

# MedWatch – Hybrid System Documentation

Architecture • Features • Data Flow • Backend + Frontend Integration (Manager + Developer Friendly)

MedWatch is a real-time MDR infection surveillance and contact-tracing platform. This hybrid document provides both high-level explanations for management and clear technical insights for developers. It covers system architecture, authentication, real-time mapping, contact graph logic, MDR alert engine, and full backend–frontend integration.

## 1. System Overview

MedWatch transforms hospital infection control from reactive to proactive. It uses RFID movement data, smart algorithms, and real-time visualization to identify MDR risks before outbreaks spread. The system adapts to low-tech and high-tech hospitals alike.

## 2. Full Architecture Pipeline

RFID → ESP32 → Google Sheet → Apps Script → MedWatch Backend API → MySQL → React Frontend → Socket.IO (Realtime). This pipeline enables instant ingestion, processing, visualization, and alerting.

## 3. Authentication (JWT + Role-Based Access)

Users login through the React frontend. Backend verifies credentials and issues a JWT. Zustand stores the token, Axios attaches it automatically, and Socket.IO authenticates with it. Roles: Admin (full control), Doctor (clinical tools), Staff (limited access).

## 4. Frontend Integration (React + Zustand + Axios)

The frontend uses Zustand for state management and Axios for API communication. PrivateRoute protects pages based on role. Login and logout flows are fully synchronized with backend authentication.

## 5. RFID Ingestion Flow

Google Sheets + Apps Script periodically POSTs rows to /api/ingest/sheet. Backend stores raw\_events and triggers contact computation + real-time updates.

## 6. Contact Graph Engine

Backend examines room/time overlaps between individuals. Overlapping intervals create edges in contact\_edges. This model powers identification of primary, secondary, and tertiary exposures.

## 7. MDR Case Detection

When a patient is marked MDR-positive, the backend performs BFS on the contact graph, identifying exposure depth. Alerts are created and sent to relevant staff in real time.

## 8. Real-Time Map Engine

GET /api/map/live provides each person's last known position and risk color (red/yellow/green).  
Socket.IO emits updates instantly when movement occurs or contact graph changes.

## 9. Admin Map Layout System

Admins upload a hospital blueprint and define room coordinates. Frontend uses this layout with real-time data to render an interactive infection-risk map.

## 10. Patient Search & Tracing

Doctors can search by UID or name. Backend returns profile, movement history, recent exposures, and risk level. This accelerates tracing and containment efforts.

## 11. Socket.IO Live Updates

Backend emits: • rawevent:new – new movement detected • contacts:computed – graph updated • map:update – real-time map changes • alert:new – MDR exposures  
Clients authenticate and join UID rooms to receive personal alerts.

## 12. Postman Test Suite (17 Steps)

17 backend tests validate authentication, ingestion, contact graph, MDR alerts, map data, patient search, and realtime behavior. This ensures system reliability end-to-end.

## 13. Team Workflow Summary

• Backend Team: APIs, database, contact algorithms, MDR alerts, sockets. • Frontend Team: React UI, dashboards, map visualization, patient search, alerts UI. • Integration Team: API wiring, Axios, role-based routing, socket listeners. This structure ensures clear division of responsibilities.