Context Diagram (Existing System)

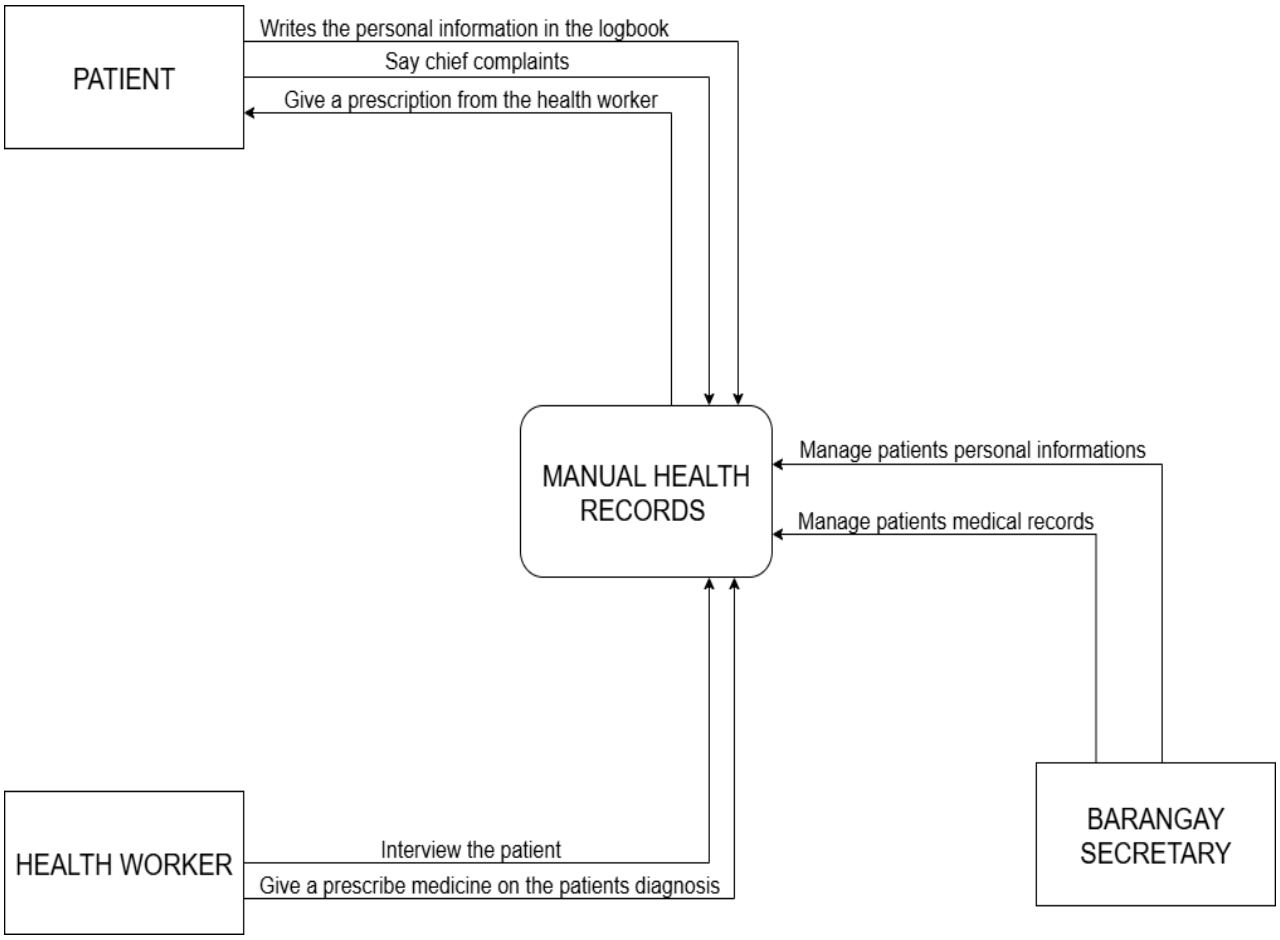


Figure 3. The Context Diagram of the Existing System

Figure 3. The Context Diagram of the Existing System Figure 3 showed the diagram of the existing system illustrates the high-level interactions between external entities and the central "Manual Health Records" process. This diagram depicts three primary external entities: "Patients" , "Health Workers" and “Barangay Secretary”. Patients interact with the system by providing their "chief complaints" and receiving a "prescription from the health worker" and “Barangay Secretary” manages the patient personal information and medical records. They also

have their "personal information written in the logbook," which is then integrated into the "Manual Health Records." On the other hand, Health Workers "interview the patient" and "give a prescribed medicine on the patients diagnosis," with these activities directly influencing and being influenced by the "Manual Health Records." The diagram highlights the manual nature of the current system, where all information flows through and is managed within the "Manual Health Records" entity.

Context Diagram (Proposed System)

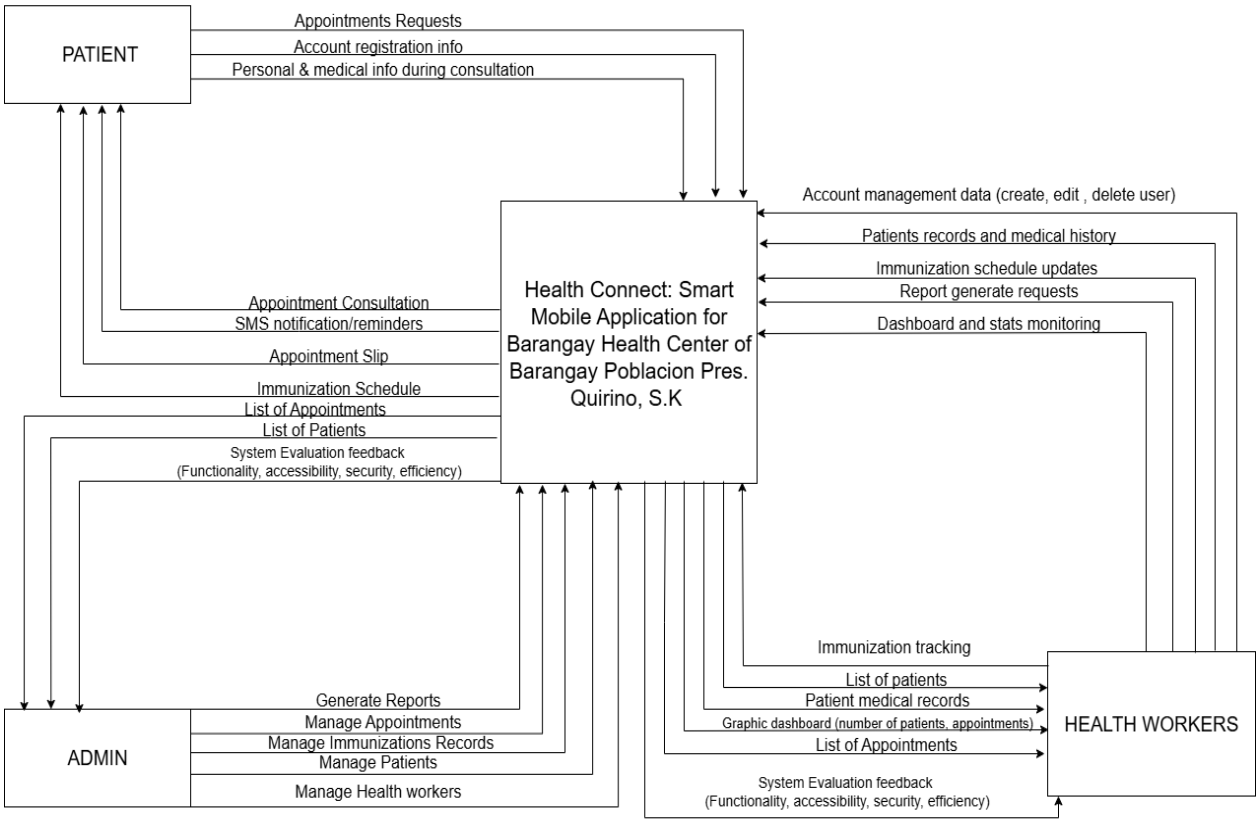
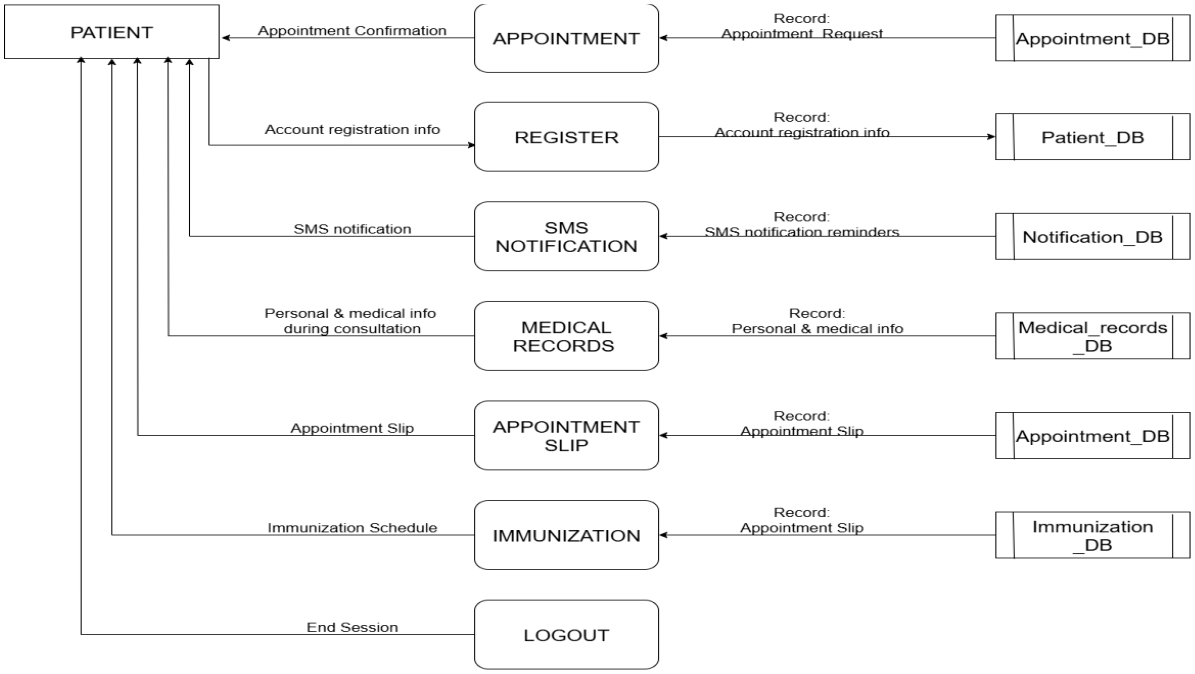


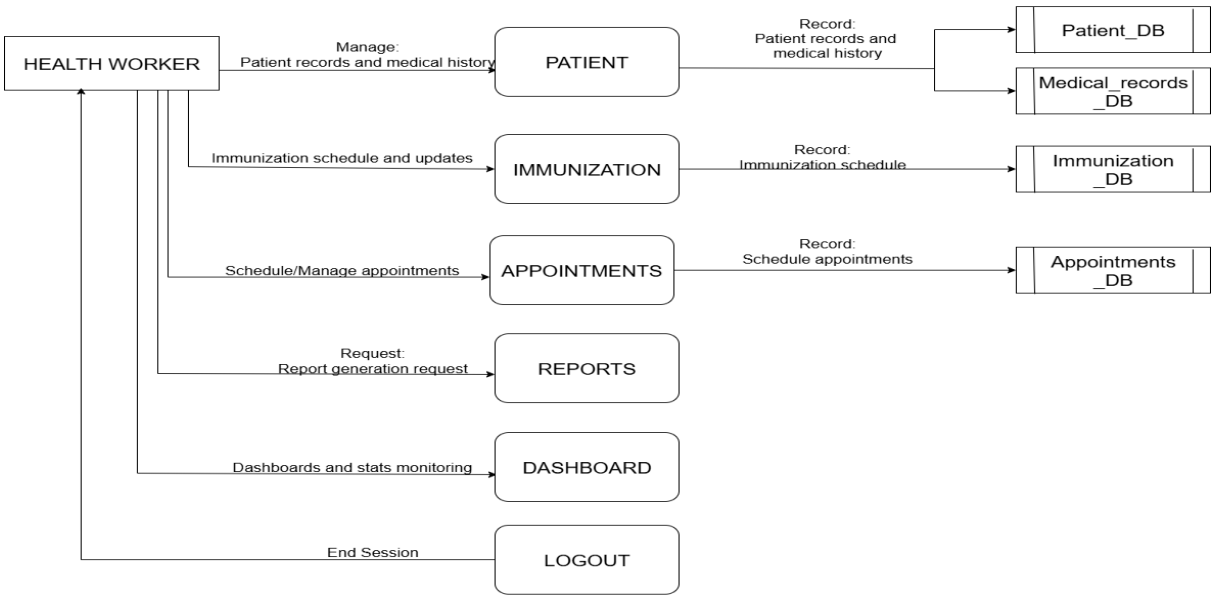
Figure 5. The Context Diagram of the Study

Figure 5 illustrates the core functions and interactions between the main users' admin , patients , health workers and the system. It emphasizes how the mobile application serves as a centralized platform for managing health worker accounts, handling patient records, and monitoring immunization schedules. Patients can register accounts, submit appointment requests, and receive SMS notifications and appointment slips. Health workers, on the other hand, can manage patient medical records, update immunization schedules, and monitor system statistics through a dashboard that visualizes patient and appointment data. Admin on the other hand manage health workers , patients accounts , monitor system statistics through a dashboard that visualizes patient and health worker data. The system also supports automated report

generation, allowing health workers to generate lists of patients, and appointments. Additionally, the system is designed to be evaluated based on functionality, security, accessibility, and efficiency. This context diagram highlights the automated and integrated nature of the system, reducing manual processes and improving communication between patients and health professionals.

DATA FLOW DIAGRAM (Logical)





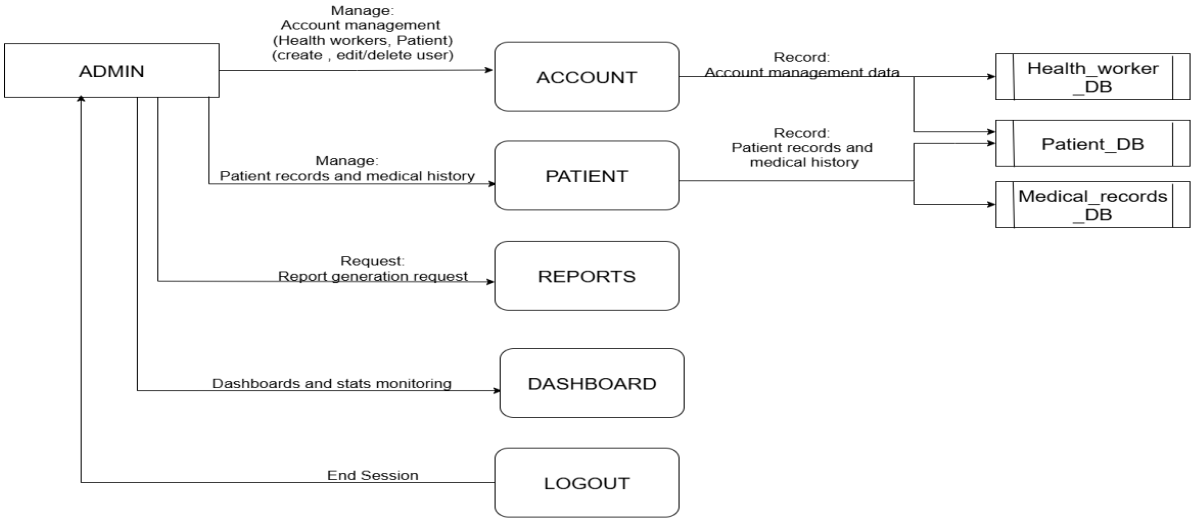


Figure 6. The Logical Data Flow Diagram of the System

Figure 6 shows the healthcare management system designed around two primary user roles: Patients , Health Workers and Admin. Patients initiate interactions by registering accounts storing their information in the Patient_DB and requesting appointments, which are logged in the Appointment_DB. During consultations, they provide personal and medical details that are recorded in the Medical_record_DB. Patients can also submit feedback and chat messages, and receive SMS notifications such as appointment confirmations managed through the Notification_DB. Additionally, they access immunization schedules from the Immunization_DB, view their medical records, and generate appointment slips for upcoming visits.

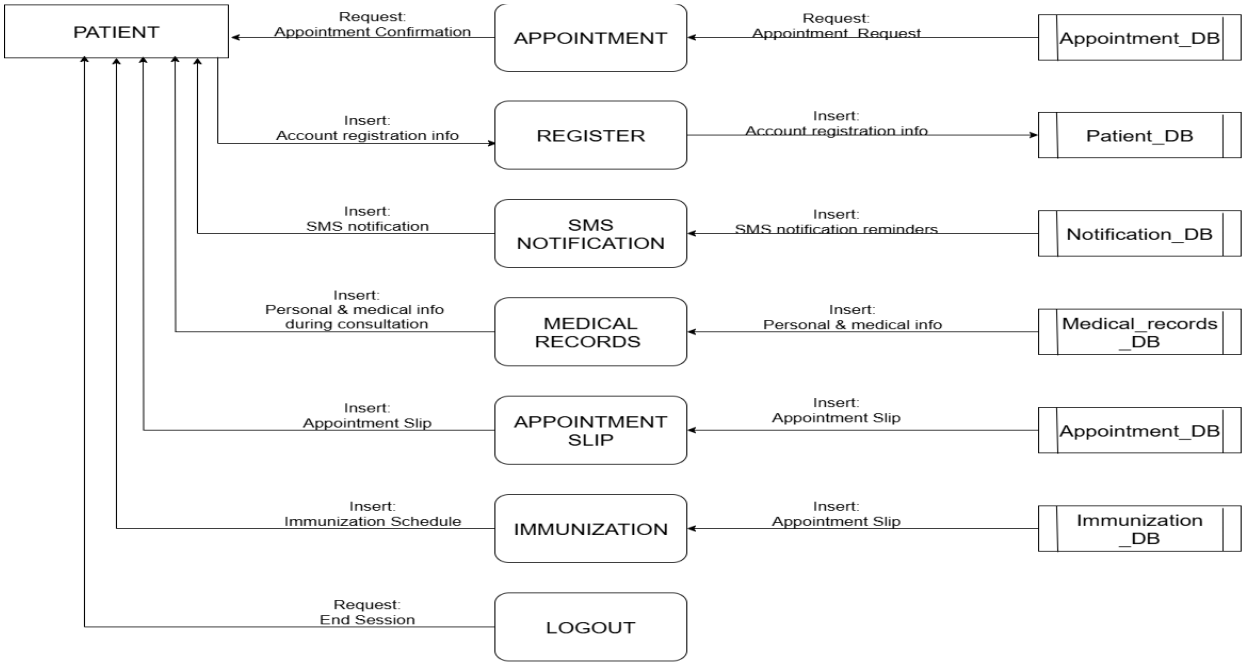
Health Workers, on the other hand, manage a broader range of system workflows. They oversee appointments, review and update patient profiles in the Patient_DB, and maintain detailed medical histories in the Medical_record_DB.

Admin , on the other hand, manage health workers account and patients account , personal information and medical records in the Health_worker_DB, Patient_DB and Medical_records_DB, request a report generation and monitor dashboard and statistics (health worker , patients data) .

The system emphasizes smooth data synchronization across eight core databases: Patient_DB (registration), Medical_record_DB (health history), Appointment_DB (bookings), Immunization_DB (vaccine scheduling), Notification_DB (SMS alerts), and Health_worker_DB (staff credentials). Key processes include appointment coordination (linking patient requests to

health worker schedules), medical data consolidation (updating records during consultations), and automated notifications (sending SMS confirmations). As a logical DFD, this model abstracts away technical implementation details such as infrastructure or security and focuses instead on the flow of data, user interactions, and storage mechanisms to represent a streamlined, patient-centric healthcare workflow.

DATA FLOW DIAGRAM (Physical)



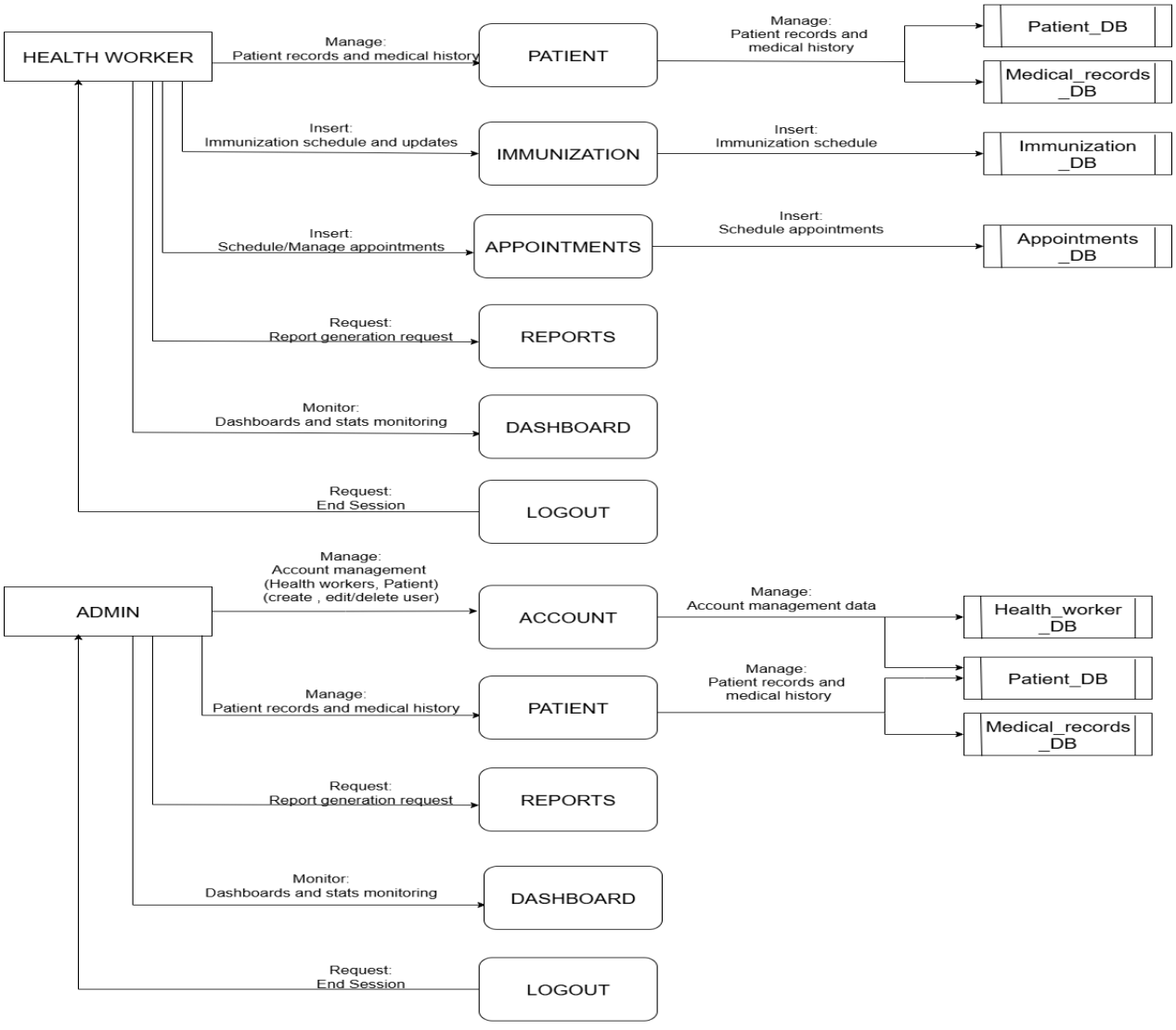


Figure 7. The Physical Data Flow Diagram of the System

Figure 7 presents an implementation-specific view of a healthcare management system, building upon the logical model by incorporating concrete components, technologies, and workflows. It restrains the triple-user structure of Patients, Health Workers and Admin, but adds physical elements such as SMS notifications, graphical dashboards, and explicit interactions with system databases.

For Patients, the workflow begins with appointment booking and registration through the Appointment module. During registration, personal details are stored in the Patient_DB, while confirmed bookings result in the creation of an Appointment Slip stored in the Appointment_DB and trigger automated SMS alerts via the Notification_DB. Patients can access their historical health data from the Medical_record_DB and review vaccination schedules stored in the Immunization_DB. Patient sessions conclude with a logout process.

Admin , on the other hand, manage health workers account and patients account , personal information and medical records in the Health_worker_DB, Patient_DB and Medical_records_DB, request a report generation and monitor dashboard and statistics (health worker , patients data) .

Health Workers manage a range of operational tasks. They handle appointments via the Appointment_DB, access demographic data from the Patient_DB, and consult or update medical histories in the Medical_record_DB during consultations. Vaccine administration and monitoring are handled through the Immunization_DB. Analytical functions are supported through Graphical

Dashboards, which visualize data trends such as appointment volumes and medicine inventory levels. A System Evaluation module helps monitor system performance. Reports are generated as needed, and sessions end with a logout.

Several key physical components enhance system functionality. The SMS Notification System, integrated with the Notification_DB, automates appointment alerts. Real-time Graphical Dashboards render analytics such as immunization coverage and resource utilization. All core data is stored in centralized databases, including Patient_DB (registration), Appointment_DB (bookings), Medical_record_DB (health records), Immunization_DB (vaccines), and Health_worker_DB (staff credentials).

The Implementation-Specific Workflow for patients typically follows the sequence: Registration → Appointment → SMS Alert → Consultation → Medical Record Access → Logout.

For admin, on the other hand: Manage Accounts (health workers , patients) → Patient Profile Check → Medical Record Update→ Dashboard Analytics→ Logout.

For health workers, it proceeds as: Appointment Review → Patient Profile Check → Medical Record Update→ Dashboard Analytics → Logout.

This physical DFD distinguishes itself from the logical model by introducing physical artifacts like the SMS gateway (Notification_DB), appointment slip generation, and interactive dashboards.

Ultimately, the purpose of this physical DFD is to illustrate how the system operates in a practical environment. Patients benefit from digital appointment slips and SMS alerts, while

health workers leverage dashboards for data-driven decisions and integrate inventory management with medical processes. The databases represent physical data structures such as tables or collections and support consistent, secure, and user-friendly workflows. As a comprehensive technical blueprint, this DFD guides developers in deploying a fully functional, patient-centric healthcare system.