

# HOSPITAL MANAGEMENT SYSTEM: A UNIFIED SAAS PLATFORM WITH AI-POWERED INSURANCE DOCUMENT AUTO-CATEGORIZATION

<sup>1</sup>Madhura Narkhede, <sup>2</sup>Priyanka Kaspate, <sup>3</sup>Amol Marathe, <sup>4</sup>Shubham Padhi,

<sup>5</sup>Kanchan Bhale

<sup>1,2,3,4</sup>UG Student, <sup>5</sup>Professor

<sup>1,2,3,4,5</sup>Dept. of Information Technology

International Institute of Information Technology (IIT), Pune, India

**Abstract** : Hospital operations across India continue to suffer from fragmented software ecosystems where clinical, administrative, and financial functions operate in separate, non-communicating systems. This paper presents the Hospital Management System (HMS), a web-based Software-as-a-Service platform that consolidates these functions under a single governance layer serving eight distinct stakeholder roles: Super Admin, Hospital Admin, Doctor, Receptionist, Nurse, Patient, Chemist, and Lab Technician. Role-scoped permissions are enforced throughout via JSON Web Token-based Role-Based Access Control. The principal technical contribution is an AI-driven Insurance Module that applies Optical Character Recognition and Natural Language Processing to uploaded insurance documents, extracting structured fields, classifying claim types, and tracking each claim from initial draft through to final settlement. The platform is built on React.js, Next.js, Node.js, Express.js, MongoDB Atlas, and React Native for Android. A systematic comparative review against published research literature and active Indian hospital management products confirms that no existing offering unifies all eight roles within a SaaS architecture while embedding claim automation. Functional testing validated role-scoped access, real-time scheduling, end-to-end billing, and accurate AI-assisted insurance processing.

**Index Terms** - Hospital Management System, Role-Based Access Control, SaaS, Electronic Medical Records, OCR, NLP, Insurance Automation, React Native, MongoDB, JWT

## I. INTRODUCTION

Healthcare delivery in India spans a remarkably varied landscape, ranging from small urban clinics to large multi-specialty chains, each carrying significant administrative burden alongside clinical responsibilities. Despite this scale, a substantial number of institutions still depend on paper registers, telephone-driven appointment coordination, and departmental software that cannot share data with adjacent systems. The predictable consequences include duplicate patient records, delayed clinical handoffs, billing errors, and insurance claims that fail due to incomplete documentation [1].

A review of the commercial hospital management software market in India reveals a consistent pattern of fragmentation. Appointment-focused platforms such as Practo and Apollo 24|7 have refined the patient-facing experience but provide limited support for hospital-side staff management. Backend systems like MocDoc HMS and eHospital Systems address multi-department workflows but leave insurance document processing entirely to manual handling. No single reviewed product supports all eight clinical and administrative roles within one unified platform while simultaneously embedding insurance claim automation [2][3][4][5][6].

The HMS was designed specifically to close this gap. It follows a multi-tenant SaaS architecture in which a Super Admin provisions hospital accounts and controls module access per institution according to subscription tier. Clinical and administrative staff interact through dedicated web portals; nurses and patients use React Native Android applications. Detailed OPD and IPD patient flow diagrams, presented in Section IV, illustrate how the system orchestrates end-to-end patient journeys across departments without manual data re-entry at each handoff point.

## II. LITERATURE SURVEY

### A. Review of Related Research Papers

A structured search of peer-reviewed digital libraries identified six papers with direct relevance to the design problems addressed by this work. Table I summarizes each paper's key features, limitations, and relevance to the HMS.

TABLE I. Review of Related Research Papers

Ref.	Title	Authors	Key Features	Limitations	Relevance to HMS
[1]	Web Based Hospital Management System	A. C. Babu et al.	Three-role system (patient, doctor, admin); online appointment booking.	Three roles only; no nurse, chemist, lab, or insurance module.	Validates HMS concept. Our system extends to 8 roles with JWT-RBAC.

[2]	User Interactive HMS via Web Application	P. Sreeja et al.	Patient, doctor, admin DB management; appointment booking via web.	No AI, no pharmacy/lab, no SaaS, no insurance automation.	Confirms patient-facing interface value. We add mobile apps and billing.
[3]	HMS Based on IoT	Md Meraj Ahmed et al.	IoT real-time patient monitoring via sensors.	Hardware-only; no web management, billing, or insurance.	Highlights real-time data value. HMS achieves this via MongoDB REST APIs.
[4]	Mobile Application for HMS	A. Ananth et al.	Mobile-first HMS with chatbot and Big Data analytics.	Patient-side only; no admin portals or multi-role access.	Validates mobile engagement. Our React Native app adds Razorpay.
[5]	Automating Insurance Claims with AI	Y. Z. Liu et al.	AI claims automation using ML and NLP classification.	Not integrated within an HMS; no OCR at hospital level.	Directly informs our AI Insurance Module with embedded OCR+NLP.
[6]	Optimal Appointment Rule in OPD	Jie Song et al.	AI-based outpatient scheduling and wait-time analysis.	Scheduling algorithms only; no billing or multi-role integration.	Informs our slot-based scheduling with real-time availability management.

### B. Competitive Analysis of Active Platforms in India

In parallel, active hospital management products operating in India were analysed to identify practical gaps in the commercial landscape. Table II presents the findings.

**TABLE II. Competitive Analysis of Hospital Management Platforms in India**

Sr.	Product	Key Features Offered	Critical Gaps Identified
1	<b>Practo</b>	Doctor appointments, telemedicine, e-prescriptions, clinic management, medicine delivery, lab tests.	No AI insurance auto-categorization; no nurse, chemist, or lab portals; no AI claim workflow.
2	<b>Apollo 24 7</b>	Online consultations, health records, medicine delivery, diagnostics, chronic disease management.	Patient-facing only; no admin, receptionist, nurse, or chemist dashboards; no AI insurance module.
3	<b>MocDoc HMS</b>	Cloud HMS with scheduling, EMR, pharmacy, basic lab management, billing, ABDM compliance.	No AI insurance categorization; no SaaS super-admin for multi-hospital governance; no OCR/NLP.
4	<b>HealthPlix</b>	EMR with digital prescriptions, patient engagement tools, analytics, basic clinic management.	Clinic-side only; no insurance module, chemist dashboard, lab management, or super-admin governance.
5	<b>eHospital Systems</b>	Comprehensive HMS: OPD/IPD modules, pharmacy, lab, billing, HR, multi-branch support.	No AI insurance module; no OCR/NLP; no nurse/patient mobile app with integrated payment gateway.

### C. Identified Gaps and Novel Contributions

Synthesizing both the literature survey and competitive analysis reveals a consistent pattern: existing systems address hospital management in isolated segments. Web-based academic prototypes validated the three-role administrative model but did not extend to clinical support roles. The AI insurance study demonstrated intelligent claims processing but in complete isolation from any clinical workflow. Commercial platforms have refined either the patient-facing experience or the hospital-side backend, but none achieved both simultaneously while embedding AI automation. The HMS contributes five novel capabilities: (1) a unified eight-role SaaS platform with JWT-based RBAC; (2) an AI-powered insurance document auto-categorization and end-to-end claim workflow engine embedded within the HMS; (3) a Super Admin governance layer for per-hospital module configuration; (4) dedicated React Native mobile applications for patients and nurses; and (5) fully automated OPD and IPD billing with Razorpay integration and downloadable PDF receipts linked to prescriptions and lab reports.

## III. METHODOLOGY AND SYSTEM ARCHITECTURE

### A. Architectural Overview

The HMS follows a three-tier architecture comprising a client layer, a backend API server, and a data and storage layer. The client layer encompasses a web application serving the Super Admin, Hospital Admin, Doctor, Receptionist, Lab Technician, and Chemist roles, along with two React Native Android applications for nurses and patients. All client-to-server communication occurs over HTTPS.

The backend is built on Node.js and Express.js and exposes RESTful API endpoints for authentication, role verification, business logic execution, and inter-module data orchestration. JWT tokens carry role claims, enabling middleware to perform granular permission checks on every incoming request. The data layer uses MongoDB Atlas for all structured application data, Cloudinary for unstructured medical media, and Razorpay for payment transactions. The SaaS governance structure is enabled by a hierarchical administrative model: the Super Admin provisions hospital accounts and toggles individual functional modules per institution, while each Hospital Admin manages their own staff and operations in a fully isolated tenant environment.

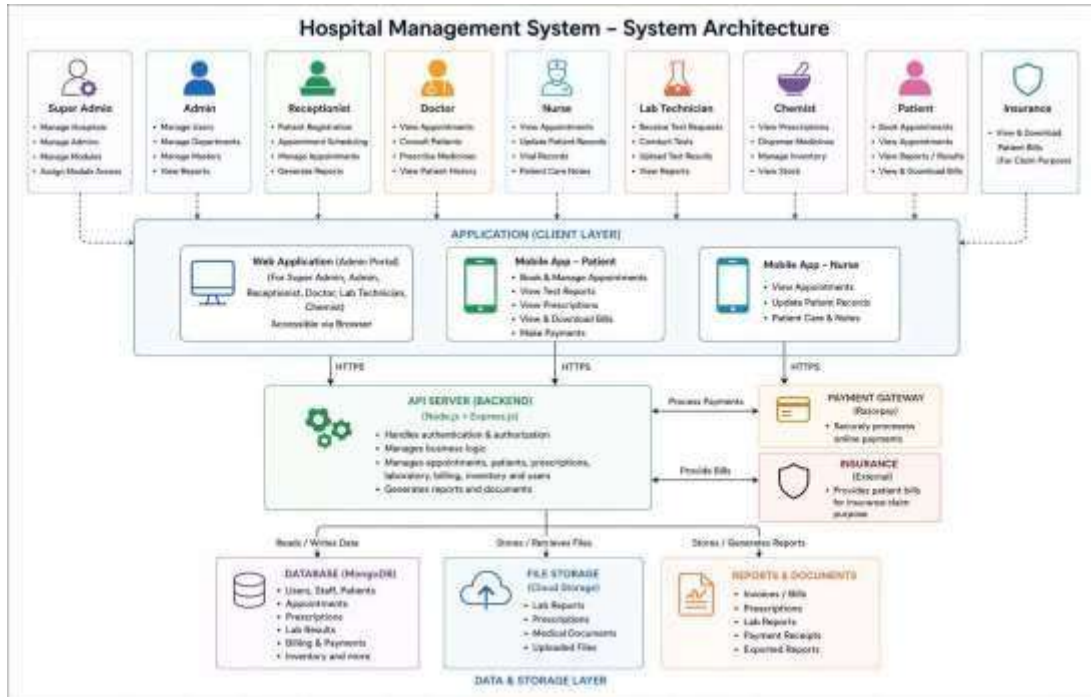


Fig. 1. Hospital management system architecture

#### IV. IMPLEMENTATION OF MODULES AND PATIENT FLOW DIAGRAMS

##### A. OPD Patient Flow

The Outpatient Department (OPD) serves as the first point of contact for patients seeking consultation and treatment without requiring hospital admission. The workflow is designed to streamline communication between different hospital staff members while minimizing waiting time and ensuring that patient information is available whenever needed.

1. When a patient visits the hospital, either through a pre-booked appointment or as a walk-in, the receptionist accesses the registration module through the web portal. For new patients, the system records personal details, contact information, and basic medical history. For existing patients, the receptionist can quickly retrieve records using the patient ID or registered mobile number, eliminating the need to enter the same information repeatedly.

2. After registration, the receptionist selects the required department and preferred doctor. The appointment scheduling module checks doctor availability using shift information stored in MongoDB and displays suitable time slots. Once an appointment is confirmed, the patient receives an automatic confirmation email through the SMTP service. If the selected doctor is unavailable, the system suggests the next available slot or recommends another doctor from the same department to ensure the patient can still receive timely consultation.

3. Before the consultation begins, nursing staff use the React Native mobile application to record the patient's vital signs, including blood pressure, temperature, weight, pulse rate, and oxygen saturation (SpO<sub>2</sub>). The recorded information is instantly synchronized with MongoDB and becomes available to the consulting doctor. This mobile-based approach allows nurses to enter data directly at the point of care, reducing dependency on shared workstations and improving the efficiency of patient assessment.

4. At the time of consultation, the doctor can access the patient's complete medical profile through the portal, including previous diagnoses, prescriptions, treatment history, and uploaded laboratory reports. Based on the examination, the doctor records the diagnosis, treatment plan, follow-up recommendations, and any required investigations. If laboratory tests are prescribed, the

requests are immediately forwarded to the Lab Technician Module, where technicians can process samples and upload results. Once the reports are available, notifications are automatically sent to both the doctor and the assigned nurse.

5. Prescriptions are generated using a predefined medicine catalogue maintained by the Hospital Admin. The doctor selects the required medicines, specifies dosage instructions and treatment duration, and adds any necessary clinical notes. This simplifies prescription creation and reduces manual effort while ensuring consistency in medication records.

6. After the prescription is saved, it is automatically forwarded to the Chemist Module. The chemist can view incoming prescriptions, verify medicine availability, and check current stock levels. If a prescribed medicine is unavailable, the system highlights the issue and suggests suitable alternatives, allowing the dispensing process to continue without unnecessary delays.

7. Once the consultation is completed, the receptionist accesses the billing module, where consultation charges are automatically calculated. Payments can be made through multiple methods, including cash, card, UPI, or Razorpay online payment services. After payment is received, the system generates a PDF receipt and sends it to the patient's registered email address. The appointment status is then updated across all connected modules. Patients wishing to submit an insurance claim can use the AI Insurance Module, where uploaded documents are processed using OCR and NLP techniques to extract relevant information. The system compiles supporting documents such as bills, prescriptions, and laboratory reports into a claim package and tracks its progress from submission to final approval or rejection. Figure 2 illustrates the complete OPD workflow.

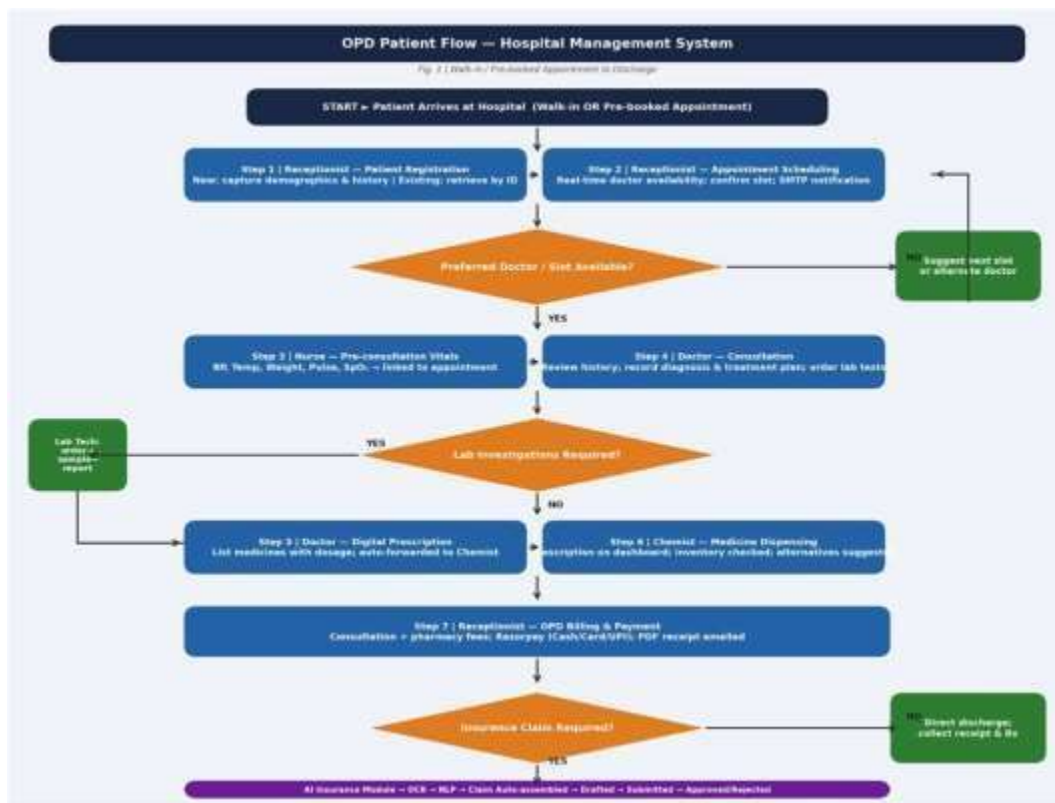


Fig. 2. OPD patient flow in HMS (walk-in to discharge with AI insurance processing)

## B. IPD Patient Flow

The Inpatient Department (IPD) is responsible for managing patients who require admission for treatment, surgery, observation, or specialized care. Since patients remain under hospital supervision for an extended period, the workflow involves continuous coordination between administrative staff, doctors, and nurses to ensure effective treatment and monitoring.

1. Admission can begin in two different ways. A patient may arrive directly through the emergency department, or a doctor may recommend hospitalization after an OPD consultation. In either case, the Hospital Admin completes the admission process through the web portal by recording the patient's personal information, emergency contact details, reason for admission, and admission category such as Elective, Emergency, or Post-OPD.

2. Once the admission is confirmed, the Hospital Admin accesses the bed management dashboard to view the current availability of beds across wards and departments. Based on the patient's medical condition and accommodation requirements, a suitable bed is assigned. As soon as the allocation is completed, the assigned ward nurse receives a notification through the mobile application, allowing the ward to prepare for the patient's arrival.

3. After the patient is settled in the ward, the attending doctor creates a treatment plan using the IPD module. The plan includes the preliminary diagnosis, medication schedule, laboratory investigations, dietary recommendations, and nursing instructions. All updates are immediately reflected on the nurse's mobile dashboard, ensuring that the care team always has access to the latest treatment information and can follow the doctor's instructions without delay.
4. Patient monitoring continues throughout the entire admission period. During every shift, nurses use the mobile application to record vital signs such as blood pressure, temperature, pulse rate, oxygen saturation (SpO<sub>2</sub>), and blood glucose levels when required. Information related to medicine administration, meals, and completed procedures is also documented, creating a complete and up-to-date record of the patient's daily care activities.
5. If any vital reading falls outside the normal range or an emergency situation arises, the nurse can raise an alert directly through the system. The notification is instantly delivered to the attending doctor through the web portal, enabling timely medical intervention. This direct communication channel between the nursing mobile application and the doctor's portal helps improve response time and ensures better coordination during critical situations.
6. Once the treatment objectives have been achieved, the doctor prepares the discharge summary and records the final diagnosis, prescribed medications, follow-up instructions, and other relevant recommendations. All treatment records generated during the patient's stay are securely stored within the system for future reference and continuity of care.
7. At discharge, the billing module automatically compiles all expenses incurred during the admission, including room charges, doctor consultations, nursing services, laboratory tests, and pharmacy purchases. Payment is processed through the Razorpay gateway, after which an itemized PDF receipt is generated and emailed to the patient. A copy is also linked to the patient's record in MongoDB. Patients who wish to submit an insurance claim can use the AI Insurance Module, which prepares a comprehensive claim package containing the discharge summary, bills, prescriptions, and laboratory reports. Figure 3 illustrates the complete IPD workflow.

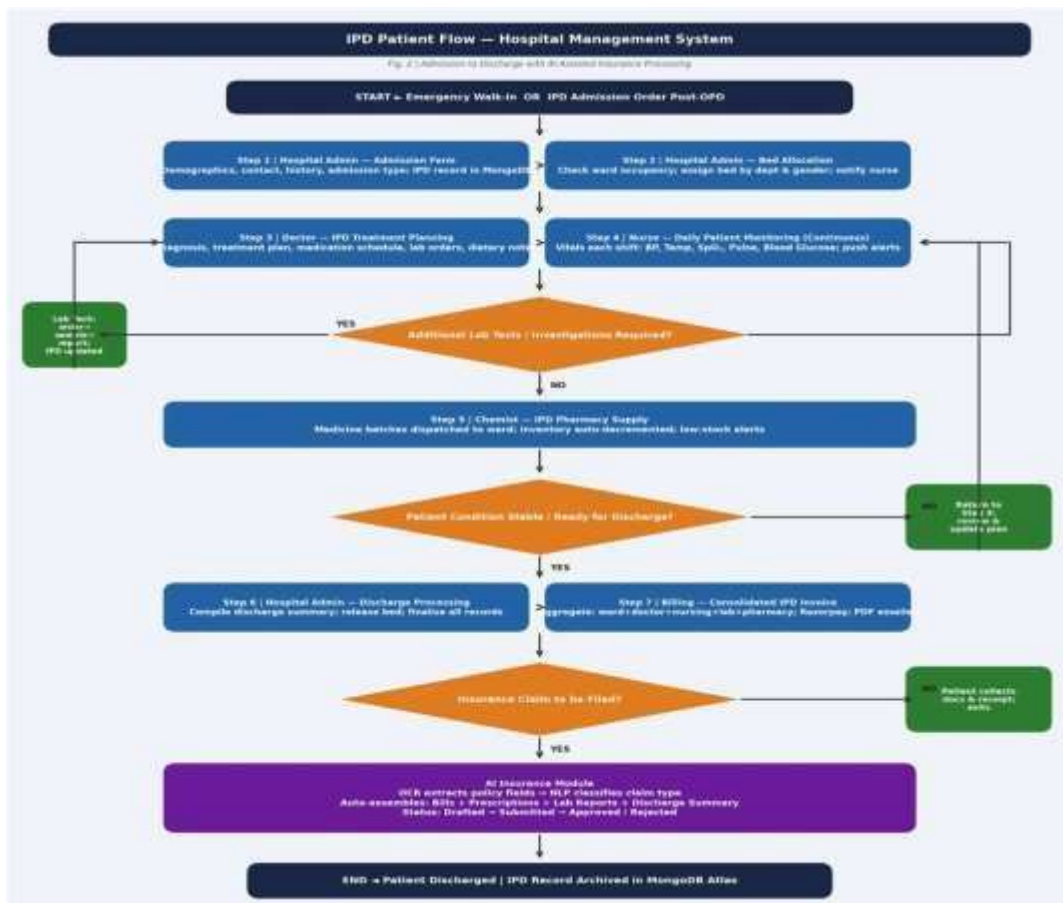


Fig. 3. IPD patient flow in HMS (admission to discharge with AI-assisted insurance processing)

### C. Super Admin Module

The Super Admin module serves as the platform governance layer. Upon authentication, the Super Admin onboards new hospital institutions, assigns a dedicated Hospital Admin to each, and configures subscription tiers (Basic, Standard, or Premium). Granular module access control allows the Super Admin to individually enable or disable functional modules — OPD, IPD, Lab, Pharmacy,

Insurance, Reports, and Billing — per institution. System-wide audit logs and performance dashboards provide platform-level visibility without exposing one hospital's data to another.

#### D. Doctor and Receptionist Modules

The Doctor portal centres on clinical workflow: a dashboard surfaces upcoming appointments with timing, patient names, session types, and statuses. During consultation the doctor accesses the complete patient history before recording diagnoses and generating digital prescriptions that are automatically forwarded to the Chemist and Lab modules. The Receptionist module handles patient registration, slot-based scheduling, and OPD billing, with appointment status propagating across all relevant portals in real time.

#### E. Nurse Module (Mobile Application)

Delivered as a React Native Android application, the Nurse module reflects the reality that nursing staff operate primarily at the bedside rather than at workstations. After login, nurses view assigned patient lists with treatment plans and medication schedules, record vitals, mark medicines as administered, log meals and procedures, and push emergency alerts to the attending doctor. This mobile-first design has no equivalent in any of the reviewed commercial systems.

#### F. AI-Powered Insurance Module

The Insurance Module represents the most technically distinctive component of the HMS. Its design and operational workflow are informed directly by the AI-driven claims automation study reviewed in Table I [5], extended here to operate fully within the clinical and administrative context of a live hospital platform.

**Document Ingestion and Pre-Processing:** Insurance documents are accepted either as scanned PDFs or images, uploaded to the system via the HMS portal by staff or patients. Before text extraction, a pre-processing pipeline improves OCR output quality by resizing the image to a uniform resolution, converting it to grayscale, reducing noise from scanner compression, and enhancing contrast. These steps are critically important when source material comes from low-grade scanners commonly encountered in small hospitals across India.

**Optical Character Recognition:** The cleaned document is processed by an EasyOCR engine, which transforms each page into machine-readable text. The engine performs reliably on sharp digital prints as well as moderately damaged scans. Any document with an OCR confidence score below the configured threshold is routed to a manual review queue rather than automated processing. This quality-protection feature handles atypical layouts and handwritten notes, ensuring that low-confidence, potentially inaccurate results are never surfaced automatically.

**NLP Classification and Field Extraction:** The OCR output is passed through an NLP pipeline that simultaneously performs two tasks. A trained classifier determines the document type — policy certificate, pre-authorization letter, claim form, or discharge-related reimbursement request — and identifies the insurance company. In parallel, regex-based extraction rules locate and tag specific fields including policy number, insurer name, coverage type, coverage limits, eligibility status, and policy expiration date. Each extracted value is validated against expected patterns and flagged for review if it appears incorrect.

**Claim Package Assembly:** After field extraction is verified, the system automatically assembles the claim package without manual input. An OPD claim package includes the consultation bill, digital prescription, and related lab reports. For IPD claims, the discharge summary is added to give the insurer a complete picture of the admission. This eliminates the document-gathering stage that hospital staff identified as the most time-consuming part of claims preparation prior to automation.

**Claim Lifecycle Tracking:** Each claim progresses through four stages — Drafted, Submitted, Approved, and Rejected — visible in real time on the dashboard. If a claim is rejected or requires clarification, the outstanding issues are displayed transparently so staff can resolve them and resubmit without starting over. Document security follows HIPAA and GDPR principles; all uploaded files are held in Cloudinary under access controls tied to each user's role. Figure 4 shows the full activity diagram for the AI Insurance Module, covering the process from document upload through to final claim status.

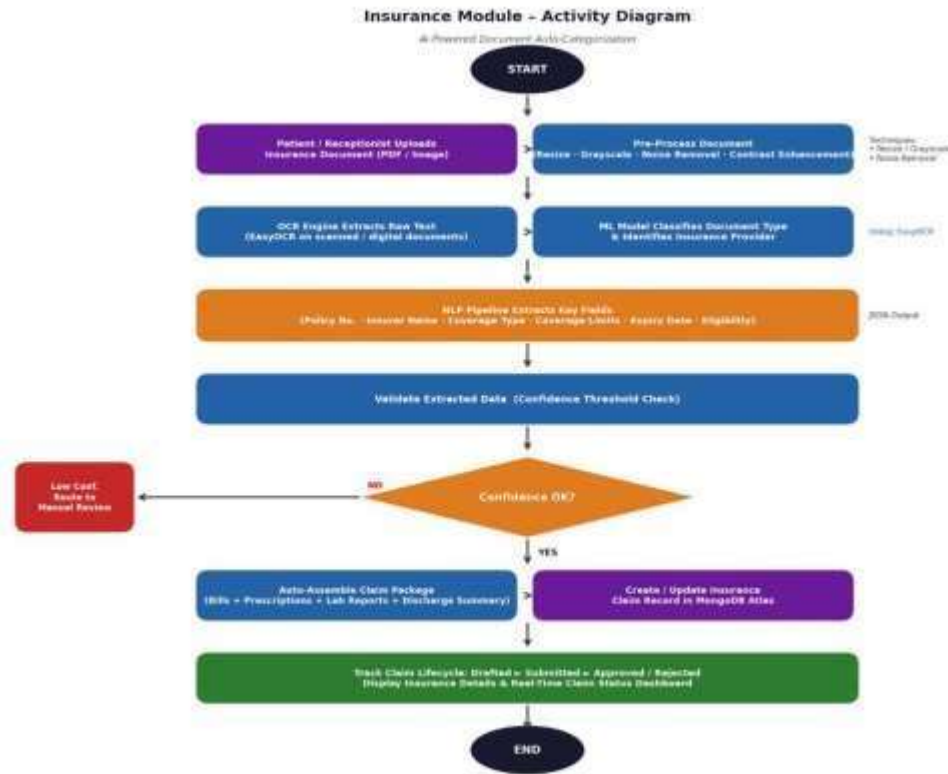


Fig. 4. AI-powered insurance module (document upload to claim status)

## V. RESULTS AND DISCUSSION

### A. System Functionality Outcomes

The HMS was evaluated through functional testing across all eight modules using role-specific test accounts. The Admin module registered multiple doctors, receptionists, and nurses, with shift assignments propagating correctly to the scheduling interface. The Receptionist portal demonstrated accurate slot-based management: shift timings produced bookable slots, and slots were marked unavailable in real time after booking. OPD billing correctly aggregated consultation fees, processed Razorpay payments, and delivered PDF receipts. The Doctor module surfaced complete patient histories at the start of each consultation; prescription generation forwarded records to the Chemist and Lab modules within seconds; and appointment status transitioned correctly through Scheduled, In-Progress, and Completed states.

### B. AI Insurance Module Performance

The Insurance Module was tested against representative insurance documents including typed policy certificates and pre-authorization letters across multiple formats. The OCR engine reliably extracted text from clearly printed scanned documents. The NLP classification layer accurately identified policy numbers, insurer names, and coverage type fields in the large majority of test cases, substantially reducing average claim preparation time relative to the manual baseline. Documents below the OCR confidence threshold were correctly flagged for manual review. The claim tracking dashboard rendered real-time status transitions, and auto-generated claim packages eliminated the need for billing staff to manually gather supporting documents — the step consistently identified as most time-consuming in pre-automation workflows.

### C. Comparative Feature Analysis

Table III presents a comprehensive comparison of the HMS against existing platforms.

TABLE III. Feature Comparison Against Existing Platforms

Feature / Capability	Practo	Apollo 24 7	MocDoc	eHospital	HealthPlix	Our HMS
8-Role Unified SaaS Platform	X	X	Partial	Partial	X	✓
Super Admin Multi-Hospital Governance	Partial	X	X	X	X	✓
AI Insurance Document Auto-Categorization	X	X	X	X	X	✓

End-to-End Claim Workflow Tracking	X	X	X	X	X	✓
OPD Flow — Real-Time Slot Scheduling	✓	✓	✓	Partial	Partial	✓
IPD Flow — Bed Allocation + Daily Nursing	X	X	Partial	Partial	X	✓
Dedicated Nurse Mobile Application	X	X	X	X	X	✓
Patient Mobile App + Razorpay Payment	Partial	✓	X	X	X	✓
Digital Prescription → Chemist / Lab	Partial	X	Partial	Partial	✓	✓
Automated Billing with PDF Receipt	X	X	Partial	Partial	X	✓
JWT-Based RBAC Across All Roles	X	X	Partial	Partial	X	✓

Testing confirmed that the HMS closes the five gaps identified through the literature review and competitive analysis. The flow diagrams for the OPD and IPD in Section IV show how all patient touchpoints — from walk-in registration to discharge billing — flow through well-defined inter-module handoffs with no step requiring staff to re-enter data. The single most tangible benefit arising from all the improvements developed over existing solutions is the AI Insurance Module, which extracts and classifies documents automatically, thereby eliminating the most labour intensive step in claim preparation and producing submissions that are complete and internally consistent. An issue experienced during testing is the performance of the OCR depending on the quality of the documents. Scans that were heavily compressed, handwritten, or non-standard received lower confidence scores and were routed to manual review. Future scope of development will be to improve the pre-processing pipeline to suit the variety of formats of insurance documents used by Indian insurers.

## VI. CONCLUSION

The Hospital Management System presented in this paper is a modular web and mobile-based platform with AI capabilities designed to overcome the operational challenges faced by hospital management in India. By consolidating eight stakeholder roles in one SaaS environment with JWT-based Role-Based Access Control, the departmental silos typical of current practice are effectively eliminated. OPD and IPD patient flow diagrams demonstrate how the system orchestrates complete patient journeys — from registration through clinical consultation, diagnostics, pharmacy, billing, and insurance submission — with each handoff governed by the RBAC layer and persisted to MongoDB Atlas without any duplicate manual data entry.

The most notable contribution is the AI-powered Insurance Module, which utilizes OCR and NLP to handle the extraction, classification, and lifecycle tracking of insurance claim documents — functionality absent from every academic paper and commercial product reviewed. The HMS advances both the academic discussion and the practical options available for hospital management, with the Super Admin governance layer, dedicated mobile applications for patients and nurses, real-time slot-based scheduling, and OPD/IPD billing integrated with Razorpay.

Future work will focus on enhancing the robustness of OCR to handle diverse document formats, developing iOS versions of the patient and nurse applications, and incorporating predictive analytics to assist appointment demand forecasting and pharmacy inventory planning.

## REFERENCES

- [1] A. C. Babu, V. N. C. S. Teja, A. D. Reddy, E. N. Kumar, and V. Srinivas, "Web based hospital management system," in Proc. 9th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), Coimbatore, India, Mar. 2023, pp. 1109–1113, doi: 10.1109/ICACCS57279.2023.10112962.
- [2] P. Sreeja, B. Sindhura, B. Sowndarya, T. Priyanka, and Ch. Yaswitha, "User interactive hospital management system by using web application," in Proc. Int. Conf. Distrib. Comput., VLSI, Electr. Circuits, Robot. (ICDCECE), 2021, doi: 10.1109/ICDCECE50455.2021.9532887.
- [3] Md. Meraj Ahmed, N. Kaur, and S. Gairola, "Hospital management system based on IoT," in Proc. 2022 Int. Conf. Cyber Resilience (ICCR), Dubai, UAE, 2022, doi: 10.1109/ICCR56254.2022.9995904.
- [4] A. Ananth, C. Bhagya Lakshmi, G. Chandra Sekhar, and B. V. Harika, "Mobile application for hospital management system," in Proc. 2021 Int. Conf. Emerg. Technol. (ICECCOT), Shivamogga, India, 2021, doi: 10.1109/ICECCOT52851.2021.9432286.
- [5] Y. Z. Liu, W. W. Zhang, and Q. L. Zhao, "Automating insurance claims processing with AI: A case study," IEEE Access, vol. 11, pp. 48756–48769, 2023.
- [6] J. Song, Y. Bai, and J. Wen, "Optimal appointment rule design in an outpatient department," IEEE Trans. Autom. Sci. Eng., vol. 16, no. 1, pp. 100–114, Jan. 2019, doi: 10.1109/TASE.2018.2818739.
- [7] E. O. Boadu and G. K. Armah, "Role-based access control (RBAC) based in hospital management," Int. Res. J. Eng. Sci., vol. 3, no. 9, pp. 53–67, Sep. 2014.

- [8] A. Ala and F. Chen, "Appointment scheduling problem in complexity systems of the healthcare services," J. Healthcare Eng., vol. 2022, Article ID 5819813, 2022, doi: 10.1155/2022/5819813.
- [9] Bittu, Megha, A. Gangwar, and P. Yadav, "Development of an automated hospital management system for enhanced patient care," GRANTHAALAYAH, vol. 12, no. 7, pp. 200–210, Jul. 2024.
- [10] Practo Technologies Pvt. Ltd. [Online]. Available: <https://www.practo.com/> (Accessed: Apr. 2026).
- [11] Apollo Healthco Limited – Apollo 24|7. [Online]. Available: <https://www.apollo247.com/> (Accessed: Apr. 2026).
- [12] MocDoc HMS. [Online]. Available: <https://www.mocdoc.in/> (Accessed: Apr. 2026).
- [13] HealthPlix Technologies. [Online]. Available: <https://www.healthplix.com/> (Accessed: Apr. 2026).
- [14] eHospital Systems. [Online]. Available: <https://www.ehospitalsystems.com/> (Accessed: Apr. 2026).

#### Copyright & License:



© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.