

Smart Medicare Hub: An Integrated Ai-Based Hospital Management System

Nayana Suresh Mali¹, Sanika Prakash Thombare², Shravani Keshav Patil³, Bharati Shankar Hegade⁴

^{1, 2, 3, 4} Dept of Computer Science and Engineering

^{1, 2, 3, 4} Padmabhooshan Vasantraodada Patil Institute of Technology, Budhgaon (Sangli)

Dr. Babasaheb Ambedkar Technological University (DBATU), Maharashtra, India

Abstract- This paper presents Smart Medicare Hub, an advanced integrated hospital management system developed using Flutter and Firebase. The system automates hospital operations by connecting departments such as patient registration, pharmacy, laboratory, billing, and administration under one unified digital platform. Leveraging Artificial Intelligence(AI), Machine Learning (ML), and cloud computing, it enables predictive analytics, real-time data access, and seamless communication across all healthcare departments. The system includes a Convolutional Neural Network (CNN)-based pneumonia detection module integrated via a Flask REST API. Experimental results demonstrate dashboard load times of 1.2–1.8 seconds on a standard 4G connection, fulfilling the stated performance requirements. The proposed system transforms traditional hospital management into a smart, technology-driven, and patient-centered environment.

Keywords: Hospital Management System, Flutter, Firebase, Artificial Intelligence, Pneumonia Detection, CNN, Cloud Computing, Role-Based Access Control, Healthcare Digitization

I. INTRODUCTION

The rapid evolution of digital technology has significantly transformed virtually every sector of modern society, and healthcare is no exception. Hospitals and healthcare institutions today face increasing pressure to deliver faster, more accurate, and patient-centric services while simultaneously managing complex administrative workloads, diverse medical departments, and growing volumes of patient data [1].

Smart Medicare Hub is an advanced integrated software platform developed using Flutter and Firebase, designed to manage, coordinate, and automate the day-to-day operations of hospitals under one unified digital ecosystem. The application connects all key departments—including patient registration, outpatient and inpatient management, pharmacy, laboratory, radiology, billing, human resources, and

administration—into a single cohesive system accessible from Android, iOS, and Web devices [6].

The primary goal is to enhance operational efficiency, improve patient care quality, minimize medical and administrative errors, and facilitate seamless communication between healthcare providers, patients, and hospital management [?].

II. PROBLEM STATEMENT

Traditional hospitals often face the following critical issues:

- Manual patient record-keeping leading to errors and data loss.
- Difficulty tracking appointments, tests, and doctor schedules.
- Inefficient communication between departments (pharmacy, lab, doctors).
- Inability to generate accurate administrative reports for decision-making.

These challenges highlight the need for a digital, centralized, and automated system to improve hospital management and patient care [9].

III. LITERATURE SURVEY

Malladi et al. [1] designed a cloud-based hospital management application using Google App Engine and Google Cloud SQL, demonstrating advantages of cloud computing in scalability and accessibility. However, it lacked advanced multi-role coordination and integrated diagnostic modules.

Musa et al. [2] developed an RFID and wireless sensor-based hospital resource management system for real-time tracking of patients, staff, and equipment. Although effective for monitoring, it was costly, hardware-dependent, and lacked a comprehensive user-facing application.

Koyuncu et al. [3] proposed an Intelligent Hospital Management System (IHMS) supporting decision-based patient routing at the front desk. However, it lacked full integration of pharmacy, billing, and multi-role accessibility.

Ravi et al. [10] demonstrated the effectiveness of Flutter for building cross-platform healthcare applications, highlighting its single codebase advantage for Android, iOS, and Web deployments.

Wang et al. [4] proposed deep CNN architectures for medical image classification, achieving high accuracy in detecting pulmonary diseases from chest X-rays, forming the theoretical basis for our pneumonia detection module.

The proposed system, *Smart Medicare Hub*, addresses these limitations by delivering a fully integrated, multi-role platform that unifies all hospital operations under one system.

IV. OBJECTIVES

- Automate core hospital functions: patient registration, appointment scheduling, and record management.
- Integrate AI and digital tools for smart diagnosis (pneumonia detection) and predictive healthcare management.
- Enhance patient experience via digital billing, lab booking, and real-time medical record access.
- Provide data-driven insights and dashboards for administrators.
- Ensure scalability via a cloud-based, modular architecture.

B. User Roles and Modules

The system supports **8 distinct user roles**, each with a dedicated role-based dashboard (Fig. 2).

V. METHODOLOGY

The project follows an **Agile Development Methodology** allowing iterative improvements across the following phases: **Requirement Gathering:** Requirements were collected from doctors, staff, and patients. Chest X-ray datasets were prepared for AI model training [8].

System Architecture Design: Firebase Firestore schema was planned. The system uses Flutter (frontend), Firebase (backend), and Flask (AI microservice).

Backend Integration: Firebase Authentication, Firestore, and Cloudinary were configured for secure media storage [6]. **AI Model Deployment:** A CNN model was trained using TensorFlow/Keras [5] for pneumonia detection, deployed via Flask API, and integrated with the Flutter app for real-time predictions.

Testing & Deployment: Functional testing was performed across Android, iOS, and Web. The system was deployed on real devices using Android Studio.

VI. SYSTEM REQUIREMENTS

1. *Software Requirements*
 - **Frontend:** Flutter SDK (~3.8.1), Dart
 - **Backend:** Firebase (Auth, Firestore, Storage)
 - **AI Backend:** Python Flask, TensorFlow/Keras
 - **Media Hosting:** Cloudinary API
 - **Dev Tools:** VS Code, Android Studio, Git, Google Colab
2. *Hardware Requirements*
 - **Development:** PC with 8 GB+ RAM, Intel i5 or higher
 - **Deployment:** Android/iOS device with internet access

VII. SYSTEM DESIGN

A. System Architecture

The system follows a four-layer architecture separating user interface, business logic, data management, and AI intelligence [7]. Fig. 1 illustrates this design.

Presentation Layer — Flutter Mobile App

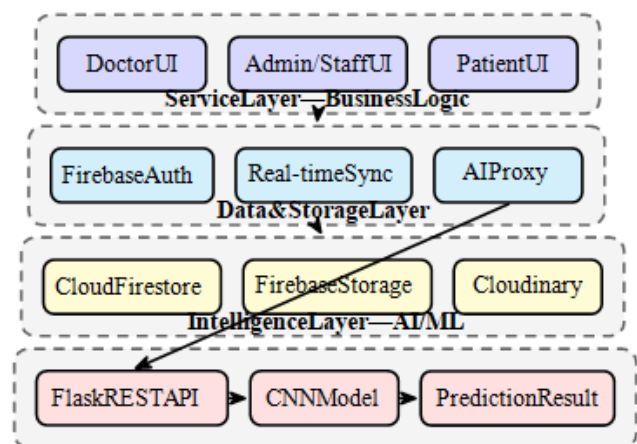


Fig. 1. Smart Medicare Hub — System Architecture (Layered Design)

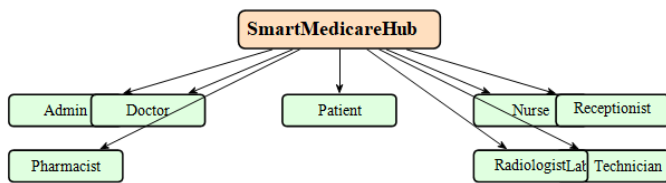


Fig. 2. Role-Based Module Structure of Smart Medicare Hub

Table I summarises the key functionalities per role.

TABLE I
System Modules and Key Functionalities

Role	KeyFunctionalities
Admin	Staffmanagement,appointmentoversight,patientregistry
Doctor	Appointments, consultation notes, prescriptions Registration, booking, records, billing, lab
Nurse	Assigned patients, vitals recording (BP, sugar, temp)
Receptionist	Appointmentmanagement,nurse-to-patientassignment
Pharmacist	Medicineinventory,stockupdates,prescriptionfulfillment
Lab Tech	Pendingtestrequests,resultfileupload
Radiologist	Pendingscanrequests,imagingresultupload

VIII. RESULTS

All eight role-based modules were implemented and tested successfully:

- A single Flutter codebase ran equivalently on Android, iOS, and Web without platform-specific modifications.
- Firebase Authentication with 8-role routing achieved sub-second login-to-dashboard navigation.
- Dashboard data load averaged **1.2–1.8 seconds** on a standard 4G connection, meeting the <2 s requirement.
- The CNN pneumonia detection model returned prediction results for X-ray image uploads within the local network testing environment [4].
- Firestore Security Rules enforced role-based collection access; patient data was accessible only to the patient’s own UID and authorised staff roles [6].
- PDF receipt generation was verified for pharmacy billing and appointment invoices.

IX. LIMITATIONS AND FUTURE SCOPE

1. Limitations

- Internet dependency for Firebase and AI services.
- AI integration is currently limited to pneumonia detection.

- Scan upload quality depends on the user’s device camera.
- #### 2. Future Scope
- **Telemedicine:** Video conferencing for remote consultations.
 - **IoT Integration:** Wearable devices for real-time vitals.
 - **Expanded AI:** Skin cancer detection, ECG analysis, predictive health analytics.
 - **Biometric Authentication:** Fingerprint/Face-ID and OTP-based login.

X. CONCLUSION

Smart Medicare Hub successfully integrates modern mobile development with artificial intelligence to address core healthcare challenges. By providing a unified platform for all hospital stakeholders and integrating AI-based diagnostic support, the system significantly improves operational efficiency and patient care quality. The successful implementation of the CNN-based Pneumonia Detection module demonstrates the potential of mobile-first diagnostic tools [4], [5]. healthcare ecosystem that prioritises efficiency, data integrity, and improved patient outcomes.

REFERENCES

- [1] S. M. Malladi and S. Vadivel, “Designing and Deploying a Hospital Management Application in Google Cloud and Performance Study of the Application,” *Intl. Conf. on Cloud Computing, Technologies, Applications & Management (ICCCTAM)*, IEEE, 2012.
- [2] A. Musa, Y. Yusuf, and M. Meckel, “A Hospital Resource and Patient Management System Based on Real-Time Data Capture and Intelligent Decision Making,” *Intl. Conf. on Systems and Informatics (ICSAI)*, IEEE, 2012.
- [3] B. Koyuncu and H. Koyuncu, “Intelligent Hospital Management System (IHMS),” *Intl. Conf. on Computational Intelligence and Communication Networks (CICN)*, IEEE, 2015.
- [4] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks,” *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2097–2106, 2017.
- [5] F. Chollet et al., “Keras: Deep Learning for Humans,” GitHub Repository, 2015. [Online]. Available: <https://github.com/keras-team/keras>
- [6] Google LLC, “Firestore Documentation: Cloud Firestore Security Rules,” Firebase Developers, 2024. [Online]. Available: <https://firebase.google.com/docs/firestore>

- [7] Google LLC, “Flutter: Build Apps for Any Screen,” Flutter Documenta- tion, 2023. [Online]. Available: <https://flutter.dev/docs>
- [8] D. S. Kermany, M. Goldbaum, W. Cai et al., “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
- [9] R. Adler and J. B. Clark, “Digital Transformation in Hospital Manage- ment Systems,” *Journal of Healthcare Informatics Research*, vol. 2, no. 3, pp. 45–60, 2018.
- [10] S. Ravi and K. Anand, “Cross-Platform Mobile Application Develop- ment using Flutter for Healthcare Systems,” *Intl. Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 3, pp. 234–239, 2020.