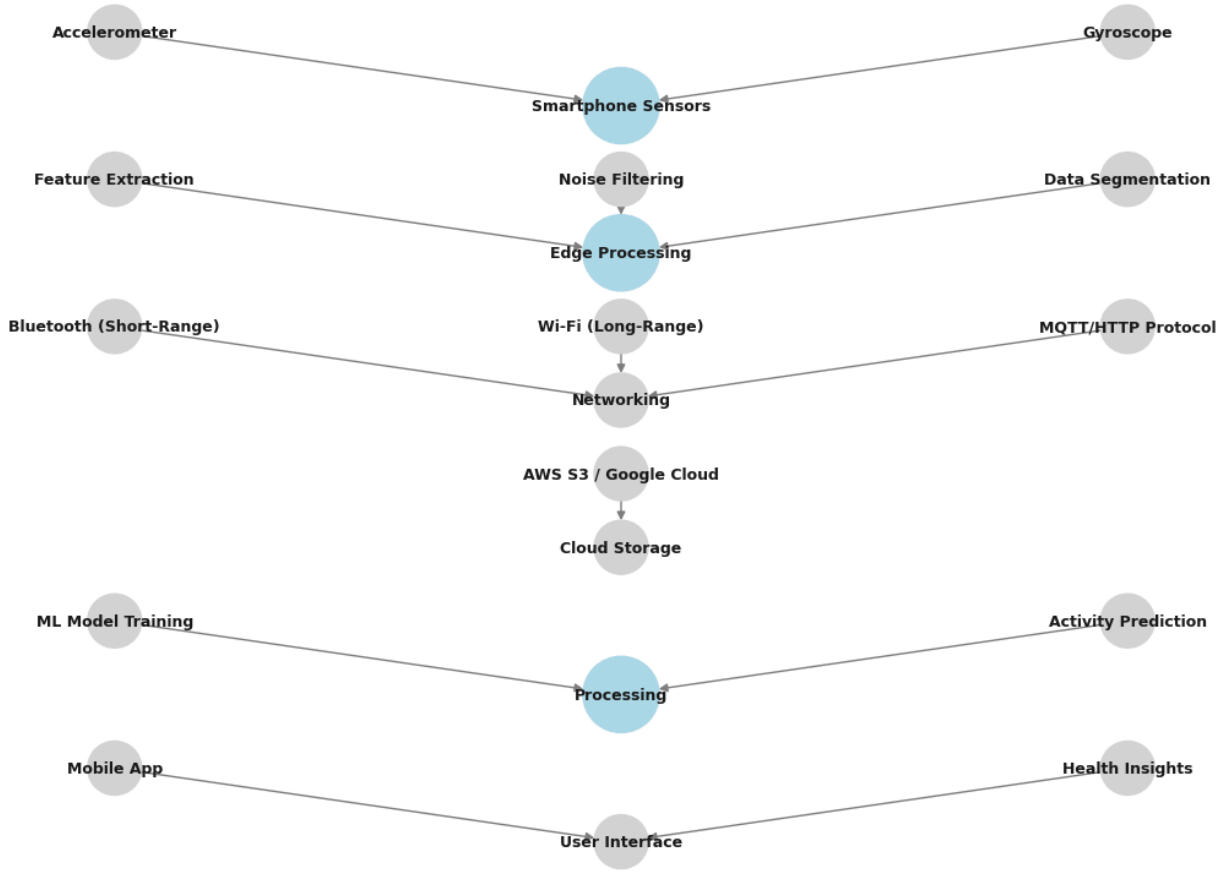
IoT System Design

Diagram
Below is the IoT System Design Diagram, which illustrates the system's components and data flow:



https://drive.google.com/file/d/1Mcdfp0UCmfV8SBa_-pnz1XoiF7pxiWfM/view?usp=drive_link

---

1. Sensors
The IoT system uses two key sensors embedded in the smartphone to collect activity data:

- **Accelerometer**:
  - **Type:** Measures linear acceleration along the X, Y, and Z axes.
  - **Specifications:**
    - Range: ±2g to ±16g.
    - Resolution: Up to 16 bits.

    - Sampling Frequency: 50 Hz.
  - **Role:** Captures movement data to detect changes in user activity patterns (e.g., walking, sitting).
  - **Limitations:**
    - Sensor noise can affect accuracy.
    - Continuous operation may drain the device's battery.
    - Requires periodic calibration to maintain accuracy.

- **Gyroscope**:
  - **Type:** Measures angular velocity along the X, Y, and Z axes.
  - **Specifications:**
    - Range: ±250°/s to ±2000°/s.
    - Resolution: Up to 16 bits.
    - Sampling Frequency: 50 Hz.
  - **Role:** Tracks rotational movements, complementing accelerometer data to classify complex activities.
  - **Limitations:**
    - Noise from small vibrations may cause inaccuracies.
    - Higher power consumption compared to accelerometers.
    - Requires periodic recalibration.

---

2. Edge Processing
The smartphone acts as the edge device in this system, performing basic data preprocessing tasks before transmission to the cloud.

- **Purpose of Edge Processing**:
  - Reduces the size of raw data, minimizing network usage.
  - Filters out noise to improve data quality.

- **Tasks Performed**:
  - Combine accelerometer and gyroscope readings into a single magnitude metric:
    \[
    Magnitude = \sqrt{X^2 + Y^2 + Z^2}
    \]
  - Apply a low-pass filter to remove noise.
  - Segment time-series data into windows for activity classification (e.g., 5-second intervals).

- **Requirements**:
  - Minimum hardware specifications:
    - 2 GB RAM.
    - Quad-core processor.
  - Software for real-time computation (e.g., lightweight Python libraries like NumPy and SciPy).

---

3. Networking
The processed data is transmitted from the smartphone to cloud storage via two networking options:

- **Short-Range Communication**:
  - **Method:** Bluetooth.
  - **Role:** Transfers data to nearby edge gateways or other devices.
  - **Limitations:** Limited range (10–30 meters).

- **Long-Range Communication**:
  - **Method:** Wi-Fi.
  - **Role:** Sends data directly to cloud storage.
  - **Limitations:** Dependent on network stability and bandwidth availability.

- **Messaging Protocol**:
  - HTTP or MQTT is used to communicate between the smartphone and cloud storage.
  - MQTT is preferred for low-latency real-time updates.

- **Transmission Frequency**:
  - Processed data is transmitted every 10 seconds to ensure near real-time updates.

- **Security Measures**:
  - Data is encrypted during transmission using HTTPS to ensure privacy and security.

---

4. Data Storage and Processing
The system uses cloud storage and processing services to manage and analyze data.

- **Storage**:
  - **Platform:** AWS S3 or Google Cloud Storage.
  - **Scalability:** Supports distributed storage to handle growing datasets.
  - **Backup and Redundancy:** Implements automated backups to prevent data loss and ensure reliability.

- **Processing**:
  - **Platform:** Python-based tools such as TensorFlow, Pandas, and NumPy.
  - **Tasks Performed:**
    - Train machine learning models for activity classification.
    - Generate insights, such as predicting sedentary behavior trends.

- **Insights Generated**:
  - Example: Predict sedentary behavior patterns and send alerts to encourage movement.
  - Classify activities in real time (e.g., walking, sitting).
  - Provide personalized health recommendations based on activity patterns (e.g., reminders to exercise).

Summary
This IoT system integrates multiple components, including sensors, edge devices, networking, and cloud processing, to deliver real-time health insights. The careful design of each element ensures efficiency, scalability, and reliability. Advanced security measures, periodic backups, and redundancy further enhance the system's robustness and usability.