**REAL TIME HEALTHCARE MONITORING SYSTEM (ONLINE)**

PROBLEM STATEMENT: A Real-Time Health Monitoring System using Java, software project aimed at continuously tracking and analyzing the vital signs and health data of individuals.

OBJECTIVE: The objective of this project is to design and implement a Java-based application that collects, processes, and displays health data in real-time. The system may also trigger alerts or notifications if the readings fall outside the safe range, enabling timely intervention.

Summary: The aim is to create an online healthcare management system for patients to book appointments, doctors to manage schedules, and administrators to oversee operations. Each user type will have a dedicated dashboard for> managing their respective activities.

User Types:

1. Admin:

○ Manages: Users, appointments, and system settings.

2. Doctor:

○ Manages: Schedules, patient records, and appointments.

3. Patient:

○ Books and manages: Appointments, views medical history.

The following functional components will be available for users:

1. User Management:

○ Input: create/update/delete to confirm successful user creation/update/deletion.

○ Output: confirmation message for successful

○ Functionality: able to manage user accounts and roles.

2. Appointment Management:

○ Input: appointment details.

○ Output: appointment schedules.

○ Functionality: expects scheduling of appointment, and to manage availability.

3. System Settings:

○ Input: Configuration settings.

○ Output: confirmation message for successful settings update.

○ Functionality: manage settings across the entire system platform.

These are the functional components that shall be available to the doctor:

1. Schedule Management:

○ Input: details of schedule.

○ Output: updated schedules.

○ Functionality: manage and modify the appointment schedule.

2. Patient Records:

○ Input: details of the patient.

○ Output: medical records of the patient.

○ Functionality: access and update the medical records of the patient.

3. Appointment Management:

○ Input: appointment details.

○ Output: confirmation sent to and acknowledgement from.

○ Functionality: be able to confirm or deny appointments of patients.

Functionalities for Patient:

1. Appointment Booking:

○ Input: preferred date and time.

○ Output: appointment confirmation.

○ Functionality: able to book and manage the appointments.

2. Medical History:

○ Input: patient details.

○ Output: medical history.

○ Functionality: view and manage personal medical history.

3. Profile Management:

○ Input: in case of a change, name, Email, Password information etc.

○ Output: profile updated successfully.

Technical Stack:

* Programming Language: Java.
* Database: MySQL, SQLite, or Mongo DB (using JDBC for connectivity).
* Networking: Java Networking APIs.
* Data Processing: Java Streams, Java.

This project can be a robust solution for continuous health monitoring, offering convenience and potentially life-saving insights through early detection and timely alerts.