



HIMS

Hospital Information Management System

Comprehensive Technical Documentation for System Administrators
Enterprise-grade healthcare management platform with integrated clinical workflows,
billing, inventory, and interoperability features.

Version: 1.0.0

Platform: Laravel 12.0 + Vue 3

Date: January 2026

CONFIDENTIAL - FOR INTERNAL USE ONLY

Table of Contents

1. Executive Summary

2. System Architecture

2.1 High-Level Architecture

2.2 Technology Stack

2.3 Multi-Tenant Architecture

2.4 Security Architecture

3. System Requirements

3.1 Hardware Requirements

3.2 Software Requirements

3.3 Network Requirements

4. Module Documentation

4.1 Patient Management

4.2 OPD Module

4.3 IPD Module

4.4 Appointments

4.5 Laboratory

4.6 Radiology

4.7 Operation Theater

4.8 Pharmacy

4.9 Inventory

4.10 Billing

5. Database Schema

5.1 Entity Relationship Diagram

5.2 Core Tables

5.3 Module Tables

6. API Architecture

6.1 Authentication

6.2 Endpoint Summary

6.3 Response Formats

7. Integration & Interoperability

7.1 ABHA/ABDM Integration

7.2 FHIR R4 API

7.3 HL7 Messaging

8. Security & Compliance

9. Deployment Guide

10. Maintenance & Operations

11. Troubleshooting

Appendices

1. Executive Summary

HIMS (Hospital Information Management System) is an enterprise-grade, cloud-ready hospital management platform designed to streamline clinical and administrative workflows across healthcare facilities. Built on modern web technologies, it provides a comprehensive solution for managing patient care, clinical operations, billing, inventory, and regulatory compliance.

1.1 Key Features

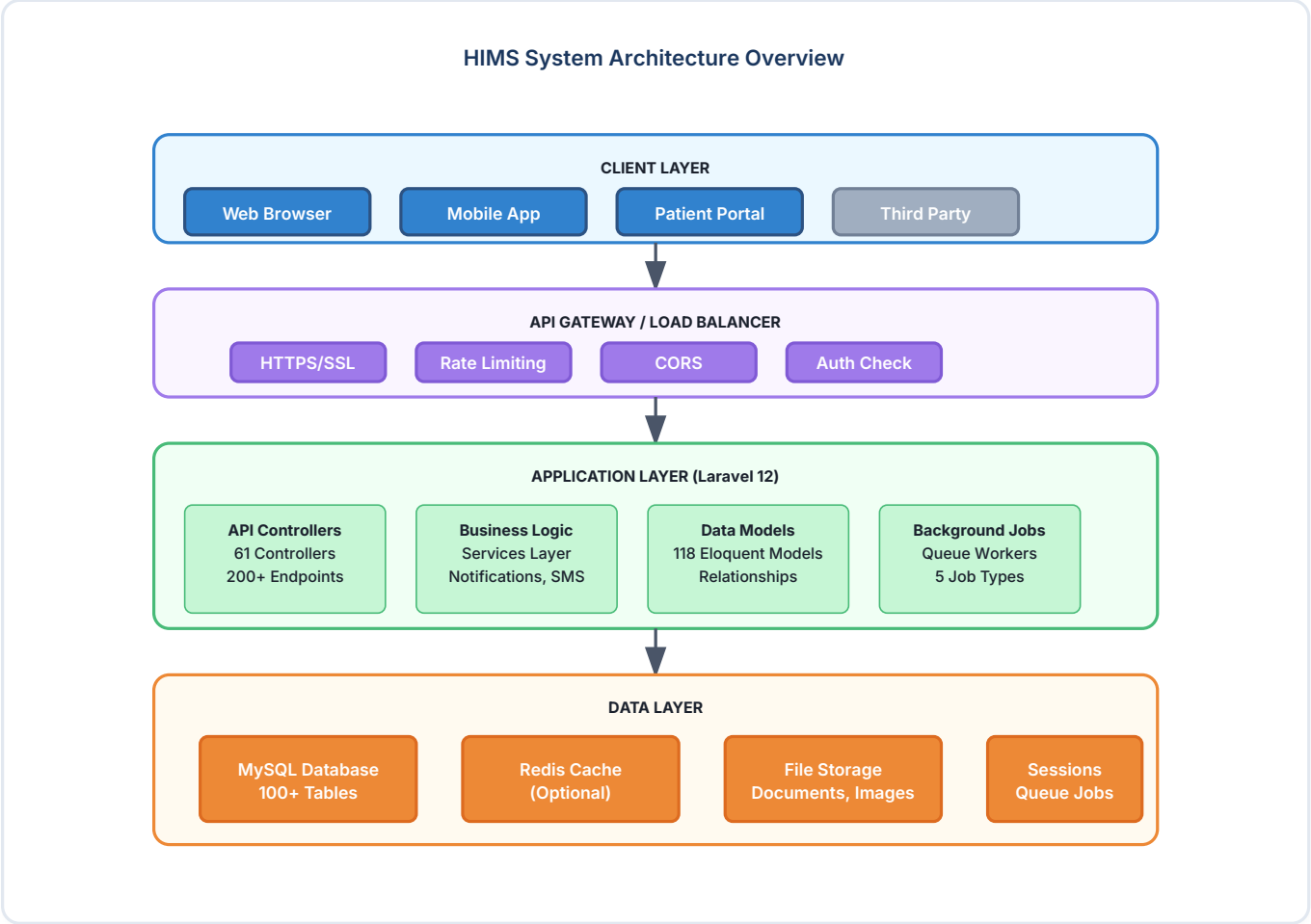
Multi-Tenant Support for multiple hospitals with complete data isolation
Clinical Workflows OPD, IPD, Lab, Radiology, OT, Pharmacy modules
Billing & Revenue Comprehensive billing with insurance support
RBAC Security Role-based access control with granular permissions
Interoperability FHIR R4, HL7, ABHA/ABDM integration
Patient Portal Self-service appointments and record access

1.2 Platform Statistics

API Controllers	61	RESTful API endpoints for all operations
Data Models	118	Eloquent models representing business entities
API Endpoints	200+	Complete CRUD and workflow operations
Database Tables	100+	Normalized relational schema
Vue Components	54+	Frontend views and components
Functional Modules	20	Integrated hospital management modules

2. System Architecture

2.1 High-Level Architecture



2.2 Technology Stack

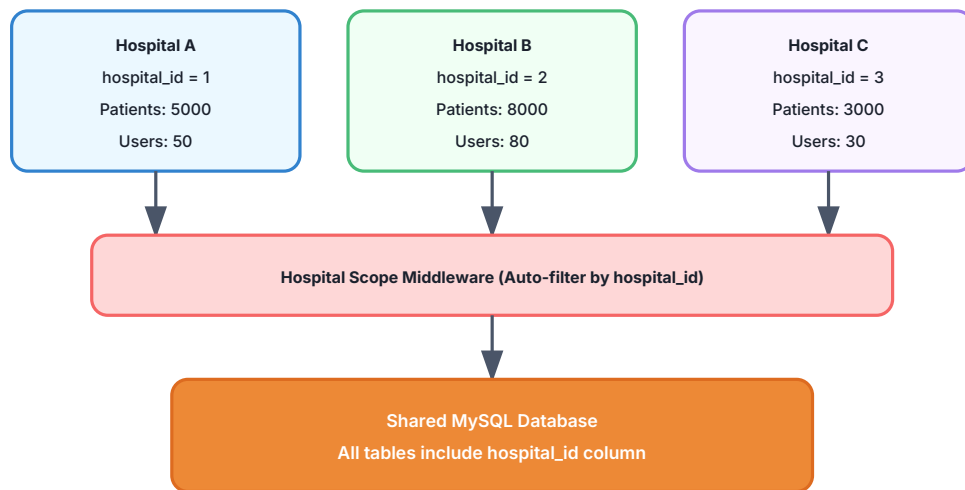
Backend	PHP	8.2+	Server-side programming language
	Laravel	12.0	Web application framework
	Laravel Sanctum	4.0	API token authentication
	Eloquent ORM	-	Database abstraction layer
Frontend	Vue.js	3.5.26	Reactive UI framework
	Pinia	3.0.4	State management
	Tailwind CSS	4.0	Utility-first CSS framework
	Vite	7.0.7	Build tool and dev server
	MySQL	8.0+	Primary relational database

Database	Redis	7.0+	Cache and queue (optional)
Server	Nginx	1.18+	Web server / reverse proxy
	PHP-FPM	8.2	PHP process manager

2.3 Multi-Tenant Architecture

HIMS implements a single-database multi-tenant architecture where all hospitals share the same database with data isolation achieved through a `hospital_id` foreign key on all relevant tables.

Multi-Tenant Data Isolation Model

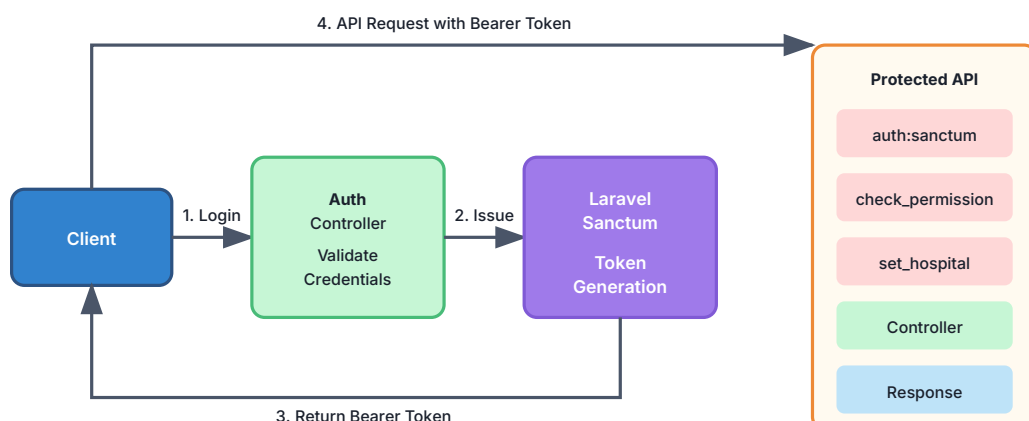


Implementation Details:

- `BelongsToHospital` trait automatically sets `hospital_id` on create
- `HospitalScope` global scope filters all queries by current hospital
- `SetCurrentHospital` middleware sets context from authenticated user
- Super Admins can access all hospitals and switch between them

2.4 Security Architecture

Authentication & Authorization Flow



3. System Requirements

3.1 Hardware Requirements

CPU Cores	2 cores	4 cores	8+ cores
RAM	4 GB	8 GB	16+ GB
Storage	50 GB SSD	200 GB SSD	500+ GB NVMe
Network	100 Mbps	1 Gbps	10 Gbps
Concurrent Users	Up to 50	Up to 200	500+

3.2 Software Requirements

Operating System

- Ubuntu 22.04 LTS (Recommended)
- Debian 12
- RHEL 9 / Rocky Linux 9
- CentOS Stream 9

Web Server

- Nginx 1.18+ (Recommended)
- Apache 2.4+ with mod_rewrite
- PHP-FPM 8.2

PHP Extensions

- php-bcmath, php-ctype, php-curl
- php-dom, php-fileinfo, php-json
- php-mbstring, php-openssl, php-pdo
- php-pdo_mysql, php-tokenizer
- php-xml, php-zip, php-gd

Database

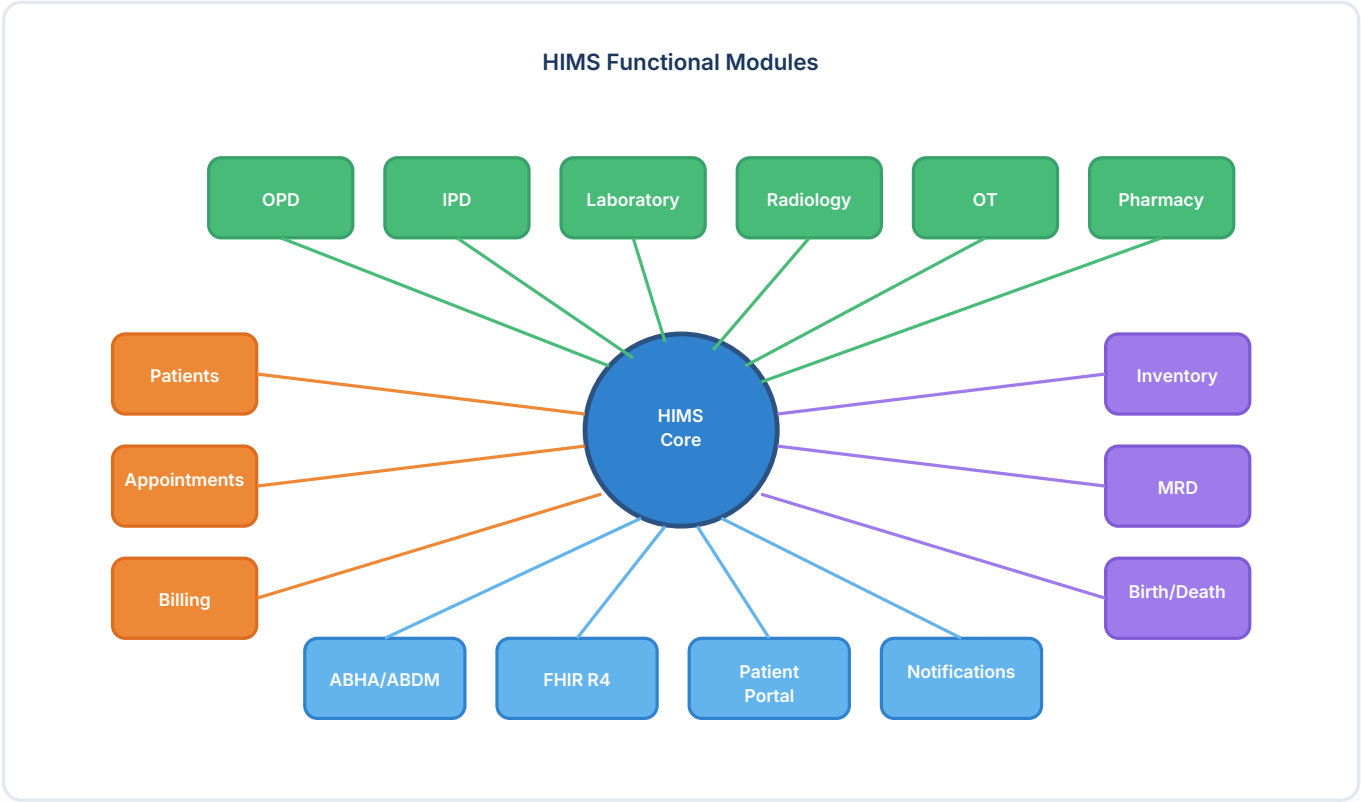
- MySQL 8.0+ (Recommended)
- MariaDB 10.6+
- Redis 7.0+ (Optional, for caching)

3.3 Network Requirements

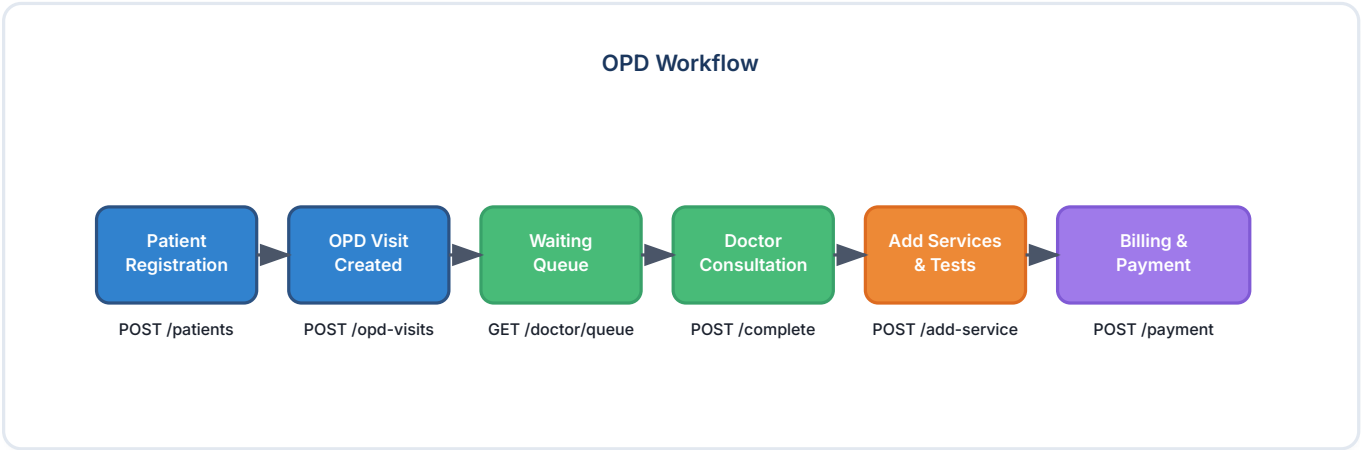
443	HTTPS	Web Application	Public
80	HTTP	Redirect to HTTPS	Public
3306	TCP	MySQL Database	Internal Only
6379	TCP	Redis (if used)	Internal Only
22	SSH	Server Administration	Restricted

4. Module Documentation

4.1 Module Overview



4.2 OPD Module (Outpatient Department)

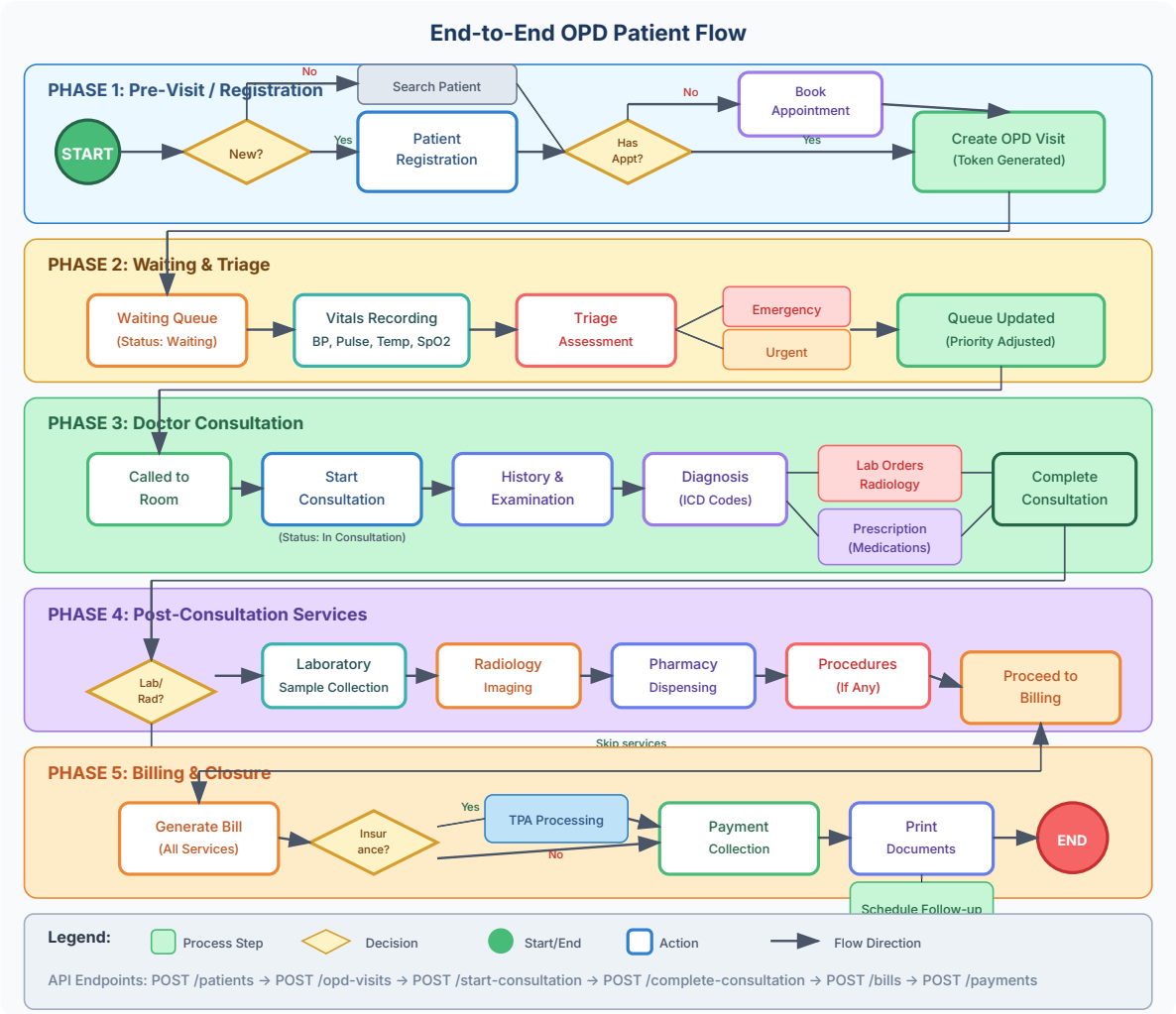


/api/opd-visits	GET	List OPD visits with filters
/api/opd-visits	POST	Create new OPD visit
/api/opd-visits/{id}/start-consultation	POST	Start doctor consultation

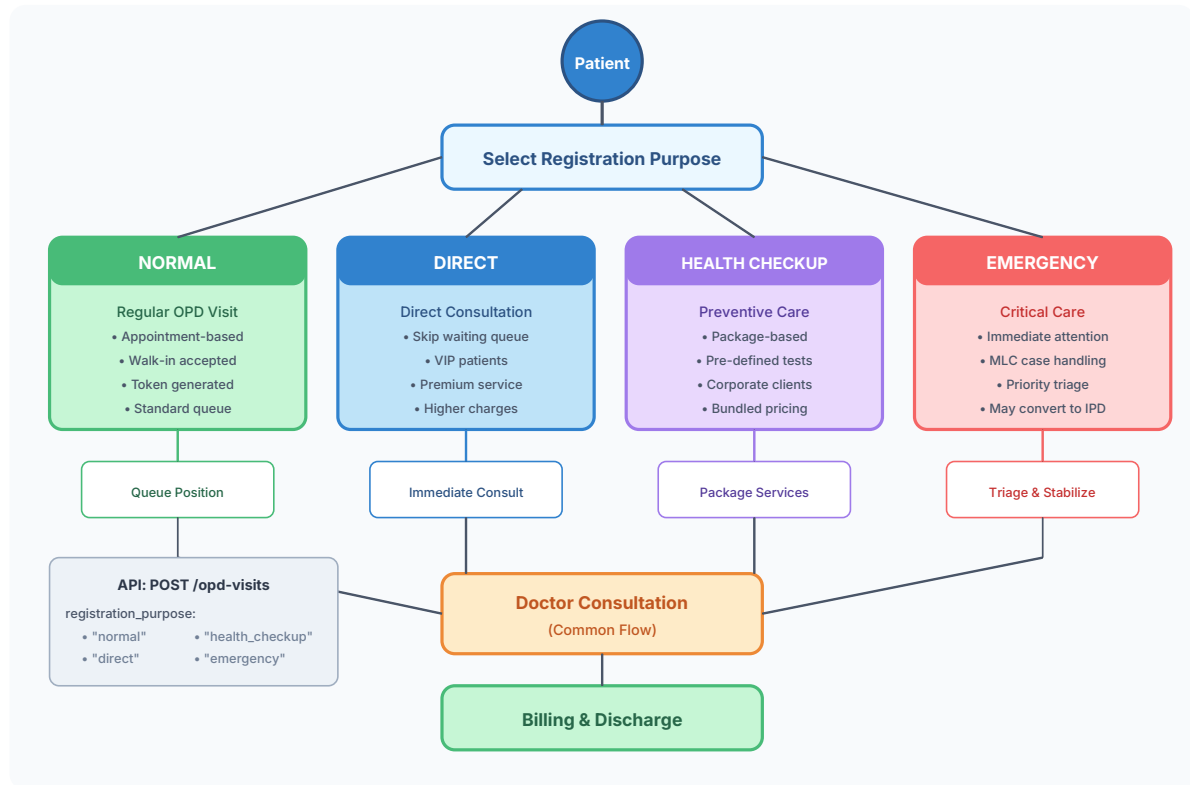
/api/opd-visits/{id}/complete-consultation	POST	Complete with diagnosis
/api/opd-visits/{id}/add-service	POST	Add billable service
/api/opd-visits/{id}/add-investigation	POST	Order lab/radiology tests
/api/opd-visits/{id}/payment	POST	Record payment
/api/opd-visits/{id}/case-paper	GET	Generate case paper
/api/opd-visits/doctor/{id}/queue	GET	Get doctor's today queue

4.2.1 Detailed OPD Patient Flow Diagrams

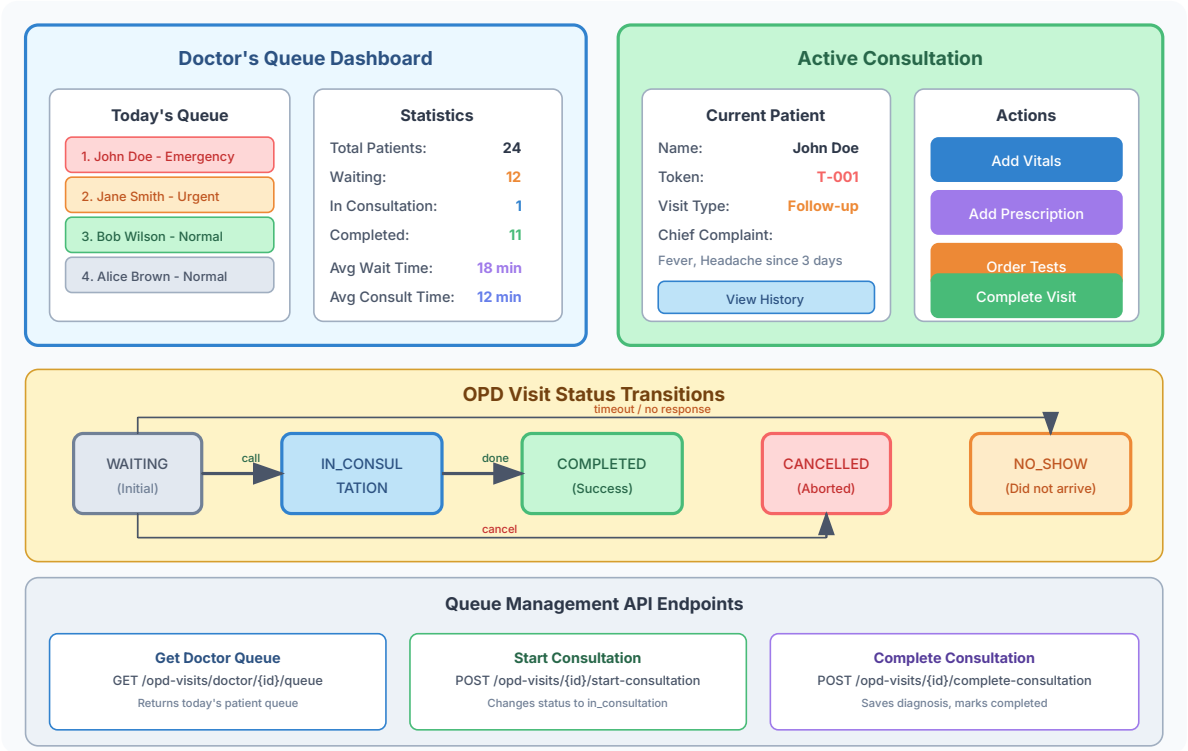
Complete OPD Patient Journey



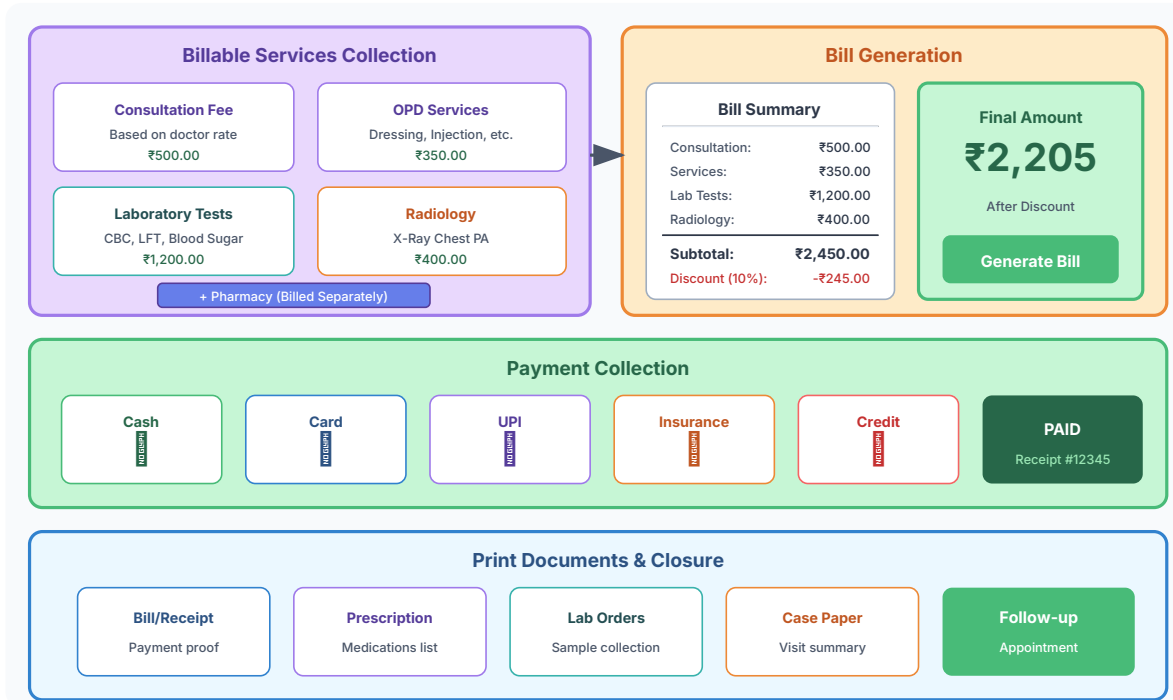
OPD Registration Types & Pathways



Doctor Queue & Consultation Management



OPD Billing & Payment Workflow

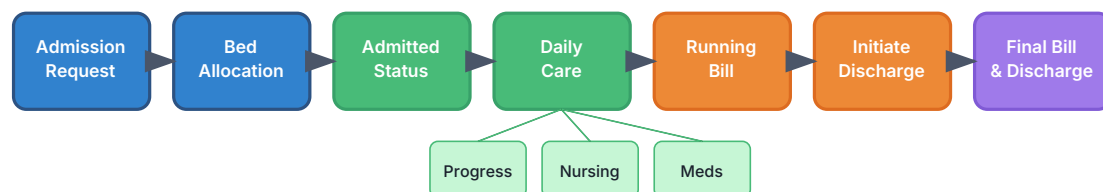


OPD Workflow Key Points

- **Token System:** Automatic token generation on OPD visit creation with configurable prefix per department
- **Queue Priority:** Emergency and urgent cases automatically prioritized in doctor queue
- **Free Follow-up:** System tracks follow-up validity based on skill-set configuration (typically 3-7 days)
- **Multi-Service:** Single visit can include consultation, procedures, lab tests, radiology, and pharmacy
- **Real-time Updates:** Queue status and patient flow visible on dashboards in real-time
- **Insurance Integration:** TPA pre-authorization and claim submission workflow supported

4.3 IPD Module (Inpatient Department)

IPD Admission Workflow

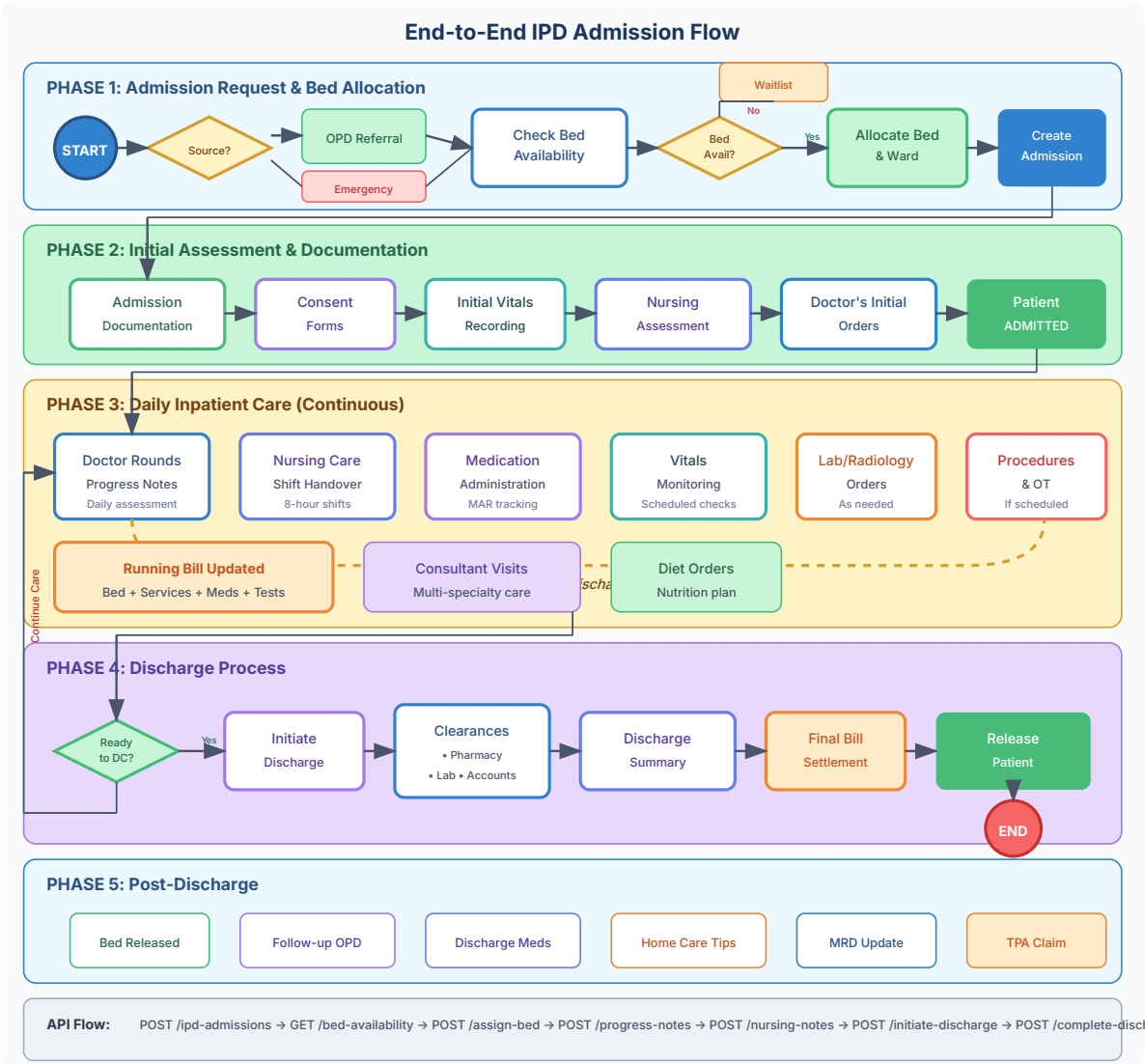


Key Features:

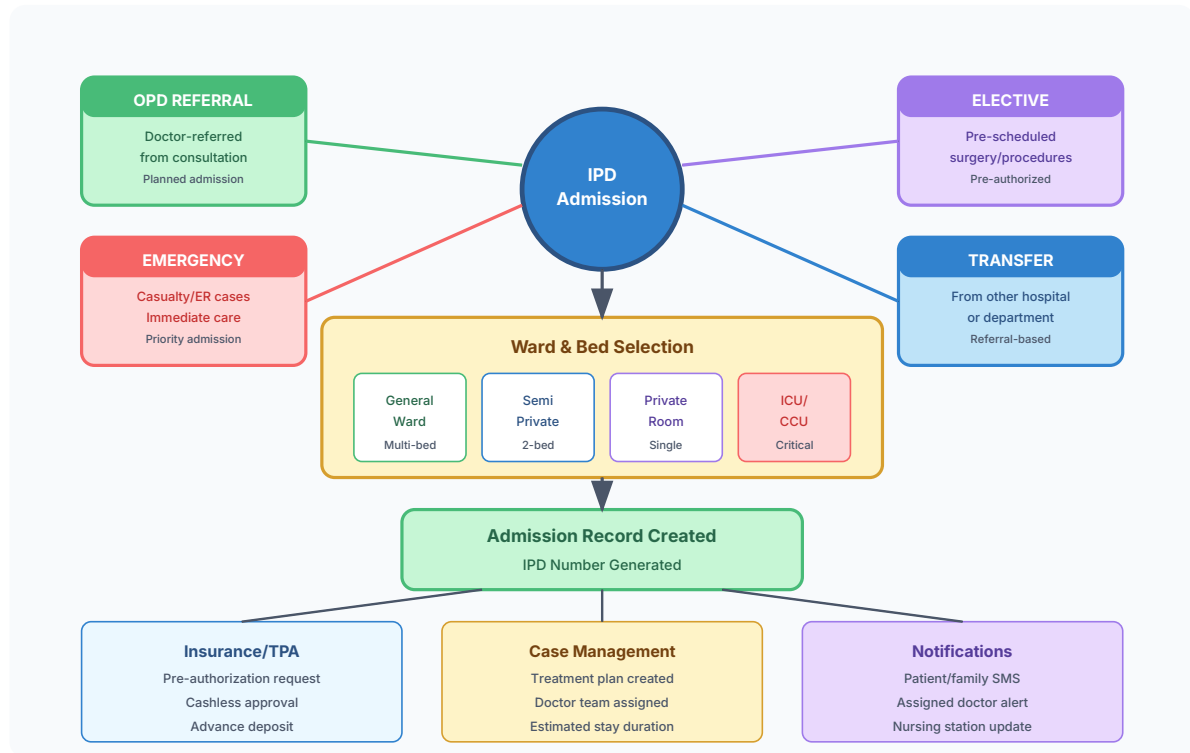
- Bed availability tracking across wards
- Progress notes by doctors
- Nursing charts with vital monitoring
- Medication administration tracking
- Running bill calculation
- Bed transfer management
- Discharge workflow with summary

4.3.1 Detailed IPD Admission Flow Diagrams

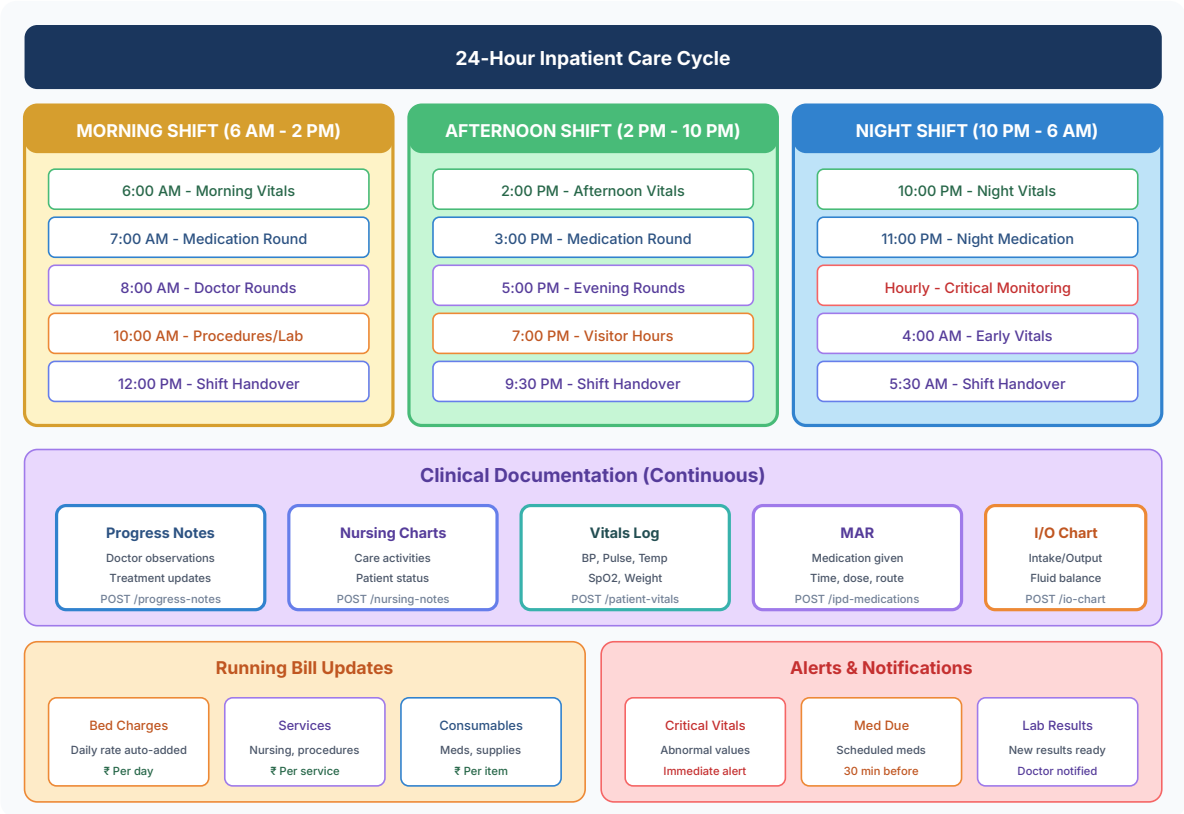
Complete IPD Patient Journey



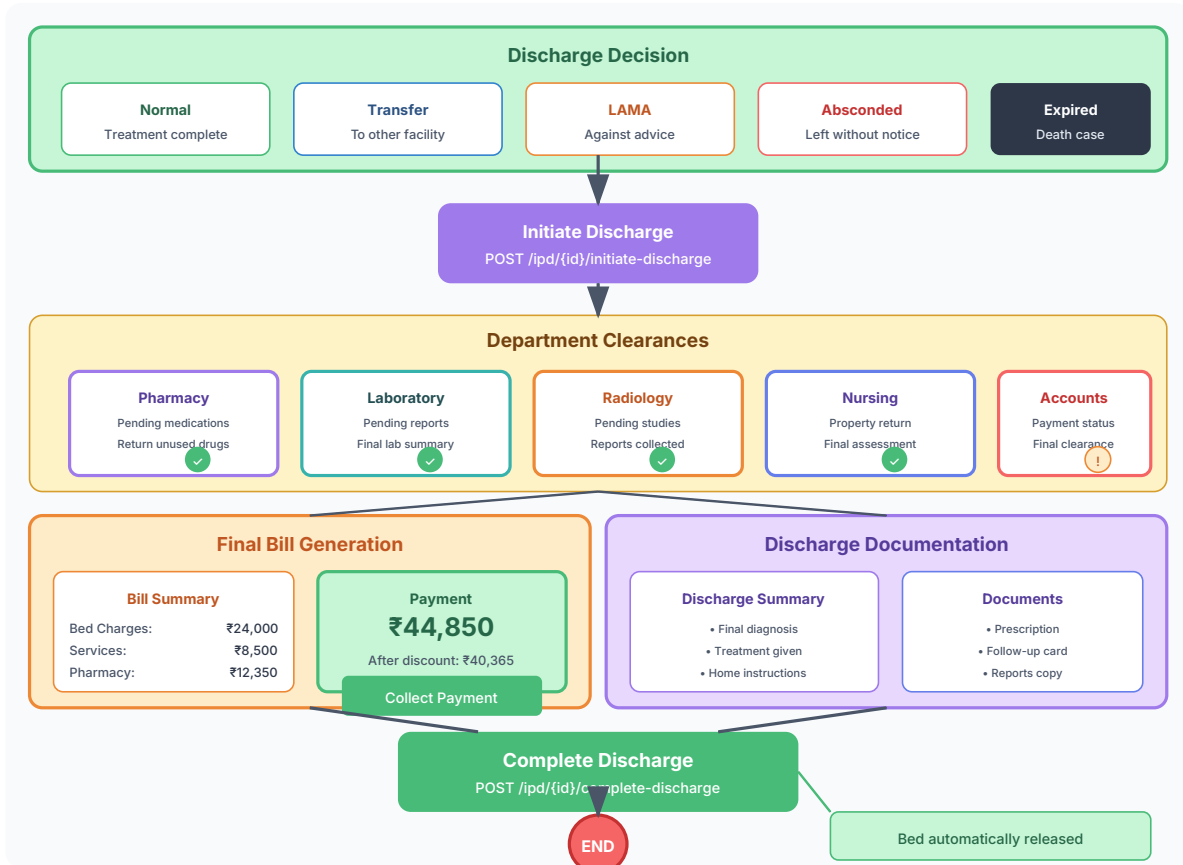
IPD Admission Types & Sources



Daily Inpatient Care & Clinical Management



IPD Discharge Workflow



IPD Workflow Key Points

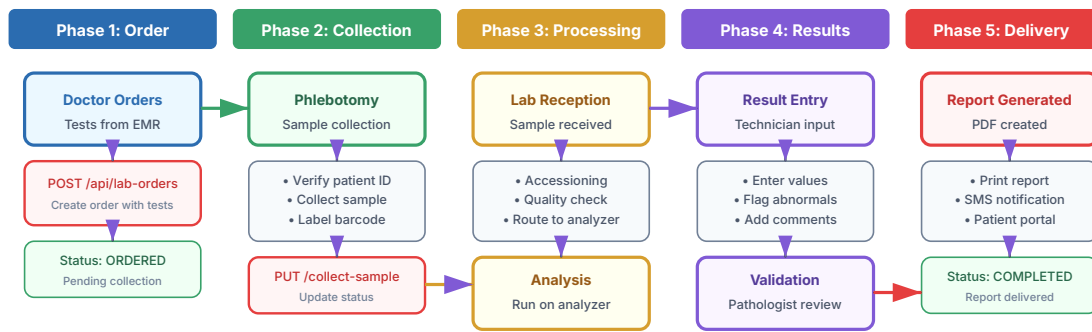
- **Admission Types:** OPD referral, Emergency, Elective (scheduled surgery), Transfer from other facility
- **Bed Management:** Real-time availability tracking, automatic bed release on discharge, transfer between wards
- **Running Bill:** Automatic daily bed charges, service-wise billing, pharmacy charges linked to dispensing
- **Clinical Documentation:** Progress notes, nursing charts, vitals monitoring, medication administration records (MAR)
- **Shift Handover:** 8-hour nursing shifts with mandatory handover documentation
- **Discharge Types:** Normal, Transfer, LAMA (Leave Against Medical Advice), Absconded, Expired
- **Clearance Workflow:** Multi-department clearance required before final discharge (Pharmacy, Lab, Radiology, Nursing, Accounts)
- **Insurance:** Pre-authorization on admission, interim bills for long stays, final TPA claim on discharge

4.4 Laboratory Module

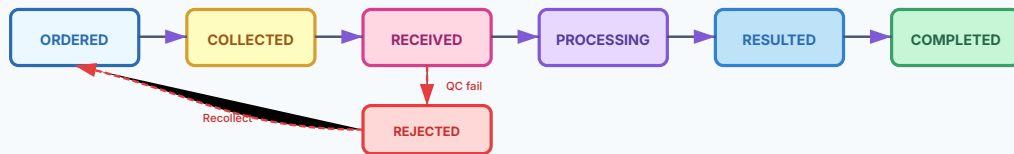
The Laboratory module manages the complete lifecycle of diagnostic tests from order to report delivery, supporting multiple test categories, sample types, and integration with clinical workflows.

4.4.1 Laboratory Workflow Diagrams

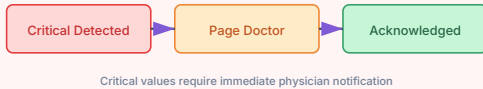
Complete Laboratory Test Journey



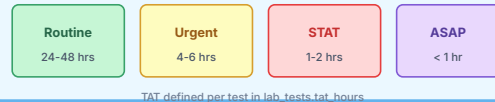
Order Status Transitions



Critical Value Alert Workflow



Turnaround Time (TAT) Monitoring



Key API Endpoints

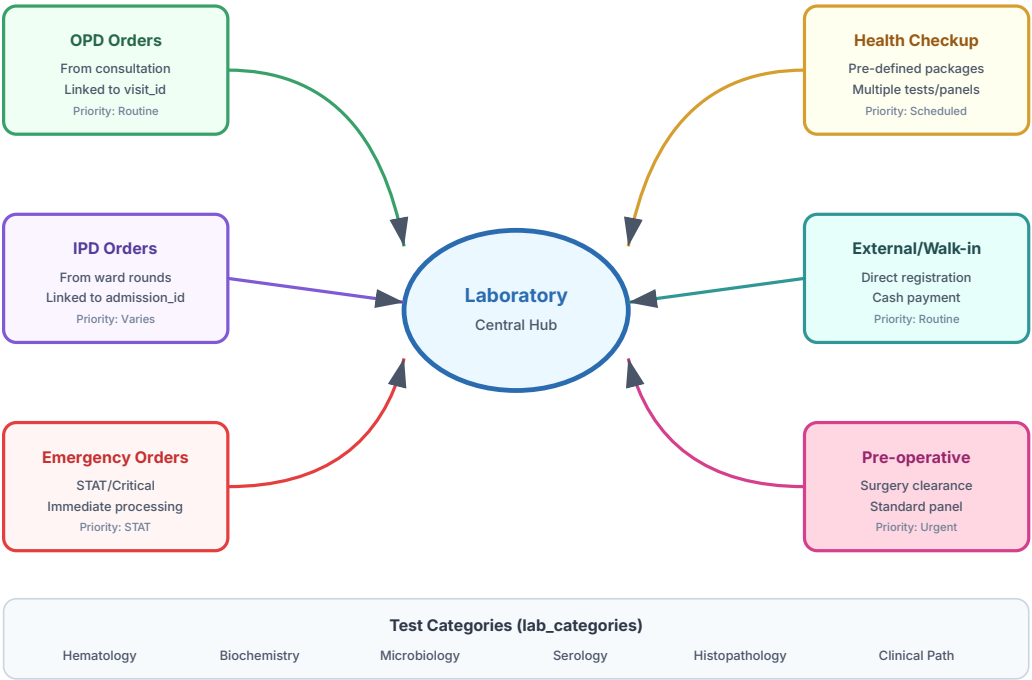
POST /api/lab-orders

PUT /api/lab-orders/{id}/collect

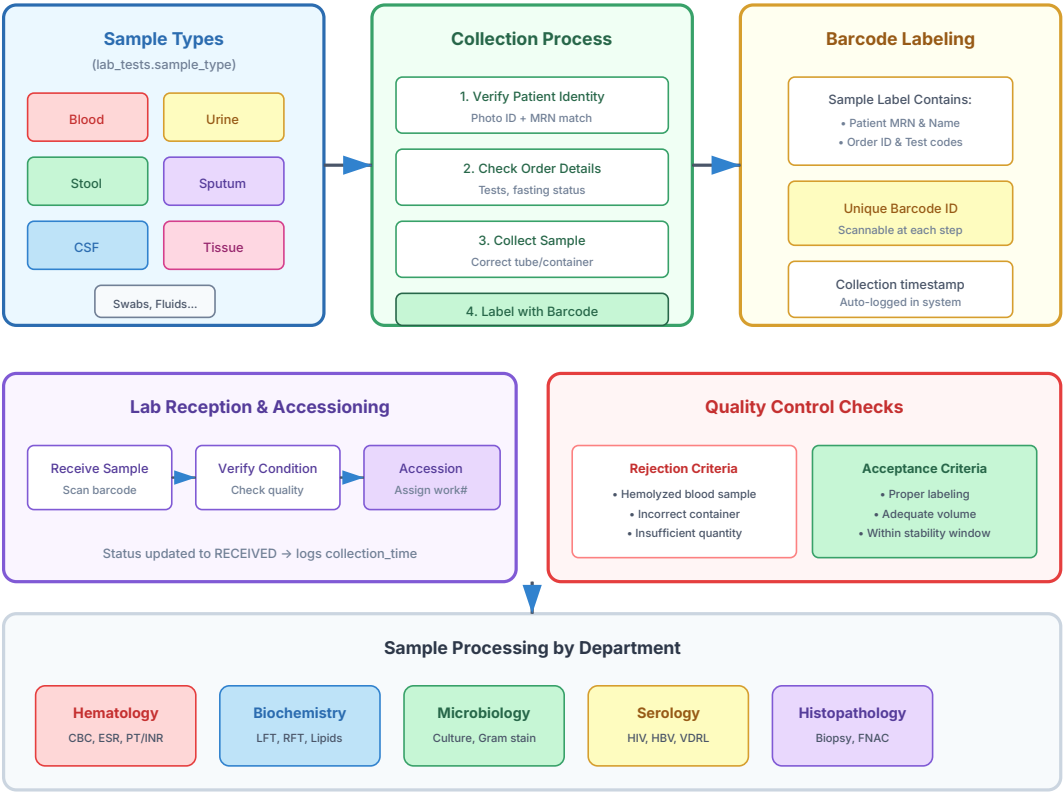
PUT /api/lab-orders/{id}/result

GET /api/lab-orders/{id}/report

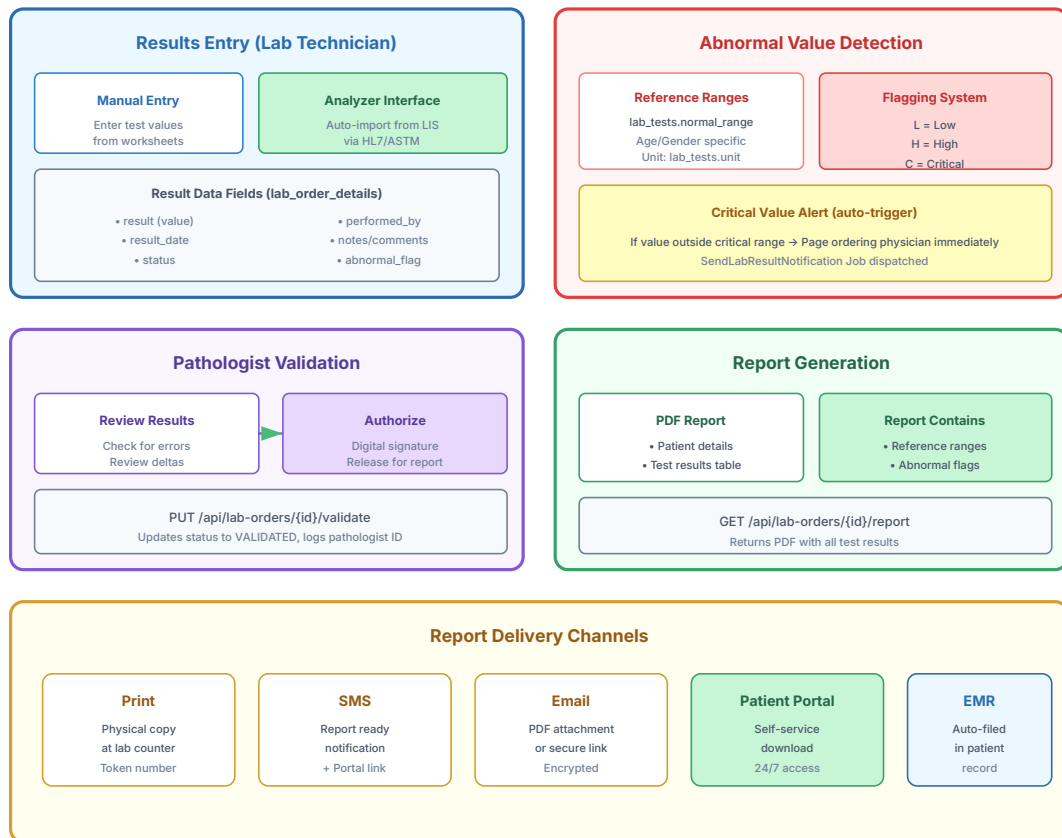
Laboratory Order Sources & Pathways



Sample Collection & Processing Workflow



Results Entry & Report Generation Workflow



Laboratory Workflow Key Points

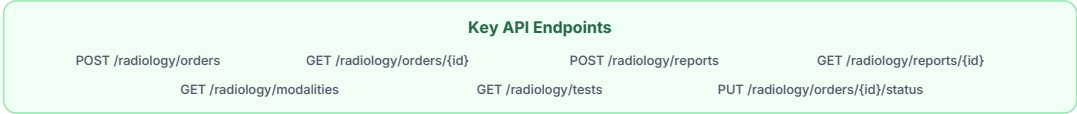
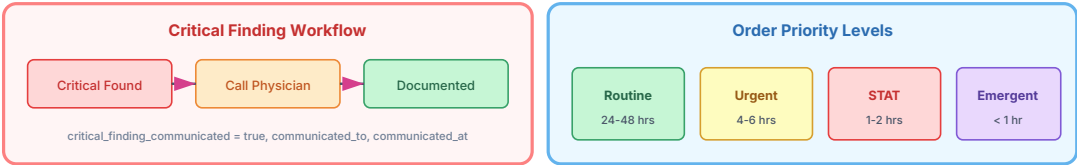
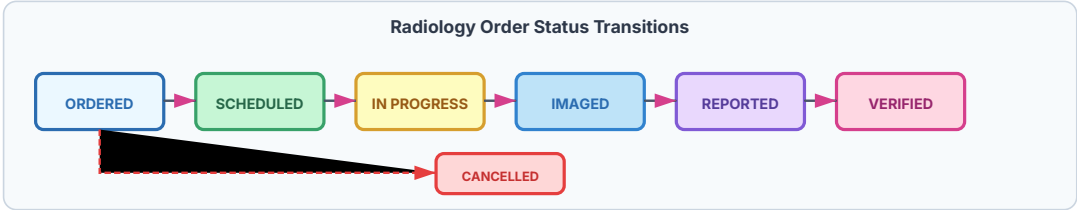
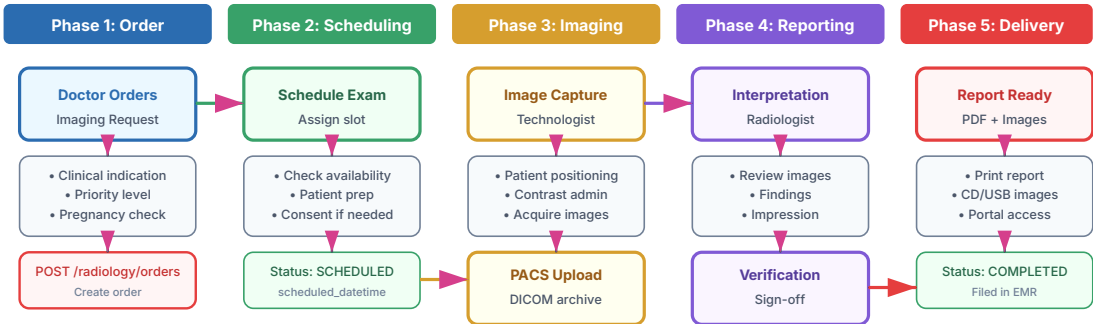
- **Order Sources:** OPD consultation, IPD ward rounds, Emergency STAT orders, Health checkup packages, External walk-ins
- **Sample Types:** Blood (serum, plasma, whole blood), Urine, Stool, Sputum, CSF, Tissue biopsies, Swabs
- **Turnaround Time (TAT):** Routine (24-48 hrs), Urgent (4-6 hrs), STAT (1-2 hrs), ASAP (<1 hr) - defined in lab_tests.tat_hours
- **Quality Control:** Sample rejection for hemolysis, insufficient quantity, improper labeling, expired stability
- **Barcode Tracking:** Unique barcode per sample, scanned at collection → reception → processing → result entry
- **Critical Values:** Auto-alert to ordering physician via SendLabResultNotification Job, requires acknowledgment
- **Validation:** Two-tier validation - Technician entry → Pathologist review and authorization
- **Report Delivery:** Print, SMS, Email, Patient portal, Auto-filed in EMR

4.5 Radiology Module

The Radiology module manages diagnostic imaging from order to report, supporting multiple modalities (X-Ray, CT, MRI, Ultrasound), PACS integration, and critical finding alerts.

4.5.1 Radiology Workflow Diagrams

Complete Radiology Imaging Journey



Radiology Modalities & Order Sources

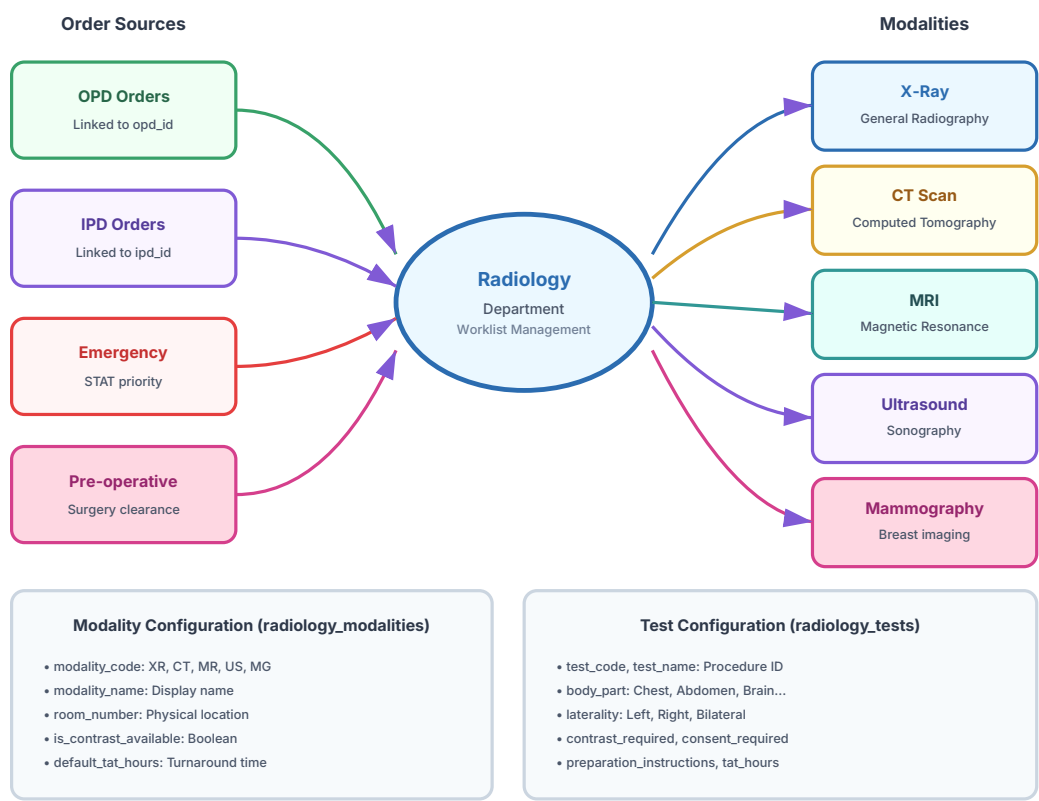
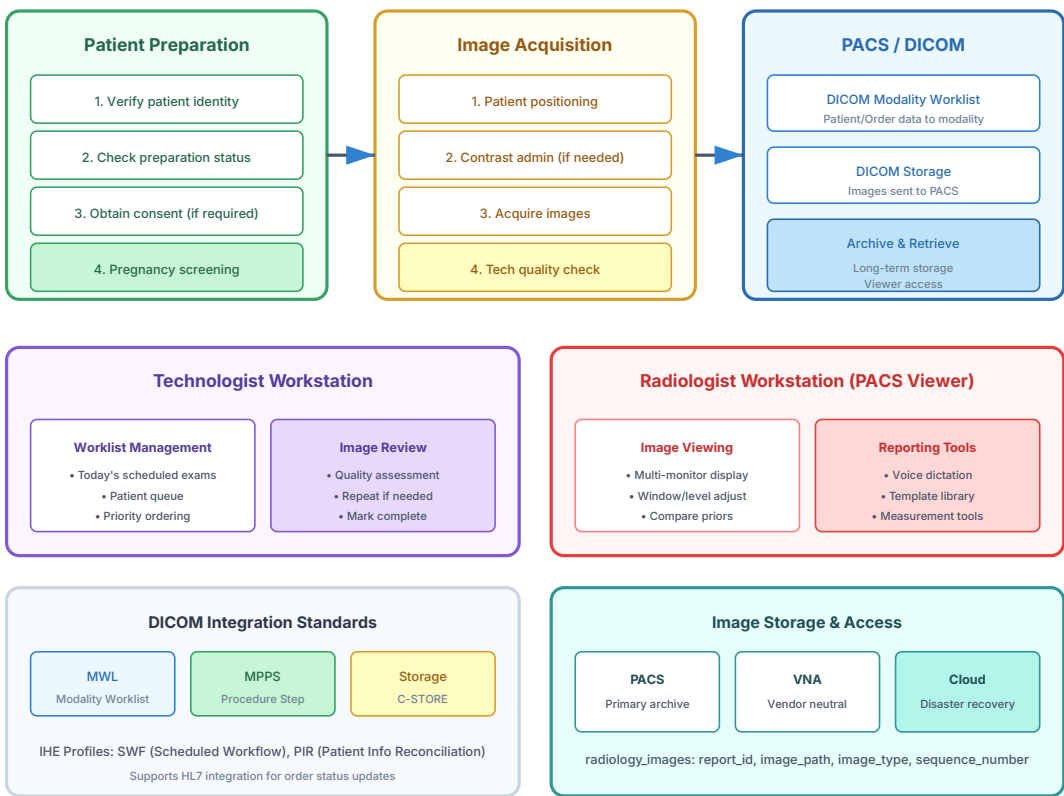
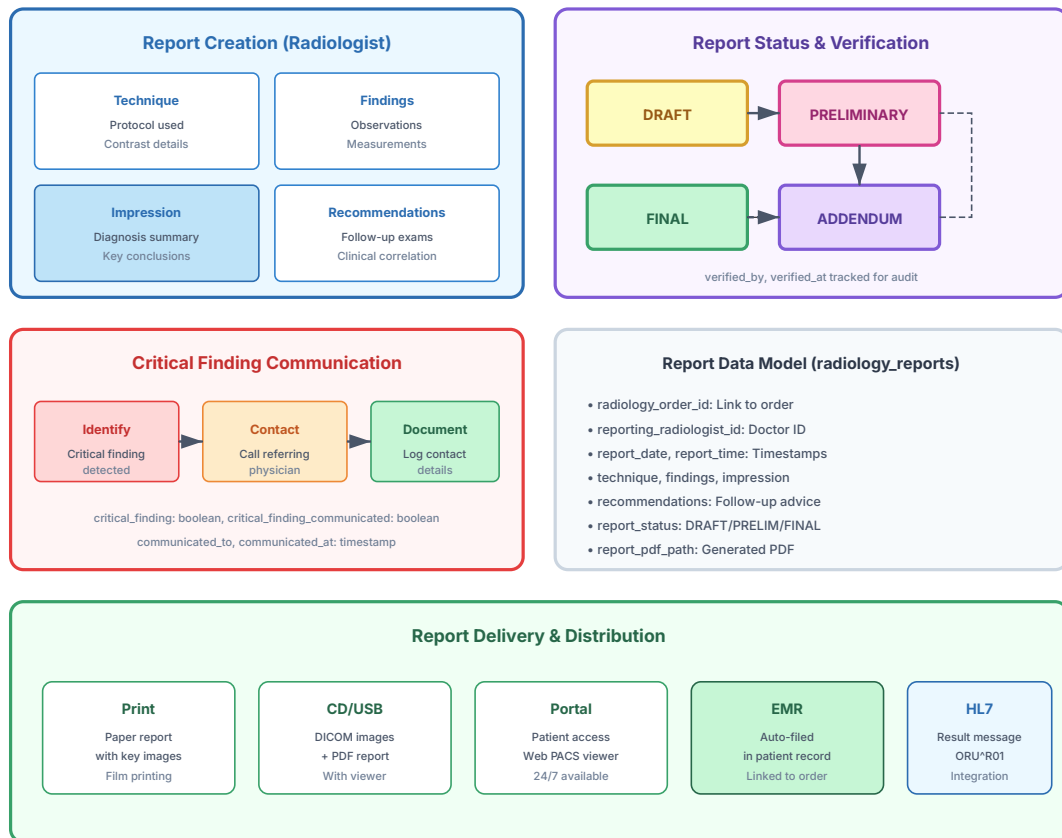


Image Acquisition & PACS Integration Workflow



Radiology Reporting & Verification Workflow



Radiology Workflow Key Points

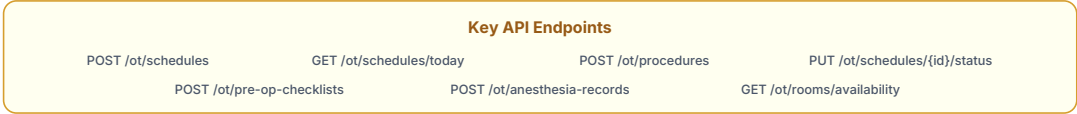
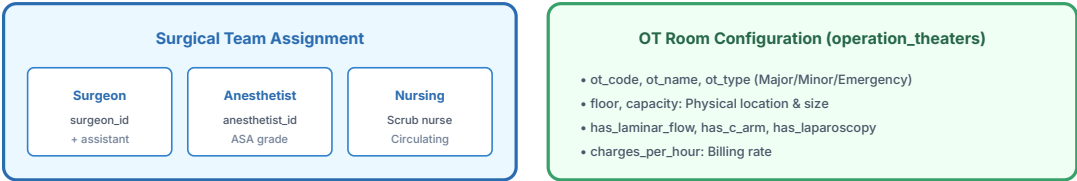
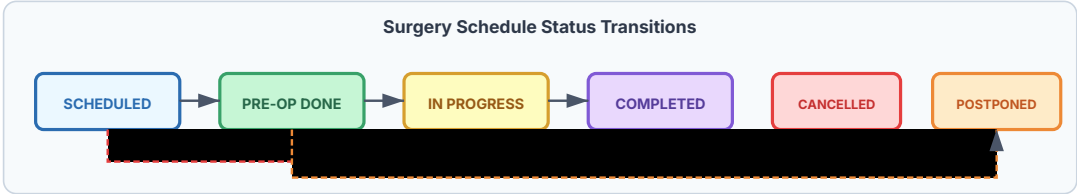
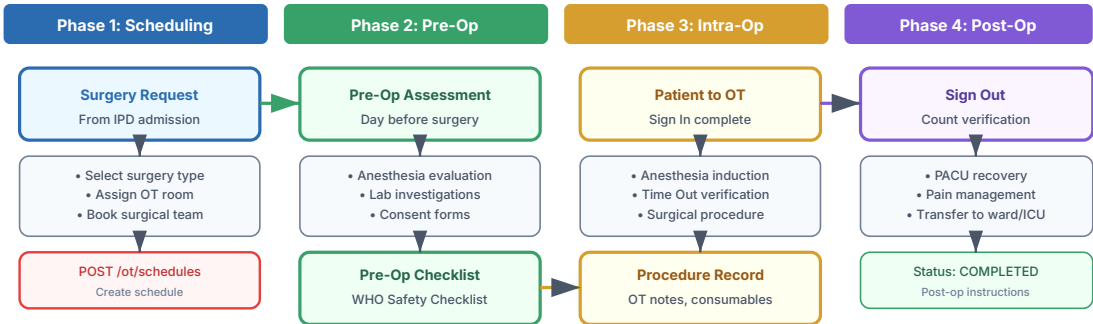
- **Modalities**: X-Ray, CT, MRI, Ultrasound, Mammography, Fluoroscopy - each with room assignment and TAT configuration
- **Order Sources**: OPD (opd_id), IPD (ipd_id), Emergency, Pre-operative clearance
- **Priority Levels**: Routine (24-48 hrs), Urgent (4-6 hrs), STAT (1-2 hrs), Emergent (<1 hr)
- **PACS Integration**: DICOM MWL for worklist, MPPS for procedure status, C-STORE for image archival
- **Contrast Handling**: `contrast_required` per test, `with_contrast` per order detail, `contrast_rate` billing
- **Critical Findings**: Immediate physician notification with mandatory documentation (`communicated_to, communicated_at`)
- **Report Status**: DRAFT → PRELIMINARY → FINAL, with ADDENDUM support for amendments
- **Verification**: Two-level sign-off - Reporting radiologist + Verifying radiologist (`verified_by, verified_at`)

4.6 Operation Theater Module

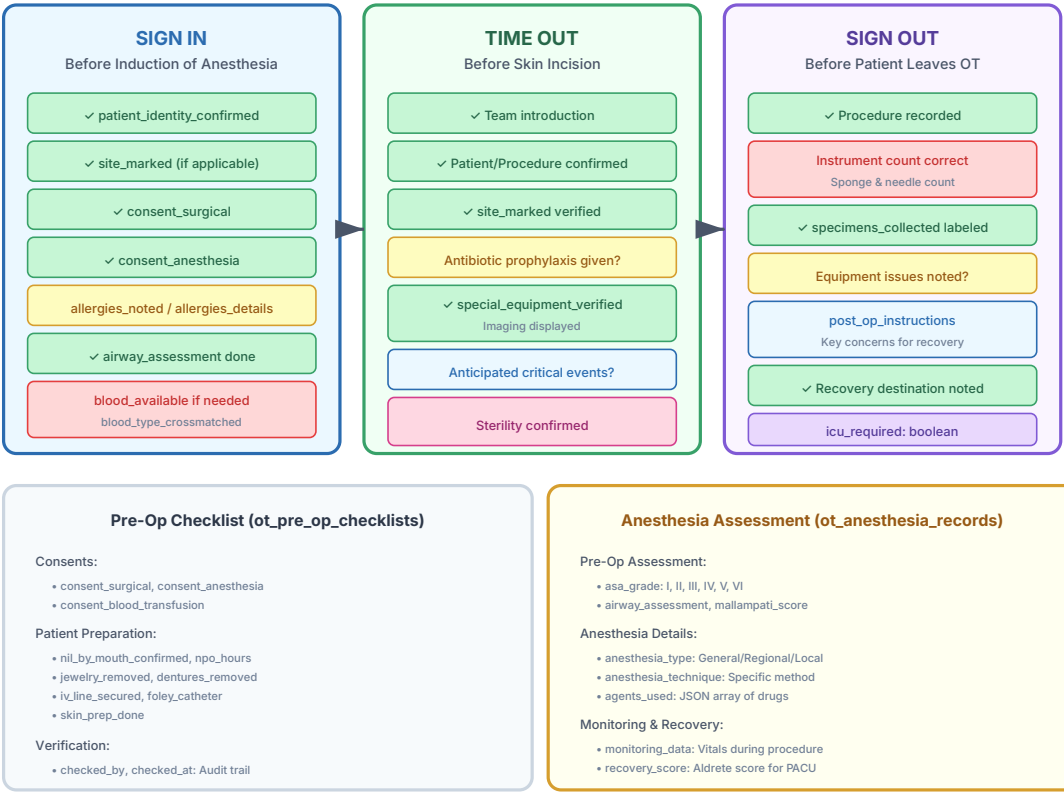
The Operation Theater module manages the complete surgical workflow from scheduling to post-operative care, including WHO surgical safety checklist compliance, anesthesia records, and consumables tracking.

4.6.1 Operation Theater Workflow Diagrams

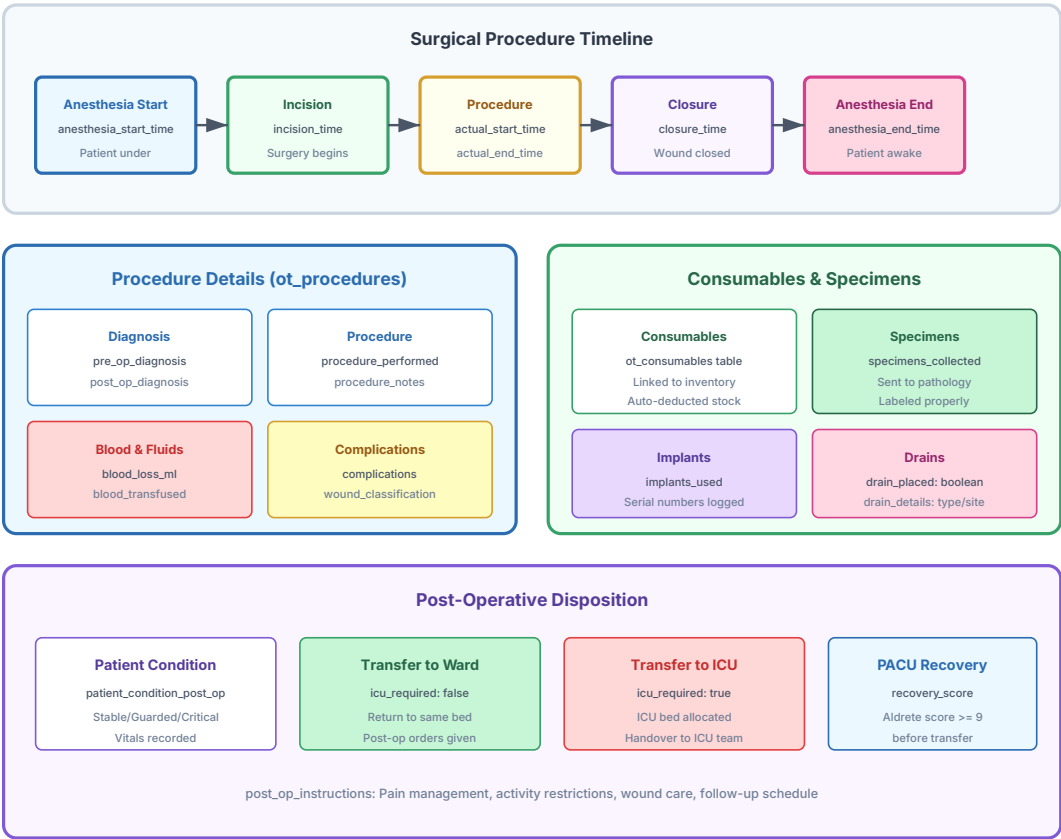
Complete Surgical Journey



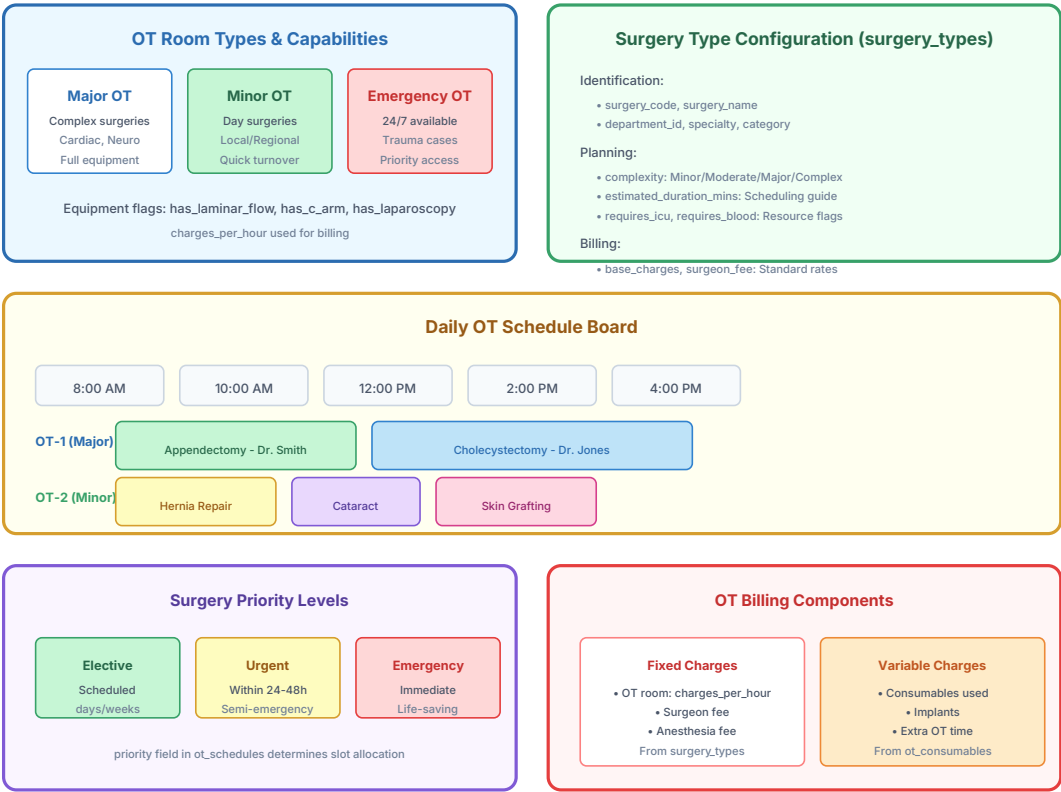
WHO Surgical Safety Checklist Workflow



Intra-Operative Documentation & Procedure Record



OT Resource Management & Scheduling



4.7 Inventory Module

Procurement Workflow



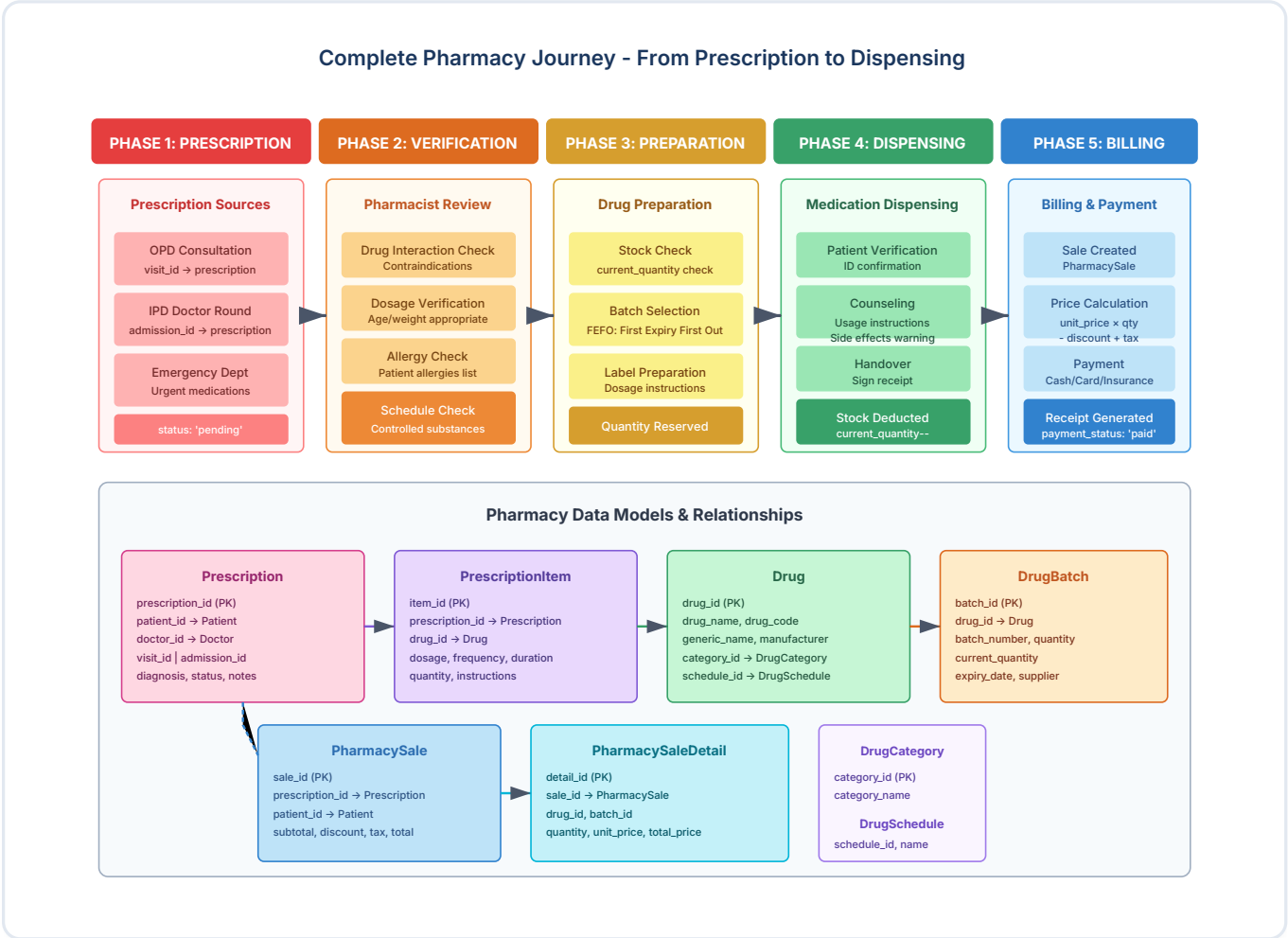
Key Features:

- Multiple store/warehouse management
- Item categorization with hierarchy
- Supplier/vendor management
- Indent workflow with approvals
- Purchase order generation
- Goods receipt note (GRN) processing
- Low stock alerts
- Expiry tracking
- Stock movement history

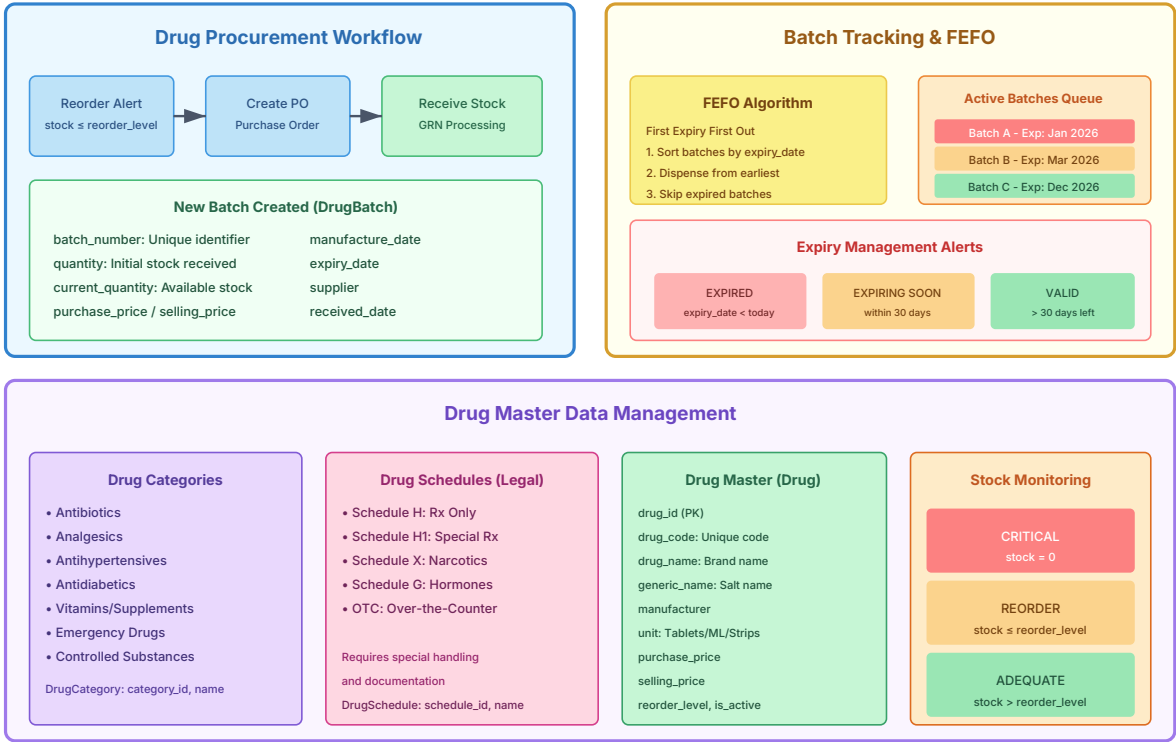
4.8 Pharmacy Module

The Pharmacy Module provides comprehensive drug inventory management, prescription processing, and medication dispensing workflows with batch tracking, expiry management, and controlled substance handling.

4.8.1 Pharmacy Workflow Diagrams



Drug Inventory Management - Procurement, Batch Tracking & Expiry Control



Drug Master Data Management

Drug Categories

- Antibiotics
- Analgesics
- Antihypertensives
- Antidiabetics
- Vitamins/Supplements
- Emergency Drugs
- Controlled Substances

DrugCategory: category_id, name

Drug Schedules (Legal)

- Schedule H: Rx Only
- Schedule H1: Special Rx
- Schedule X: Narcotics
- Schedule G: Hormones
- OTC: Over-the-Counter

Requires special handling and documentation

DrugSchedule: schedule_id, name

Drug Master (Drug)

drug_id (PK)

drug_code: Unique code

drug_name: Brand name

generic_name: Salt name

manufacturer

unit: Tablets/ML/Strips

purchase_price

selling_price

reorder_level, is_active

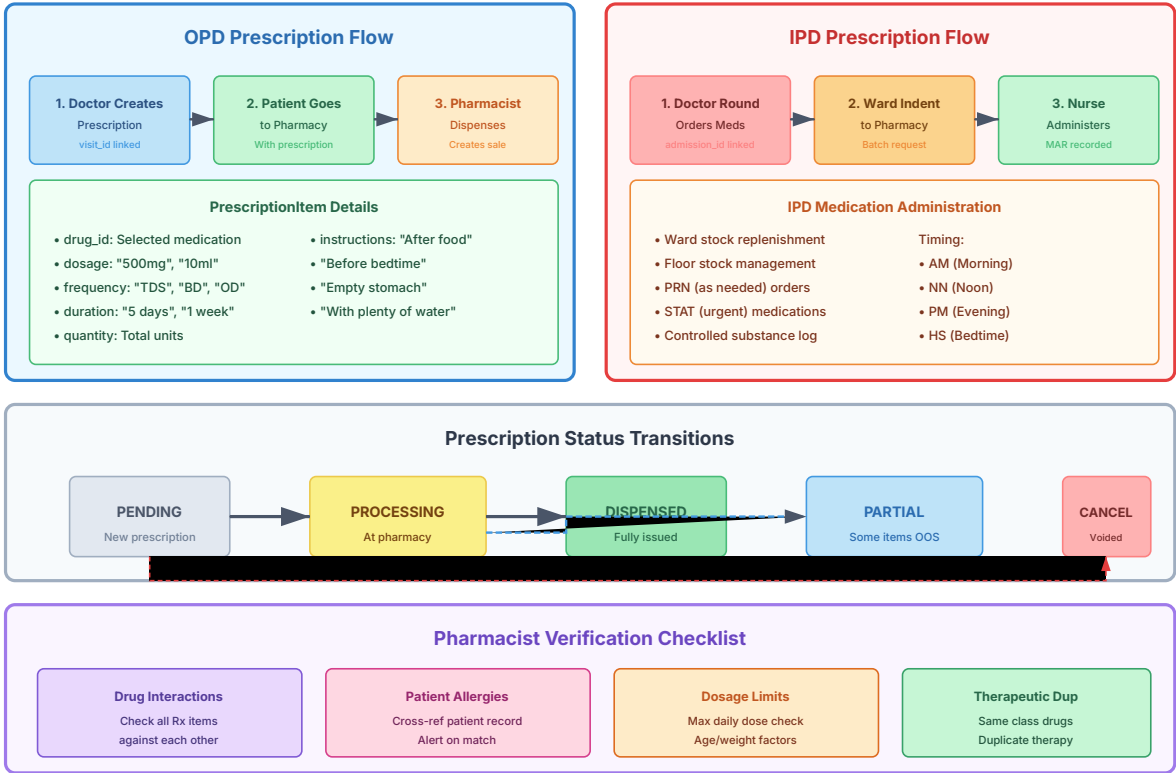
Stock Monitoring

CRITICAL
stock = 0

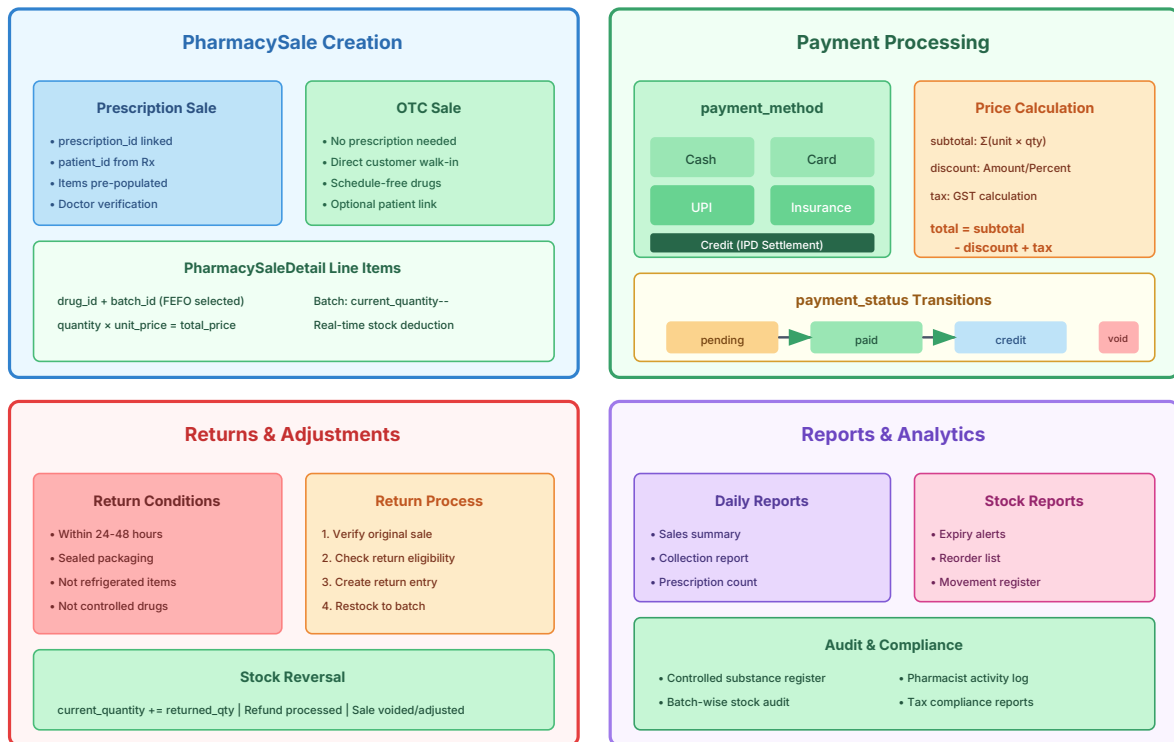
REORDER
stock ≤ reorder_level

ADEQUATE
stock > reorder_level

Prescription Processing - OPD & IPD Medication Workflows



Pharmacy Billing & Sales - Payment Processing & Returns



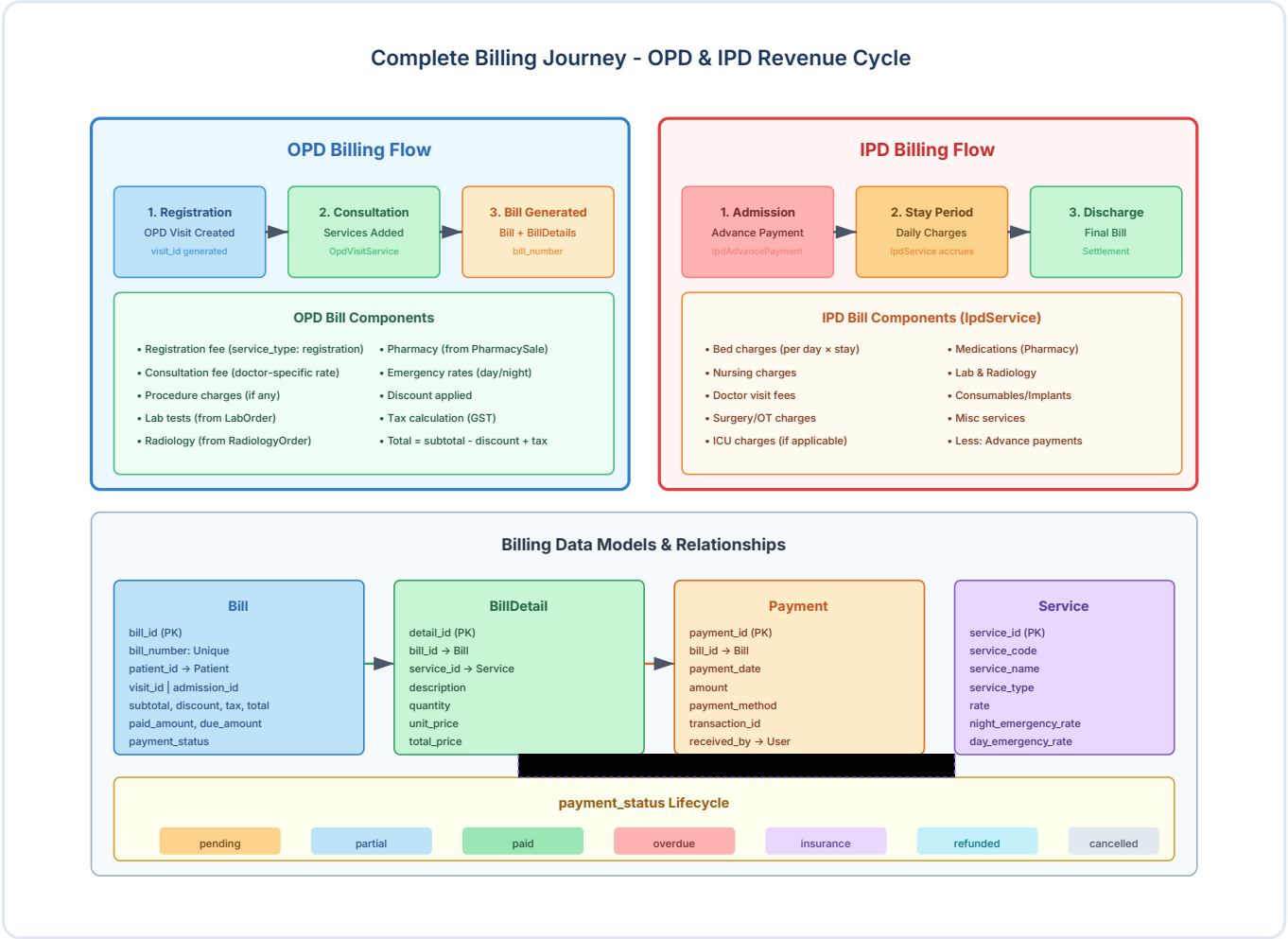
Pharmacy Workflow Key Points

- **Prescription Sources:** OPD (visit_id), IPD (admission_id), Emergency - all create Prescription records with PrescriptionItems
- **Drug Master:** Maintains drug_code, generic_name, manufacturer with category (Antibiotics, etc.) and schedule (H, H1, X, G, OTC)
- **Batch Management:** DrugBatch tracks batch_number, quantity, current_quantity, expiry_date - uses FEFO (First Expiry First Out)
- **Stock Monitoring:** Automatic reorder alerts when stock \leq reorder_level, expiry alerts for items expiring within 30 days
- **Pharmacist Verification:** Drug interactions, patient allergies, dosage limits, therapeutic duplications checked before dispensing
- **Sales Processing:** PharmacySale links prescription_id, calculates subtotal - discount + tax = total, supports multiple payment methods
- **Stock Deduction:** Real-time current_quantity decrement on dispense, increment on returns (if eligible)
- **IPD Integration:** Ward stock indents, floor stock management, Medication Administration Record (MAR)

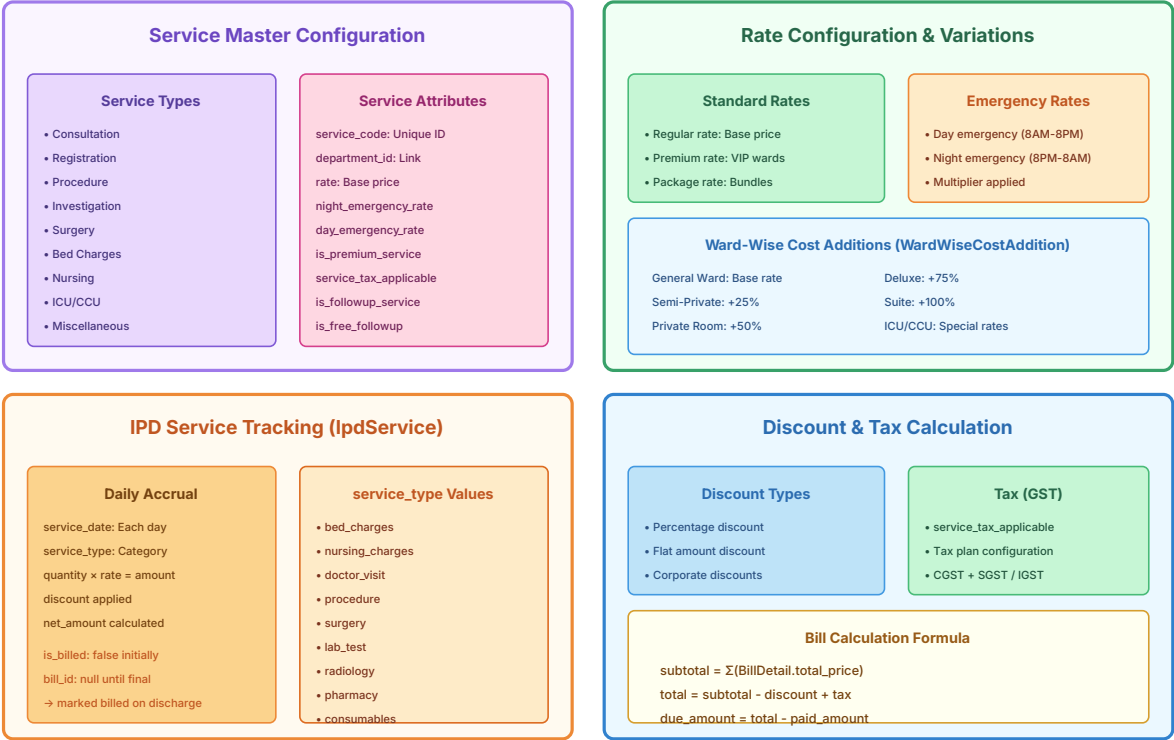
4.9 Billing Module

The Billing Module manages comprehensive hospital revenue cycle including OPD/IPD billing, service charges, payment processing, insurance claims, and financial reporting with multi-mode payment support.

4.9.1 Billing Workflow Diagrams



Service & Charges Management - Pricing, Rates & Categories



IPD Service Tracking (IpdService)

Daily Accrual

service_date: Each day
service_type: Category
quantity x rate = amount
discount applied
net_amount calculated

is_billed: false initially
bill_id: null until final
→ marked billed on discharge

service_type Values

- bed_charges
- nursing_charges
- doctor_visit
- procedure
- surgery
- lab_test
- radiology
- pharmacy
- consumables

Discount & Tax Calculation

Discount Types

- Percentage discount
- Flat amount discount
- Corporate discounts

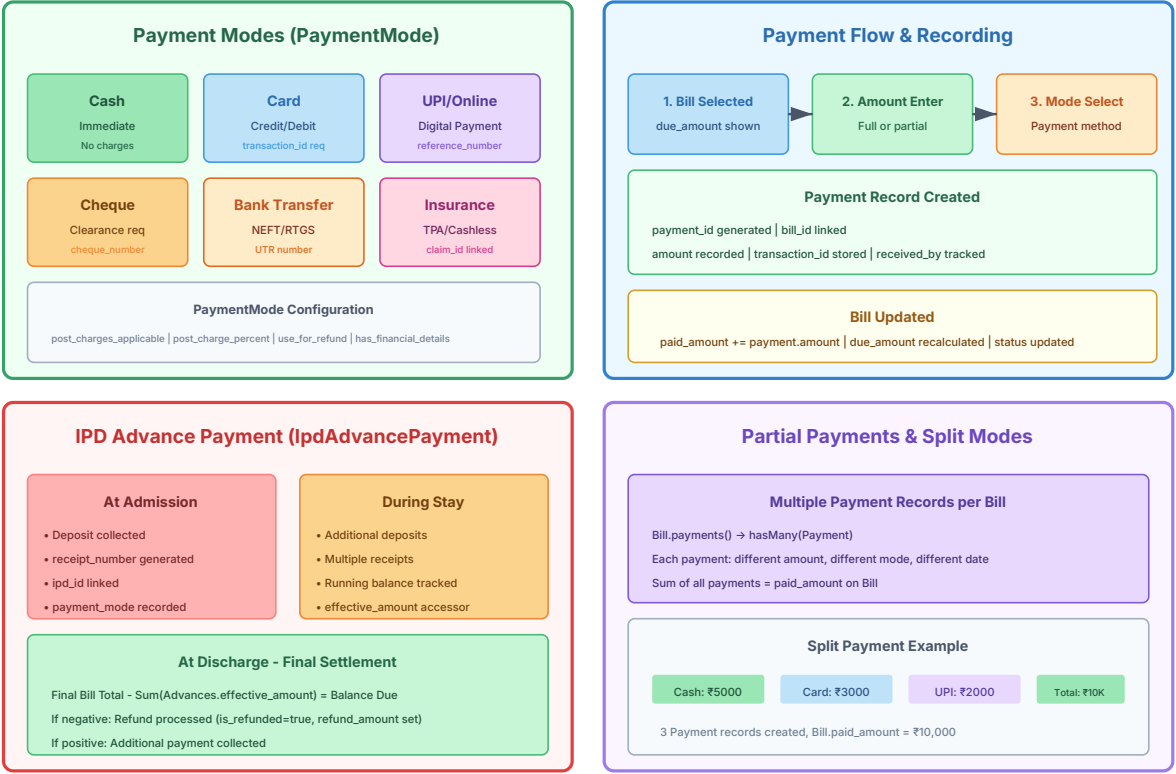
Tax (GST)

- service_tax_applicable
- Tax plan configuration
- CGST + SGST / IGST

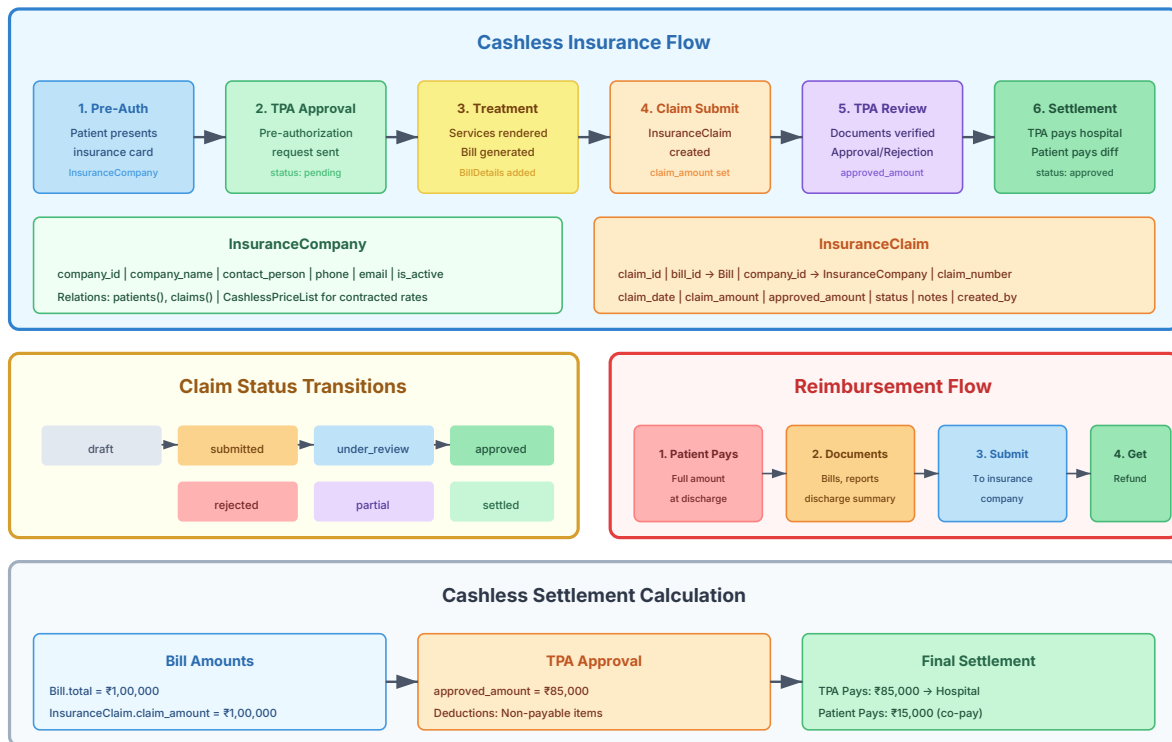
Bill Calculation Formula

subtotal = $\Sigma(\text{BillDetail.total_price})$
total = subtotal - discount + tax
due_amount = total - paid_amount

Payment Processing - Multiple Modes, Partial Payments & Advances



Insurance Claims - Cashless & Reimbursement Workflows



Billing Workflow Key Points

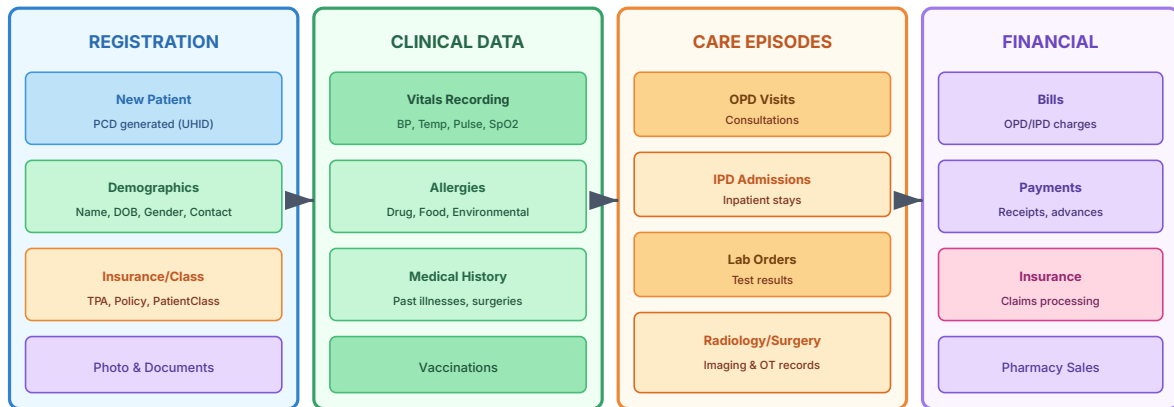
- **Bill Structure:** Bill (header) with multiple BillDetails (line items), each linked to Service with quantity × unit_price = total_price
- **OPD vs IPD:** OPD bills via visit_id (immediate), IPD bills via admission_id with IpdService accrual during stay
- **Service Rates:** Standard rate, emergency rates (day/night), ward-wise cost additions, premium service rates
- **Payment Processing:** Multiple Payment records per Bill, supports split payments across different modes
- **IPD Advances:** IpdAdvancePayment tracks deposits, deducted from final bill, refund if excess
- **Payment Status:** pending → partial → paid | overdue | insurance | refunded | cancelled
- **Insurance Claims:** InsuranceClaim links Bill to InsuranceCompany, tracks claim_amount vs approved_amount
- **Cashless Flow:** Pre-auth → Treatment → Claim → TPA Review → Settlement (TPA + Patient co-pay)

4.10 Patient Management Module

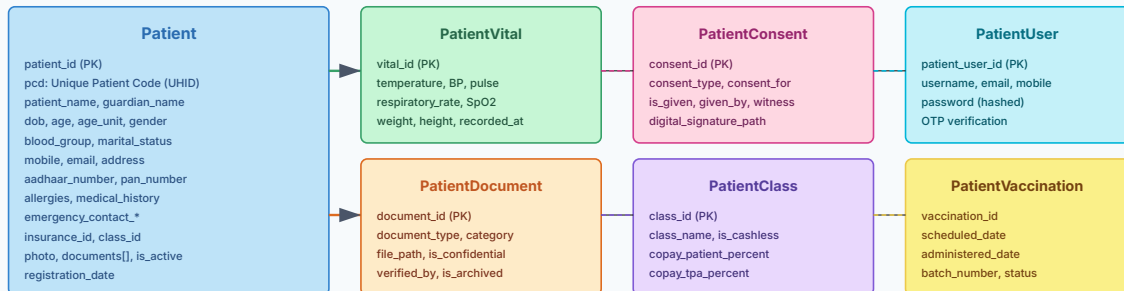
The Patient Management Module is the foundation of HIMs, handling patient registration, demographics, medical records, vital signs tracking, document management, consent forms, and patient portal access for self-service capabilities.

4.10.1 Patient Management Workflow Diagrams

Complete Patient Journey - Registration to Comprehensive Medical Record



Patient Data Model & Relationships



Patient Data Management - Demographics, Documents & Consents

Patient Demographics

Personal Information

- patient_name
- guardian_name, relation
- dob / age + age_unit
- gender (M/F/O)
- marital_status
- blood_group
- nationality, religion
- occupation
- is_bpl, is_foreigner
- photo (file path)

Contact & Address

- mobile (primary)
- phone (alternate)
- email
- subscribe_sms
- Current Address:
- address, area, city
- district, state, country
- pincode
- Permanent Address:
- permanent_* fields

Identity & Insurance

ID Documents

- aadhaar_number
- pan_number
- documents[] (JSON array)
- Passport, Voter ID, etc.

Insurance Details

- insurance_id → Company
- insurance_policy_number
- cashless_referral_no
- tpa_id

PatientClass (Billing Category)

class_id → PatientClass | class_code, class_name | is_cashless
copay_patient_percent, copay_tpa_percent | pharmacy_cash | service_tax_applicable
prior_approval_required, prior_approval_limit | grace_period_days

Document Management (PatientDocument)

Document Types

- id_proof (Aadhaar, PAN)
- insurance_card
- prescription
- lab_report
- radiology_image
- discharge_summary
- consent_form
- referral_letter
- other

Document Workflow

- Upload:
- file_path, file_name
 - file_type, file_size
 - uploaded_by → User
- Verification:
- verified_by, verified_at
- Archival:
- is_archived, archived_at
- Security:
- is_confidential_flag

Consent Management (PatientConsent)

Consent Types

- general_treatment
- surgery
- anesthesia
- blood_transfusion
- high_risk_procedure
- discharge_against_advice
- data_sharing
- research_participation
- photography

Consent Details

- Given By:
- given_by (name)
 - relationship_to_patient
 - witness_name
- Signatures:
- consent_form_path
 - digital_signature_path
- Revocation:
- revoked_at, revoked_by
 - revocation_reason

Patient Vitals & Clinical Data - Vital Signs, Allergies & Medical History

Vital Signs Recording (PatientVital)

Parameters

- temperature (°F/°C)
- BP systolic (mmHg)
- BP diastolic (mmHg)
- pulse_rate (bpm)
- respiratory_rate
- oxygen_saturation %
- weight (kg)
- height (cm)

Recording Context

- Linked To:
- visit_id (OPD)
 - admission_id (IPD)
- Metadata:
- recorded_at (datetime)
 - recorded_by → User
 - notes (free text)
- Multiple records per visit/admission

Normal Ranges

Temp: 97.8-99.1°F
BP: 90-120/60-80
Pulse: 60-100 bpm
Resp: 12-20/min
SpO2: 95-100%

- Alerts Generated:
- Critical values
 - Trend analysis
 - Early warning score

Emergency Information

Emergency Contacts

- emergency_contact_name
- emergency_contact_phone
- emergency_contact_relation

Critical Flags

- is_urgent: Priority patient
- charges_type: Emergency rates
- allergies: CRITICAL display

Allergies & Medical History

Allergies (Text Field)

- Drug Allergies:
- Penicillin, Sulfa drugs
 - NSAIDs, Aspirin
- Food Allergies:
- Nuts, Shellfish, Eggs
- Environmental:
- Latex, Iodine contrast
- ⚠ Displayed prominently

Medical History (Text)

- Past Illnesses:
- Diabetes, Hypertension
 - Heart disease, Asthma
- Past Surgeries:
- Appendectomy, C-section
- Current Medications:
- Ongoing prescriptions
- Family History noted

Vaccination Tracking (PatientVaccination)

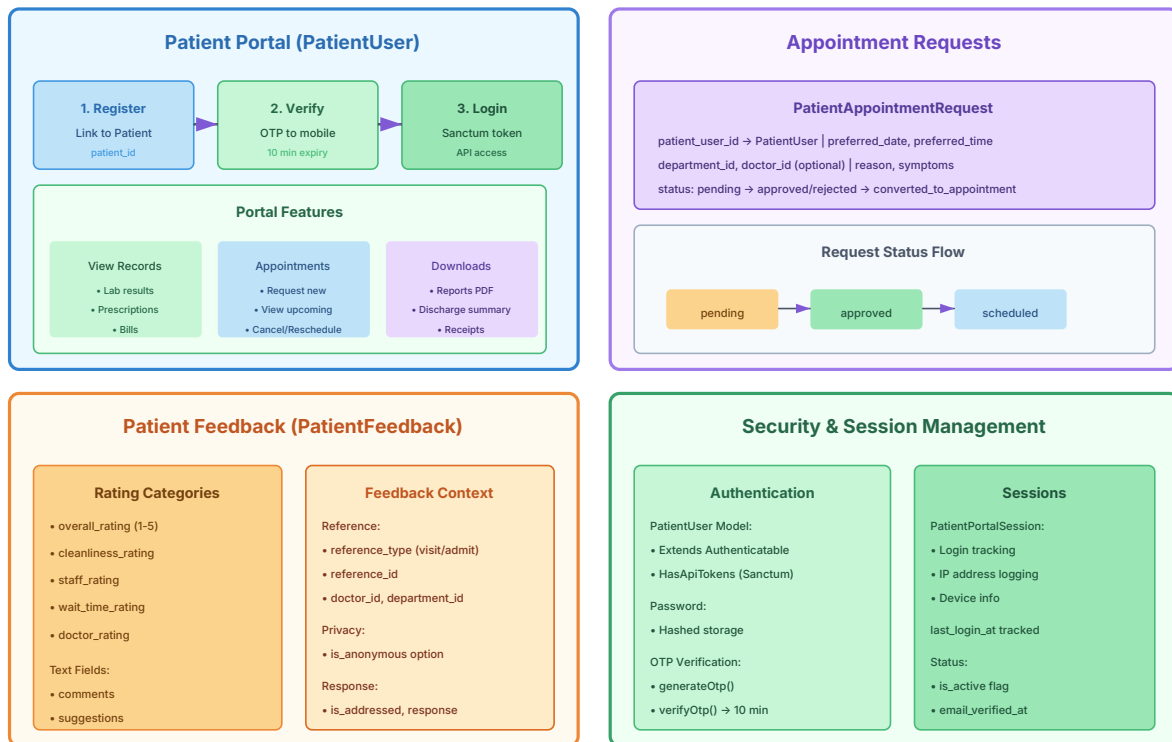
Vaccination Record

- vaccination_id → Vaccine
 - scheduled_date
 - administered_date
 - batch_number
 - administered_by
- Status:
- scheduled
 - administered

Vaccine Types

- Pediatric:
- BCG, OPV, DPT
 - Hepatitis B, MMR
- Adult:
- Influenza, Pneumonia
 - Tetanus, COVID-19
- Special:
- Rabies, Hepatitis A

Patient Portal & Engagement - Self-Service & Feedback



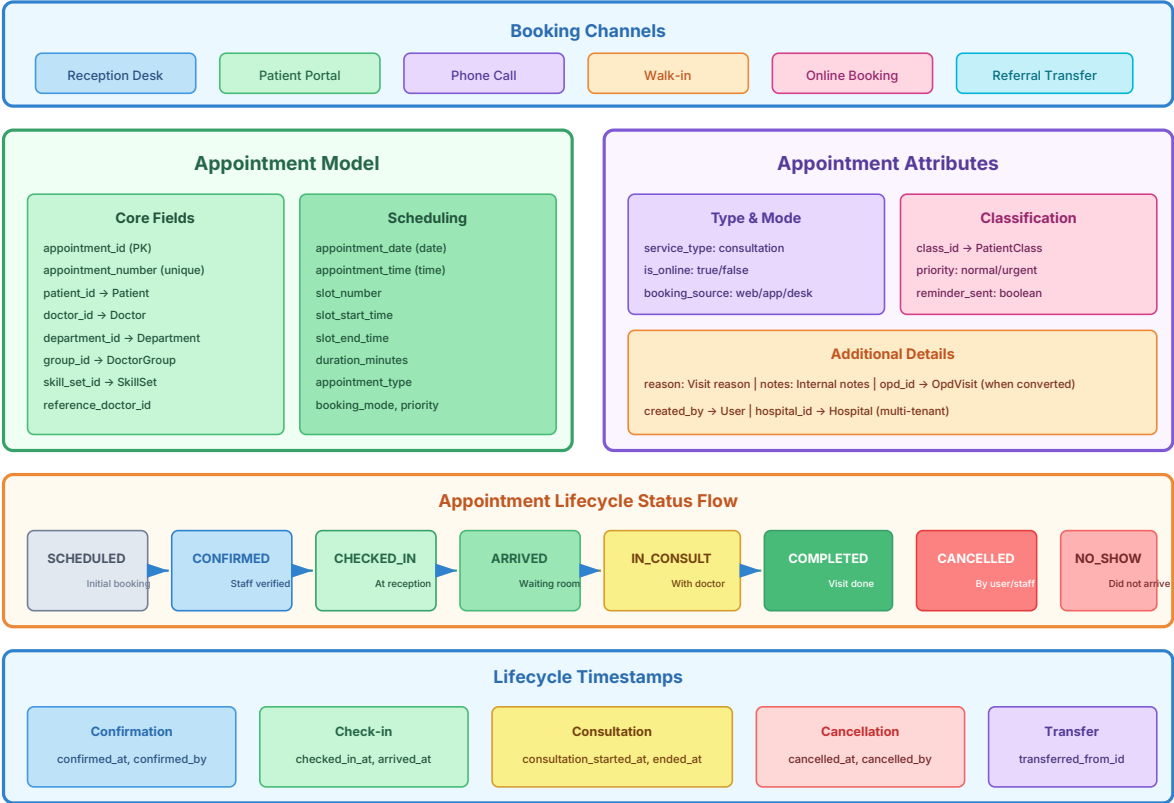
Patient Management Workflow Key Points

- **Patient Registration:** Unique PCD (Patient Code/UHID) generated, comprehensive demographics with current/permanent addresses
- **Identity Documents:** Aadhaar, PAN stored; documents[] JSON array for multiple ID proofs; PatientDocument for uploaded files
- **Insurance Integration:** insurance_id links to InsuranceCompany; class_id defines billing category (cashless, copay percentages)
- **Vital Signs:** PatientVital records temperature, BP, pulse, SpO2, weight, height - linked to OPD visit or IPD admission
- **Clinical Alerts:** allergies field displayed prominently; medical_history for past illnesses/surgeries; is_urgent flag for priority
- **Consent Management:** PatientConsent tracks consent_type (surgery, anesthesia, blood), digital signatures, revocation capability
- **Patient Portal:** PatientUser with Sanctum auth, OTP verification, appointment requests, feedback submission
- **Care Episodes:** Patient links to appointments(), opdVisits(), ipdAdmissions(), labOrders(), bills(), vaccinations()

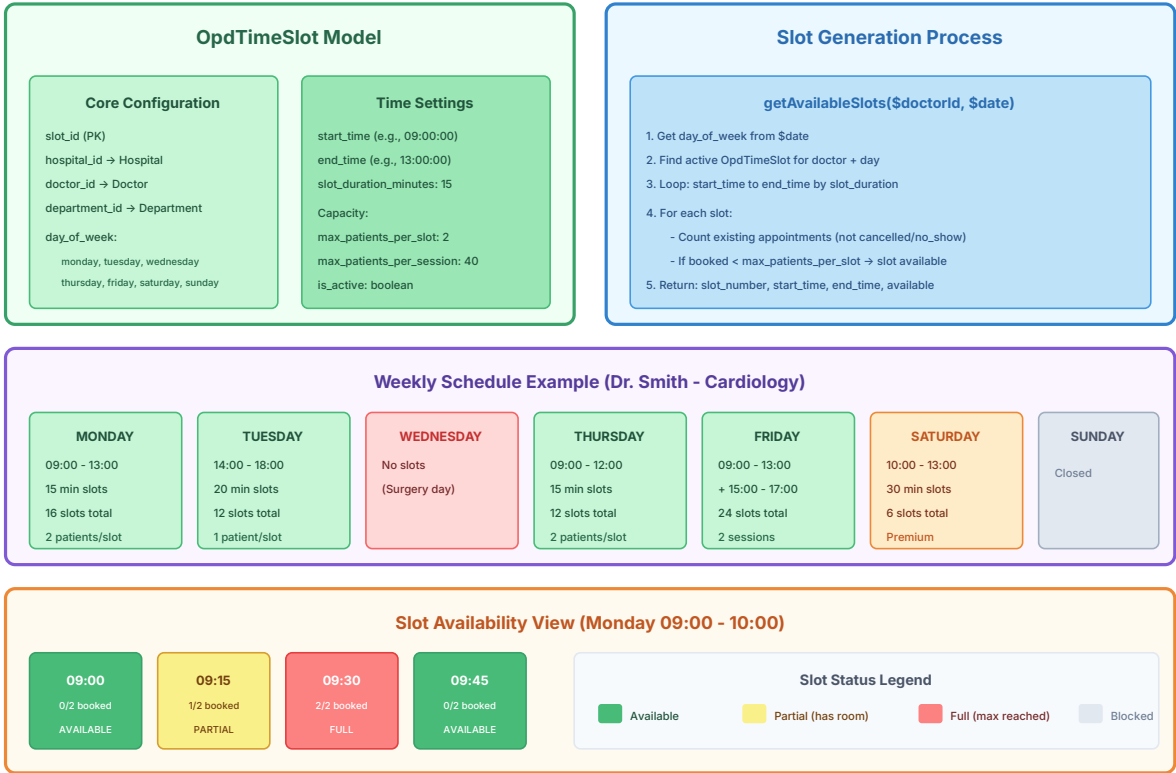
4.11 Appointment Scheduling Module

The appointment system handles patient scheduling with time slot management, multi-channel booking, appointment lifecycle tracking, cancellation/transfer workflows, and patient portal self-service requests.

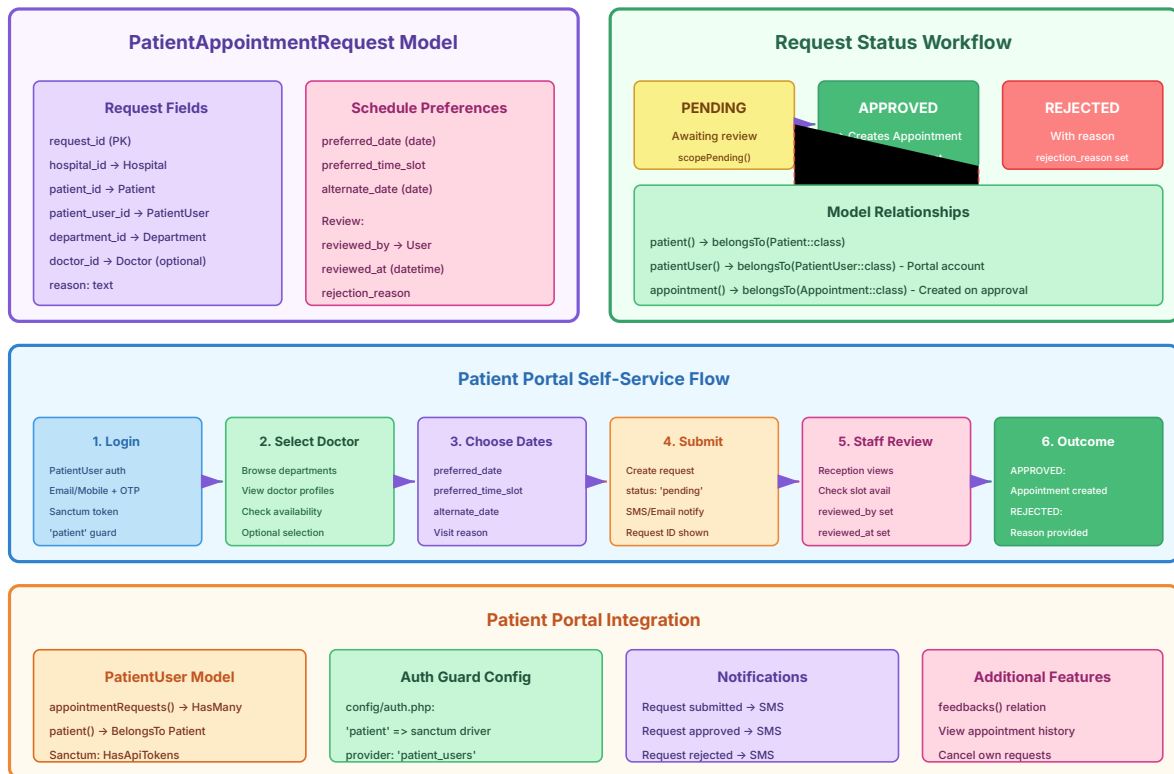
Appointment Booking & Lifecycle - Complete Patient Journey



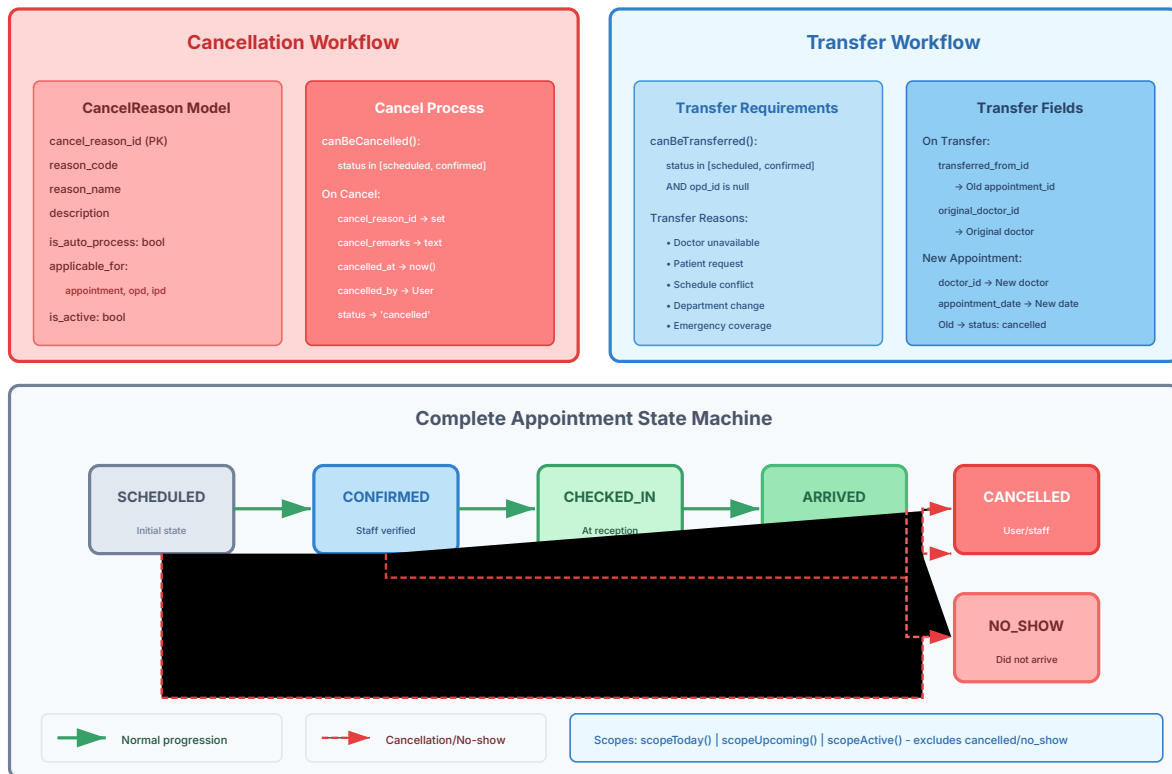
Doctor Schedule & Slot Management - OpdTimeSlot Configuration



Patient Portal Appointment Request - Self-Service Booking Flow



Appointment Status Transitions - Cancellation, Transfer & Completion



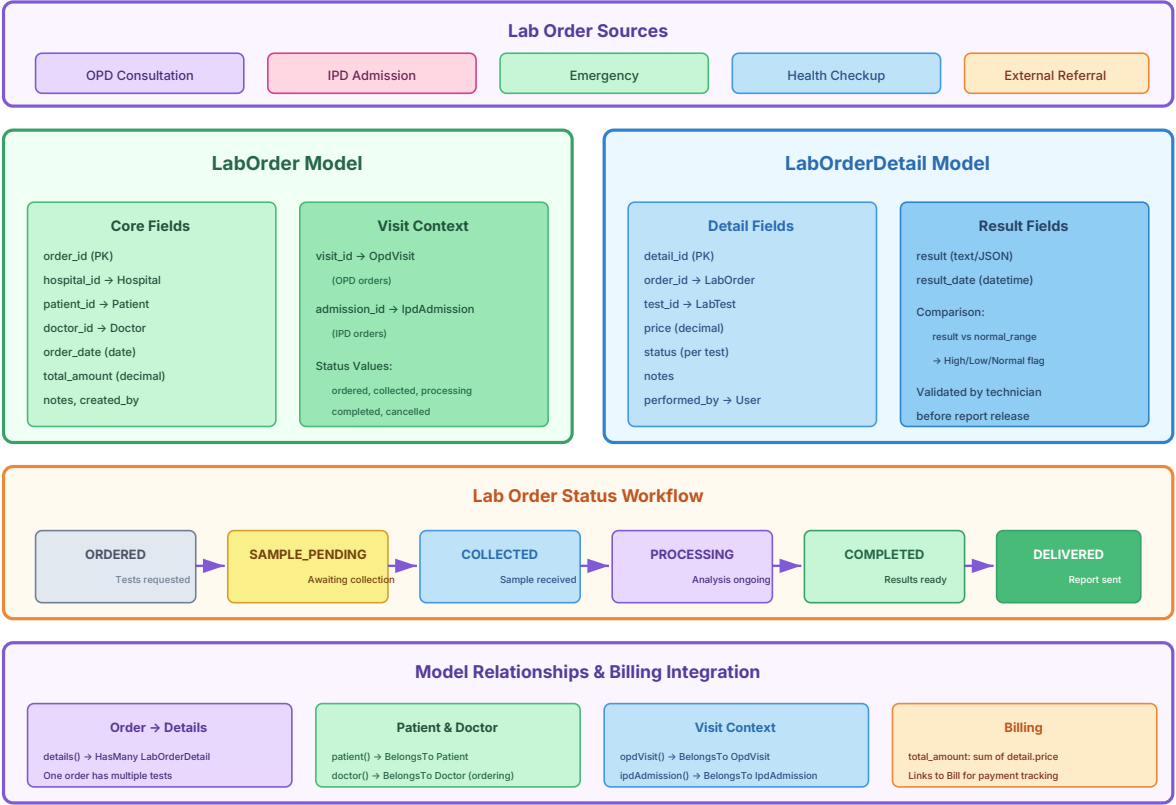
Appointment Scheduling Workflow Key Points

- **Multi-Channel Booking:** Reception desk, patient portal, phone, walk-in, online booking, referral transfers
- **Time Slot Management:** OpdTimeSlot configures doctor schedules per day_of_week with slot_duration, max_patients_per_slot
- **Slot Availability:** getAvailableSlots(\$doctorId, \$date) calculates real-time availability excluding cancelled/no_show
- **Status Lifecycle:** scheduled → confirmed → checked_in → arrived → in_consultation → completed (or cancelled/no_show)
- **Patient Portal Requests:** PatientAppointmentRequest allows self-service with preferred_date/alternate_date, pending staff review
- **Cancellation:** canBeCancelled() checks status; CancelReason provides standardized reasons with is_auto_process flag
- **Transfer Support:** canBeTransferred() allows doctor/date changes; tracks original_doctor_id and transferred_from_id
- **OPD Conversion:** opd_id links to OpdVisit when appointment converts to actual consultation

4.12 Laboratory Management Module

The laboratory system manages test ordering, sample collection, result entry, and reporting with comprehensive tracking from order creation through result delivery to patients and referring physicians.

Laboratory Order Lifecycle - From Order to Results



Lab Order Status Workflow

ORDERED

Tests requested

SAMPLE_PENDING

Awaiting collection

COLLECTED

Sample received

PROCESSING

Analysis ongoing

COMPLETED

Results ready

DELIVERED

Report sent

Model Relationships & Billing Integration

Order → Details

details[] → HasMany LabOrderDetail
One order has multiple tests

Patient & Doctor

patient() → BelongsTo Patient
doctor() → BelongsTo Doctor (ordering)

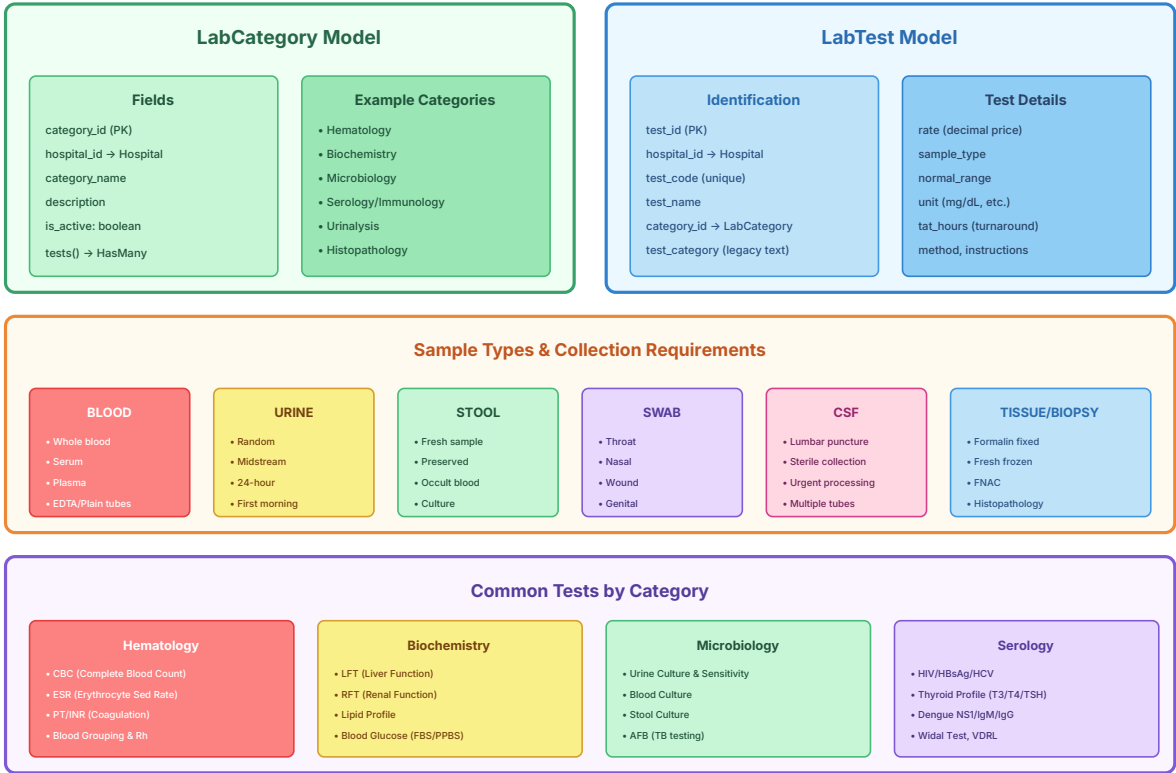
Visit Context

opdVisit() → BelongsTo OpdVisit
ipdAdmission() → BelongsTo IpdAdmission

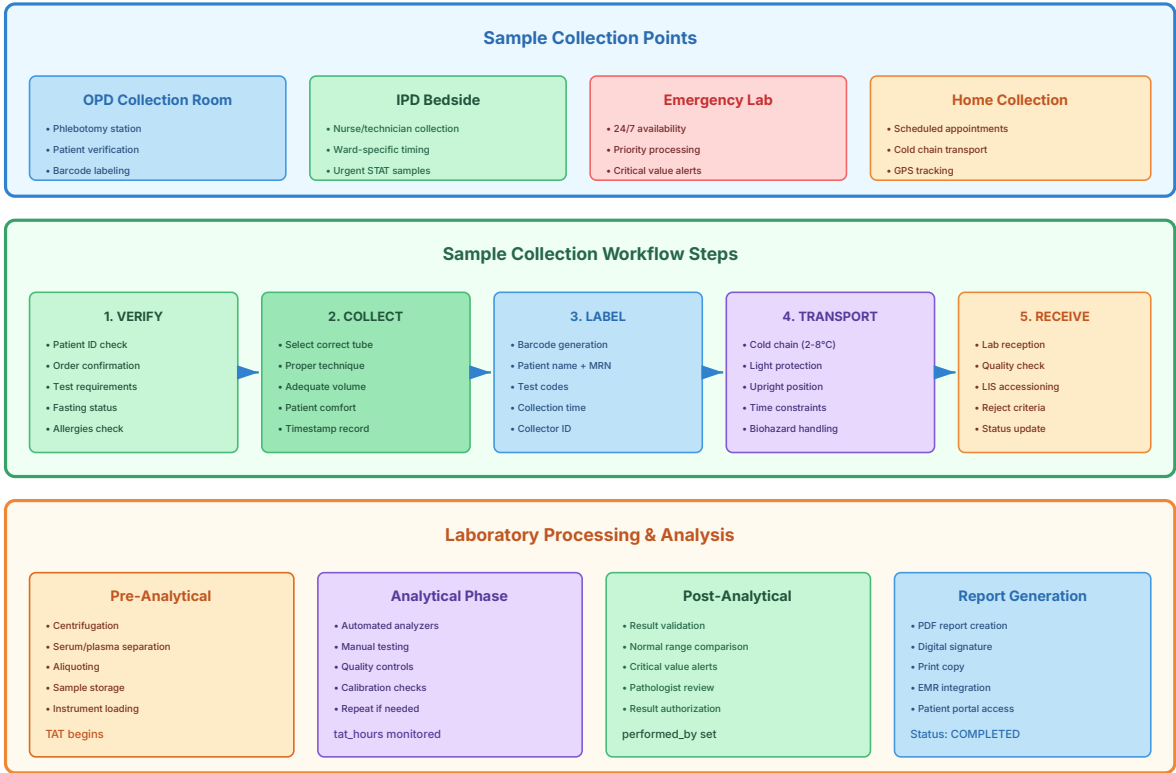
Billing

total_amount: sum of detail.price
Links to Bill for payment tracking

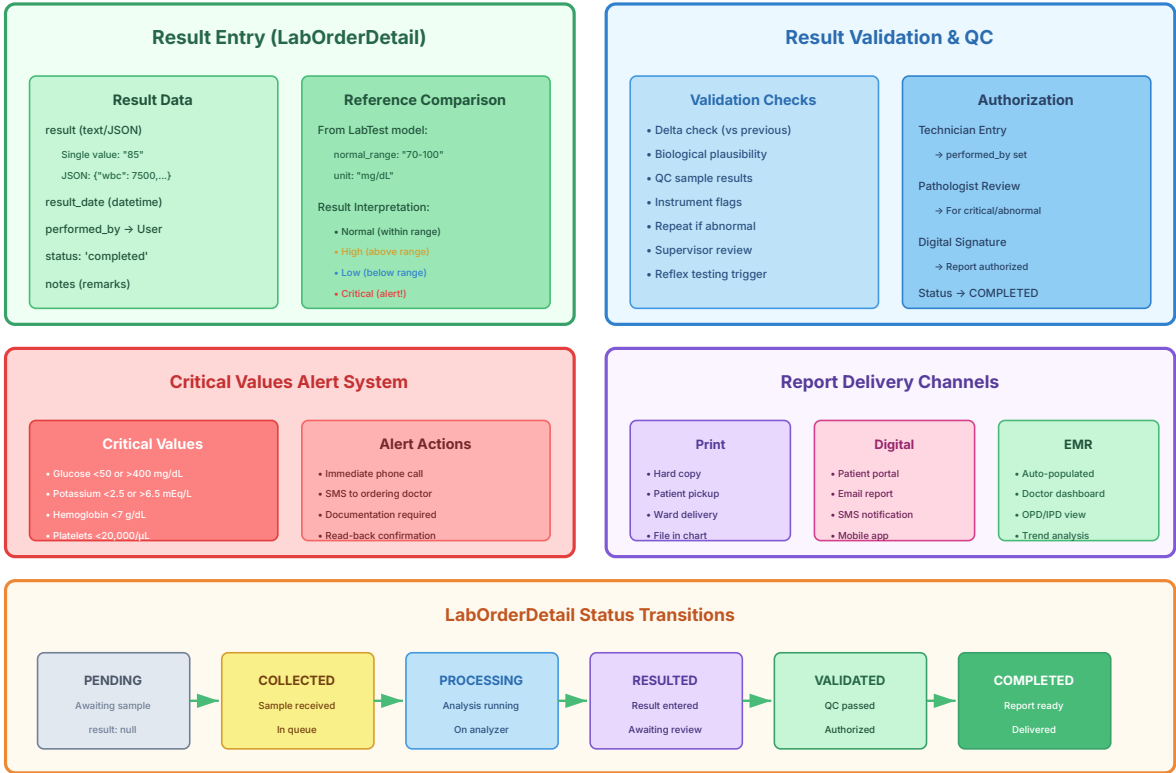
Laboratory Test Catalog - Categories, Tests & Sample Types



Sample Collection & Processing Workflow



Result Entry, Validation & Reporting Workflow



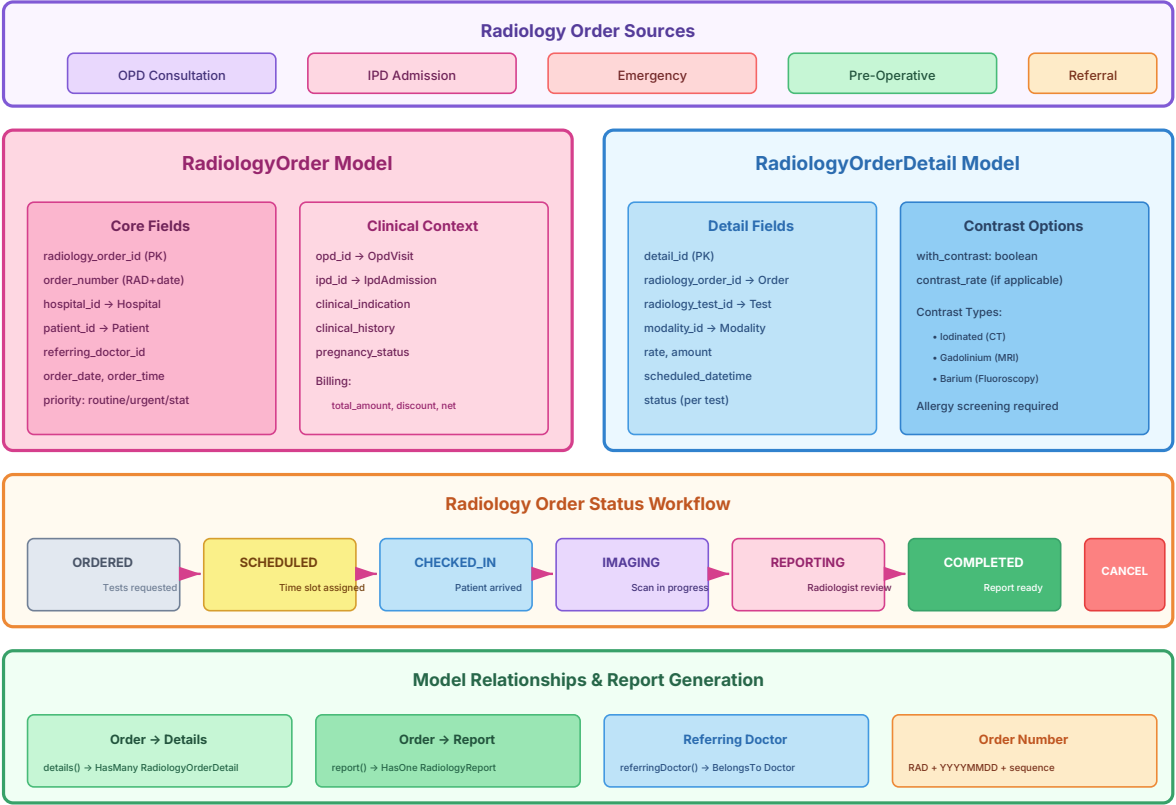
Laboratory Management Workflow Key Points

- **Test Catalog:** LabCategory organizes tests; LabTest defines test_code, rate, sample_type, normal_range, unit, tat_hours, method
- **Order Management:** LabOrder links to patient, doctor, OPD visit or IPD admission; details() contains individual tests
- **Sample Types:** Blood (serum/plasma/EDTA), Urine, Stool, Swab, CSF, Tissue - each with specific collection requirements
- **Collection Workflow:** Verify patient → Collect sample → Label with barcode → Transport with cold chain → Receive at lab
- **Processing Phases:** Pre-analytical (centrifugation, aliquoting) → Analytical (testing) → Post-analytical (validation, reporting)
- **Result Entry:** LabOrderDetail stores result, result_date, performed_by; compared against normal_range for interpretation
- **Critical Values:** Immediate phone notification to ordering doctor with read-back confirmation required
- **Report Delivery:** Print, patient portal, email, SMS, and direct EMR integration for seamless clinical workflow

4.13 Radiology & Imaging Module

The radiology system manages diagnostic imaging orders, modality scheduling, PACS/DICOM integration, radiologist reporting with critical findings alerts, and image delivery to clinicians through the EMR.

Radiology Order Lifecycle - From Order to Report



Radiology Modalities & Test Catalog

RadiologyModality Model

Fields

modality_id (PK)
hospital_id → Hospital
modality_code (CT, MR, US)
modality_name
room_number
default_tat_hours

Options

is_contrast_available: bool
is_active: boolean

Relationship:
tests() → HasMany
RadiologyTest
scopeActive()

RadiologyTest Model

Identification

radiology_test_id (PK)
modality_id → Modality
test_code (unique)
test_name
body_part (chest, head...)
laterality (left/right/both)

Details & Pricing

cpt_code (billing)
rate (base price)
contrast_required: bool
contrast_rate (add-on)
consent_required: bool
tat_hours, is_active

Imaging Modality Types

X-RAY (CR/DR)

- Chest PA/Lateral
- Abdomen
- Extremities
- Spine views

CT SCAN

- CT Head/Brain
- CT Chest (HRCT)
- CT Abdomen
- CT Angiography

MRI

- MRI Brain
- MRI Spine
- MRI Knee/Joint
- MRA/MRV

ULTRASOUND

- USG Abdomen
- USG Pelvis
- Obstetric USG
- Doppler studies

MAMMOGRAPHY

- Screening
- Diagnostic
- Tomosynthesis
- BI-RADS scoring

FLUOROSCOPY

- Barium Swallow
- Barium Meal/Enema
- IVP/MCU/RGU
- Hysterosalpingography

Patient Preparation Instructions (preparation_instructions field)

Fasting Required

- CT Abdomen with contrast
 - USG Abdomen
 - Barium studies
- 6-8 hours fasting

Bladder Full

- USG Pelvis
 - Early Obstetric USG
- Drink 4-5 glasses water
1 hour before scan

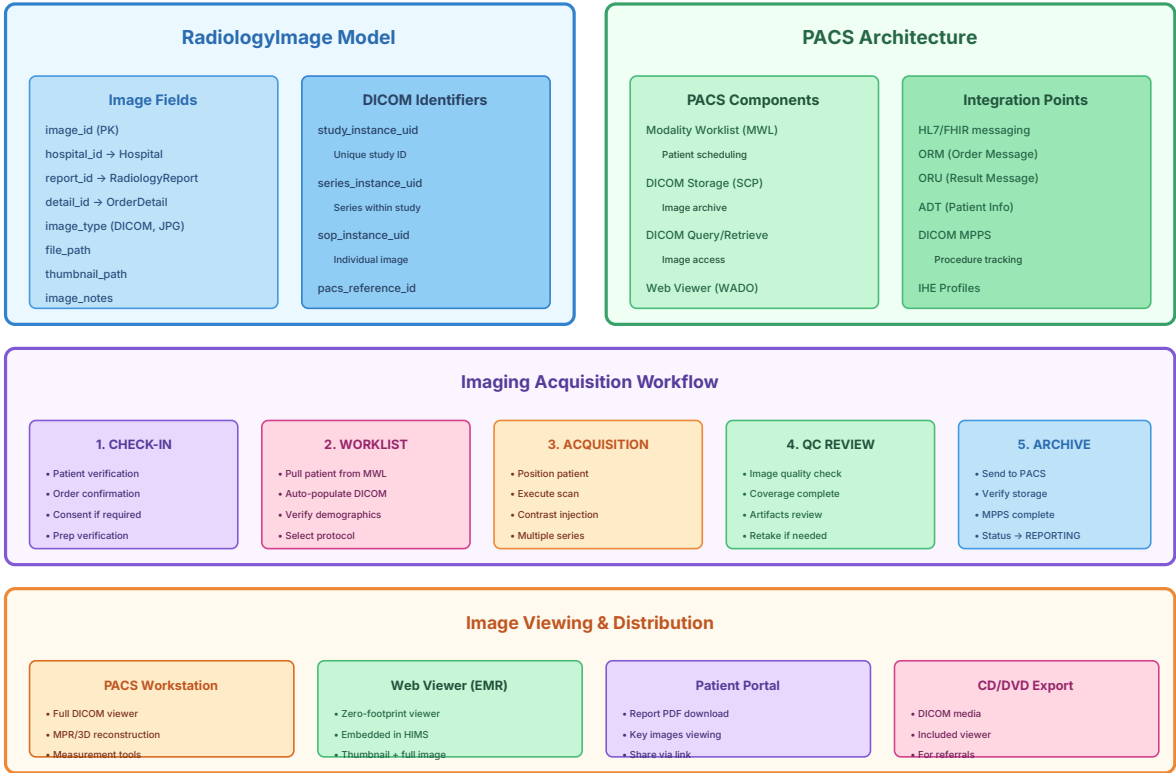
Contrast Prep

- Kidney function test (eGFR)
- Allergy history check
- Hydration protocol
- Consent signed

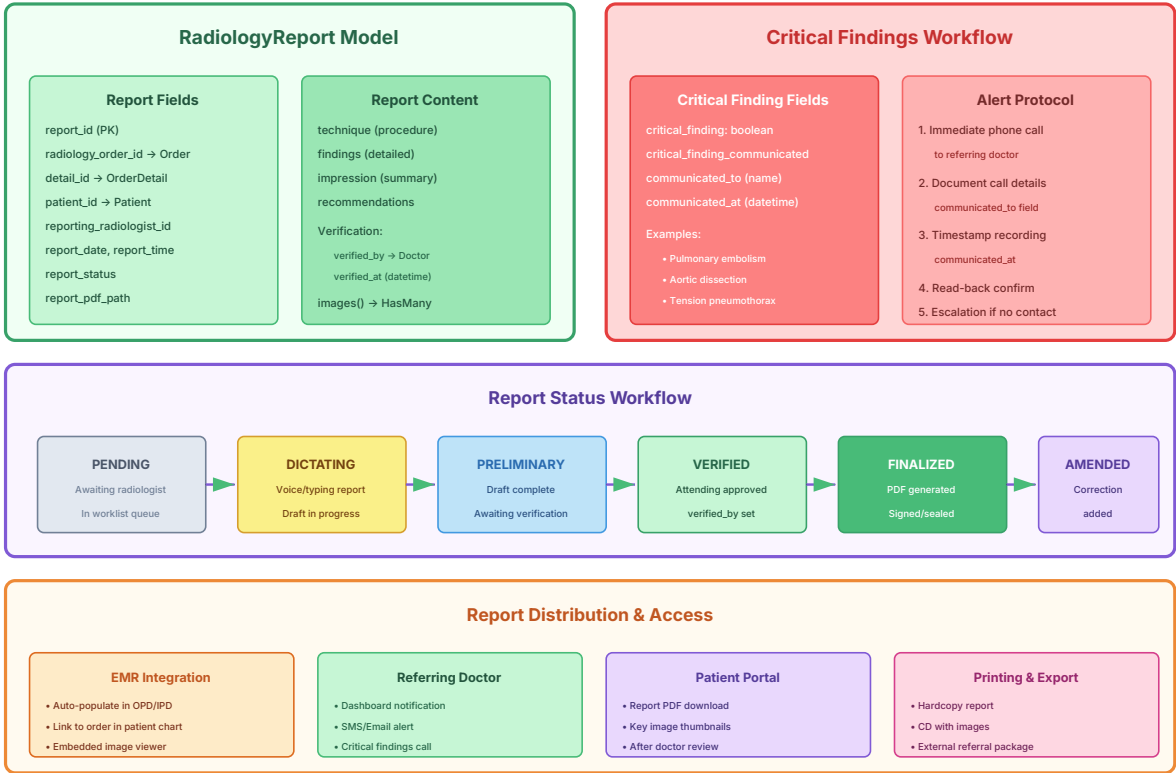
MRI Screening

- Metal implants check
- Pacemaker exclusion
- Claustrophobia assess
- Remove jewelry/metals

Imaging Workflow & PACS/DICOM Integration



Radiologist Reporting & Verification Workflow



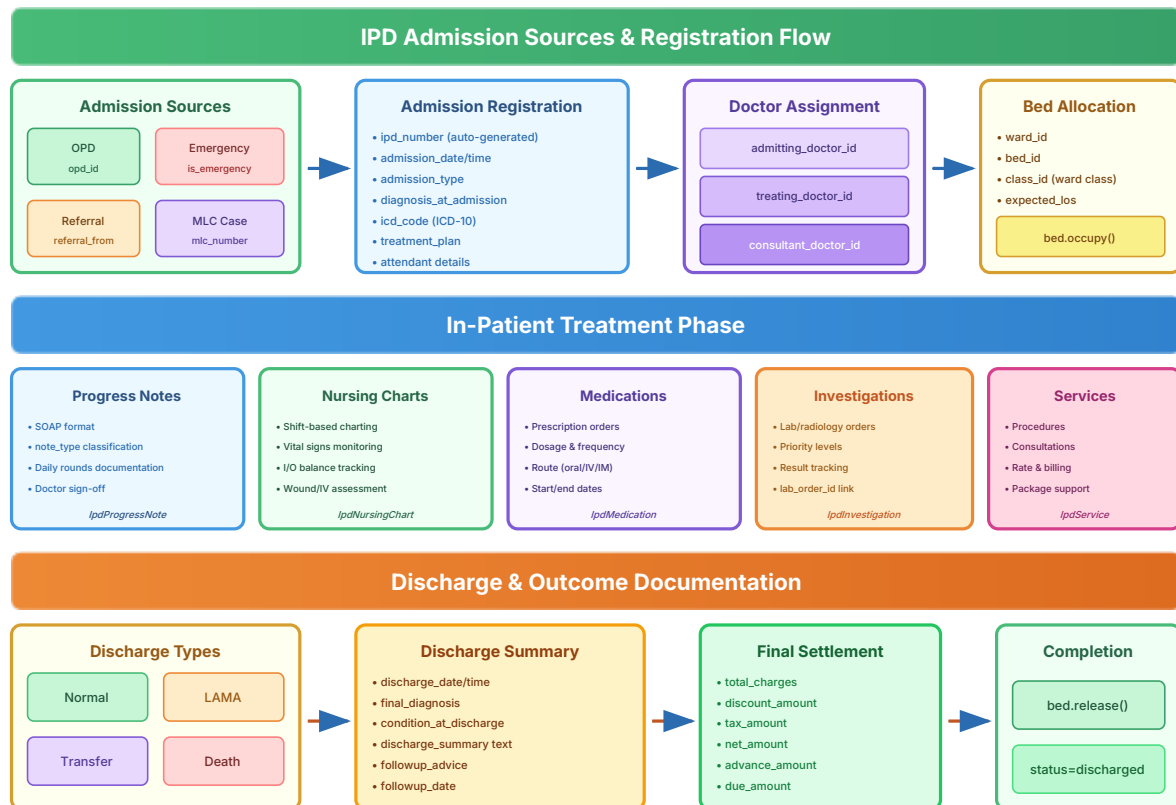
Radiology & Imaging Workflow Key Points

- **Modality Management**: RadiologyModality defines equipment (CT, MRI, X-Ray, USG) with room_number, contrast availability, and default TAT
- **Test Catalog**: RadiologyTest includes test_code, body_part, laterality, cpt_code, rate, contrast_required, consent_required, preparation_instructions
- **Order Workflow**: RadiologyOrder with priority levels (routine/urgent/stat), clinical_indication, pregnancy_status; details() for multiple tests
- **Scheduling**: RadiologyOrderDetail tracks scheduled_datetime, with_contrast, amount per test; status per individual exam
- **PACS Integration**: RadiologyImage stores DICOM UUIDs (study/series/sop_instance_uid), pacs_reference_id for seamless image access
- **Reporting**: RadiologyReport with structured format (technique, findings, impression, recommendations); verified_by for attending sign-off
- **Critical Findings**: critical_finding flag triggers immediate communication protocol with communicated_to and communicated_at documentation
- **Distribution**: Reports accessible via EMR, patient portal, print, CD export; images viewable in zero-footprint web viewer

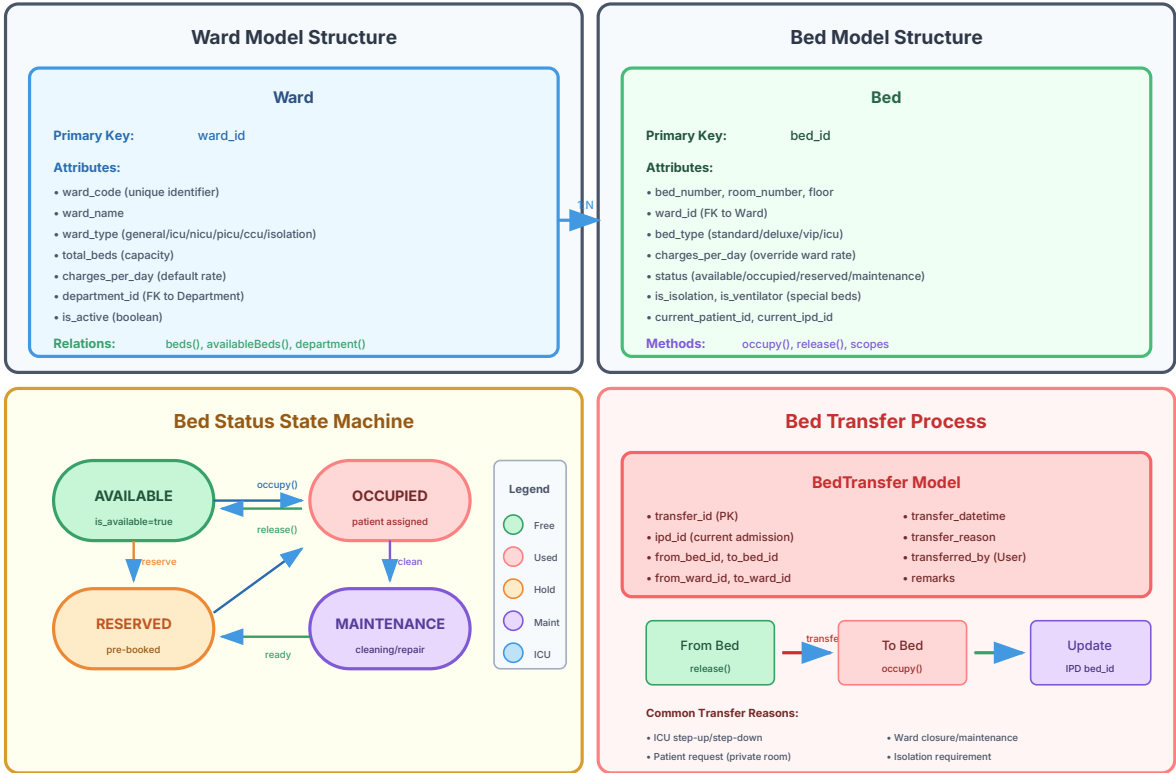
4.14 IPD (In-Patient Department) Module

The IPD module manages patient admissions, ward/bed allocation, clinical documentation, and discharge workflows for hospitalized patients.

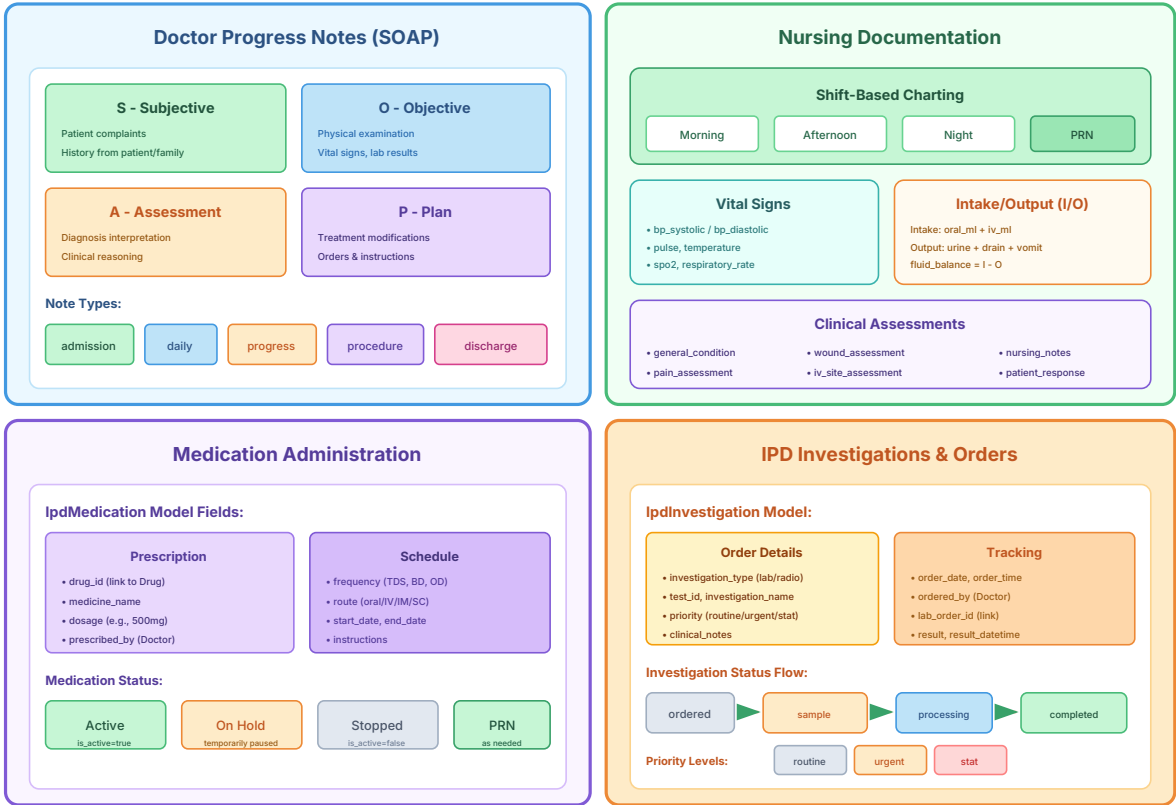
IPD Admission & Discharge Lifecycle



Ward & Bed Management System



Clinical Documentation Workflow



Medication Administration

IpdMedication Model Fields:

Prescription

• drug_id (link to Drug)

• medicine_name

• dosage (e.g., 500mg)

• prescribed_by (Doctor)

Schedule

• frequency (TDS, BD, OD)

• route (oral/IV/IM/SC)

• start_date, end_date

• instructions

Medication Status:

Active
is_active=true

On Hold
temporarily paused

Stopped
is_active=false

PRN
as needed

IPD Investigations & Orders

IpdInvestigation Model:

Order Details

• investigation_type (lab/radio)

• test_id, investigation_name

• priority (routine/urgent/stat)

• clinical_notes

Tracking

• order_date, order_time

• ordered_by (Doctor)

• lab_order_id (link)

• result, result_datetime

Investigation Status Flow:

ordered

sample

processing

completed

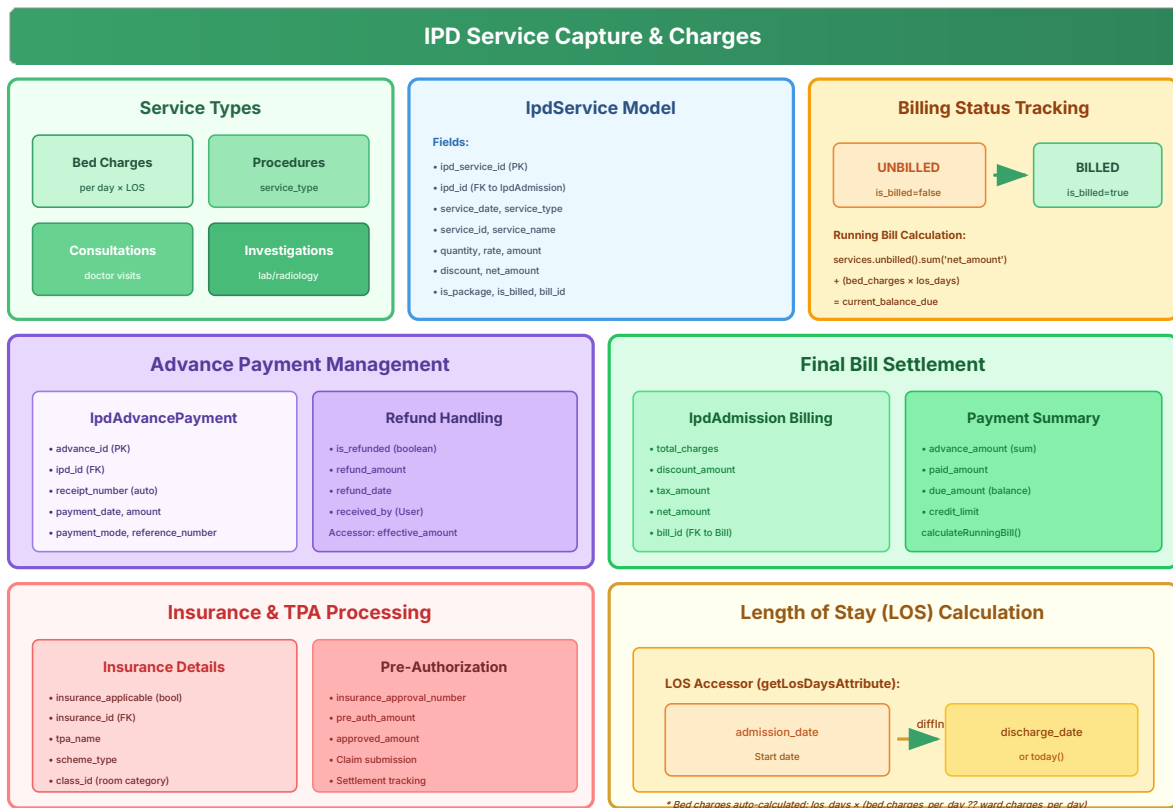
Priority Levels:

routine

urgent

stat

IPD Billing & Services Workflow



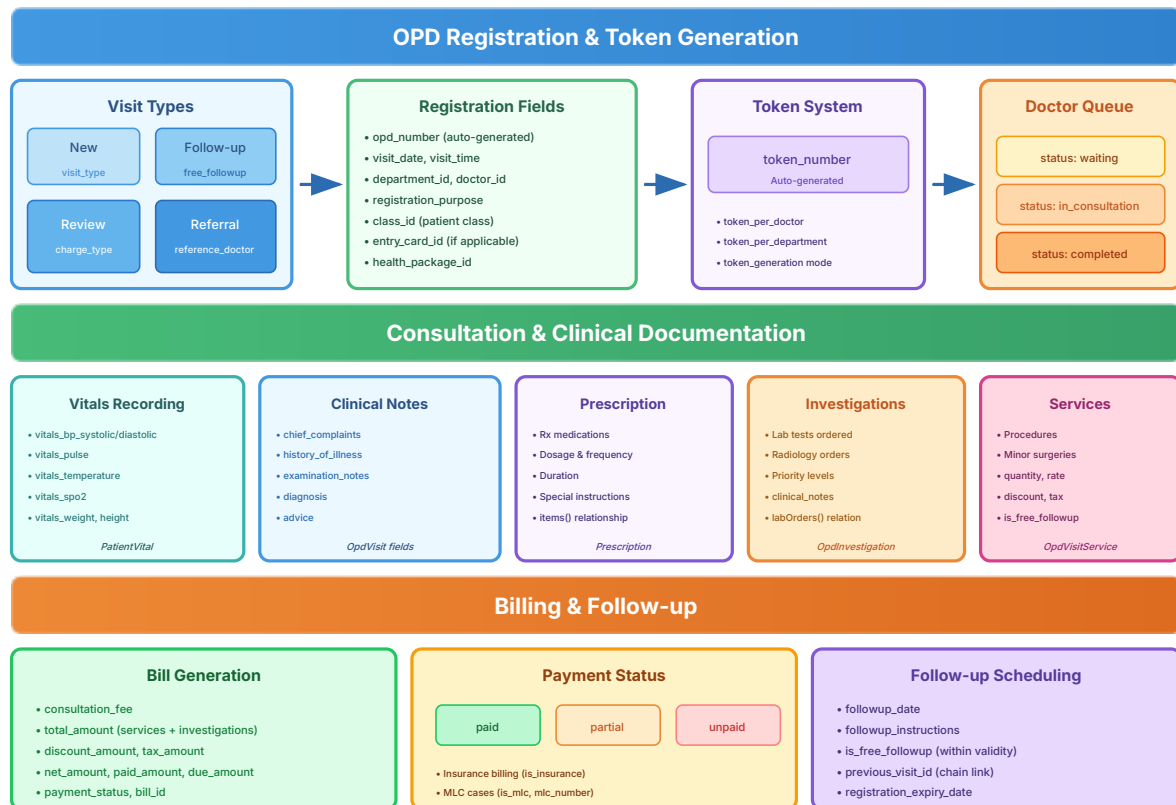
IPD Module Workflow Key Points

- **Admission Sources:** IpdAdmission supports OPD referral (opd_id), emergency (is_emergency), MLC cases (mlc_number), and external referrals
- **Doctor Roles:** Three distinct doctor assignments - admitting_doctor_id, treating_doctor_id, consultant_doctor_id for clear responsibility
- **Bed Management:** Ward → Bed hierarchy with status tracking (available/occupied/reserved/maintenance); occupy() and release() methods for state management
- **Clinical Documentation:** IpdProgressNote uses SOAP format (Subjective, Objective, Assessment, Plan) with note_type classification
- **Nursing Charts:** IpdNursingChart tracks shift-based vitals, I/O balance (fluid_balance accessor), wound/IV assessments
- **Medications:** IpdMedication with drug linkage, dosage/route/frequency, active/stopped status, and date ranges
- **Investigations:** IpdInvestigation with priority levels (routine/urgent/stat), status workflow, and lab_order_id integration
- **Services & Billing:** IpdService captures all charges; calculateRunningBill() provides real-time balance; IpdAdvancePayment tracks deposits with refund support
- **Insurance:** Full TPA workflow with pre-authorization, approved amounts, and scheme tracking
- **Discharge:** Multiple discharge types (normal/LAMA/transfer/death) with comprehensive summary, follow-up scheduling, and bed release

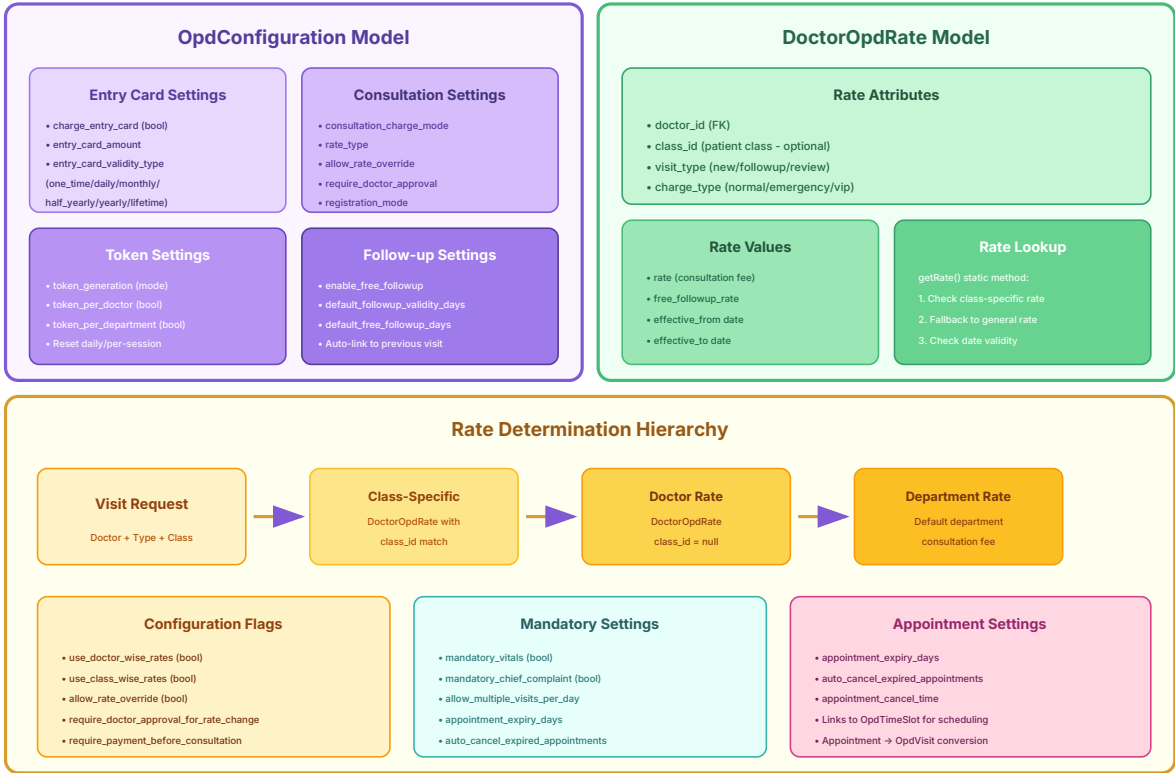
4.15 OPD (Out-Patient Department) Module

The OPD module manages outpatient visits, consultations, doctor queues, prescriptions, and billing for walk-in and appointment-based patients.

OPD Visit Registration & Patient Flow



OPD Configuration & Rate Management



Doctor Schedule & Time Slot Management

OpdTimeSlot Model

Primary Key: slot_id

Attributes:

- doctor_id (FK to Doctor)
- department_id (FK to Department)
- day_of_week (monday/tuesday/.../sunday)
- start_time, end_time
- slot_duration_minutes (e.g., 15, 20, 30)
- max_patients_per_slot
- max_patients_per_session
- is_active (boolean)

Weekly Schedule Example

MonTueWedThuFriSatSun

Morning (09:00-13:00)

09-13 20 slots	09-13 20 slots	OFF	09-13 20 slots	09-12 15 slots	10-13 15 slots	OFF
-------------------	-------------------	-----	-------------------	-------------------	-------------------	-----

Evening (17:00-20:00)

17-20 15 slots	OFF	17-20 15 slots	17-20 15 slots	17-19 10 slots	OFF	OFF
-------------------	-----	-------------------	-------------------	-------------------	-----	-----

Morning

Evening

Off

Slot Duration: 15 min

Slot Generation Methods

getAvailableSlots(\$doctorId, \$date)

- Get day_of_week from date
- Fetch OpdTimeSlot records for doctor + day
- Generate time slots based on duration
- Check Appointment count for each slot
- Return slots with available capacity

generateSlots() Instance Method

- Parse start_time and end_time
- Calculate slot intervals
- Generate slot_number sequence
- Return array of time ranges

Queue & Token Display

Token #1
Waiting
09:00

Token #2
In Consult
09:15

Token #3
Completed
09:30

OpdVisit Status Flow

registered

waiting

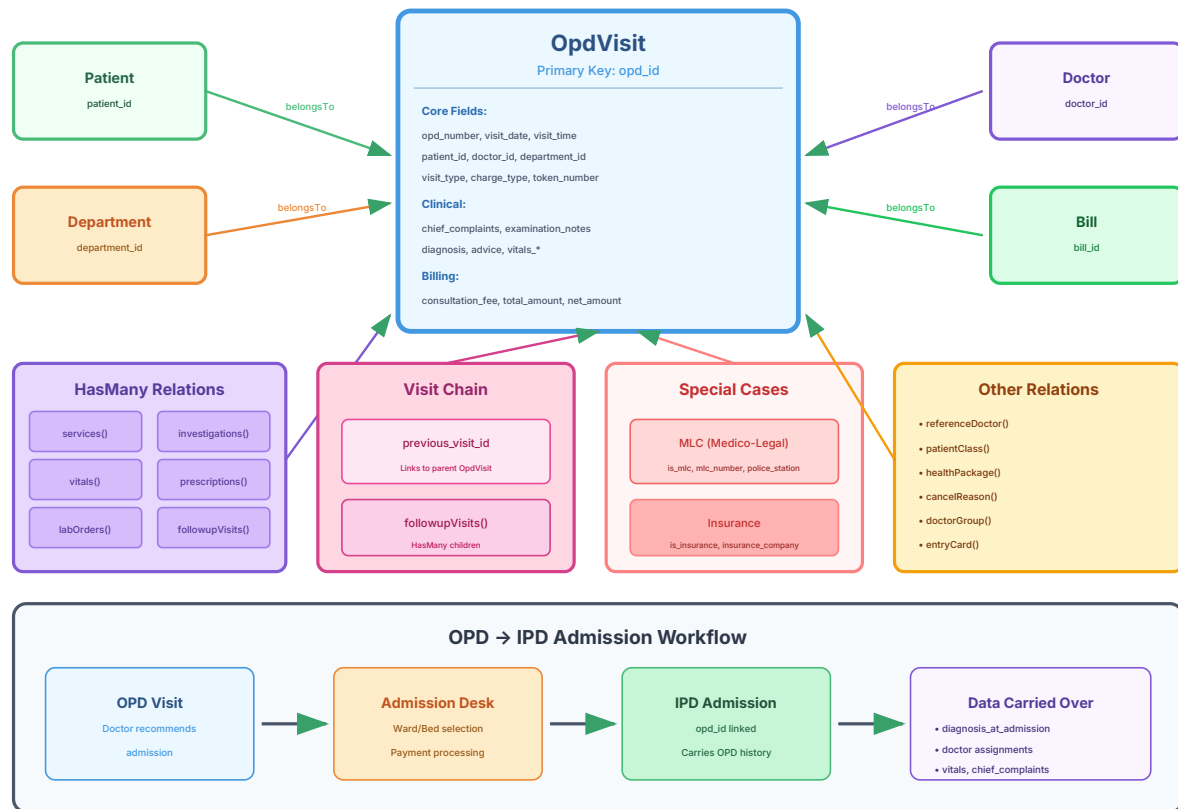
in_consult

cancelled

✓

• cancel_reason_id tracks cancellation reason

OPD Visit Model & Relationships

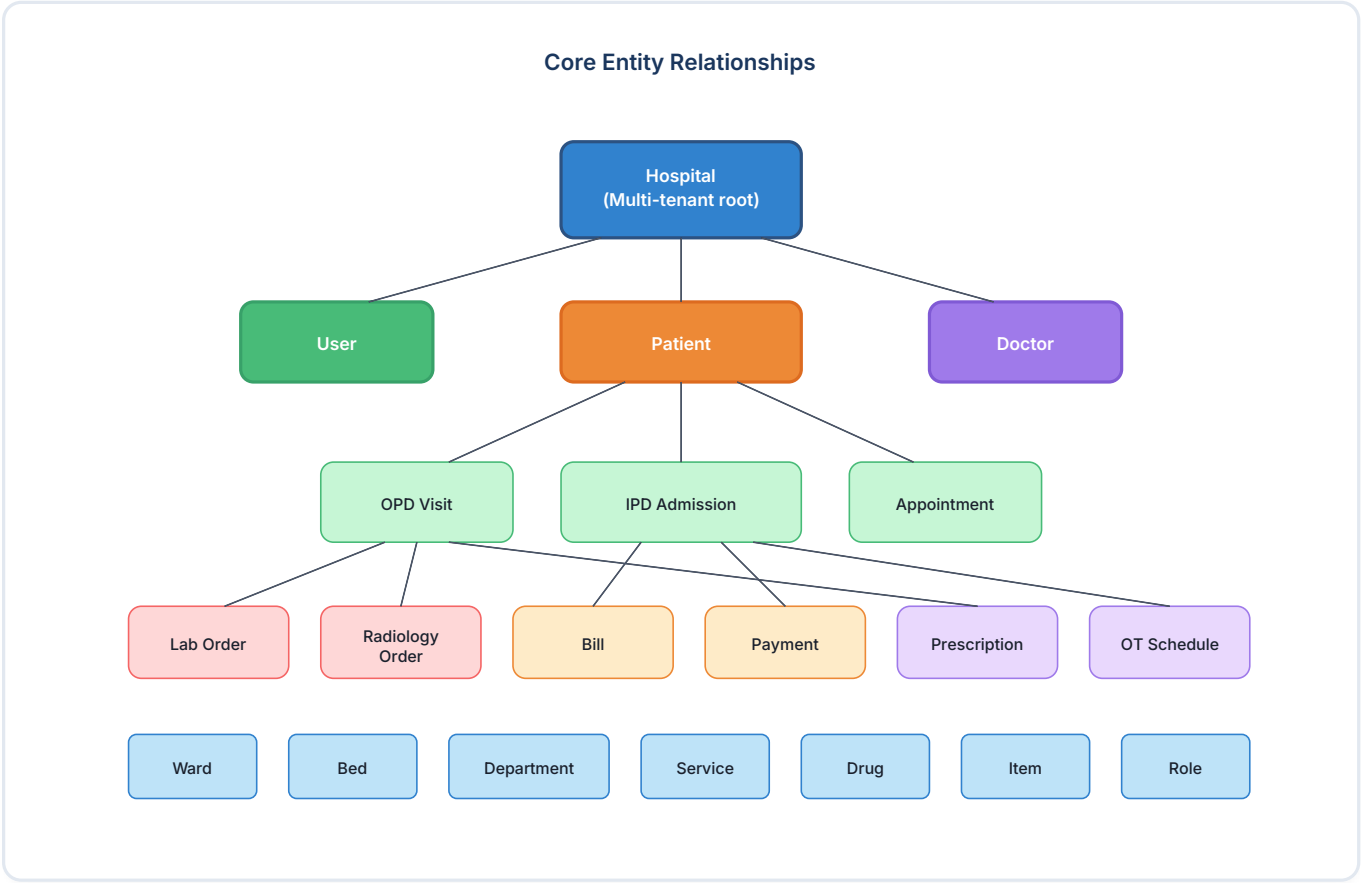


OPD Module Workflow Key Points

- **Visit Types:** OpdVisit supports new, follow-up, and review visits with charge_type variations (normal/emergency/VIP)
- **Token System:** Configurable token generation per doctor or department via OpdConfiguration; auto-incrementing token_number
- **Rate Management:** DoctorOpdRate enables doctor-specific and class-specific consultation fees with date-based validity
- **Time Slots:** OpdTimeSlot defines doctor schedules per day with slot_duration_minutes and max_patients_per_slot
- **Clinical Documentation:** Vitals captured inline (vitals_*) or via PatientVital model; chief_complaints, examination_notes, diagnosis, advice
- **Prescriptions:** Linked via prescriptions() HasMany relationship with PrescriptionItem for individual medications
- **Investigations:** OpdInvestigation tracks lab/radiology orders with priority levels; links to LabOrder via labOrders()
- **Follow-up Chain:** previous_visit_id creates visit chains; is_free_followup based on OpdConfiguration.default_free_followup_days
- **Entry Cards:** Optional entry_card_id with configurable validity (one_time to lifetime) via OpdConfiguration
- **IPD Conversion:** OPD visits can be converted to IPD admissions with opd_id linkage for continuity of care

5. Database Schema

5.1 Entity Relationship Overview



5.2 Core Tables Summary

Core	hospitals, users, roles, permissions	Multi-tenant and authentication
Patient	patients, patient_classes, patient_vitals, patient_documents	Patient registry and demographics
Clinical	opd_visits, ipd_admissions, appointments, prescriptions	Clinical transactions
Doctor	doctors, doctor_groups, doctor_opd_rates, skill_sets	Physician management
Laboratory	lab_categories, lab_tests, lab_orders, lab_order_details	Lab operations
Radiology	radiology_modalities, radiology_tests, radiology_orders, radiology_reports	Imaging operations
OT	operation_theaters, ot_schedules, ot_procedures, surgery_types	Surgery management
Pharmacy	drugs, drug_batches, drug_categories, pharmacy_sales	Pharmacy operations
Inventory	items, stores, suppliers, purchase_orders, indents, grn	Stock management
Billing	services, bills, bill_details, payments, payment_modes	Revenue cycle

IPD	wards, beds, ipd_progress_notes, ipd_nursing_charts, ipd_medications	Inpatient care
Registration	birth_registrations, death_registrations	Vital events
MRD	medical_record_requests, mrd_file_movements, icd_codes	Medical records
Integration	abha_registrations, fhir_resources, hl7_messages	Interoperability

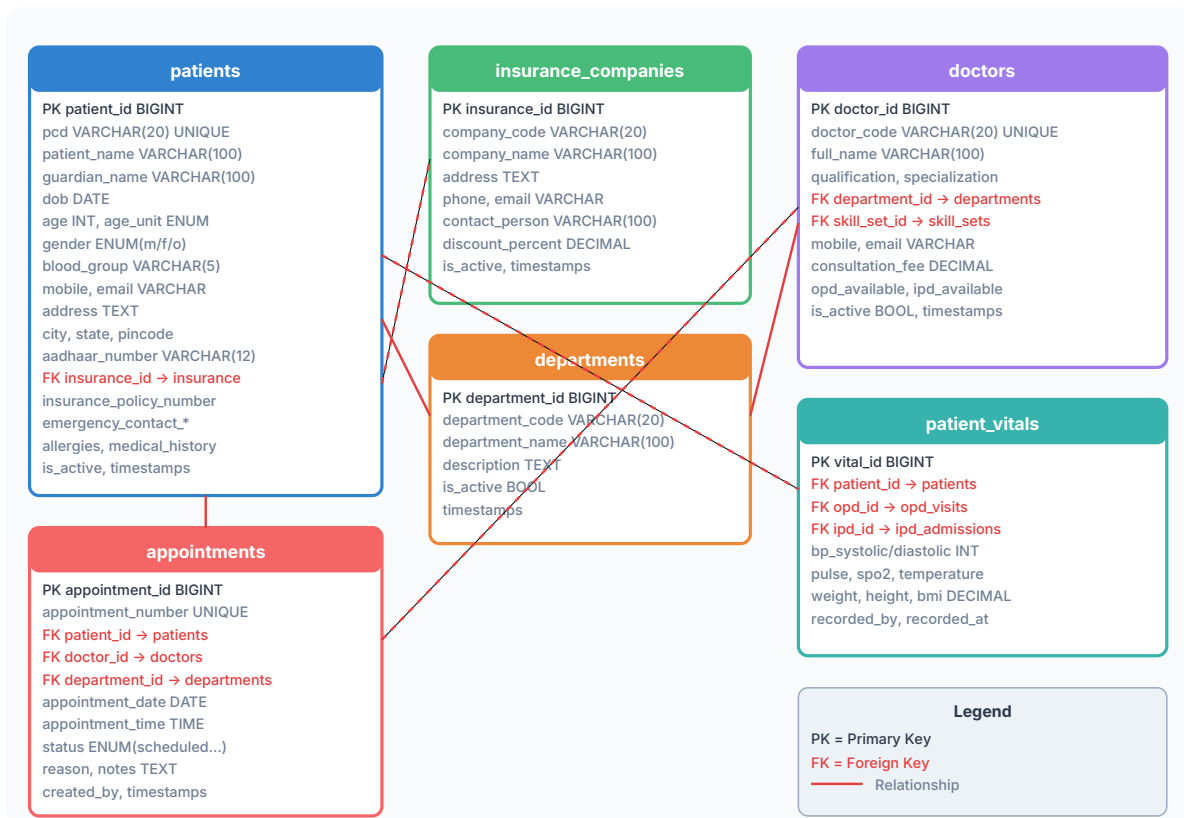
5.3 Migration Files

create_users_table	-	users, sessions, password_reset_tokens
create_hospitals_table	-	hospitals
create_hims_tables	34.5 KB	Core HIMMS entities (30+ tables)
create_opd_masters_tables	20.3 KB	OPD master data
create_opd_transaction_tables	10.7 KB	OPD transactions
create_rbac_tables	-	roles, permissions, pivots
create_notification_tables	5.6 KB	SMS gateways, templates, logs
create_radiology_tables	9 KB	Radiology module
create_ot_tables	14.1 KB	Operation theater
create_inventory_tables	18.4 KB	Inventory management
create_birth_death_tables	11 KB	Vital registration
create_mrd_tables	9.7 KB	Medical records
create_patient_portal_tables	7.3 KB	Patient portal
create_abha_tables	6.8 KB	ABHA integration
create_fhir_tables	6.2 KB	FHIR/HL7

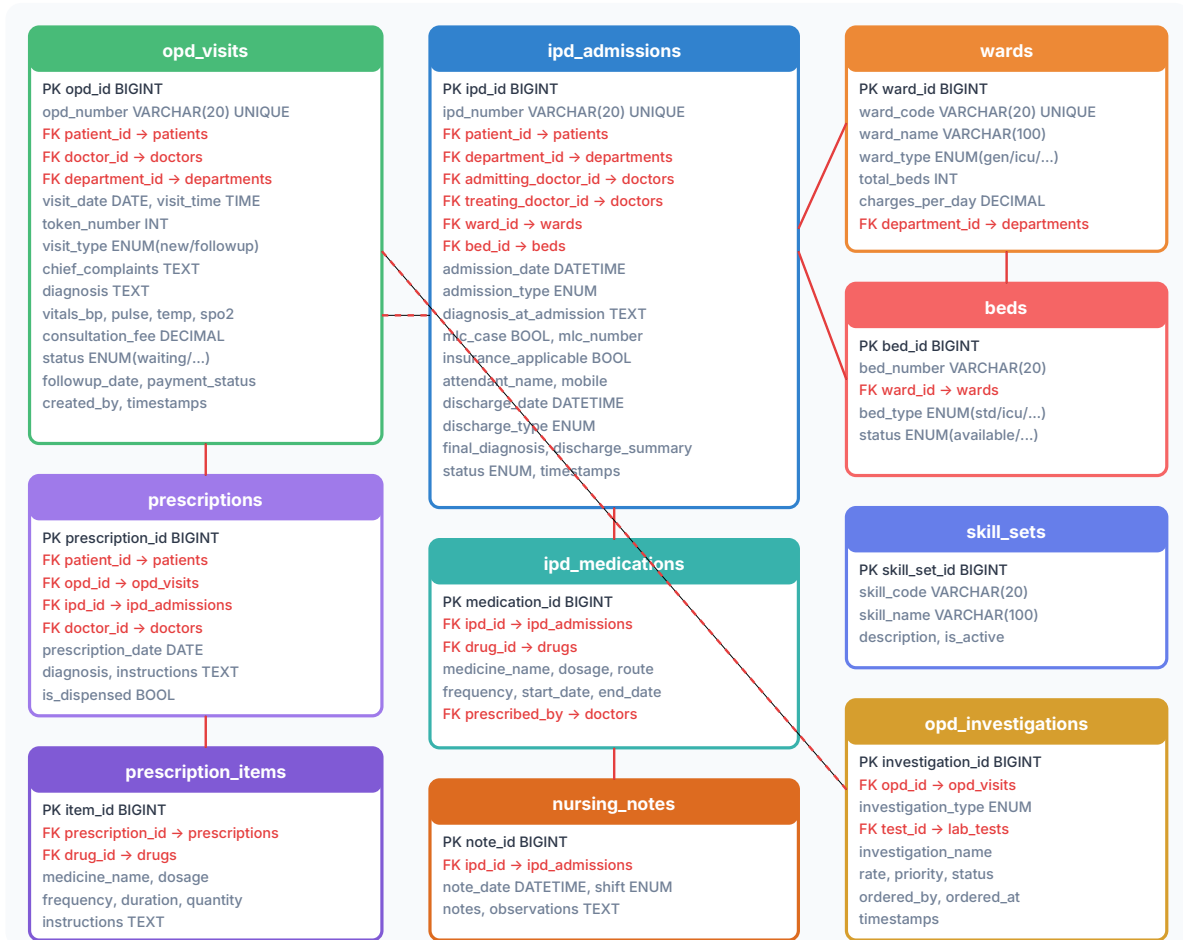
5.4 Detailed Entity Relationship Diagrams

The following diagrams illustrate the detailed database schema with table structures, primary keys, foreign keys, and relationships.

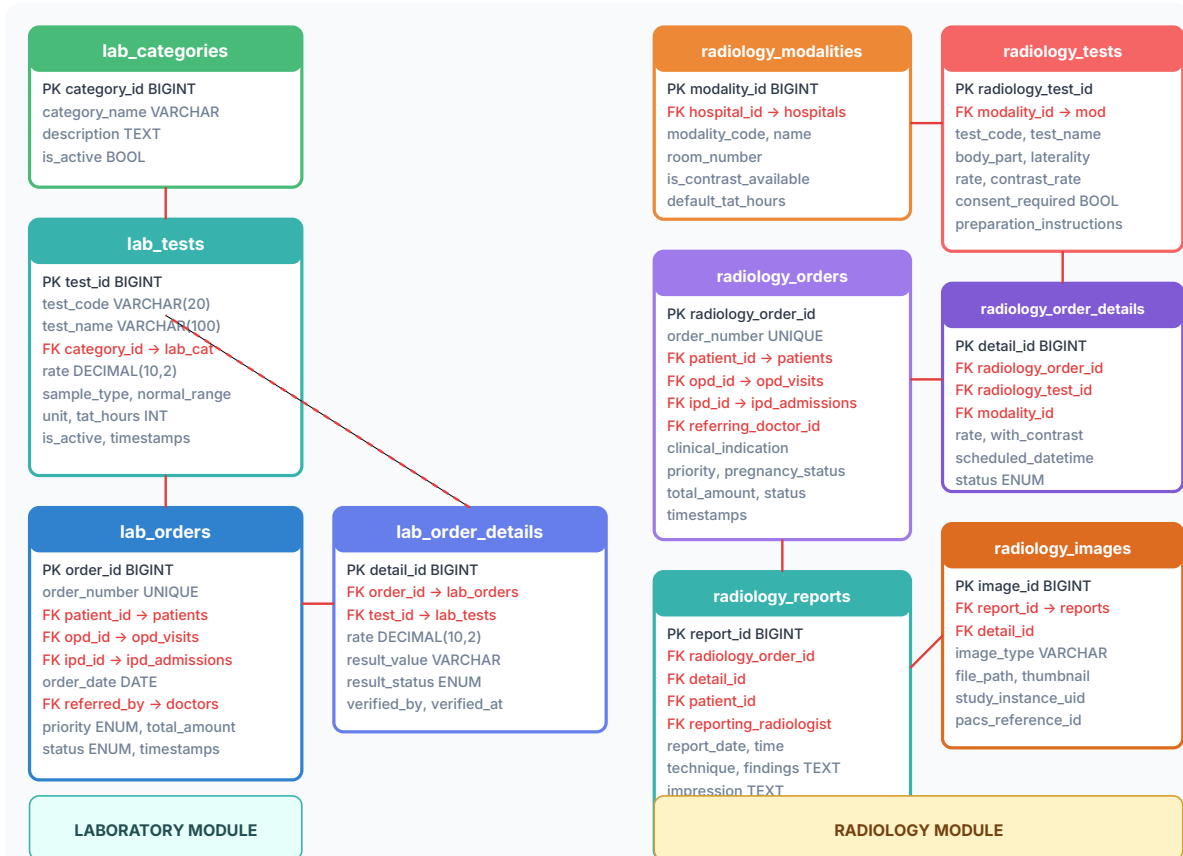
Patient Management Schema



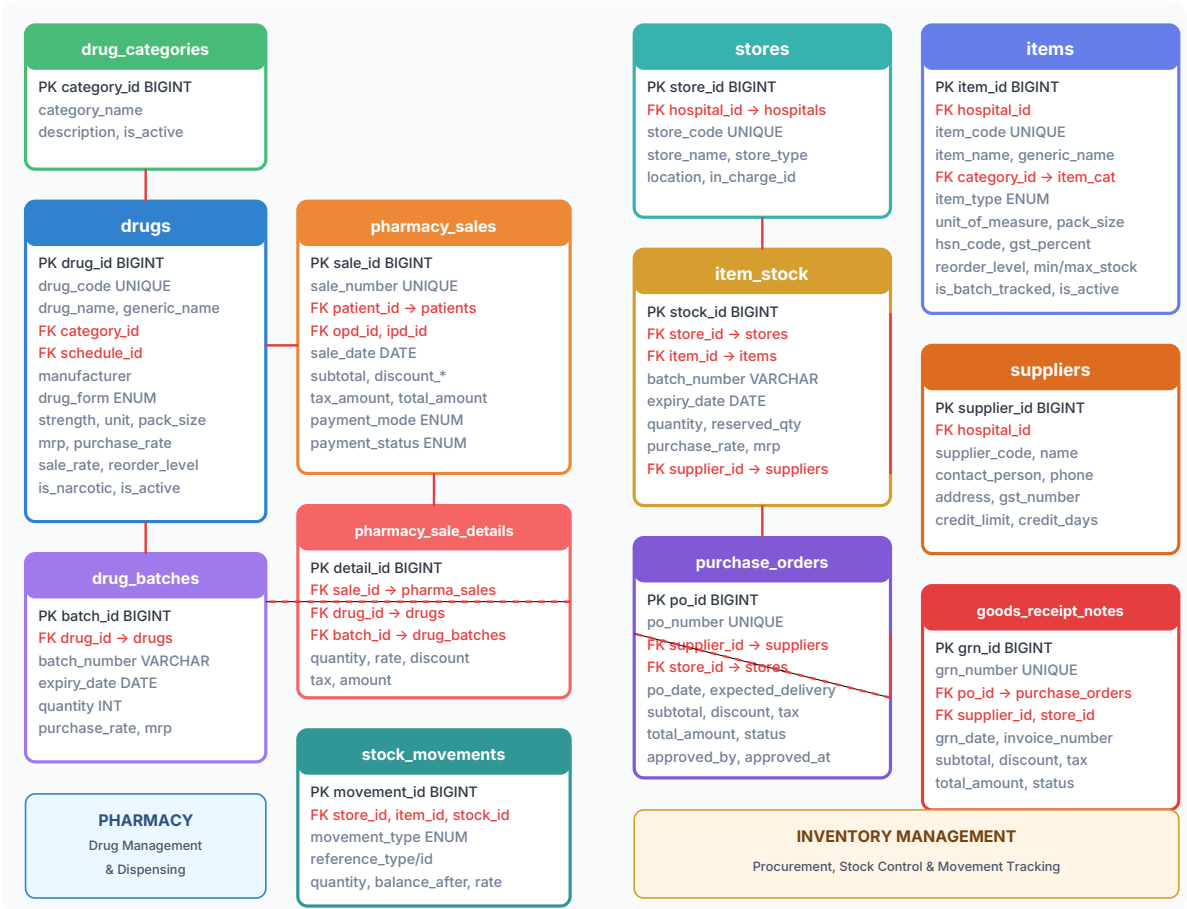
Clinical Operations Schema (OPD & IPD)



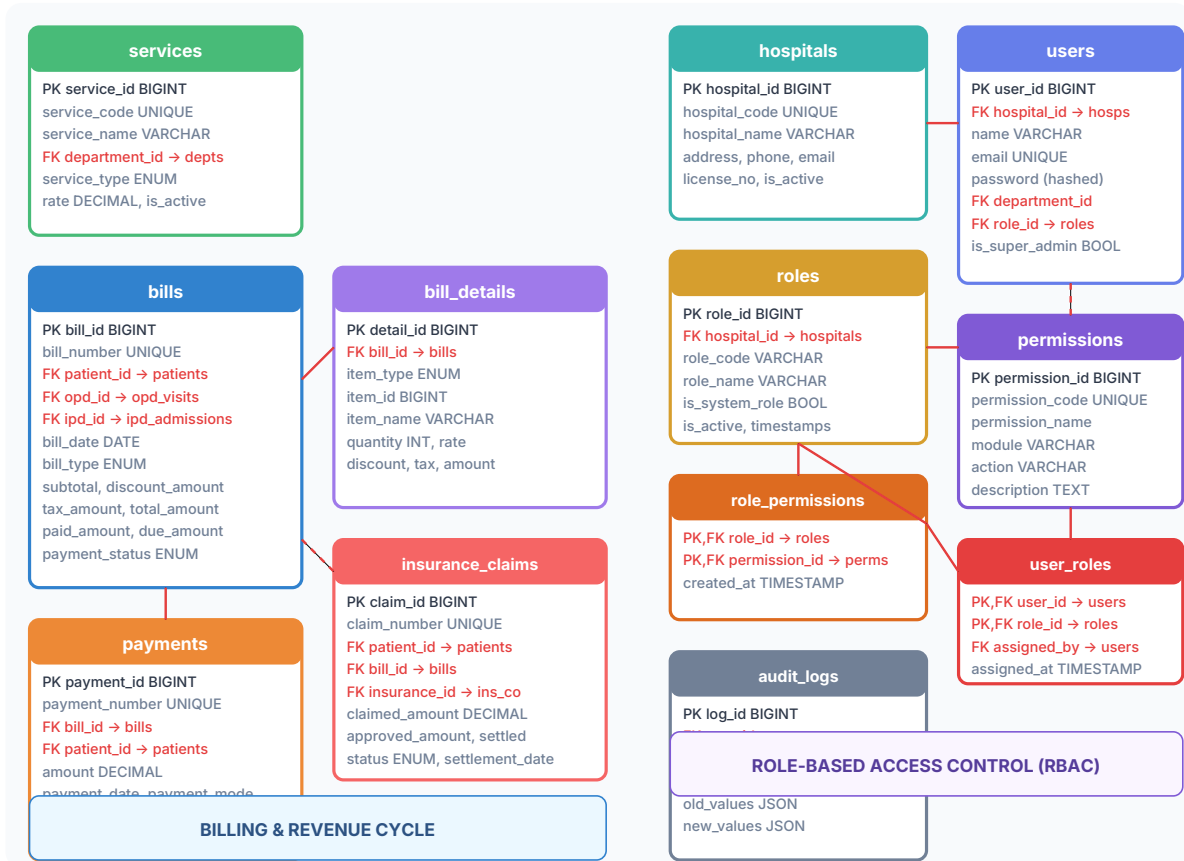
Laboratory & Radiology Schema



Pharmacy & Inventory Schema



Billing & Access Control Schema



5.5 Key Relationships Summary

Entity	Related Entities	Relationship Type	Primary Key
hospitals	users, patients, doctors, all modules	One-to-Many	hospital_id
patients	appointments, opd_visits, ipd_admissions, bills	One-to-Many	patient_id
doctors	appointments, opd_visits, prescriptions	One-to-Many	doctor_id
opd_visits	prescriptions, lab_orders, radiology_orders	One-to-Many	opd_id
ipd_admissions	ipd_medications, nursing_notes, bills	One-to-Many	ipd_id
lab_orders	lab_order_details	One-to-Many	order_id
radiology_orders	radiology_order_details, radiology_reports	One-to-Many	radiology_order_id
bills	bill_details, payments	One-to-Many	bill_id
roles	role_permissions, user_roles	Many-to-Many	role_id
stores	item_stock, stock_movements, transfers	One-to-Many	store_id
wards	beds	One-to-Many	ward_id
drugs	drug_batches, prescription_items	One-to-Many	drug_id

Multi-Tenancy Implementation

All major tables include a `hospital_id` foreign key that references the `hospitals` table. This enables:

- Complete data isolation between hospitals
- Single database shared across all tenants
- Global scopes automatically filter queries by current hospital
- Super admin users can switch between hospitals

6. API Architecture

6.1 Authentication

HIMS uses Laravel Sanctum for API token authentication with two separate guards:

sanctum	Staff (User model)	Authorization: Bearer {token}	Admin, doctors, nurses, staff
patient	Patient (PatientUser model)	Authorization: Bearer {token}	Patient portal access

Login Request Example

```
POST /api/login
Content-Type: application/json

{
  "email": "admin@hospital.com",
  "password": "password123"
}

Response:
{
  "success": true,
  "token": "1|abc123xyz...",
  "user": {
    "id": 1,
    "name": "Admin User",
    "email": "admin@hospital.com",
    "current_hospital_id": 1
  }
}
```

6.2 API Endpoint Summary

Authentication	/api	5	login, logout, user, change-password, switch-hospital
Hospitals	/api/hospitals	6	CRUD + stats
Patients	/api/patients	12	CRUD + history, search, VIP, documents
Appointments	/api/appointments	15	CRUD + confirm, check-in, cancel, reschedule
OPD	/api/opd-visits	12	CRUD + consultation workflow
IPD	/api/ipd-admissions	23	CRUD + case sheet operations
Laboratory	/api/lab-*	8	Categories, tests, orders, results
Radiology	/api/radiology/*	16	Full radiology workflow
OT	/api/ot/*	16	Scheduling, procedures

Pharmacy	/api/drug-*	8	Drugs, batches, sales
Inventory	/api/inventory/*	18	Full procurement cycle
Billing	/api/services, bills	10	Services, billing, payments
RBAC	/api/roles	7	Roles, permissions
Notifications	/api/notifications	8	Gateways, templates, send
Patient Portal	/api/patient-portal	15	Self-service features
FHIR	/api/fhir/r4	7	FHIR R4 resources
ABHA	/api/abha	9	Health ID integration

6.3 Response Formats

Success Response

```
{
  "success": true,
  "data": { ... },
  "message": "Operation successful"
}
```

Paginated Response

```
{
  "data": [ ... ],
  "links": {
    "first": "https://api/resource?page=1",
    "last": "https://api/resource?page=10",
    "prev": null,
    "next": "https://api/resource?page=2"
  },
  "meta": {
    "current_page": 1,
    "last_page": 10,
    "per_page": 15,
    "total": 150
  }
}
```

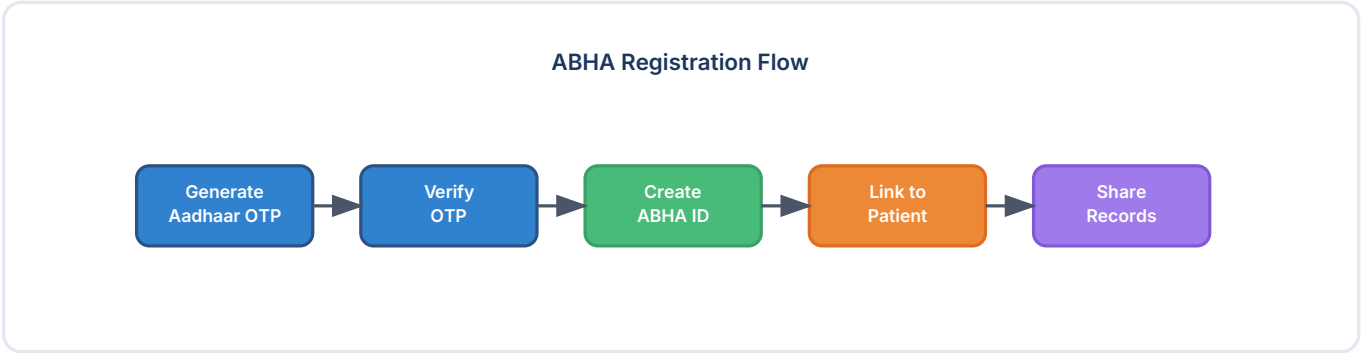
Validation Error (422)

```
{
  "success": false,
  "message": "The given data was invalid.",
  "errors": {
    "email": ["The email field is required."],
    "password": ["The password must be at least 8 characters."]
  }
}
```

7. Integration & Interoperability

7.1 ABHA/ABDM Integration (India)

HIMS integrates with India's Ayushman Bharat Digital Mission (ABDM) for health ID management:



7.2 FHIR R4 API

HIMS exposes FHIR R4 compliant endpoints for healthcare interoperability:

CapabilityStatement	/api/fhir/r4/metadata	GET (Server capabilities)
Patient	/api/fhir/r4/Patient	Search, Read
Observation	/api/fhir/r4/Observation	Search (Vitals, Lab results)
DiagnosticReport	/api/fhir/r4/DiagnosticReport	Search (Lab, Radiology)
MedicationRequest	/api/fhir/r4/MedicationRequest	Search (Prescriptions)

7.3 HL7 Messaging

Support for HL7 v2.x message processing for legacy system integration:

- ADT (Admit/Discharge/Transfer) messages
- ORM (Order) messages
- ORU (Results) messages
- Message parsing and generation
- Transaction logging

8. Security & Compliance

8.1 Role-Based Access Control

super_admin	System administrator	All permissions, manage hospitals
hospital_admin	Hospital administrator	All hospital operations, user management
doctor	Physician	OPD, IPD, prescriptions, orders
nurse	Nursing staff	Vitals, nursing charts, medication admin
receptionist	Front desk	Registration, appointments, basic billing
lab_technician	Laboratory staff	Lab orders, sample collection, results
radiologist	Radiology staff	Radiology orders, reports, images
pharmacist	Pharmacy staff	Drug dispensing, sales
billing_staff	Billing department	Bills, payments, refunds
store_manager	Inventory manager	Stock, indents, purchase orders

8.2 Security Features

Authentication

- Token-based API authentication
- Password hashing (bcrypt)
- Session management
- OTP verification (patient portal)

Authorization

- Role-based access control
- Permission-based restrictions
- Hospital scope isolation
- Middleware protection

Data Protection

- HTTPS/TLS encryption
- Database encryption at rest
- Sensitive data masking
- Audit logging

API Security

- Rate limiting
- CORS configuration
- Input validation
- SQL injection prevention

8.3 Compliance Considerations

Healthcare Data Protection:

- Patient data encryption and access controls
- Audit trail for all data modifications
- Consent management for data sharing
- Role-based data visibility
- Secure file storage for medical documents

9. Deployment Guide

9.1 Production Deployment Steps

1. Server Preparation

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install required packages
sudo apt install nginx php8.2-fpm mysql-server nodejs npm -y

# Install PHP extensions
sudo apt install php8.2-{bcmath,ctype,curl,dom,fileinfo,json,mbstring,openssl,pdo,pdo_mysql,tokenizer,xml,zip,gd} -y
```

2. Clone and Configure

```
cd /var/www/html
git clone <repository> hims-laravel
cd hims-laravel
composer install --optimize-autoloader --no-dev
cp .env.example .env
php artisan key:generate
```

3. Database Setup

```
# Create database
mysql -u root -p -e "CREATE DATABASE hims_production;"

# Configure .env with database credentials
# Run migrations
php artisan migrate --force
php artisan db:seed
```

4. Build Frontend

```
npm install
npm run build
```

5. Set Permissions

```
sudo chown -R www-data:www-data /var/www/html/hims-laravel
sudo chmod -R 755 /var/www/html/hims-laravel
sudo chmod -R 775 storage bootstrap/cache
```

6. Optimize for Production

```
php artisan config:cache
php artisan route:cache
php artisan view:cache
php artisan optimize
```

9.2 Environment Configuration

```
# Application
APP_NAME="HIMS"
APP_ENV=production
APP_DEBUG=false
APP_URL=https://hims.example.com

# Database
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=hims_production
DB_USERNAME=hims_user
DB_PASSWORD=secure_password

# Cache & Queue
CACHE_STORE=database
QUEUE_CONNECTION=database
SESSION_DRIVER=database

# Security
SANCTUM_STATEFUL_DOMAINS=hims.example.com
```

10. Maintenance & Operations

10.1 Backup Procedures

```
# Database backup
mysqldump -u hims_user -p hims_production | gzip > backup_$(date +%Y%m%d).sql.gz

# Files backup
tar -czf files_backup_$(date +%Y%m%d).tar.gz /var/www/html/hims-laravel/storage
```

10.2 Monitoring

Laravel Log	storage/logs/laravel.log	Application errors and debug
Nginx Access	/var/log/nginx/access.log	HTTP request logging
Nginx Error	/var/log/nginx/error.log	Server errors
MySQL Error	/var/log/mysql/error.log	Database errors

10.3 Common Maintenance Commands

```
# Clear all caches
php artisan optimize:clear

# Rebuild caches
php artisan optimize

# Check queue status
php artisan queue:failed

# Retry failed jobs
php artisan queue:retry all

# Run migrations
php artisan migrate --force

# View routes
php artisan route:list
```

10.4 Update Procedure


```
# 1. Enable maintenance mode
php artisan down --message="System upgrade in progress"

# 2. Backup database
mysqldump -u hims_user -p hims_production > pre_update_backup.sql

# 3. Pull latest code
git pull origin main

# 4. Update dependencies
composer install --optimize-autoloader --no-dev
npm install && npm run build

# 5. Run migrations
php artisan migrate --force

# 6. Clear and rebuild caches
php artisan optimize:clear
php artisan optimize

# 7. Restart services
sudo systemctl restart php8.2-fpm
sudo systemctl restart hims-queue

# 8. Disable maintenance mode
php artisan up
```

11. Troubleshooting

11.1 Common Issues

500 Internal Server Error	Permission issues, config error	Check storage permissions, run php artisan config:clear
401 Unauthorized	Invalid or expired token	Re-authenticate, check token in request header
Database connection error	Wrong credentials, MySQL down	Verify .env settings, check MySQL status
Queue not processing	Worker stopped	Restart queue worker service
Session issues	Cookie/session config	Clear sessions, check SANCTUM_STATEFUL_DOMAINS
Slow performance	Missing indexes, N+1 queries	Enable query logging, add database indexes

11.2 Debug Commands

```
# Check Laravel log
tail -f storage/logs/laravel.log

# Test database connection
php artisan tinker
>>> DB::connection()->getPdo();

# Check PHP configuration
php -i | grep memory_limit

# Test specific route
php artisan route:list --path=patients

# Check queue worker
sudo systemctl status hims-queue
```

Appendices

A. API Endpoint Quick Reference

Authentication	POST /login, /logout, /change-password, /switch-hospital
Patients	GET/POST /patients, /patients/{id}, /patients-search
Appointments	GET/POST /appointments, /appointments/{id}/confirm, /cancel
OPD	GET/POST /opd-visits, /start-consultation, /complete-consultation
IPD	GET/POST /ipd-admissions, /progress-notes, /nursing-charts
Lab	GET/POST /lab-orders, /lab-tests, /lab-categories
Radiology	GET/POST /radiology/orders, /reports, /modalities
OT	GET/POST /ot/schedules, /procedures, /theaters
Billing	GET/POST /services, /bills, /payments
Inventory	GET/POST /inventory/items, /indents, /purchase-orders

B. Environment Variables Reference

APP_NAME	Application name	HIMS
APP_ENV	Environment (local/production)	production
APP_DEBUG	Debug mode	false
DB_CONNECTION	Database driver	mysql
CACHE_STORE	Cache driver	database
QUEUE_CONNECTION	Queue driver	database
SESSION_DRIVER	Session driver	database

C. Contact & Support

Technical Support: support@hims.example.com
API Documentation: https://hims.example.com/api-docs/
System Status: https://status.hims.example.com

HIMS - Hospital Information Management System

Version 1.0.0 | January 2026

This document is confidential and intended for authorized personnel only.