Smart Intruder Alert System (SIAS)

An IoT-Based Home Security Solution

Group Members

Aitsam Atif (BSCE22012)
Umair ul Hassan (BSCE22032)
Muhammad Khizer naveed (BSCE22042)

The Need for a Modern, Accessible Security Solution

Flaws of Traditional Security

High Cost Significant investment in hardware and professional installation.
Subscription Fees Reliance on proprietary monitoring services with mandatory monthly fees.
Complex Installation Often requires hardwiring and invasive setup procedures.
Limited Access Lack of simple, direct remote monitoring for the user.

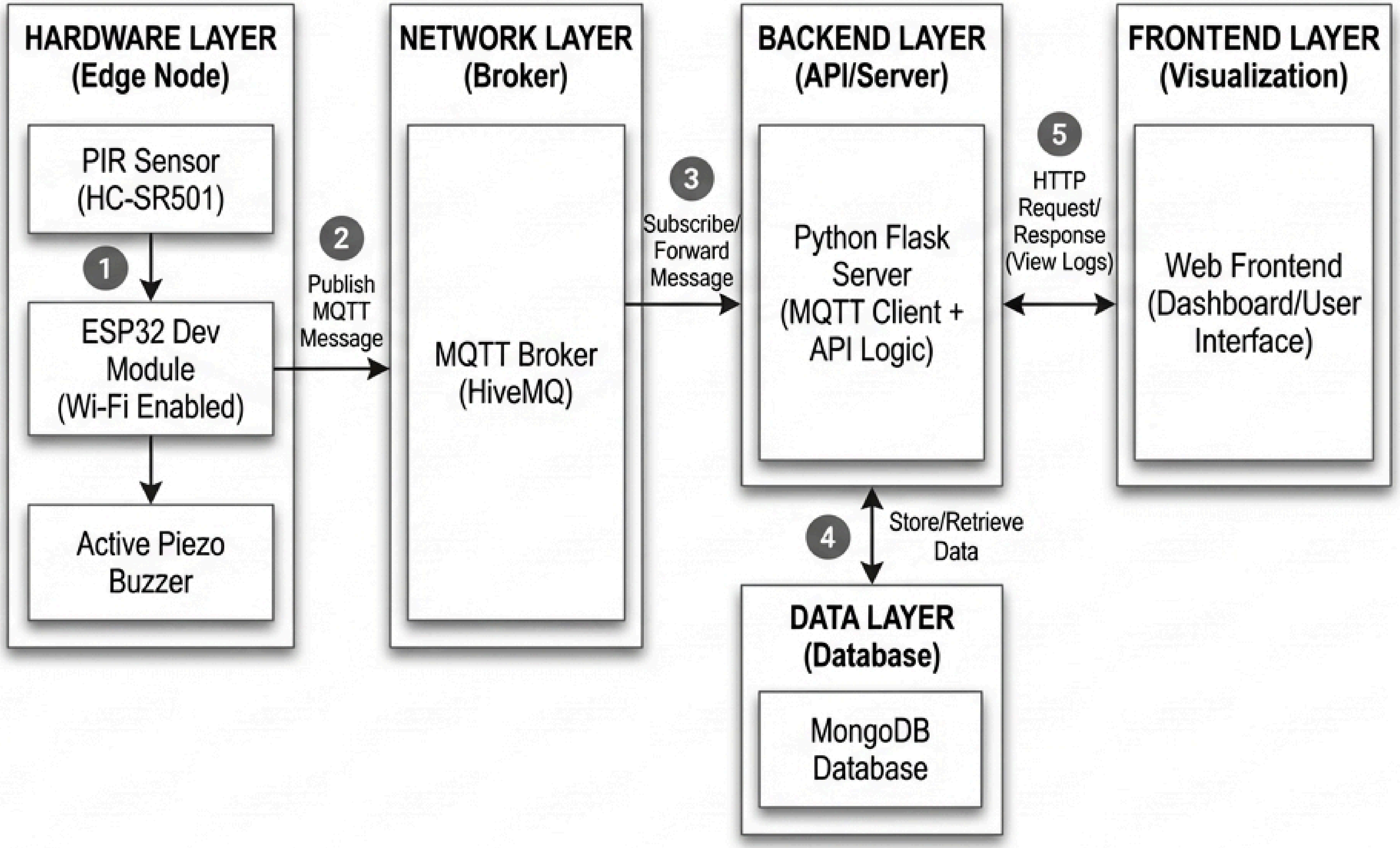
Our System’s Core Principles

Smart & Wireless Utilizes ESP32 and MQTT for seamless IoT connectivity.
Cost-Effective Built with affordable, off-the-shelf components without recurring fees.
Remote Monitoring Web-based dashboard to view security logs from anywhere.
Real-Time Alerts Instant local alarm via a buzzer and persistent cloud-based data logging.

System Architecture: From Motion Detection to User Alert

A 5-step process from sensing to visualization.

1. **SENSE:** A PIR Sensor detects motion and sends a HIGH signal to the ESP32.
2. **PUBLISH:** The ESP32 publishes an alert message to a specific topic on the HiveMQ MQTT broker.
3. **PROCESS:** A Python Flask server, subscribed to the topic, receives the message.
4. **STORE:** The server logs the event details (timestamp, location) into a MongoDB database.
5. **VISUALIZE:** The user accesses a web dashboard which fetches and displays the complete security log from the server.



HARDWARE LAYER (Edge Node)
PIR Sensor (HC-SR501)
1
ESP32 Dev Module (Wi-Fi Enabled)
Active Piezo Buzzer
2
Publish MQTT Message
NETWORK LAYER (Broker)
MQTT Broker (HiveMQ)
3
Subscribe/ Forward Message
BACKEND LAYER (API/Server)
Python Flask Server (MQTT Client + API Logic)
5
HTTP Request/ Response (View Logs)
FRONTEND LAYER (Visualization)
Web Frontend (Dashboard/User Interface)
4
Store/Retrieve Data
DATA LAYER (Database)
MongoDB Database

NotebookLM

Project Timeline & Current Status

Hardware and connectivity layers are complete. Backend development is underway.

Week	Phase	Status
Week 1	Hardware & Circuit Assembly	**In Progress**
Week 2	ESP32 Programming (Wi-Fi & MQTT)	**In Progress**
Week 3	Backend & Database Setup (Flask & MongoDB)	**In Progress**
Week 4	Full System Integration & Frontend Development	□ **Upcoming**
Week 5	Final Testing & Documentation	□ **Upcoming**

Next Immediate Step:
Finalize the Python server logic to process MQTT messages and establish the connection to the MongoDB database for data persistence.

Technical Foundations & References

Our project is built upon robust, industry-standard technologies and documentation.

Core Protocols & Standards

- MQTT Essentials - HiveMQ
- HiveMQ Public MQTT Broker

Hardware & Microcontroller

- ESP32 Technical Reference Manual - Espressif Systems
- PubSubClient Arduino Library

Software & Backend

- Flask Web Framework Documentation
- Eclipse Paho MQTT Python Client
- MongoDB PyMongo Driver Documentation