Goal Statement

Project title: SleePose

Team members: Shareef / Cherry / Samar / Toby

Describe the Problem

Problem Statement:

Sleep posture and head alignment are essential factors in sleep quality and long-term musculoskeletal health. Many individuals suffer from poor sleep posture, which can lead to chronic neck, shoulder, or back pain. Existing solutions primarily focus on manual pillow adjustments or wearable devices to monitor vitals, but there is little integration of high-resolution posture sensing with an automatically adjustable pillow.

***SleePose** aims to solve this problem by integrating:*

1. A network of “pixels” (pressure or capacitive sensors) within both the pillow and bed sheet to detect sleep posture.

2. Vitals monitoring via a wearable device for deeper insight into sleep quality and physiological stress.

3. A bottom, height-adjustable layer of the pillow that can adapt based on posture, vitals, and an ML model’s predictions to maintain optimal head and neck alignment throughout the night.

Functional Requirements

Below are the core functionalities of the SleePose system, along with target performance metrics. Each metric includes units where applicable:

1. **Posture Sensing Accuracy**

    - The sensor matrix (pressure or capacitive) must detect body/head position with at least **90% accuracy** in differentiating common sleep postures (back, side, stomach).

2. **Adjustment Response Time**

    - The adjustable layer in the pillow should reposition its height or “terrain” within **5 seconds** of detecting a significant posture shift.

3. **Sensor Matrix Resolution**

    - The bed sheet and pillow layer “pixels” should have a resolution of at least **5 cm x 5 cm** per sensor “pixel,” allowing for fine-grained capture of pressure distribution.

4. **Wearable Vitals Integration**

    - The system should be able to capture heart rate and SpO2 from a wearable device at least **once every minute** and fuse this with posture data in near-real-time.

5. **Power Consumption**

    - The entire system (pillow sensor, adjustable motors/actuators, and computing) should operate within **<10 W** on average under normal usage to remain safe and energy-efficient.

6. **Data Connectivity**

    - The system must transmit data (posture and vitals) to a local processing unit or a companion app via Bluetooth or Wi-Fi at least **1 upload per second** for near-real-time visualization.

7. **Weight/Load Handling**

    - The adjustable layer mechanism should support an adult’s head weight of up to **10 kg** (to cover pressure points plus any incidental load) without mechanical failure.

8. **Noise Level**

    - The adjustment mechanism’s noise should not exceed **30 dB** to avoid waking the user at night.

9. **Dimensions**

    - The adjustable pillow layer thickness range should vary from **0 mm to 40 mm** (adjustable height) to accommodate different preferences.

Engineering Design

5.1 Overall System Architecture

1. **Sensors**
   - **Pressure or Capacitive Patches** on both the bedsheet and the pillow.
   - Conductive thread runs through each “pixel,” creating an X-Y sensor matrix.
2. **Pillow Structure**
   - Two layers:
     - **Top (Comfort) Layer** – Traditional pillow material for user comfort (e.g., memory foam or latex).
     - **Bottom (Adjustable) Layer** – An array of small, motor-driven supports (or an air bladder array) that can raise or lower to align the user’s head and neck optimally based on sensor feedback.
3. **Wearable Device**
   - A standard smart band or smartwatch that measures vitals (heart rate, SpO2, etc.).
   - Communicates with the control unit for data fusion.
4. **Control and Processing**
   - **Local Microcontroller or Single Board Computer (e.g., Raspberry Pi, ESP32)** to:
     - Aggregate data from sensor matrices in bed sheet and pillow.
     - Receive vitals from wearable over Bluetooth/Wi-Fi.
     - Run the Machine Learning (ML) inference to determine the user’s posture and recommended pillow height adjustments.
   - **Adjustable Layer Controller** – Motors or actuators that respond to the commands from the microcontroller to raise or lower individual “pixels.”
5. **Software Stack**
   - **Firmware** (C/C++ or MicroPython for an ESP32, for instance) to handle sensor data polling and actuator control.

- **ML Model** – Possibly a small neural network or decision tree model running locally or on an edge device, trained to correlate posture data + vitals with optimum pillow shape.
- **User Interface** – A companion mobile or desktop application that displays posture, sleep quality metrics, and recommended settings.

5.2 Physical Design Sketch

Pillow Layout - 4 Layers

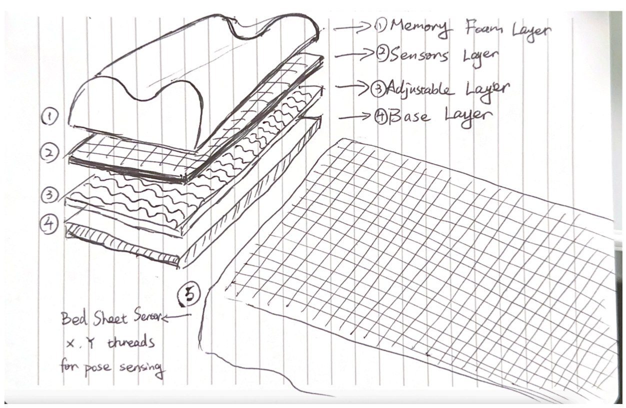
Top Comfort Layer | Sensor Layer | Adjustable Layer | Base Plate

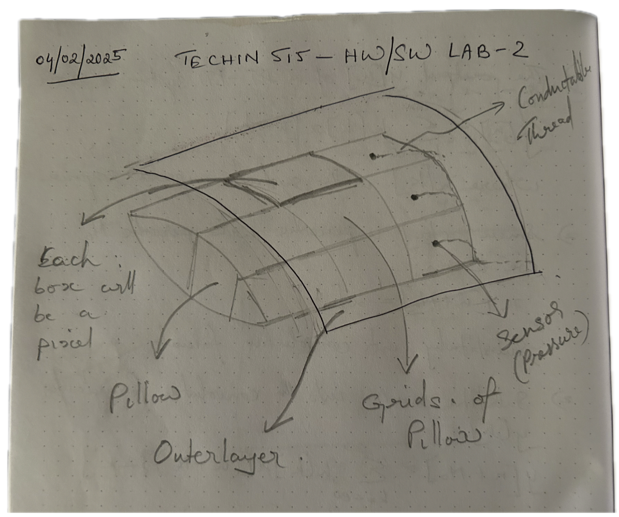
The Adjustable layer will again be a grid of inflatable /deflatable pockets, which when needed (based on the sensor attached to it) can be actuated to a certain position.
The actuated level of the grid pouch in combination with the other grid pouches will determine the shape of the pillow.

The shape of the pillow over time will be analysed through Machine Learning to determine the optimum.

Bedsheet Sensor Design - 1 Layer

Conductive Thread pixel pressure Matrix



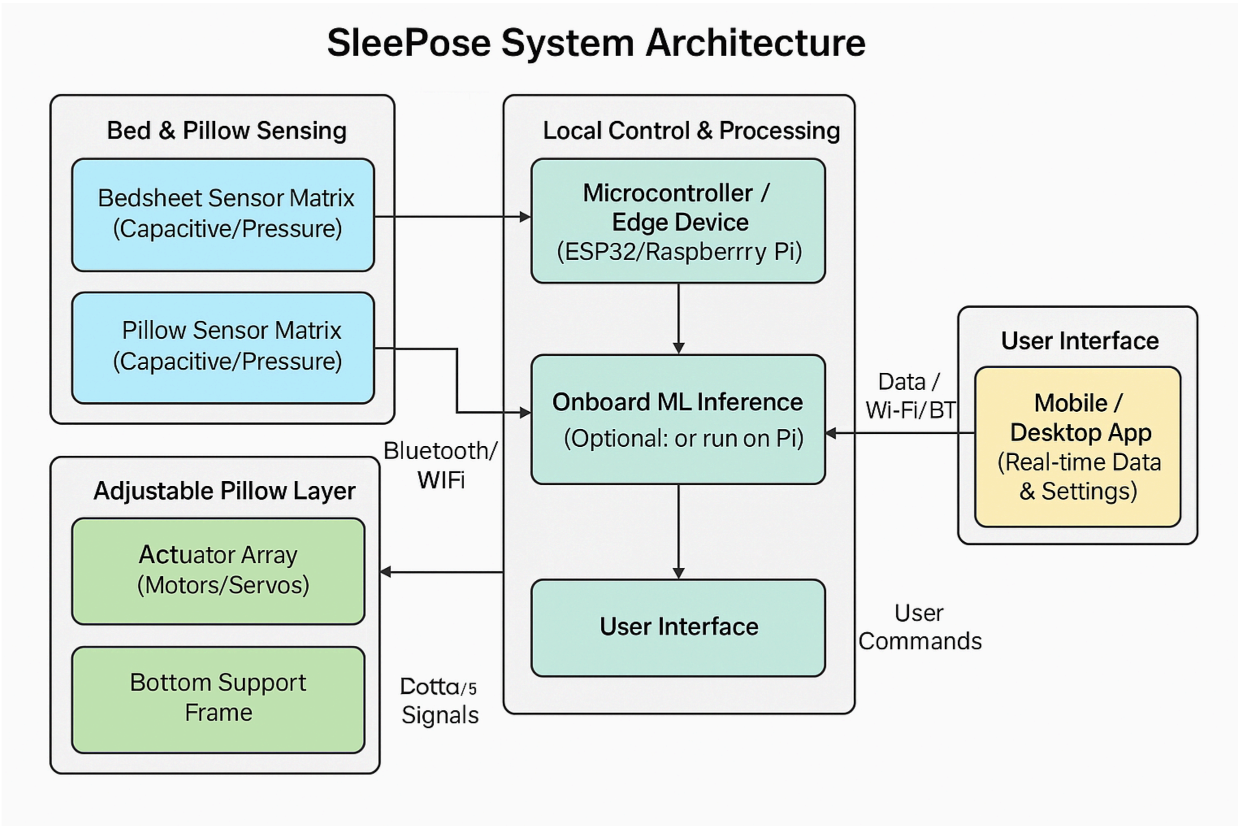


1. **User Interface Wireframe**
    - **Dashboard**: Sleep posture status, posture accuracy/score.
    - **Vitals Panel**: HR, SpO2, respiratory rate trends.
    - **Adjustment Status**: Shows real-time pillow profile.
    - **Settings**: For calibrating sensor sensitivity, ML model parameters, and scheduling.

5.3 Components and Integration

1. **Commercial Components**
    - Conductive thread (silver-plated nylon or equivalent).
    - Small servo motors (lower cost) or linear actuators (higher accuracy) for pillow height adjustment.
    - Smart wearable device (standard smartwatch or fitness band).
    - Microcontroller (ESP32 or Raspberry Pi).
2. **Custom-Built Components**
    - **Sensor Matrix Fabric Layer**: Sewn with conductive thread to create discrete “pixels.”
    - **Pillow Structure**: Custom foam shape or layered design that houses adjustable columns and keeps them aligned/stable.
3. **Key Design Specifications**
    - **Sensor Matrix Operating Voltage**: ~3.3 V or 5 V.
    - **Microcontroller**: Should have enough ADC channels or an external ADC multiplexer to read all sensor “pixels.”
    - **Motor Torque**: Sufficient to lift/adjust the pillow’s load within 5 seconds.
    - **Mechanical Range**: Each support column must extend/retract a specified range (0 to 40 mm).
    - **Thermal Management and Safety:** Given the nature of bedding and the device being next to the user’s head, we will incorporate the following safety measures:
        1. Low voltage (5 V) to minimize heat generation
        2. Ventilated compartments
        3. Use flame-retardant foam

System architecture



6. GitHub

https://github.com/shareefjasim/HWSW_II

7. Testing Plan

1. **Posture Sensing Accuracy Test**
   - **How**:
     - Have multiple volunteers lie in various positions on the bed or pillow.
     - Manually record the actual posture, compare it with the system's detected posture.
   - **What**:

        - Compare the system's classification of posture to the ground truth (visual/manual observation).
    - **Equipment**:
        - Volunteer participants, camera or manual observation records.
    - **Success Criteria**:
        - ≥90% posture detection accuracy over a set of test positions.

2. **Adjustment Response Time**
    - **How**:
        - Introduce a sudden shift in posture.
        - Measure time from posture change detection to completion of pillow adjustment.
    - **What**:
        - Response time in seconds.
    - **Equipment**:
        - Stopwatch or software timer.
    - **Success Criteria**:
        - Average response time ≤ 5 seconds.

3. **Sensor Matrix Resolution & Data Quality**
    - **How**:
        - Place standardized weights or shapes at known positions on the sheet/pillow.
        - Verify the sensor output correlates to the correct grid location within ±5 cm.
    - **What**:
        - Accuracy of localization (difference in cm between actual and reported position).
    - **Equipment**:
        - Test weights, measuring tape, reference grid.
    - **Success Criteria**:

- Proper detection of location within ±1 pixel error margin (±5 cm).

4. **Wearable Vitals Integration**
    - **How**:
        - Stream wearable HR/SpO2 data into the microcontroller.
        - Validate data frequency matches the spec (≥1 reading/min).
    - **What**:
        - Check for data packet timestamps, continuity, and correctness (no dropped or corrupted data).
    - **Equipment**:
        - Wearable device, microcontroller, data logging software.
    - **Success Criteria**:
        - Consistent data reception with minimal (<5%) data packet loss.
5. **Power Consumption**
    - **How**:
        - Use a power meter to measure average consumption over an 8-hour period.
    - **What**:
        - Power usage in watts (W).
    - **Equipment**:
        - A watt meter or bench power supply with real-time monitoring.
    - **Success Criteria**:
        - <10 W average consumption over the measurement period.
6. **Noise Level**
    - **How**:
        - Place a decibel meter near the pillow while the motors adjust the bottom layer.
    - **What**:
        - dB measurement during operation.
    - **Equipment**:

        - Decibel meter (or phone app with calibration).
    - **Success Criteria**:
        - Maximum noise ≤30 dB at normal usage distance.

8.1 Timeline of Tasks and Responsibilities

Task Description	Start Week	End Week
Finalize Sensor Matrix Design	Week 1	Week 2
Conductive Thread Integration	Week 2	Week 3
Build/Assemble Pillow Adjustable Layer	Week 3	Week 4
Microcontroller Firmware Setup	Week 5	Week 6
ML Model Training & Integration	Week 5	Week 7
System Integration & Debugging	Week 7	Week 8
Preliminary Testing & Validation	Week 9	Week 9
Final Documentation & Presentation	Week 10	Week 10

- ***Weeks 1–2**: Focus on the **design phase**, ensuring Cherry has time to complete the sensor matrix design.*
- ***Weeks 3–4**: The **fabrication and physical integration** phase; Shareef and Samar begin working in parallel.*
- ***Weeks 5–6**: **Firmware and data communication logic** is developed by Toby, while Cherry simultaneously trains and integrates the ML model.*

- ***Weeks 7–8**: The **system is brought together and debugged** with input from all members.*

- ***Week 9**: Dedicated to **testing and validation**.*

- ***Week 10**: Wrap up with **final documentation, polishing, and presentation**.*

8.2 Estimate Remaining Purchases

An example purchasing table with placeholder items, quantities, costs, and suppliers:

Item	Number	Cost (USD)	Supplier
Conductive Thread Spool (100m)	2	20	SparkFun/Adafruit
Servo Motors (micro, high-torque)	10	100	Amazon
Foam/Memory Foam Pillow Base	2	50	Local Supplier
ESP32 Dev Board	2	30	Adafruit
ADC Multiplexer (for sensor reading)	2	15	Digi-Key
Miscellaneous Electronics (wires, etc.)	1	20	Digi-Key/Amazon
Wearable Device (smart band)	1	50	Local Store