐ As ADMIN: all buttons visible
- ☐ Back to List button returns to `/patients`

Edit Patient

- [ ] Open edit form → all fields pre-populated
- [ ] Patient ID not editable (shown in header only)
- [ ] Change address and submit → success toast
- [ ] Detail page shows updated data
- [ ] Record Info shows Last Updated By and timestamp
- [ ] Cancel discards changes

Status Management

- [ ] As ADMIN: Deactivate Patient → confirmation modal appears
- [ ] Cancel modal → patient stays ACTIVE
- [ ] Confirm deactivation → status changes to INACTIVE
- [ ] INACTIVE patient not in Active filtered list
- [ ] Activate patient → no confirmation, immediate
- [ ] Status changes back to ACTIVE
- [ ] As RECEPTIONIST: no Deactivate/Activate button visible

Vitals History

- [ ] As DOCTOR: Record Vitals button visible; record new entry → appears in table and latest readings
- [ ] As NURSE: Record Vitals button visible and functional
- [ ] Empty state: `'No vitals recorded'` shown
- [ ] After 1 entry: Latest readings row shows (no chart yet)
- [ ] After 2+ entries: Recharts trend chart appears
- [ ] History table paginates at 10 records
- [ ] As RECEPTIONIST: Record Vitals button **not visible**

Insurance

- [ ] As RECEPTIONIST: Add Insurance → modal opens, fill all fields, save → appears in card
- [ ] Primary badge shown on primary insurance record
- [ ] Edit insurance → modal pre-populated, update saves correctly
- [ ] Delete insurance → confirmation → record removed
- [ ] As DOCTOR: Add Insurance button **not visible**

Family Relationships

- [ ] Add SPOUSE relationship between two patients → appears in both patients' profiles with colour tag
- [ ] Add PARENT/CHILD relationship → verify inverse type (PARENT ↔ CHILD)
- [ ] Click related patient name → navigates to that patient
- [ ] Remove relationship → removed from both patients' profiles
- [ ] As NURSE: Add Relationship button **not visible**

Photo Upload

- [ ] No photo: initials avatar shown
- [ ] Upload JPEG ≤2MB → photo replaces avatar
- [ ] Upload .gif or >2MB file → error message
- [ ] Delete photo → avatar restored
- [ ] As DOCTOR: upload button **not visible**; photo still viewable

Patient ID Card

- [ ] All roles: Print ID Card button visible on detail page

- [ ] Modal shows card with name, ID, DOB, blood group, emergency contact, QR code
- [ ] Click Print → browser print dialog; only card is printed
- [ ] QR code: scan returns Patient ID (e.g. `P2026001`)

Duplicate Detection

- [ ] Register two patients with same name + DOB → yellow alert on detail page
- [ ] `'N possible duplicate patients detected'` message
- [ ] Click View Duplicates → modal lists duplicate patients
- [ ] Click duplicate patient link → navigates to their profile
- [ ] Patient with unique details → no alert shown

CSV Export

- [ ] As RECEPTIONIST: Export CSV → `patients_export.csv` downloaded
- [ ] Apply filters then Export CSV → CSV respects all filters
- [ ] As DOCTOR: Export CSV button **not visible**

HIPAA Audit Trail

- [ ] After registration: `PATIENT_CREATE` in audit_logs
- [ ] After viewing detail: `PATIENT_READ` in audit_logs
- [ ] After editing: `PATIENT_UPDATE` in audit_logs
- [ ] After deactivating: `PATIENT_DEACTIVATE` in audit_logs
- [ ] After recording vitals: `VITALS_CREATE` in audit_logs
- [ ] After adding insurance: `INSURANCE_CREATE` in audit_logs
- [ ] After uploading photo: `PHOTO_UPLOAD` in audit_logs
- [ ] As ADMIN: Audit Trail button visible → modal shows colour-coded action entries
- [ ] Attempt `UPDATE` on `audit_logs` → immutability error

Sign Out

- [ ] Click Sign Out → redirected to `/login`
- [ ] Visit `/patients` after sign out → 401 error shown

21. Known Limitations & Out-of-Scope Items

The following items are intentionally not implemented in the Patient Module — they are explicitly deferred to other modules:

Item	Owner Module	PRD Reference
Multi-Factor Authentication (MFA)	Auth Module	§7 HIPAA Technical Safeguards
15-minute session timeout	Auth Module	§7 HIPAA Technical Safeguards
Production HTTPS/TLS termination	Infrastructure/Nginx	§6 Security NFR
User management (create/assign roles)	Auth Module	§10 Dependencies
Appointment scheduling	Appointment Module	§9 Out of Scope
EMR / clinical notes	EMR Module	§9 Out of Scope
Billing & payment processing	Billing Module	§9 Out of Scope
Reporting and analytics dashboards	Reporting Module	§9 Out of Scope

Production Deployment Blocker: This module MUST NOT be deployed to production without a HIPAA-compliant Auth Module providing MFA and session timeout. The dev-login endpoint (`DevAuthController`) issues JWTs without MFA and must be removed or disabled before production deployment.

Document prepared by the Ai Nexus Engineering Team · Patient Module v1.2.0 · February 2026 · CONFIDENTIAL — Internal Use Only
Previous version: v1.1.0 (9 features) · This version: v1.2.0 (16 features — added Vitals, Insurance, Family Relationships, Photo Upload, ID Card, Duplicate Detection, CSV Export, Audit Trail UI Modal)