WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Hospital Management Software

[1]Santhoshkumar. M, [2]Sugunraj. K

[1] RVS Educational Trust Groups of Institutions, Department of Computer Science and Engineering
Mail: santhoshkumar040803@gmail.com

ABSTRACT

In today's fast-paced world, hospitals face numerous challenges managing operations effectively. This project aims to provide a robust Hospital Management System using Python, KivyMD, and SQLite, featuring multi-role login (Receptionist, Doctor, Medical Staff, and Patient), appointment scheduling, email OTP verification, and treatment report retrieval. The system streamlines medical data handling, enhances communication, and facilitates efficient patient care management. This paper elaborates on the development, features, architecture, and implementation of the system.

Keywords : Hospital Management System, Python, KivyMD, SQLite, Multi-role Login, OTP Verification.

I. Introduction

Efficient hospital administration plays a critical role in the delivery of quality healthcare services. Traditional hospital management often involves paper-based systems, which are prone to errors, data loss, and inefficiency. Our proposed Hospital Management System (HMS) digitalizes and automates essential functions, including patient appointments, login verification, treatment management, and more. The system's architecture is designed for scalability, reliability, and ease of use, leveraging modern technologies such as Python, KivyMD for the UI, and SQLite for backend storage. The system also incorporates role-based dashboards and email-based OTP verification for secure access.

II. Literature Review

Various hospital management systems have been proposed and implemented in the past, using different frameworks and platforms. Systems like Medixcel EMR and OpenMRS focus on enterprise-level deployments. However, such systems often require high-end infrastructure and trained personnel. Our proposed system is lightweight and ideal for small to medium hospitals. It is inspired by successful models but emphasizes simplicity, portability, and ease of deployment. It also integrates modern mobile-friendly UIs using KivyMD and offers local database support for offline capability.

III. System Architecture

The Hospital Management System follows a modular architecture with a centralized SQLite database. Each module is responsible for a specific functionality:• Login System – Validates users based on roles: Receptionist, Doctor, Medical Staff, or Patient.• Appointment Module – Receptionists can schedule and manage appointments with predefined doctors.• OTP Verification – Patients authenticate using email-based OTP during login.• Treatment Report – Doctors can enter and patients can view medical treatment reports.• Dashboard – Each role has a dedicated dashboard showing relevant actions.The user interface is built using KivyMD, which offers cross-platform compatibility and a modern look.

IV. Methodology

The system is implemented in Python using KivyMD for frontend UI and SQLite as the backend database. The workflow begins with role-based login followed by dashboard redirection. Receptionists can book appointments using dropdown menus for doctors, while doctors can input treatment details. Medical staff may view and manage general information, and patients log in using OTP received via email. The OTP mechanism is implemented using SMTP integration. Data persistence is achieved using SQLite, and all inputs are validated to prevent errors.

V. Implementation and Features

• Multi-Role Login: Distinct interfaces and access rights for Receptionist, Doctor, Medical Staff, and Patient.• Appointment Booking: Receptionists can enter patient number, select a doctor from a dropdown, and specify visit reason.• Email OTP Login: Patients receive and verify OTPs to access their dashboards.• Treatment Entry & Retrieval: Doctors record treatments; patients can retrieve them later.• SQLite Database: Stores all data locally, enabling offline use and ease of access.• Consistent UI: Use of KivyMD components like MDToolbar, MDCard ensures a professional look.

VI. Results and Discussion

The system was tested across multiple scenarios and user roles. The OTP verification mechanism worked reliably, sending emails within seconds. Appointment booking was validated with error handling for duplicate and invalid entries. Doctors could efficiently input treatment data, which patients could view post-login. SQLite ensured quick data transactions, even on mobile setups. The modularity of the code makes it easy to extend functionality, e.g., to integrate SMS APIs or cloud databases in future versions.

VII. Conclusion

This Hospital Management System meets the essential needs of small to mid-sized clinics and hospitals by providing a centralized, role-based, and secure platform for managing medical records, appointments, and communication. With Python and KivyMD, it offers a responsive UI and smooth functionality. The inclusion of OTP-based patient login and treatment data retrieval further strengthens its utility and security. The system can be extended with features such as cloud backup, mobile app packaging, and AI-based diagnostics for future enhancement.

VIII. Future Work

While the system meets its initial goals, several areas remain for future improvement. These include:• Integration with cloud databases for remote access.• Role-based access control with logging and audit features.• SMS notification support for appointment reminders.• Packaging as an Android app using Buildozer.• Incorporating analytics and health dashboards for hospital management.• Multilingual support for wider reach in rural areas.

REFERENCES

[1] T. G. Berndt and T. L. Lush, “Hospital Information Systems: A Review of Literature,” Health Informatics Journal, vol. 10, no. 3, pp. 223–237, Sep. 2004.

[2] M. Arif and J. K. Mohamed, “Development of Patient Management System Using Python and SQLite,” International Journal of Computer Applications, vol. 180, no. 8, pp. 5–9, Jan. 2018.

[3] A. Meier, SQL Performance Explained, 5th ed., San Francisco, CA: Percona Press, 2018.

[4] KivyMD Developers, “KivyMD: Material Design Components for Kivy,” GitHub Repository, 2024. [Online]. Available: https://github.com/kivymd/KivyMD

[5] SQLite Consortium, “SQLite Documentation,” 2024. [Online]. Available: https://www.sqlite.org/docs.html

[6] World Health Organization, “Digital Health,” WHO, 2023. [Online]. Available: https://www.who.int/health-topics/digital-health

[7] Fast2SMS, “Fast2SMS API Documentation,” 2024. [Online]. Available: https://www.fast2sms.com/dev-api

[8] Python Software Foundation, “Python 3.13.2 Documentation,” 2024. [Online]. Available: https://docs.python.org/3.13/

[9] Kivy Organization, “Kivy Documentation (v2.3.1),” 2024. [Online]. Available: https://kivy.org/doc/stable/

[10] A. Mohan and R. Reddy, “Healthcare Management System Using Kivy and SQLite,” International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 5, pp. 1054–1060, May 2020.