Fourth Trimester Health

Developing a wearable and environmental sensor system for the postpartum period.

Rapid Prototyping of Computer Systems
Spring 2025

Faculty: Nikolas Martelaro, Asim Smailagic
Editors: Adrian Cantu, Nadia Palar, Fu-Cheng Pan, Ember Shan, Ray Xia, Yuchen Dai

April 30, 2025

Table of Contents

Table of Contents 2
1. Overview 8
2. Problem Definition 9
3. Initial Solution Concepts 10
3.1 Baseline Scenario 10
Current State of Postpartum Care 10
Reimagined/Preferred State of Postpartum Care 10
3.2 Initial Visionary Scenario 11
Value Flow Model 11
Short-Term Care 12
Long-Term Care 13
Identifying Problem Areas 14
3.3 Revised Visionary Scenario 14
Storyboards 14
Discharge and Entering Postpartum Period 14
Sensor Setup 15
Model Building 15
Front End 16
Provider Side 16
3.4 User Archetypes 17
4. System Architecture 19
5. Subsystems 20
5.1 HCI 20
5.1.1 Research on the Postpartum Period 20
5.1.2 Symptoms & Complications for Various Postpartum Risks 25
5.1.3 Functional Requirements 26
HCI Subteams 29
5.1.4 Service Design 29
Research 29
Prototyping 30
Prototype: Support Notifications 31
Purpose 31
Methodology 31
Recommendations 33

Prototype: Privacy + Data Sharing 33
Purpose 33
Methodology 33
Findings 34
Recommendations 35
Prototype: Emotion Tracking 35
Data Collection Approach + Onboarding Design 35
Severity Scores 38
Anomaly Notification Phrasing 38
Anomaly Severity Score Examples 39
Example Push Notifications: 42
Emotional Input 43
Reflection Questions Based on Edinburgh Postnatal Depression Scale 44
5.1.5 Data Visualization 45
Research 45
Synthesis & Development of Design Principles 46
Provider and User Interface Development 47
Integration 50
5.1.6.1 User Interface–Patient Mobile App 51
5.1.6.2 User Interface–Provider Website 57
Provider Research 57
Using Research to Inform Flows 58
Transitioning from Lo-Fi to Hi-Fi Prototypes 60
Prototype at the Conclusion of Phase 2 63
5.1.7 Wearable HCI 63
5.1.7-1 Interaction Flow Diagram 63
5.1.7-2 Product Exterior and Material Selection 65
5.1.7-3 Determining User Interface Hardware 66
5.1.7-4 Designing User Interface, Cues, and Feedback Mechanism 66
5.1.7-5 Scenario Simulation and User Feedback Mechanisms 68
5.1.7-6 Prototype Video Production 69
5.1.7-6 Final Prototype design 70
5.1.7-8 Final Prototype printing and assembly 71
5.1.8 Environment HCI 72
5.1.8.1 Research 72
5.1.8.2 User Flow 73
5.1.4.3 Base Station Physical Design 74

Ideation & Concept 74
Iteration 1 (Phase 2) 75
Iteration 2 (Phase 2) 76
Final Iteration (Phase 3) 76
5.1.8.4 Base Station Interface Design 77
Wireframe 77
Iteration 1 (Phase 2) 79
Final Iteration (Phase 3) 81
5.1.8.5 Pill Box Design 84
Ideation & Concept 84
Iteration 1 (Phase 2) 85
Final Iteration (Phase 3) 86
5.1.8.6 Final Rendering 89
Phase 2 89
5.1.8.7 Reflection 89
5.2 Provider/User Software 90
5.2.1 Initial Research (Product/Feature Matrix) 90
5.2.1.1 Website Development 90
5.2.1.2 Mobile App Development 92
5.2.2 Final Design Decisions 94
5.2.2.1 Backend Framework: NodeJS and TypeScript 95
5.2.2.2 App Frontend: Flutter 95
5.2.2.3 Website Frontend: React with TypeScript 95
5.2.3 Expected Functionality 96
5.2.3.1 Development Outline 96
5.2.4 Phase 2 Implementation Status 96
5.2.4.1 Task Outline 97
5.2.4.2 Mobile App Frontend Progress 98
5.2.4.3 Website Frontend Progress 99
5.2.4.4 Backend Progress 100
5.2.5 Final Implementation 101
5.2.5.1 Task Outline 101
5.2.5.2 Mobile App Frontend Progress 102
5.2.5.4 App Integrations 107
5.2.5.5 Website Frontend Progress 107
5.2.5.6 Backend 109
5.3 Cloud/Data/Communication Systems 110

5.3.1 Functional Requirements 110
5.3.2 System Architecture 110
5.3.3 Services 111
5.3.3.1 EC2 111
5.3.3.2 Database 113
5.3.3.3 Lambda Function 115
5.3.3.4 S3 Bucket 115
5.4 Intelligence 116
5.4.1 Data Processing 117
5.4.1.1 Data Formatting 117
Raw Data Format and Parameters 117
5.4.1.2 Abnormal Values 121
5.4.1.3 Data Preprocessing Workflow 123
Generated datasets 123
Data Preprocessing Workflow 123
5.4.1.4 Public Datasets 125
5.4.1.5 Planned Data Post Processing (if applicable) 125
Evaluating ML Model Predictions 125
Evaluating LLM Model Responses 126
5.4.2 ML Models 126
5.4.2.1 Data Pipeline Overview 126
5.4.2.2 Anomaly Detection Framework 127
5.4.2.3 Implementation and Operational Results 128
5.4.2.4 Integration with the Overall System Architecture 129
5.4.2.5 Further Improvement of Anomaly Detection Algorithm (Phase 3) 129
5.4.3 LLM 130
5.4.3.1 Prompt Engineering Framework 131
5.4.3.2 Research 132
5.4.3.3 LLM Scripts 132
Anomaly_notification.py 133
Weekly_report.py 135
5.4.3.4 Anomaly Notification Language Design 137
5.4.3.5 Input/Output Cloud Data 138
Anomaly Data Table 138
Weekly Report Data Table 139
Patient Data Table 139
5.5 Wearable Hardware/Firmware 140

5.5.1 System Architecture 141
5.5.2 Microcontroller and Firmware Architecture 142
5.5.3 Sensors 144
5.5.3.1 Primary Wearable Sensor Suite 145
5.5.3.1.1 Pulse Oximeter 145
5.5.3.1.2 Thermistors 145
5.5.3.1.3 IMU 145
5.5.3.2 Secondary Wearable 145
5.5.4 User Interface 146
5.5.4.1 Haptic Motor 146
5.5.4.2 LEDs 146
5.5.4.3 Screen 146
5.5.4.4 Push Buttons 147
5.5.6 Communication 148
5.5.6.1 Primary Wearable to Phone 148
5.5.6.2 Secondary Wearable to Phone 148
5.5.7 Printed Circuit Board 148
5.5.7.1 Schematics 148
5.5.7.2 Layout 150
5.5.7.3 Final Circuit Board 151
5.6 Environmental Hardware/Firmware 151
5.6.1 System Architecture 151
5.6.2 Initial Research 153
5.6.2.1 Main Computer 153
5.6.2.2 Micro-Controllers 153
5.6.2.3 Wireless Protocols 154
5.6.2.4 Air Quality Sensor 154
5.6.2.5 Camera 155
5.6.2.6 Medication Adherence 156
5.6.3 Final Design Decisions 157
5.6.3.1 Base Station 157
5.6.3.2 Ambient Condition Subsystem 158
5.6.3.3 Camera Subsystem 160
5.6.3.4 Medication Adherence Subsystem 162
5.6.4 Final Implementation 162
5.6.4.1 Base Station 163
5.6.4.2 Ambient Conditions and Air Quality Substations 164
5.6.4.3 Medication Adherence Substation 165

6 Design Process 166
Summary Work Logs 166
HCI 166
Hourly Task Breakdown 169
Provider/User Software 169
Hourly Task Breakdown 173
Cloud/Data/Communication Systems 173
Hourly Task Breakdown 178
Intelligence 178
Hourly Task Breakdown 181
Wearable Hardware/Firmware 181
Hourly Task Breakdown 182
Environmental Hardware/Firmware 182
Hourly Task Breakdown 184
Task Dependencies (view in Google Sheet) 184
7 Reflections **189**
Phase 1 189
Provider/User Software 189
Cloud/Data/Communication Systems 189
Provider/User HCI 189
Wearable HCI 190
Environmental Hardware/Firmware 191
Intelligence 191
Wearable Hardware/Firmware 191
Phase 2 192
Provider/User Software 192
Cloud/Data/Communication Systems 193
Service HCI 193
Data Visualization HCI 193
User Interface HCI 193
Wearable HCI 194
Environment HCI 195
Environmental Hardware/Firmware 195
Intelligence 196
Wearable Hardware/Firmware 196
Phase 3 196

Provider/User Software 196
Cloud/Data/Communication Systems 197
Service HCI 197
Data Visualization HCI 198
User Interface HCI 198
Wearable HCI 198
Environment HCI 199
Environmental Hardware/Firmware 199
Intelligence 200
Wearable Hardware/Firmware 201
Blog Posts **202**
Provider/User Software 202
Cloud/Data/Communication Systems 203
Service HCI 204
Provider Data Visualization HCI 206
Patient Data Visualization HCI 207
User Interface HCI 207
Wearable HCI 208
Environment HCI 209
Environmental Hardware/Firmware 210
Intelligence 211
Wearable Hardware/Firmware 215
References 217
Section 1 Overview and Problem Definition 217
Section 5.2 Provider/User Software 218
Intelligence 218
Section 5.1.4 Service Design 219

1. Overview

Partnered with 99P Labs, a subset of the Honda Research Institute, focused on producing and ideating innovative solutions for a better world. Our prompt was to create a monitoring ecosystem – wearable and environmental – that leverages data analytics and machine learning to produce a digital twin of the user. The system's core functionality should focus on utilizing this digital twin to generate insight and direction for improving health and wellbeing.

2. Problem Definition

Our resolution was to produce a fully-integrated and holistic monitoring system to empower and support users during the postpartum period. This grew from our shared sentiment to focus our solution on post-discharge care. The category of post-discharge care includes a rich supply for potential areas of focus. The postpartum period presented a compelling problem space for ideating an impactful product for improving health and wellbeing.

The postpartum period is traditionally defined as the 12 weeks following labor where an individual's body and self is reacclimating to a state that mirrors pre-pregnancy (Paladine et al 485). It represents a critical transition, and there is a growing momentum to refer to the postpartum period as the fourth trimester due to the significant challenges and developments that accompany it (ibid). Specifically, a mother's health is still in a precarious state and can leave her/them vulnerable to a variety of health conditions and risks. Individuals can experience hypertension, eclampsia, hemorrhages, postpartum depression, cardiomyopathy, along with multiple other ailments (Paladine et al 486). Unfortunately, despite the serious complications that can arise during the postpartum period, care during this time is often overlooked. Women routinely visit their doctors during their pregnancy, but they typically only have one visit to their obgyn in the postpartum period, usually at the 6 weeks mark (Eyal and Freihart 2024). Additionally, symptoms indicative of a serious health condition are normalized and deemed to be an expected part of this experience.

The impact of this disproportionately applied attention and marginalization of symptoms has been severe. 53% of pregnancy-related deaths occur during the postpartum period, and frustratingly, 80% of these deaths are preventable ("Four in 5 pregnancy-related deaths" 2022). Women of color face a higher risk, as Black women are three times more likely to die a pregnancy-related death than white women (Matthews et al 2024). Critical health conditions are going untreated: around 50% of women with postpartum depression are not diagnosed, jeopardizing the health and wellbeing of the mother (Sudhanthar et al 2019). Overall, rates of maternal mortality and morbidity are on the rise in the United States (Gunja et al 2024). As a country that operates on the precipice of tech innovation and medical advancement, this is not acceptable. The care gap in the postpartum period is a systemic issue, but the creation of intentional tech can act as a key impediment to the progression of this issue. Our solution will ensure that a mother's health is not overlooked, but instead meticulously tracked and assessed, allowing for the prevention and detection of serious health conditions and ensuring a mother receives the support she/they needs and deserves.

3. Initial Solution Concepts

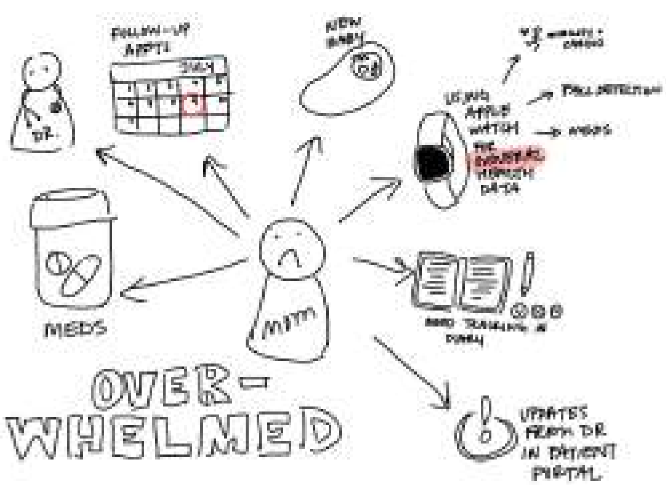
3.1 Baseline Scenario

Our baseline scenario attempted to illustrate the value of the solution that we envisioned. Below, you will see the current state of postpartum care as compared to the preferred or reimagined state that our solution strives to achieve.

Current State of Postpartum Care

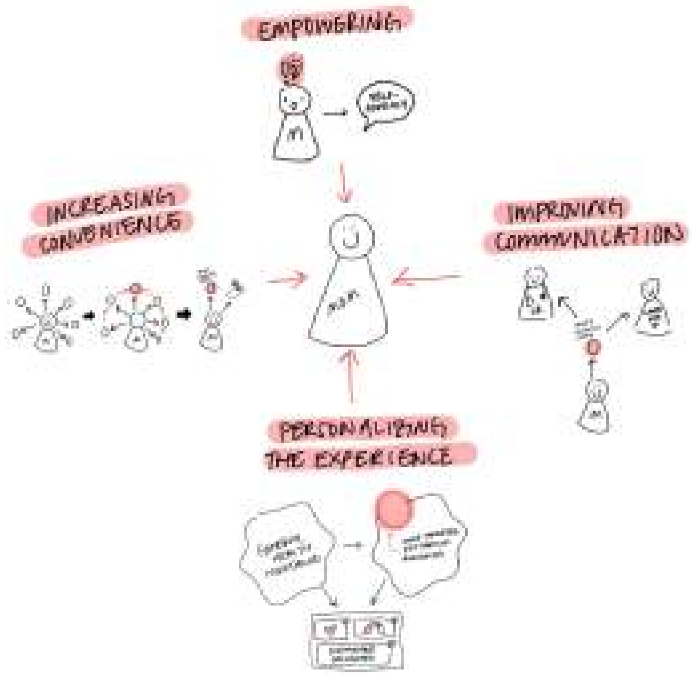
When discharged from the hospital, the user suddenly finds themselves faced with numerous responsibilities and new health symptoms to monitor. They now have a new baby to care for and may need to remember scheduled medications and follow-up appointments. They must keep track of information coming from multiple sources, potentially including general health data from a watch, a diary to track their mood, and the patient portal through which their doctor communicates with them.

The result is fragmented information, often lacking context, which places a higher cognitive load on the user.



Reimagined/Preferred State of Postpartum Care

With our holistic postpartum care solution, we centralize mom and her health by leveraging environmental and wearable sensors to detect anomalies in the mother's health that may warrant a visit to the doctor. This solution would organize all of mom's health information in one dashboard, increase understanding and awareness of postpartum-related health issues, promote self-advocacy, enable data sharing with loved ones and providers, and personalization that makes mom feel like she is in control of her own health.

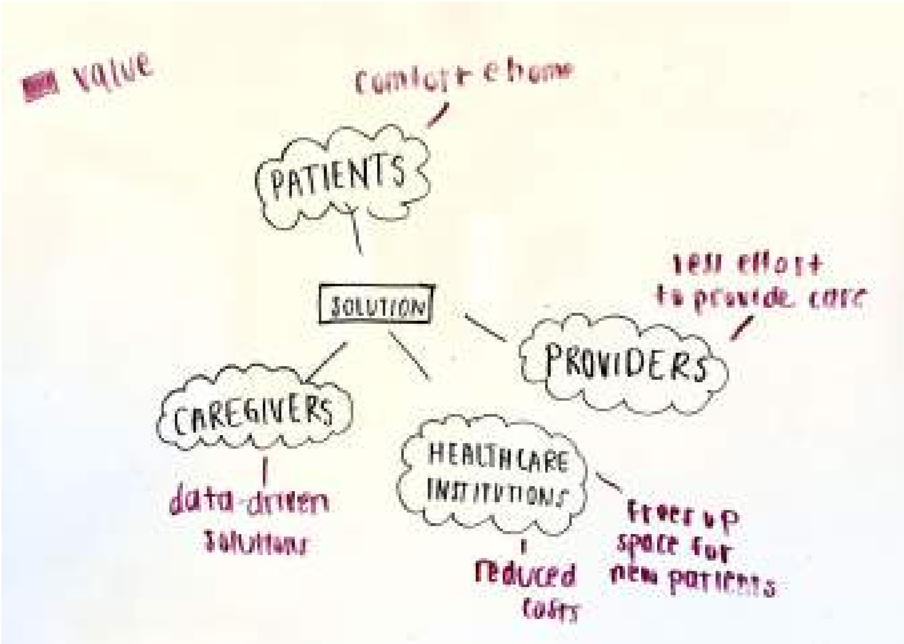


3.2 Initial Visionary Scenario

With this in mind, we developed our initial visionary scenarios in which we spent time sense-making. We created multiple models to systematically break down the problem space into manageable components.

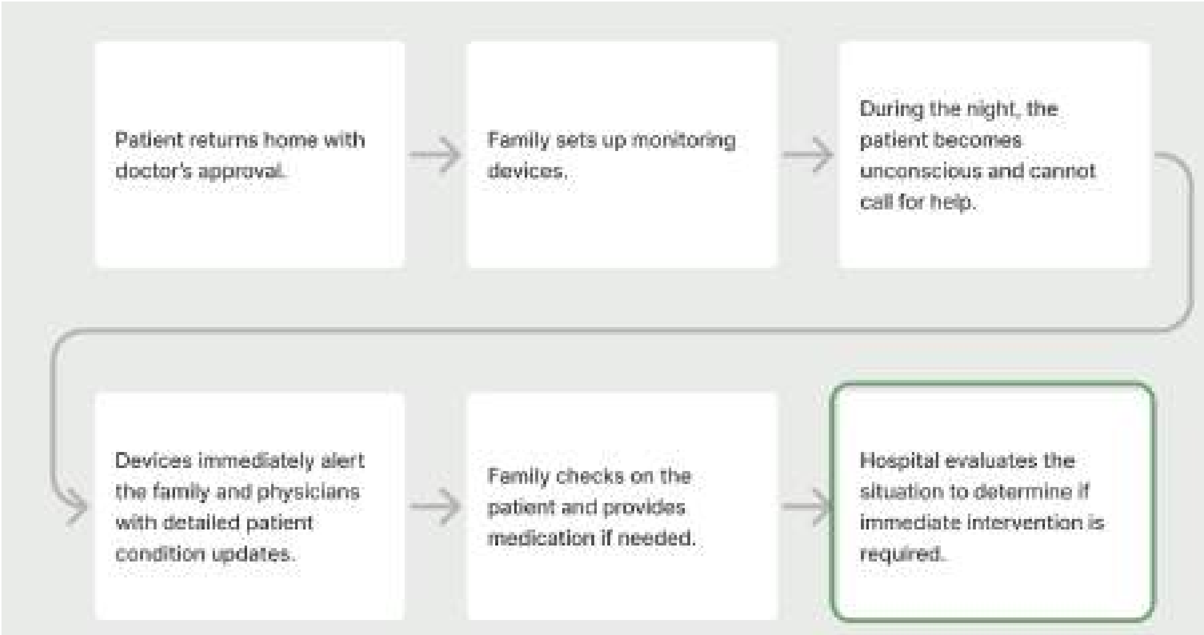
Value Flow Model

This Value Flow Model was a brainstorming exercise we conducted to identify how each of the stakeholders in the ecosystem benefit from this solution. This enabled us to view the service proposition from various angles: as a mother entering the postpartum period, a provider checking in on the mother, a caregiver, and a healthcare institution. Creating this artifact gave us a more holistic understanding of the role this service would play in each stakeholder's life.



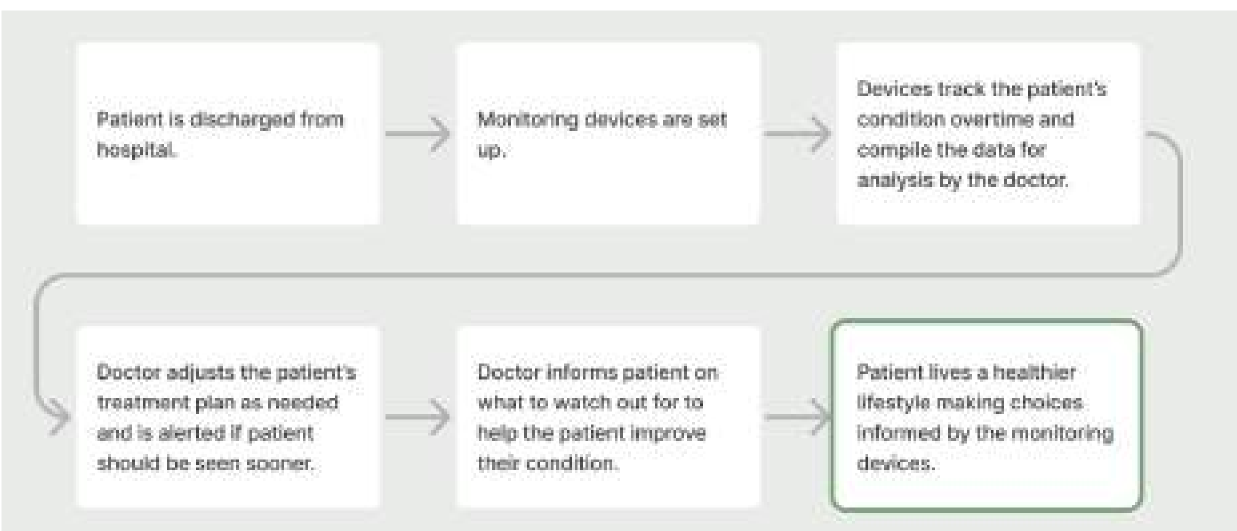
Short-Term Care

We created a flowchart for short-term and long-term care scenarios to help us brainstorm potential implementations of our solution. In providing short-term care, our solution could assist users who need monitoring for a brief period after discharge. For example, if a user is being monitored at home by our sensor stack and falls unconscious without the ability to call for help, the sensors could alert family members and providers about the user's status. The collected data could be used to determine the next steps — whether that involves medication administration or hospital readmission. From this chart, we concluded that our solution should provide important notifications that could prompt immediate action.



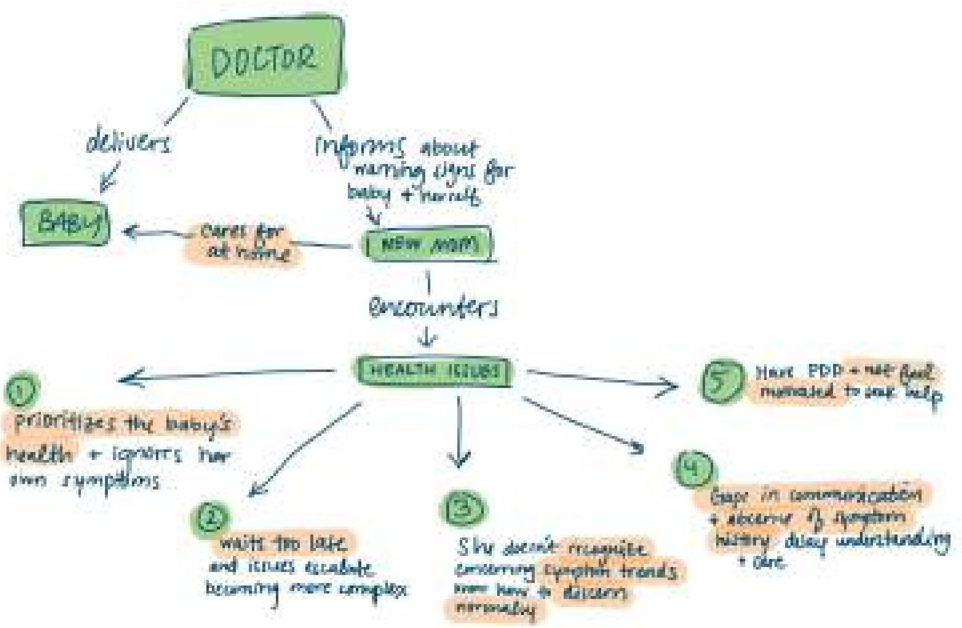
Long-Term Care

This flow chart illustrates a potential long-term care scenario with our services. It begins similarly to the short-term scenario but shifts the focus from immediate action to promoting lifestyle changes. From this chart, we conclude that in our solution, data should be collected, tracked, and visualized over time for both the user and the provider to adjust treatment plans and recommendations.



Identifying Problem Areas

Through this model, we aimed to clarify how our solution could assist users during the postpartum period. We recognized that our solution could support mothers by reducing their responsibilities and cognitive load, proactively identifying issues that can be communicated to doctors, gathering and presenting data to both the mother and the provider, and facilitating care in settings beyond the hospital. We sought to visualize a solution in our storyboards that could address these postpartum concerns in both short-term and long-term scenarios.



3.3 Revised Visionary Scenario

Storyboards

Discharge and Entering Postpartum Period

Imagine this…Mom gives birth and is discharged within a day to go home. Mom is now in the postpartum period, which puts her at *high* risk for numerous conditions (e.g., depression, hypertension, preeclampsia, deep vein thrombosis, hemorrhaging, sepsis, etc).

Sensor Setup

Environmental sensors and wearables are set up to track data points like air quality and heart rate, respectively, to gauge whether she is at risk of postpartum issues. We could also incorporate the opportunity for user-inputted metrics, such as a mood tracker, to help monitor and track the mother's mental well-being.

Model Building

Using the collected data, models identify irregularities, or anomalies, in the mother's health that may indicate the need for a doctor's visit.

Front End

Based on whether an anomaly is detected, the front-end system recommends the patient to be seen by a doctor and/or take a stretch break, nap, etc. and records the sensor data to package and send to the physician.

Front-end system sends recommendations to the mother like more sunshine and rest based on anomalies detected by sensor stack

Depending on the the urgency of the situation, system recommends a doctor's visit

Provider Side

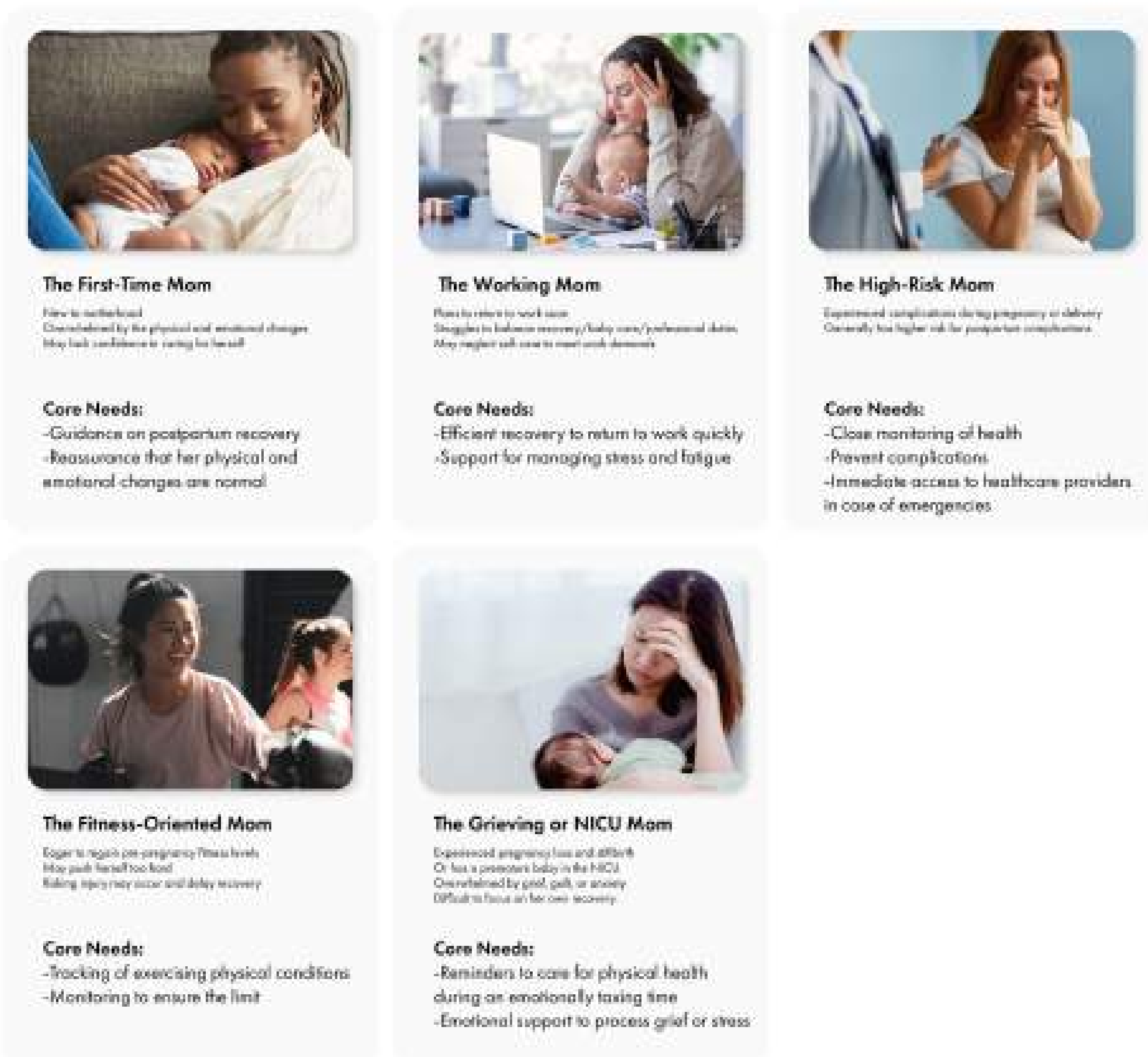
The doctor can easily review the data compiled by the sensor stack and order labs/tests (like blood and urine samples) to help determine a diagnosis. Finally, the mother can return home with an updated care plan and new metrics to look out for.

Dr. reviews collected (and visualized) data with the mother

Dr. orders appropriate lab tests to check for conditions based on the data she reviewed.

Mother returns home with an updated plan of care

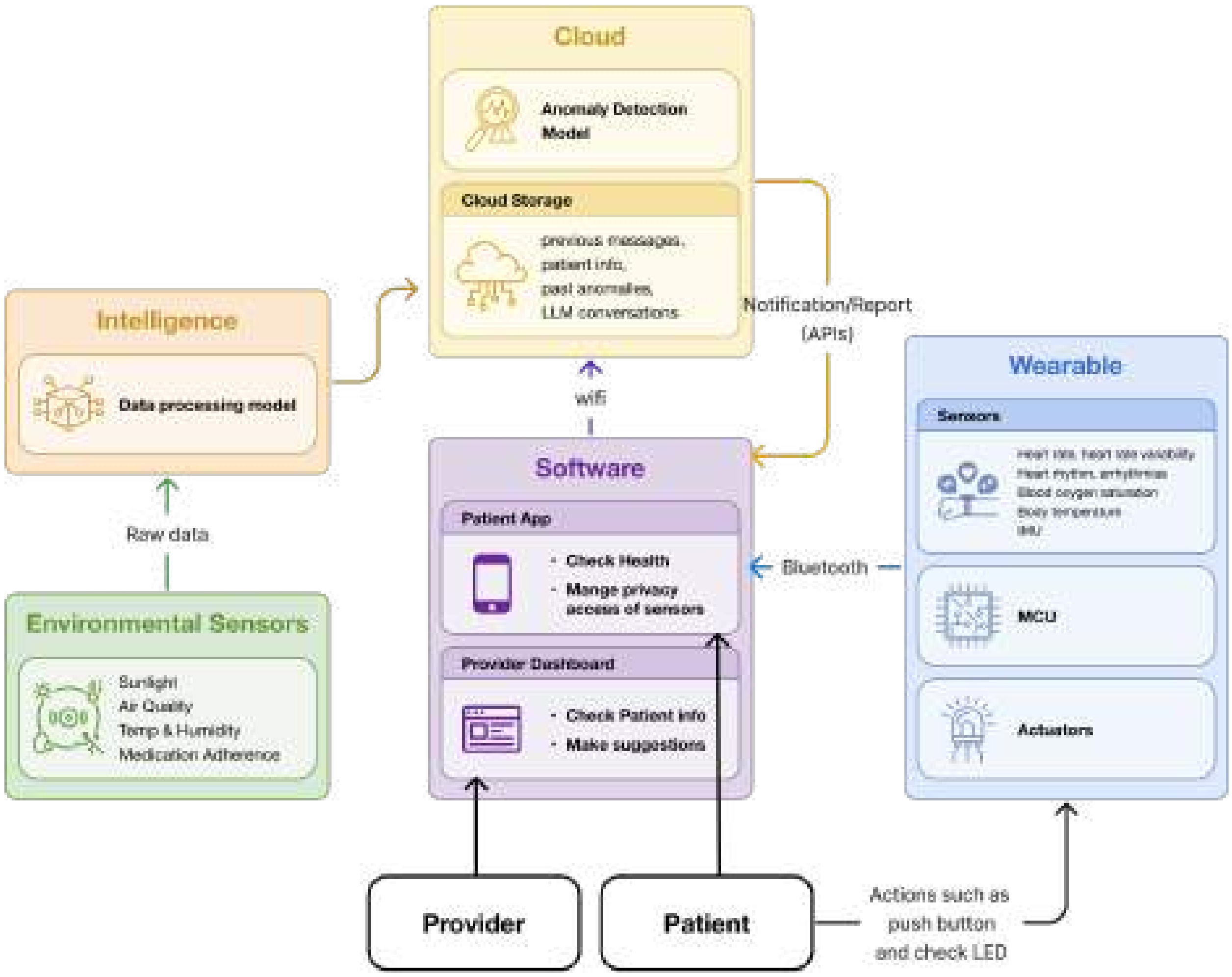
3.4 User Archetypes



We created a set of User Archetypes to empathize with diverse user needs, behaviors, and challenges. For a complex and emotionally charged domain like postpartum care, it is especially important to understand what different users' core needs are and how our product can fulfill and support these needs.

Here, we analyzed five types of mothers. The thought process behind these archetypes was to capture a wide range of postpartum experiences, from joyful to aspiring to stressful to sorrow. The first-time mom navigates through the unknowns, the working mom balances motherhood and career, the high-risk mom monitors her conditions closely, the fitness-oriented mom returns to her pre-pregnancy routines, and the grieving or NICU mom deals with profound emotional and physical challenges. By identifying the core needs and behaviors of each core user, the HCI team can start to tailor features, functionalities, notifications, and support systems. For example, a first-time mother can benefit from educational content and pushing reassurance. This way, we can ensure that the product is inclusive and compassionate.

4. System Architecture



Here's an overview of our system architecture. We offer a mobile app for patients and a web dashboard for providers. Patients will also have a wearable device that tracks multiple vitals and includes actuators for notifications if their phone is not nearby.

For data that is difficult to track via a wearable or phone—such as sunlight exposure, air quality, environmental temperature, and medication adherence—environmental sensors will be used. The raw data is processed through the intelligence model and sent to the cloud. Notifications and reports requiring attention will be sent via Wi-Fi to either the patient's phone or the provider's dashboard as needed.

5. Subsystems

5.1 HCI

5.1.1 Research on the Postpartum Period

The postpartum period, the time following childbirth, is a critical phase in maternal health. This section examines the postpartum timeline, serious health risks for both mother and child, and the role of sensor technology in monitoring and mitigating these risks. The research shared below directly informs the functional requirements for the system.

Postpartum Timeline

1. Acute Phase (6 to 12 hours after delivery)
    a. Risks: This phase is the highest risk period for eclampsia and postpartum hemorrhage, requiring immediate monitoring and intervention.
    b. Essential to monitor:
        i. Blood pressure
        ii. Heart rate irregularities
        iii. Bleeding levels
        iv. Excess swelling
2. Subacute phase (24 hours to 2-6 weeks after delivery)
    a. Risks: A critical period where cardiomyopathy, postpartum depression (PPD), and urinary incontinence commonly emerge.
3. Delayed phase (6 weeks to 6 months after delivery)
    a. Chronic conditions like pelvic floor dysfunction, painful intercourse, and uterine prolapse become apparent.

Serious Risks to the Mother

1. Postpartum cardiomyopathy: A rare but life-threatening condition where the heart muscle weakens after childbirth, leading to heart failure and reduced blood circulation
    a. Timeline: Most common one month before to 5 months after delivery
    b. Symptoms
        i. Fatigue
        ii. Heart palpitations
        iii. Shortness of breath
        iv. Swelling (edema) in feet and ankles
        v. Dry cough
        vi. Swollen neck veins

   vii. Lightheadedness
   viii. Low blood pressure (hypotension) or blood pressure that drops suddenly upon standing up
   ix. Chest pain
c. Complications
   i. Blood clots
   ii. Arrhythmias
   iii. Cardiac shock
   iv. Severe heart failure
   v. Brain injury
d. Risk Mitigation
   i. Continuous heart rate and oxygen saturation monitoring
   ii. Blood pressure monitoring
   iii. Monitoring of swelling in feet and ankles
   iv. Monitoring for blood clots

2. Postpartum Depression: A mood disorder affecting new mothers, characterized by persistent sadness, fatigue, anxiety, and difficulty bonding with the baby, often triggered by hormonal and environmental factors.
   a. Timeline: Typically develops within the first few weeks to six months postpartum
   b. Symptoms
      i. Persistent sadness or mood swings
      ii. Fatigue or low energy
      iii. Loss of interest in activities
      iv. Difficulty bonding with the baby
      v. Changes in appetite or sleep patterns
      vi. Feelings of hopelessness, guilt, or worthlessness
   c. Risk Mitigation
      i. Continuous monitoring of sleep, heart rate variability, and activity levels using wearables
      ii. Air quality monitoring
      iii. Access to mental health support and counseling
   d. Risk Factors
      i. Air pollution: PM1, PM2.5, PM10, and CO_2 concentrations pose significant health risks to postpartum mothers due to their heightened vulnerability during this period

ii. Exposure to PM10 and PM2.5 specifically during and after pregnancy was associated with increased risks of postpartum depression (Sun et al., 2023)

e. *Basal metabolic rate*:
    i. Regular metabolic testing can identify nutrient gaps or thyroid issues early.
    ii. Lower BMR may indicate reduced energy availability, nutrient deficiencies (e.g., iron, B12) Dysregulated metabolism linked to hormonal imbalances (e.g., cortisol, thyroid hormones) that may increase susceptibility to PPD (Mikkelsen et al., 2020).

f. *Heart rate:*
    i. HRV is associated with heightened stress response and autonomic nervous system dysregulation.
    ii. Lower HRV during the postpartum period correlates with increased anxiety and depressive symptoms (Osborne et al., 2019).

g. *Lower activity levels (step count):*
    i. Reduced physical activity is both a symptom and a risk factor for PPD. Sedentary behavior limits endorphin release, disrupts circadian rhythms, and worsens fatigue, creating a cyclical relationship with mood disorders (Coll et al., 2021).

3. Postpartum preeclampsia: A condition involving high blood pressure and potential organ damage after childbirth, increasing the risk of stroke, seizures, and life-threatening complications if not addressed immediately.
    a. Timeline: Can develop within 48 hours to six weeks postpartum
    b. Symptoms
        i. High blood pressure (>140/90 mmHg)
        ii. Protein in urine (proteinuria) (>300mg)
        iii. Severe headache
        iv. Changes in vision
        v. Swelling in hands, feet, face, and limbs
        vi. Nausea or vomiting
        vii. Abdominal pain
        viii. Decreased urination
        ix. Rapid weight gain
        x. Shortness of breath
    c. Complications
        i. Stroke
        ii. Seizures

    iii. Permanent organ damage (brain, liver, kidney)
    iv. Pulmonary edema (excess fluid in lungs)
    v. Blood clots
    vi. HELLP syndrome
    vii. Death
  d. Risk Mitigation
    i. Blood pressure monitoring
    ii. Urine protein tracking
    iii. Remote patient monitoring with wearable devices
    iv. Monitoring of swelling
    v. Weight monitoring
4. Postpartum hemorrhage (PPH): Severe bleeding after childbirth that can lead to hypovolemic shock, organ failure, or death if not treated promptly.
  a. Timeline
    i. Primary PPH: Within the first 24 hours postpartum
    ii. Secondary PPH: Up to 12 weeks postpartum
  b. Symptoms
    i. Excessive bleeding (>1 liter) after delivery including passing large blood clots
    ii. Sharp drop in blood pressure which causes dizziness and faintness
    iii. Low blood pressure (hypotension)
    iv. Increased heart rate (tachycardia)
    v. Decreased red blood cell count (hematocrit levels)
    vi. Pale or clammy skin
    vii. Pain and swelling in vaginal or perineal area
    viii. Bloating, bruising, or pain in the abdomen (in people who gave birth via cesarean section)
  c. Complications
    i. Hypovolemic shock (organ failure due to blood loss) that can cause falls or temporary loss of consciousness
    ii. Death
  d. Risk Mitigation
    i. Continuous blood pressure and heart rate monitoring
    ii. Wearable sensors to detect hemorrhage-related symptoms
    iii. Monitoring of swelling
5. Postpartum Hypertension: High blood pressure following childbirth that can increase the risk of stroke, seizures, and long-term cardiovascular complications.

   a. Timeline: Most common between 5-7 days postpartum but can develop up to six weeks postpartum
   b. Symptoms
      i. High blood pressure (>130/80 mmHg)
      ii. Headache
      iii. Vision changes
      iv. Chest pain
      v. Shortness of breath
   c. Complications
      i. Seizures
      ii. Stroke
      iii. Organ damage
6. Deep Vein Thrombosis: A condition where blood clots form in deep veins, usually in the legs, which can become life-threatening if they travel to the lungs and cause a pulmonary embolism.
   a. Timeline: Can develop anytime up to six weeks postpartum.
   b. Symptoms
      i. Intermittent pain and tenderness in the leg
      ii. Swelling in one leg
      iii. Redness and warmth in the affected area
      iv. Enlarged veins
      v. Leg ulcers
      vi. Severe headache (if clot travels to the brain)
      vii. Seizures
   c. Complications
      i. Pulmonary embolism: Fatal if untreated; symptoms include chest pain, shortness of breath, coughing blood, lightheadedness, and fainting
      ii. Stroke
   d. Risk Mitigation
      i. Thermal camera monitoring to detect blood clots and temperature differences between legs
      ii. Compression stockings for high-risk individuals
      iii. Laser speckle flow index monitoring (although limited research in this area makes inclusion in the sensor stack unfeasible)
      iv. Wearable smart socks with bioimpedance sensors to detect leg swelling and fluid retention (though feasibility is limited)
7. Sepsis: A severe, life-threatening response to infection that can rapidly lead to organ failure, tissue damage, and death if not treated promptly.

    a. Timeline: Can occur within hours to six weeks postpartum, particularly after surgical births or infections
    b. Symptoms
        i. Fever (>100.4°F)
        ii. Elevated heart rate
        iii. Elevated breathing rate
        iv. Chills
        v. Clammy skin
        vi. Confusion
        vii. Severe pain or discomfort
    c. Complications
        i. Organ failure
        ii. Septic shock
        iii. Death (can occur within 12 hours if untreated)
    d. Risk Mitigation
        i. Continuous temperature and heart rate monitoring
        ii. Breathing rate monitoring

Serious Risks to the Child

1. Air Pollution Exposure: Exposure to fine particulate matter (PM1, PM2.5, PM10) and carbon dioxide (CO_2) that can impair lung development, increase respiratory infections, and raise infant mortality risks during the first year of life (Nazarpour et al., 2023).
2. Ambient temperature: Higher ambient temperatures are associated with increased risk for sudden infant death syndrome (SIDS). Temperatures within the range of 16-20° C are recommended to prevent these risks according to the National Health Service of the United Kingdom.

Accurate data collection from wearable devices is heavily influenced by proper usage and adherence to specific guidelines. Research indicates that user-related factors such as incorrect device placement or inconsistent usage can lead to incomplete or inaccurate data (Cho et al., 2021)

Regarding the pillbox monitor, many mothers are prescribed medications for pain management (more common for cesarean section births).

5.1.2 Symptoms & Complications for Various Postpartum Risks

This chart maps the different symptoms and complications that correspond to common yet major conditions during the postpartum period. It highlights which physiological and clinical indicators are relevant for monitoring each condition, helping to identify early warning signs and potential complications.

	Blood Pressure	Heart Rate	Breathing Rate	Swelling	CO2	Activity	Air Quality	Falls	Shortness of Breath	Protein in Urine	Weight	Blood Clots
Postpartum Cardiomyopathy	✅	✅	✅	✅	✅			✅	✅			
Postpartum Depression	✅	✅				✅	✅				✅	
Postpartum Preeclampsia	✅	✅	✅	✅	✅			✅	✅	✅	✅	✅
Postpartum Hemorrhage	✅		✅	✅	✅			✅			✅	✅
Postpartum Hypertension	✅	✅			✅			✅	✅			
Deep Vein Thrombosis				✅	✅			✅	✅			✅
Sepsis	✅	✅	✅		✅			✅			✅	

5.1.3 Functional Requirements

		Functional Requirement	Subsystem Teams
Setup and Configuration	Device Setup	The system shall provide an intuitive setup experience, guiding users through proper installation to ensure proper functionality.	HCI, Software
	User Profile & Customization	The system shall collect baseline health data, allow users to input personal health conditions, and provide guidance on wearable usage, maintenance, and best practices.	HCI, Software
Monitoring & Data Collection	Wearable Monitoring	The system shall continuously monitor biometric data	Wearable, Wearable HCI
	Environmental Monitoring	The system shall track home environmental conditions, including: Air quality (PM1, PM2.5, PM10, CO2 concentration), temperature &	Environmental

		humidity, sunlight, and pillbox weight	
Intelligence & Data Analysis	Data Processing & Anomaly Detection	The system shall use machine learning models to analyze trends and detect anomalies in health and environmental data.	Intelligence, Cloud
		The system shall detect abnormal physiological and environmental trends based on predetermined thresholds, such as rapid heart rate changes or decreased oxygen levels (e.g., "Your elevated body temperature might be due to a fever.").	Intelligence
Alerts & Pings	Recommended Actions for the User	The system shall send real-time alerts for physiological and environmental anomalies, providing users with severity scores and recommended actions to mitigate potential health risks.	Software, Wearable, Wearable HCI
		The system shall categorize alerts by severity (e.g., 0-5 and associated colors like green = mild, yellow = moderate, red = critical).	
	Prompt Notifications	The system shall ask user if they're experiencing symptoms when anomalies above are detected	Software, Intelligence, HCI
		The system shall allow the user to confirm or decline whether they're experiencing symptoms	
	Involving a Provider	The system shall prompt users to contact their provider for recurring or concerning symptoms	Software, HCI
		The system shall automatically flag high-risk trends for inclusion in provider reports.	
	Emergency Escalation	The system shall detect critical emergencies, notify users with haptic and audible alerts, and escalate to emergency contacts and services when necessary	Wearable, Wearable HCI
Patient Platform (Mobile-Based Dashboard and Reports)	Dashboard & Health Status Overview	The system shall provide a real-time dashboard displaying current health status, recent trends, and alert history.	Software, HCI
		The system shall display when sensor data was last updated	
		The system shall present visual graphs & charts for key health metrics (e.g., heart rate trends, medication adherence).	

	Insights & Trends	The system shall generate weekly health trend reports, allowing users to compare historical data and identify patterns in their health.	
		The system shall provide weekly health trend reports with actionable insights.	
	Recommendations & Well-Being Check-Ins	The mobile app shall offer daily well-being check-ins and suggest personalized health habits.	Software, HCI
		The system shall suggest habit-building actions (e.g., reminders for medication, relaxation techniques, breathing exercises).	
Provider Platform (Web-Based Dashboard & Reports)	Patient Monitoring Dashboard	The system shall present a real-time patient monitoring interface for providers, highlighting high-risk cases and summarizing key trends and alerts.	Software, Intelligence, HCI
	Detailed Patient Reports	The system shall generate comprehensive weekly reports, summarizing patient health trends, integrating electronic health record data, and provide provider-relevant insights for informed decision-making.	
Wearable	Reminder Notifications	The system shall remind users to stand up and walk around if inactive	Wearable, Wearable HCI
		The system shall remind users to take medications if they haven't	
		The system shall remind user to check reports/messages if they haven't	
		The system shall let users dismiss reminders	
	Alert Notifications	The system shall notify user when heart rate exceeds or falls below prescribed threshold	Software, HCI
		The system shall notify user when blood pressure exceeds or falls below prescribed threshold	
		The system shall notify user when body temperature exceeds or falls below prescribed threshold	
		The system shall notify user when the battery is below 10%	

5.1.4 Service Design

Research

Patients
We conducted 7 interviews with mothers who had recently experienced the postpartum period to gain insights into their experiences and identify what support would have been helpful during this critical period. We also performed secondary research by observing discussions in online forums, such as Facebook groups and subreddits for new parents. Despite multiple attempts to engage with users in these groups, our posts were denied due to privacy concerns and measures to protect new parents from scammers. This highlighted the importance of safeguarding people during this vulnerable period.

Providers
Additionally, our subteam has conducted semi-structured interviews with 2 providers, an obgyn and a doula. Additionally, we have consulted another doctor in a more informal manner – through messages – to resolve quicker queries. We had planned to speak with an additional doula, but the doula missed our scheduled meeting and was not responsive to emails for rescheduling. Naimah, a member of the UI subteam, individually interviewed 2 providers to inform her of the design of the provider portal. In total, the HCI has been able to receive some form of feedback, input, and guidance from 5 providers.

The research objectives for these interviews was to receive the provider’s assessment of the concept, understand current frameworks for communication between provider/patient, hear input on the current mockup for the provider portal, and learn how we can design this product to equip providers to care for mothers.

Both the provider and doula were receptive to our product in terms of its approach and its purpose. Dr. Scott, the obgyn, stated that

> “the patient has to become more a part of their healthcare. And the only way to do that is to communicate with them somehow. And this platform that you guys have made would be super helpful because the patient is actually becoming involved in their care.”

This demonstrated to us that our product had the potential to empower a mother to recenter her health and wellbeing during the postpartum period. As our desire to build this product to equip mother's to feel more ownership over their health during a time where they often overlooked, this

Additionally Dr. Scott shared that,

> "we say we want them (patients) to take care of themselves, but then we (healthcare providers) don't give them the tools to take care of themselves. And, um, that's super important. So I think what you guys are trying to do with this platform is super exciting."

Again, this quote showed the value added through our product. There is a mismatch between what the hospital is saying and then what it is actually equipping patients to do. FTH fills this gap, it is a product that actively provides a mother with the necessary toolkit to understand and track their health during the postpartum period.

Overall, our interviews with providers were able to validate the idea of our concept as a whole.

Prototyping

Through our initial interviews with patients, we identified three critical aspects to address: notifications, data privacy, and emotional tracking. We decided on these areas because our interview findings underscored just how significant the impact is of postpartum mood disorders, such as depression and anxiety. New mothers shared the following insights:

- "[Once I had delivered the baby,] I was just left out there, alone."
- "Postpartum depression is rooted in a sense of isolation, loss of autonomy... [It's] a radical change to your life."
- "I went from having a very happy postpartum experience to being highly anxious, having feelings of despair, and not knowing why... Everything was overwhelming."

Although we initially stepped away from using wearable and environmental sensors to detect mood disorders in Phase 1, we realized that we couldn't ignore this crucial risk. As a result, we prioritized emotional tracking and prototyped a new proactive notification system. Instead of just detecting anomalies, this system would send supportive and validating messages to help users feel less alone.

Regarding data privacy, discussions with the cloud and intelligence teams underscored privacy concerns. We aimed to explore users' comfort levels with data sharing with the prototypes.

Prototype: Support Notifications

Purpose

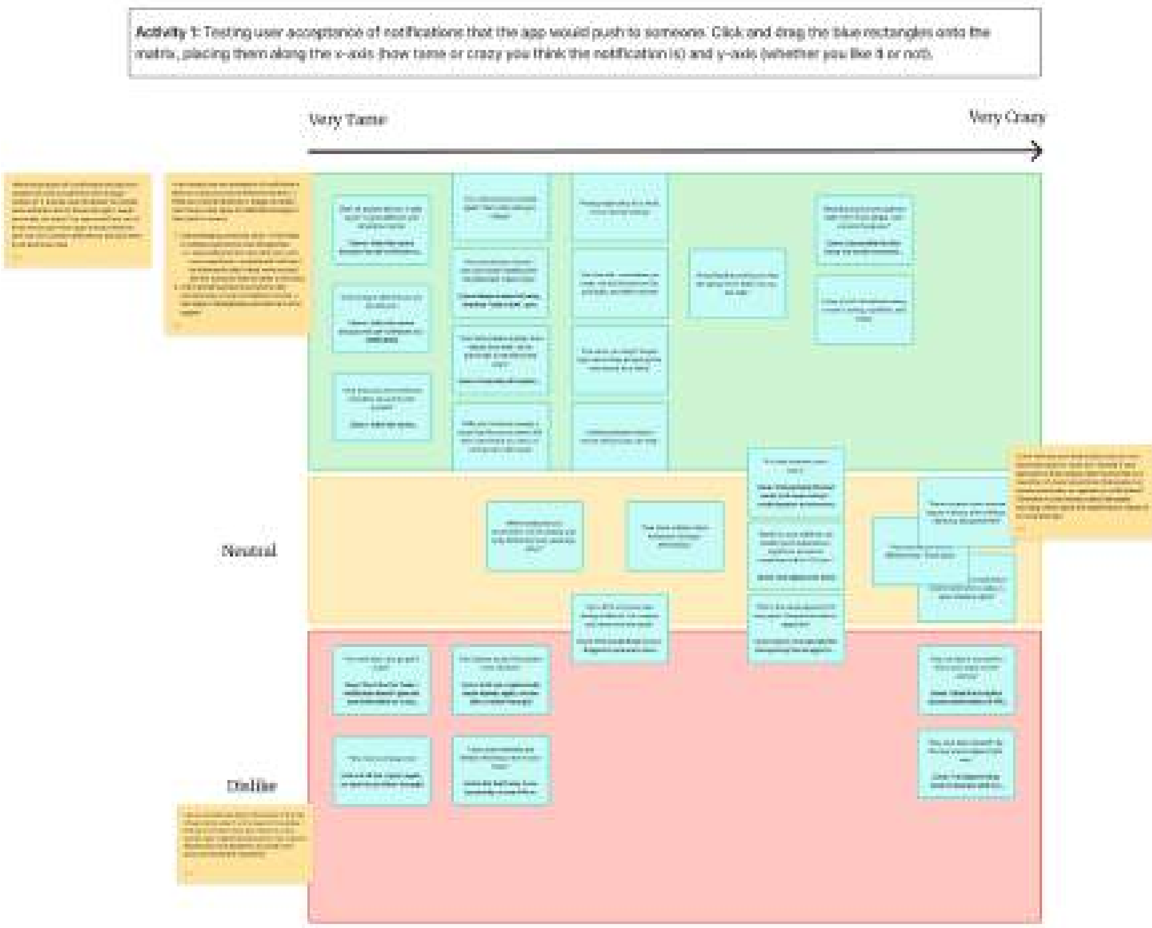
The purpose of this prototype was to understand what type of content mothers wanted in their support notifications and what tone they were receptive to in their notifications. As individuals, who have never experienced the postpartum period, we lacked a comprehensive and definitive understanding of what messaging could actually be helpful for this period. Our interviews had given us a baseline that mothers wanted a form of reassurance about the physical and emotional changes they faced and doubts/struggles they were experiencing.

Methodology

To build upon this baseline, we created a test that would allow us to determine the tone and subject matter that resonated with mothers. We first discussed what scale of tones, what scale of content, and what variety of content to include. We decided to include a scale of tone that ranged from cryptic and/or clinical to upbeat and borderline patronizing. We created a scale of content that ranged from being tame to being provocative. Our notifications covered a variety of topics from potential lingering effects of pregnancy like painful urination, changes in intimacy and dynamics with partners, the frustration of motherhood, messages of empowerment, and the monotony of motherhood. These notifications were generated using Generative AI like Claude and ChatGPT along with some notifications that were created from the team's brainstorming.

The test was conducted on a FigJam. Each notification was written on a single digital sticky. Participants were asked to place each individual sticky along a matrix of like to dislike and very tame to very crazy in accordance to their perception of the notification. The reasoning behind this matrix was to understand when a message went too far and to determine whether the tone could shift a mother's perception of the message and content and vice versa.

Figure: Support notification prototype used to evaluate users' preferences on tone and content



Findings

Our research on effective support notifications for mothers revealed several important insights about tone in terms of empathy and communication. When showing support, we found it's best to show empathy without saying anything similar to "I know how you feel" or telling mothers how they should be feeling. Additionally when trying out different ways of referencing the mother, we found that there was a slight aversion and hesitancy towards words like "mama," "babe," "love," and "champ" as they were categorized as potentially leaning into generational humor preferences and could be received as off putting or emotionally reductivist. Again, one of our intentions for prototyping was to test the limits of what mothers deemed appropriate. Something we were surprised to find is that structure is important in support notifications as well. Ending on a positive note was well received compared to ending on a negative note (even if the negative note was more so "realistic").

Beyond tone or structure, mothers wanted content to not only acknowledge difficulties, but to take the next step by providing practical suggestions or resources to address those difficulties.

This practical approach ties directly to the importance of accuracy in understanding maternal experiences. We learned it's crucial to avoid making assumptions about what mothers are going through, as there is tremendous diversity in their experiences. Incorrect assumptions about more “niche” experiences can invalidate their reality or create confusion rather than demonstrating support or system accuracy. We also learned what subjects required mindful timing of notifications, some being relevant within certain timeframes and not others (e.g. discussing marital strain specifically before birth or changes in intimacy before birth or at least a few months after birth). We also found the value of the messages does not just arise from its content or existence but from its sender. For some notifications, moms felt that if it was coming from someone they knew (a family member or friend) they would like it. When it is sent from an app that does not have a personal understanding of a user’s sense of humor, moms expressed hesitancy.

Recommendations

Support notifications should be especially careful about assuming how the mother is feeling or should be feeling (including phrasing like “you may,” “you might,” or “if you are” can help avoid making assumptions). The structure of support notification should include ending on a positive note or providing practical suggestions and resources to address mentioned difficulties. Support notifications of certain topics like marital strain or changes in intimacy should specifically not be sent to mothers during the first 6 months postpartum, and instead before their birth or after 6 months. With additional development time, a valuable feature would be allowing close family and friends to send their own messages through the support notification system.

Prototype: Privacy + Data Sharing

Purpose

We wanted to test mothers' comfort levels with data sharing across different stakeholders and data types. This research aimed to inform privacy architecture, data sharing protocols, and user customization options for controlling information visibility.

Methodology

We designed a matrix evaluation tool in FigJam where:

- Rows represented various data categories collected by the system:
    - Anomalies (blood clots, fever, falls)
    - Continuous biometric data (heart rate, oxygen levels, blood pressure)
    - Emotional tracking data
    - Environmental sensors (sunlight, sound levels)

- Physical activity (steps, movement)
- Journal entries
- Mood trends
- "Do not disturb" status

- Columns represented potential data recipients/viewers:
  - User only
  - Partner
  - Family/friends
  - Doctor
  - “The void” (data collected but not shown to anyone, including the user

Figure: Data privacy prototype used to evaluate users’ comfort levels with sharing data

Findings

Our evaluation with three recently postpartum mothers revealed significant variation in privacy preferences:

- Two participants preferred highly restricted sharing, primarily limiting data visibility to themselves and their doctors
    - Slightly more receptive to sharing emotional tracking data with their partner
    - Expressed hesitation about AI/ML model training using their personal data
- One participant demonstrated high sharing comfort, willing to share most data types with:
    - Partner
    - Selected close family/friends (specifying relationships like "sister" or "best friend")
    - Doctor
    - AI systems for improving care quality

Recommendations

The significant variation in privacy preferences required us to rethink our approach to data sharing and implement customizable privacy controls. Due to prototyping limitations, we won't be able to build comprehensive privacy rules into the current prototype, though these findings would determine how the product is developed in a real-world implementation.

For the current prototype phase, we will:

- Take a more conservative approach with the more obvious, user-facing AI features (but continue to use the existing anomaly detection ML model)
- Avoid overtly suggestive notifications based on collected data
- Minimize prescriptive or imposing interactions with users

Prototype: Emotion Tracking

After speaking with postpartum mothers, we recognized the necessity of tracking emotions, even without the capability to monitor them through the sensing system. We concluded that self-reported emotion tracking would be sufficient. To inform our approach, we conducted competitor research on other emotion tracking apps, including Finch, How We Feel, and Daylio. Although time constraints prevented us from prototyping the emotion tracking feature, we plan to complete it in early Phase 3.

Data Collection Approach + Onboarding Design

The core functionality of FTH is based upon continuous and abundant data collection, as FTH is with the mother every day tracking her health during a momentous shift in her life. Given the context and the content, the data being collected is inherently intimate. Having ethical data collection practices is always the right approach, but especially with the high level of sensitivity attached to the postpartum period.

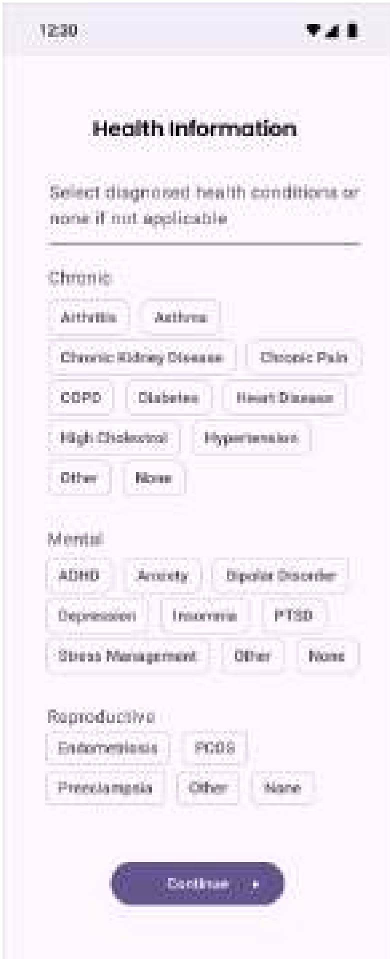
Ethical data collection refers to a MyData approach. MyData is specifically a “data management vision and a set of principles that posit the individual as the owner and controller of the generated personal data (Alorwu et al).” Examples of apps that do not follow the MyData approach are the period app, Flo, or the mental health app, BetterHelp. Flo came under fire when it was discovered that it was sharing with Facebook less-than-anonymous data about user’s menstrual cycles and family planning (Schechner and Secada 2019). BetterHelp’s controversy was similar, as it was discovered that it was sharing sensitive and supposedly-protected consumer data with third parties for advertising purposes (Cox 2024). These instances of data profiteering betray consumer trust for a cause that is extractive and does not serve the purpose of the app, improving user health and wellbeing.

Because this app incorporates the provider, there is a covered entity involved. This means that FTH has to be HIPAA compliant. This does require more data security than mHealth apps that solely involve the user and stricter restrictions on selling data (“Mobile Health Apps and HIPAA” 2021). However, there are still loopholes that companies can pursue for the purpose of selling user data (Cox 2024). The purpose of this app is to empower consumers and ensure that they feel ownership of every component of their health, this includes their data. Selling data for advertising purposes does not align with this vision and has the added consequences of jeopardizing user trust and threatening their security.

To ensure that we have our users’ trust and to ensure that users understand that the data collected is only being utilized for their benefit, we formatted the onboarding section to include very clear communication of these values and principles. There was a clear assertion that the user’s data would be their data. Practices such as third-party data sharing would not occur and we clearly asked for user’s permission to collect data and to share data. Intentionality is a core design principle of FTH, so we wanted every part of the process including, asking for data permission, to consider what is best for the user and put the needs of the user first.

The onboarding process is the first step for establishing the user experience. We wanted the onboarding process to clearly communicate to users that FTH's focus is ensuring that mothers can be more connected and understanding of their health during the postpartum period. To guide our designs, we used existing mHealth apps – examples include Clue and Tidepool – that shared similar sentiments towards data collection as inspiration. This helped us to understand how long the actual onboarding process should be, what type of questions to ask, and how to format the data privacy components.

We wanted the design to illustrate that FTH is here to make this involved process of understanding health as simple as possible. For example, for the section that included the mother inputting her pre-existing medical conditions, we made the selection bubbles and categorized by type of condition. This was done as a way to make the process of looking at medical conditions less overwhelming and to reduce effort on part of the mother with even something as simple as manually typing the conditions.



Severity Scores

In collaboration with the intelligence team, we ensured that the ML model had helpful context for anomalies from different sensors and what each severity score meant in terms of biometric data if applicable and further necessary actions (e.g. notifying providers or emergency contacts or prompting a user to call 911). We also provided further guidance for the LLM to translate severity scores to appropriate language for push notifications, using specific adjectives to correspond with severity scores (e.g. slightly, somewhat, moderately, very, critically). Some severity scores for biometrics did not make sense, for example a severity score of 1 for a fever, so are intentionally left blank.

Anomaly Notification Phrasing

Original Output

anomaly_data.json

{
    "date": "2024-03-10",
    "anomaly": "High blood pressure",
    "severity_score": 7,
    "mitigation_technique": "Monitor regularly, reduce sodium intake"
  },

Notification Language

It looks like your [anomaly: subject] is [severity score: slightly, somewhat, moderately, very, critically] [anomaly adjective: elevated, low]. Try [mitigation technique: in present tense (verbs = ing)]

Anomaly Severity Score Examples

General meaning

- 0: No Risk - no notification sent, no anomaly detected
- 1: Low Risk - "slightly" **- action: monitor / watch out**
- 2: Low-Medium Risk - "somewhat" **- action: slight adjustment**
- 3: Medium Risk - "moderately" - **action: change behavior, watch for other symptoms**
- 4: High Risk (requires immediate action (e.g., sitting down), notifies care team, alerts the user to watch out for other symptoms and seek emergency medical care if necessary) - "very" **- action: change course immediately, watch for symptoms**
- 5: High Risk + calls the ambulance (alerts you + caregiver before calling on it's own) - "critically" **- go to hospital**

High blood pressure

- 0: Blood pressure is lower than 120/80 mm Hg; No Risk/normal reading - no notification sent, no anomaly detected
- 1: The top number ranges from 120 to 129 mm Hg and the bottom number is below, not above, 80 mm Hg; slightly elevated but not concerning; user should continue monitoring
- 2: The top number ranges from 130 to 139 mm Hg or the bottom number is between 80 and 89 mm Hg; requires monitoring and lifestyle adjustments (e.g., reducing salt intake, take more rests); user should watch out for persistent headaches, facial or hand swelling
- 3: The top number is 140 mm Hg or higher or the bottom number is 90 mm Hg or higher; Moderately elevated; user is instructed to rest; user should contact healthcare provider within 24 hours if it persists

- 4: Dangerously elevated (higher than >180/120mmHg); care team has been notified; user should watch for symptoms of a hypertensive crisis (chest pain, shortness of breath, or symptoms of stroke) and seek medical care if they have them
- 5: Dangerously elevated (higher than 180/120mmHg) and user is showing signs of being incapacitated (e.g., heart attack, unresponsiveness, seizures); user is at risk of heart attack, stroke, or other life-threatening health problems; notifies caregiver; calls ambulance if user or caregiver do not cancel

Low blood pressure (1)

- 0: No Risk - no notification sent, no anomaly detected
- 1: No risk of fainting; user should consider mitigation techniques
- 2: Lower than 90/60 mm Hg; faint risk is low; user be on the lookout for feelings of dizziness or lightheadedness; user should consider mitigation techniques
- 3: Faint risk is medium; user may feel slightly dizzy or lightheaded; user should sit down or not move too quickly; user should look out for signs of shock (e.g., sweaty skin, rapid breathing, a blue skin tone, or a weak and rapid pulse) and seek immediate medical attention if they have the signs
- 4: Faint risk is high; care team has been notified; user is at risk of organs not receiving enough blood; user should immediately sit down to prevent fainting or reduce the risk of injury if they faint; if user is also exhibiting signs of shock (rapid breathing or a weak and rapid pulse), calls ambulance if user or caregiver do not cancel
- 5: Faint risk is extremely high; user is about to faint or has already fainted; calls ambulance if user or caregiver do not cancel

Low blood oxygen saturation (1)

- 0: 95-100%; No Risk - no notification sent, no anomaly detected
- 1: 92-95%, user is not in immediate risk but should look out, potentially do a breathing exercise
- 2: 90-92%
- 3: 88-90%; user should slow down, focus on breathing, call provider if not improving
- 4: 80-88%; user should rest immediately, focus on breathing, and seek immediate medical care
- 5: blood oxygen level is below 80%; user is at risk of organ damage and should seek immediate emergency medical care; calls ambulance if user or caregiver do not cancel

Heart Palpitations

- 0: No Risk - no notification sent, no anomaly detected
- 1:
- 2:

- 3: Medium Risk - May be a result of postpartum anxiety/hormone changes and do not require immediate medical attention if so no immediate medical attention is needed. If accompanied by fainting, severe shortness of breath, chest pain, and/or dizziness, seek immediate medical help
- 4: High Risk - care team has been notified
- 5:

High heart rate

- 0: 60-100 bpm; No Risk - no notification sent, no anomaly detected (heart rate 60-100 bpm)
- 1: >100 bpm while not exercising or above 220 minus the user's age while exercising; no risk for cardiac arrest; user may be slightly alert and should consider trying mitigation techniques
- 2: Risk for cardiac arrest is low; user should consider sitting down, try mitigation techniques, and be on the lookout for other symptoms
- 3: Risk for cardiac arrest is medium; user may feel they are anxious or on edge and have a racing heart; user should sit down and actively try mitigation techniques
- 4: >100 bpm accompanied with shortness of breath; care team has been notified; user may feel panicked and disoriented; user should immediately sit down and monitor symptoms and go to the emergency room immediately if they experience trouble breathing, chest pain, feel their heart pounding, or feel faint or dizzy
- 5: >100 bpm accompanied with fainting; Risk for cardiac arrest is high; user may feel they have a lack of control over their body or like they are floating; user is about to go into cardiac arrest or has already; calls ambulance if user or caregiver do not cancel

Low heart rate (1)

- 0: 60-100 bpm; No Risk - no notification sent, no anomaly detected
- 1: <60 bpm (not sleeping); slightly below normal; user should monitor for weakness or dizziness and limit strenuous activity
- 2:
- 3: user should monitor for chest pain, confusion or memory problems, dizziness or lightheadedness, feeling very tired especially during physical activity, fainting or near-fainting, shortness of breath;
- 4: <60 bpm and accompanied with heart palpitations or shortness of breath; user should seek emergency medical help if they feel faint, have difficulty breathing or have chest pain lasting more than a few minutes; care team has been notified
- 5: <60 bpm and accompanied with heart palpitations or shortness of breath; user is at risk of organ damage from not receiving enough oxygen; user should seek emergency help immediately; calls ambulance if user or caregiver do not cancel

High body temperature

- 0: No Risk - no notification sent, no anomaly detected
- 1:
- 2: slightly elevated (99°F–100.3°F); user has a low-grade fever; user should monitor temperature and stay cool and hydrated
- 3: (100.4°F–102.9°F); user should seek medical treatment within 24 hours and watch for additional symptoms (leg pain or swelling, severe stomach pain, vaginal bleeding that soaks a pad within an hour, bleeding that has large clots, bleeding from an incision site, pus at an incision site)
- 4: (103°F–104°F); user should seek immediate medical attention if they experience additional symptoms (confusion, difficulty breathing, loss of consciousness, painful urination or urine that smells bad, seizure, severe pain); notify care team;
- 5: body temperature is dangerously high (>105°F); user is at risk of organ failure and death if not treated; notifies caregiver; calls ambulance if user or caregiver do not cancel

Deep vein thrombosis

- 0: No Risk - no notification sent, no anomaly detected
- 1: N/A - no "low risk" instance of deep vein thrombosis
- 2: N/A - no "low risk" instance of deep vein thrombosis
- 3: early signs of deep vein thrombosis; patient should monitor for pain, swelling, warmth, or redness in one leg; patient should notify provider within a few days
- 4: clear signs of deep vein thrombosis (obvious clot in leg); patient should contact provider within 24 hours; care team is notified; patient is at risk of a blood clot traveling to lungs and causing a pulmonary embolism; patient should monitor for symptoms of pulmonary embolism (sudden shortness of breath, chest pain or discomfort that worsens with a deep breath or cough, lightheadedness or dizziness, fainting, rapid pulse, rapid breathing, coughing up blood)
- 5: no severity score of 5 for DVT as it does not warrant calling an ambulance

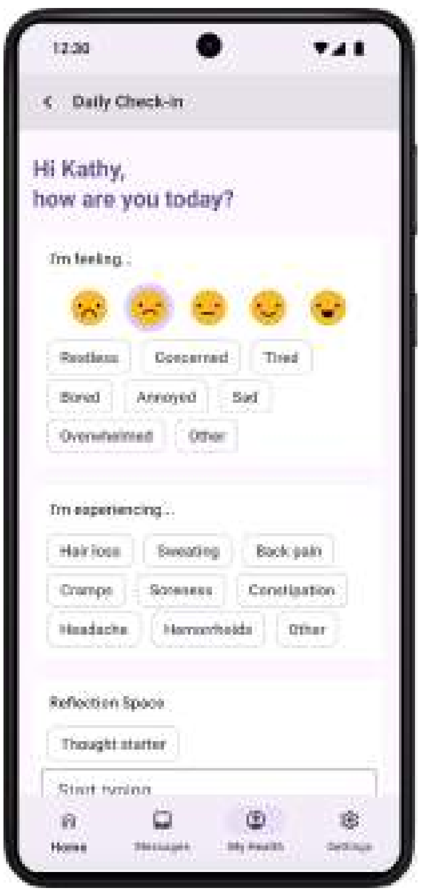
Example Push Notifications:

Severity Score 1, high blood pressure: "It looks like your blood pressure is slightly elevated. Try monitoring regularly and reducing sodium intake."
Severity Score 2, high blood pressure: " It looks like your blood pressure is somewhat elevated. Try monitoring regularly and reducing sodium intake."
Severity Score 3, high blood pressure: "It looks like your blood pressure is moderately elevated. Monitor regularly and reduce sodium intake."

Severity Score 4, high blood pressure: “It looks like your blood pressure is very elevated. Your care team and caregiver have been notified and will reach out if necessary. If you’re experiencing headaches, shortness of breath, or nosebleeds, call 911 or go to the hospital.”
Severity Score 5, high blood pressure: “It looks like your blood pressure is critically elevated. Your care team and caregiver have been notified and 911 has been called.”

Emotional Input

Postpartum depression (PPD) affects approximately 1 in 8 mothers, and it emerged as a major concern during our interviews. Many mothers shared that they were so focused on caring for their newborns that they didn’t realize they were experiencing depression until months later. This highlighted a clear need for emotional self-reflection. Through competitive analysis of other emotion-tracking apps (e.g. Apple’s State of Mind, How We Feel, and Daylio), we noticed some were overly complex. We aimed to do the opposite, keeping things simple, accessible, and attuned to the realities of early motherhood.



Recognizing the deep connection between emotional and physical health, we decided to present both types of symptoms on a single input page in the patient app. To reduce cognitive load, we designed the emotional input using a face-based Likert scale (ranging from very unhappy to very happy), which then expands into more specific emotions through progressive disclosure. This approach allows mothers to engage at their own pace, without feeling overwhelmed. All input was intentionally made optional to respect each user’s energy and capacity.

In earlier prototype testing on data privacy, we found that participants didn’t resonate with the idea of “ranting into the void" and expressing themselves in a place that didn’t save either to the cloud or locally. We decided the “void” was an added layer of complexity and was not adding value, so we included a more familiar journaling space to support self-reflection. If the mother doesn’t know where to start there is an added prompt based on the Edinburgh Postnatal Depression Scale (EPDS) to help guide deeper emotional awareness.

Reflection Questions Based on Edinburgh Postnatal Depression Scale

Looking Forward (Based on EPDS Q1 about looking forward with enjoyment)
"What's one thing you're looking forward to in the coming week? Has this changed from before?"

Joy & Laughter (Based on EPDS Q2 about enjoying things)
"Share something that made you laugh this week. Has finding joy in things felt different lately?"

Self-Blame (Based on EPDS Q3 about blaming oneself unnecessarily)
"When things don't go as planned with the baby, how do you talk to yourself about it?"

Worry & Anxiety (Based on EPDS Q4 about anxiety or worry)
"What's been on your mind most lately? How does it feel when worries come up?"

Fear or Panic (Based on EPDS Q5 about feeling scared or panicky)
"Have there been moments lately when you've felt overwhelmed? What helps ground you?"

Overwhelm (Based on EPDS Q6 about things getting on top of you)
"On a scale of 'swimming comfortably' to 'barely treading water,' how would you describe managing daily life right now?"

Sleep Quality (Based on EPDS Q7 about sleep difficulties)
"When the baby is actually sleeping, how is your sleep? What's your relationship with rest these days?"

Sadness (Based on EPDS Q8 about feeling sad or miserable)
"How has your emotional weather been this week? More sunny days, cloudy days, or rainy days?"

Tearfulness (Based on EPDS Q9 about crying)
"When was the last time you needed a good cry? What allows you to express your emotions?"

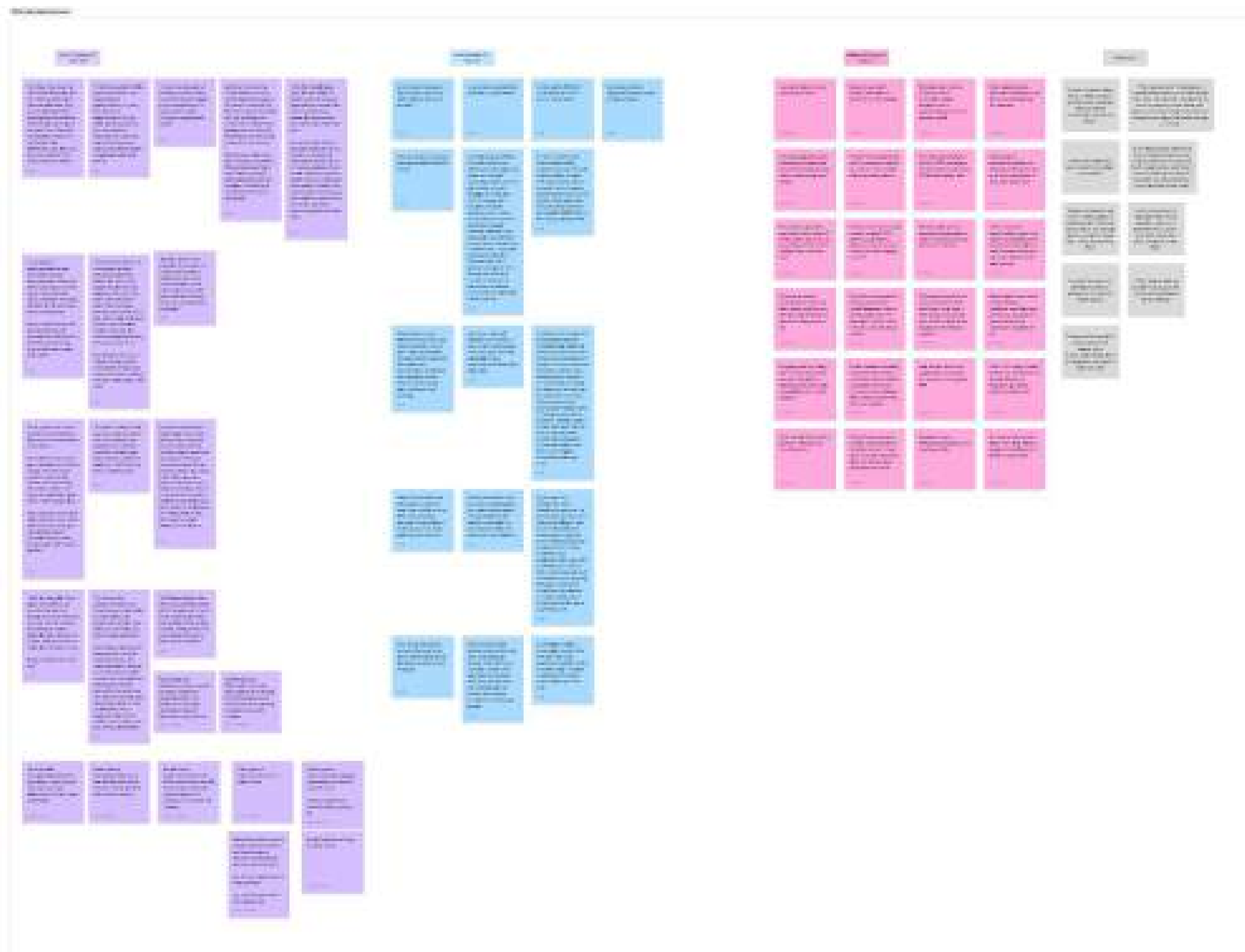
Self-Harm Thoughts (Based on EPDS Q10 about thoughts of harming oneself)
"What helps you feel connected to yourself when things get really hard? Who would you reach out to if you needed immediate support?"

5.1.5 Data Visualization

Research

As stated in the previous section, we interviewed seven mothers and five providers in the research phase of this project. These interviews were conducted in a semi-structured interview format where the interviewer asked the interviewee a variety of questions. Interviews were transcribed on FigJam sticky notes with special attention on pull quotes that were illustrative of key attitudes and behaviors. We wanted to conduct enough research so that we were not only able to appropriately categorize needs of both mothers and providers, but also rationalize each design decision effectively.

Interview Notes



Synthesis & Development of Design Principles

In an effort to synthesize this data, we performed affinity diagramming, which helped us identify common themes in the data. The themes led us to generate insights, and each insight led to the development of a key design principle and associated design recommendations.

This work revealed what we came to regard as our North Star: **minimizing cognitive load** for the mother. Through our interviews, we learned that there are so many things that mom has to do for her newborn. Given the stress of her responsibilities, a core design principle of the system should be to have minimalistic information architecture. Therefore, the system should limit the display of extraneous information and only support quick, at-a-glance health insights. Other design principles included gentle and encouraging messaging, passive data tracking and monitoring, and unalarming visuals with subtle recommendations.

Affinity Diagramming

Design Principles & Recommendations

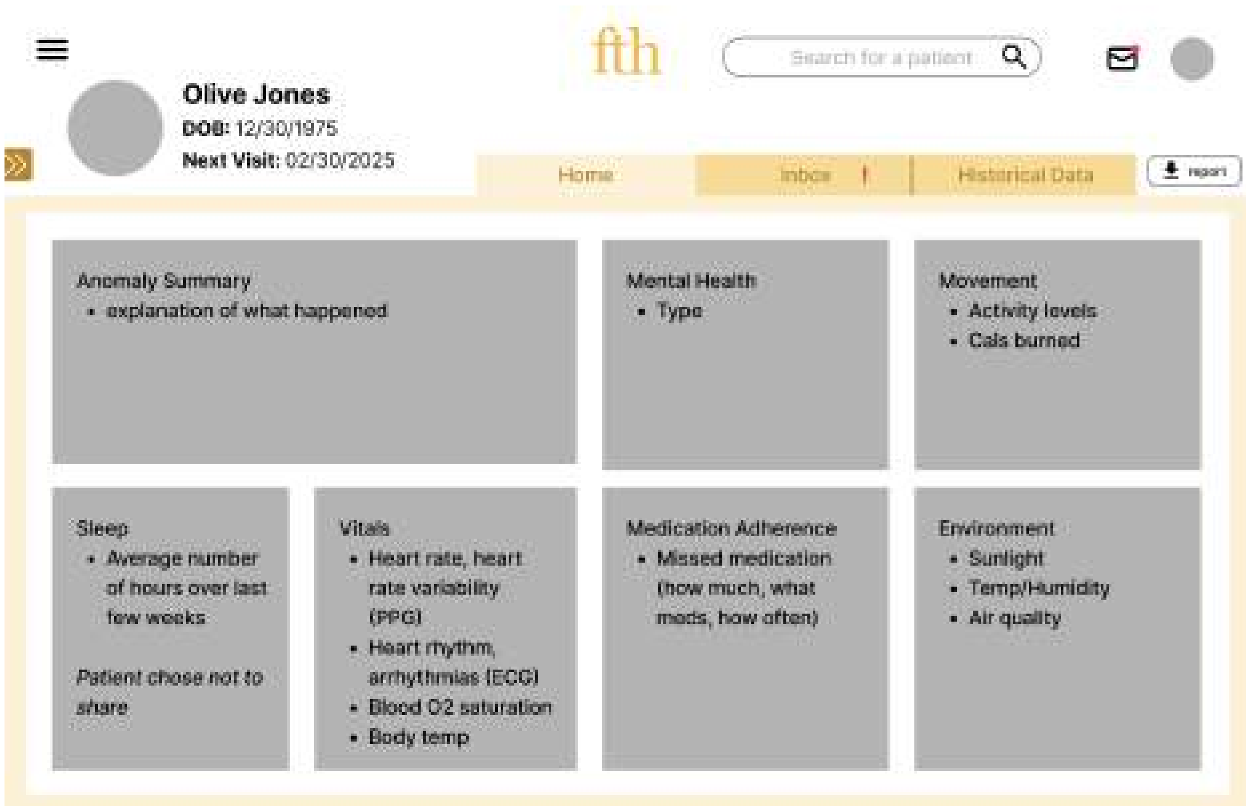
Insight	Description	Design Consideration	Design Recommendation
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

Provider and User Interface Development

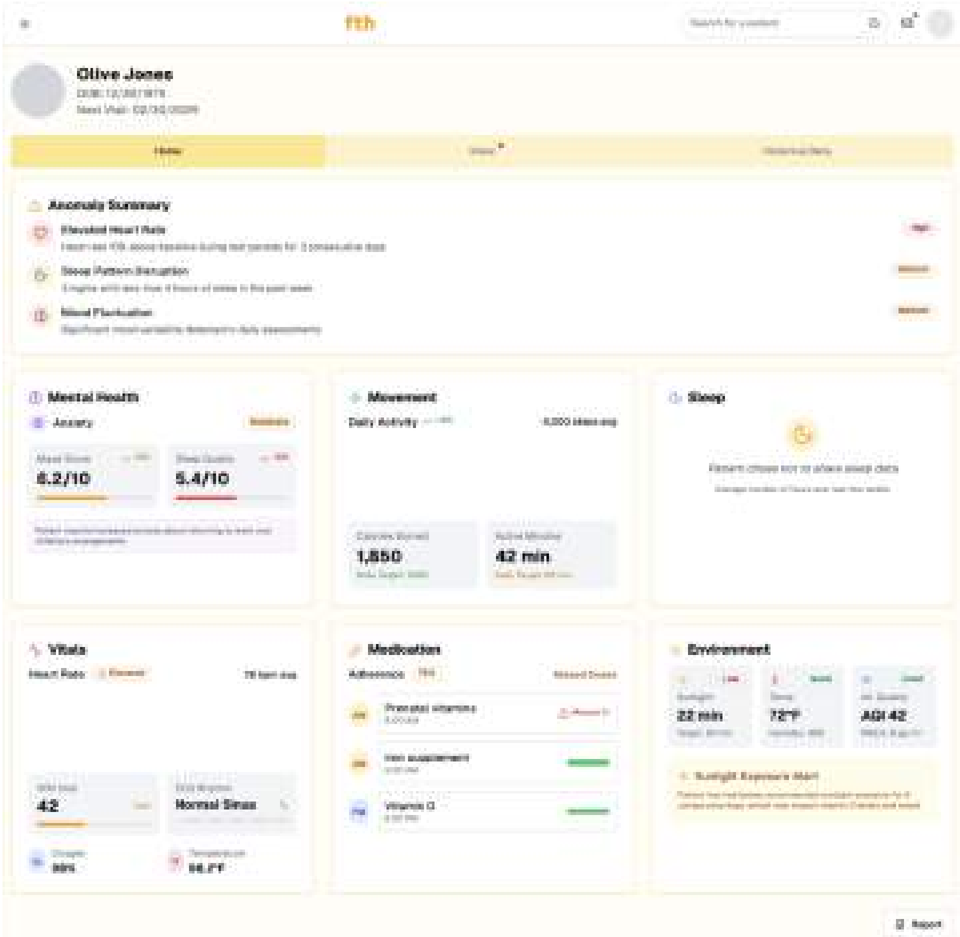
Design Principles. Using the research insights, design principles, and design recommendations, we generated a wireframe, which primarily served to outline the information architecture. Then, we pulled inspiration from Apple Health's tile-like dashboard for our primary organizational structure, which allowed us to create separate tiles for each health category (e.g., vitals - heart rate, O2 saturation, etc, movement - calories burned, environment - CO_2 levels, sunlight levels, medication adherence, and mental health).

High Fidelity. We put these two prototypes (shown below), along with our "North Star" design principle of minimal cognitive load, into a v0 prompt which asked for a screen with the associated metrics. The result (shown below) gave us a jumping off point for inspiration. Finally, we developed Figma screens and standardized the design with components for handoff to the software team (shown below). Our next steps include getting insight from providers on the design overall, including whether they would like more information "under the fold." This means that the provider would have to click on a tile to see a detailed graph with trending metrics, for example. We plan to make minimal adjustments to our design once that feedback is received.

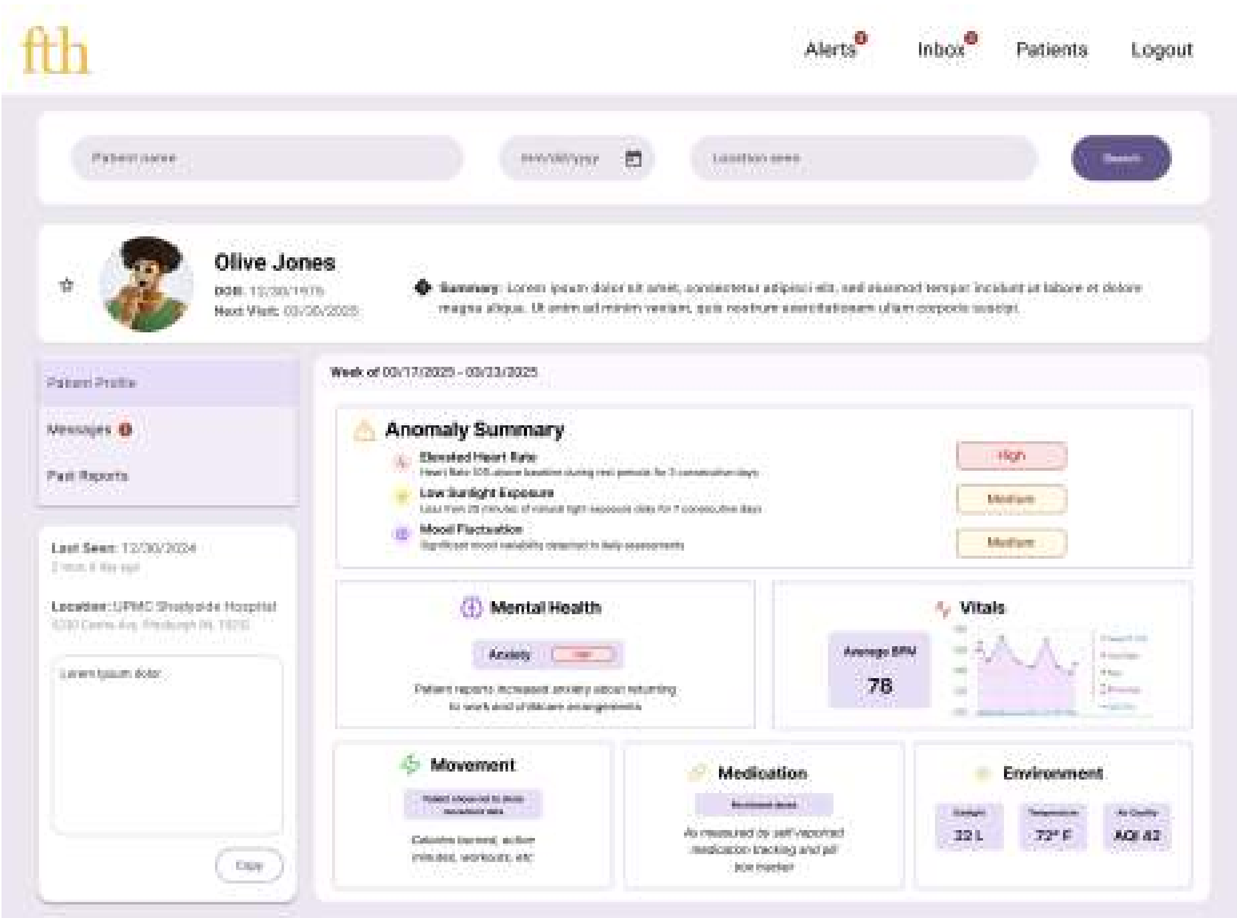
Wireframe Prototype



v0 Ideation Prototype



Final High-Fidelity Prototype



Integration

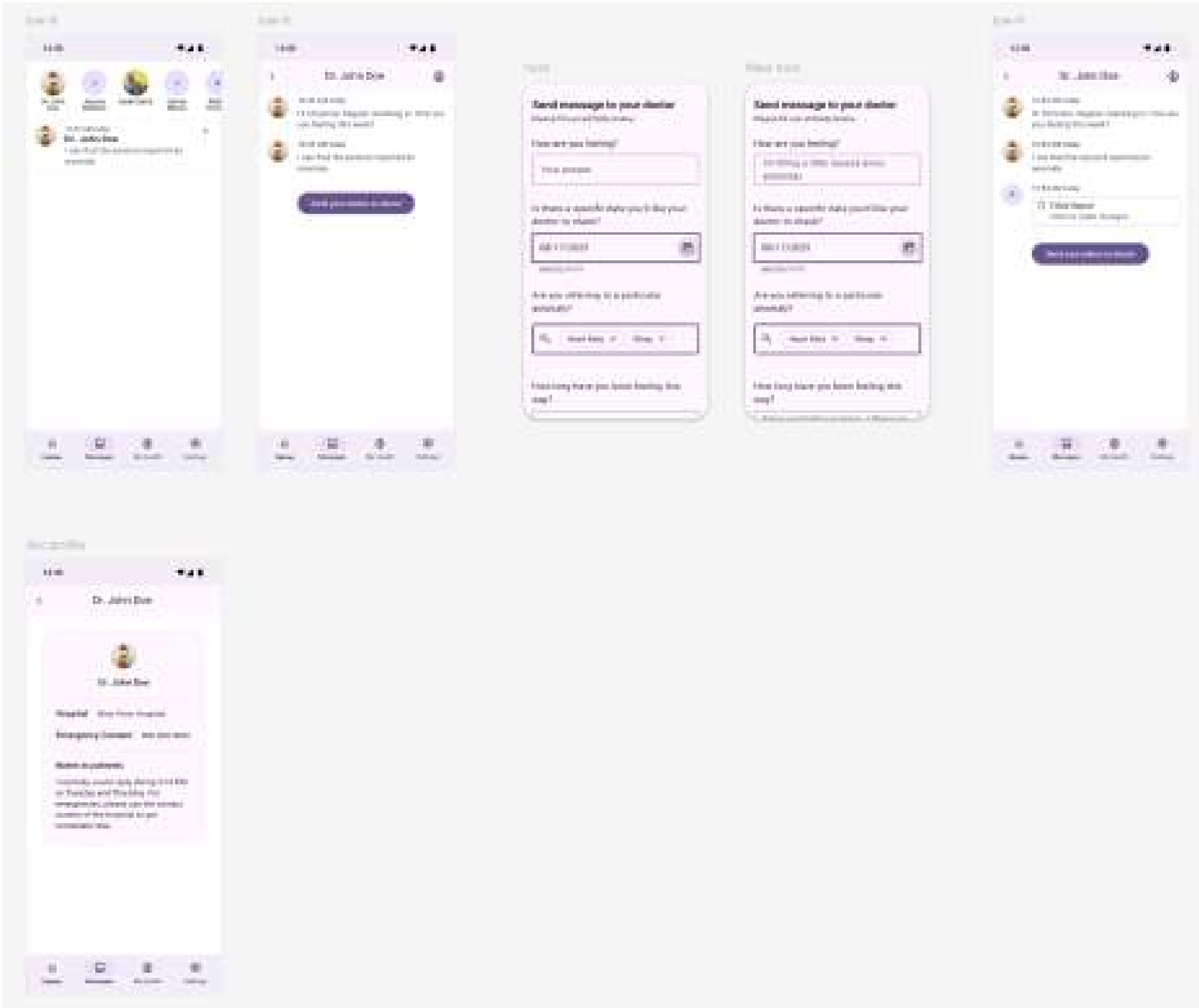
As we developed the provider dashboard, we collaborated closely with the broader HCI group and the software team to ensure our designs were clear, actionable, and technically feasible. A major challenge was determining how to present complex health data in a way that providers could quickly interpret and use to guide care. We focused on translating raw data into visual patterns and built-in prompts that supported decision-making.

To keep communication smooth and iterative, we used Figma comments as our main channel for cross-team feedback. This allowed us to engage in ongoing design discussions, resolve questions efficiently, and adjust our visuals based on feedback from the software team. These conversations helped align our designs with the backend implementation timeline and constraints.

Because some features were still being built or explored, we chose to “Wizard of Oz” certain elements, such as the messaging interface and the provider dashboard backend. This allowed us to simulate those experiences in prototypes and gather user feedback before the underlying functionality was complete.

5.1.6.1 User Interface–Patient Mobile App

This is the app architecture and the first round of low-fidelity wireframes. Initially, we decided to include Home, Inbox, Notification, and Profile screens. The Home page will contain data visualizations. The Inbox page will allow patients to reply to or message doctors. The Notifications page will display environment sensor devices, wearable data, and anomaly notifications. On the Profile page, users can upload their medical and family history.



During Phase 2, we finished the flows for patients to message their doctors. To ensure effective communication between patients and doctors, patients will be prompted to fill out a form answering questions such as how they are feeling, when the anomaly was reported, etc., to provide more context to doctors.

As Phase 2 progressed, we refined our user app's information architecture. We moved from an organization centered around Home, Inbox, Notifications, and Profile → to one organized by Home, Inbox, My Health, and Settings.

Overall, The design principles that we used revolved around simplicity, control, and empathy. For example, the Home page is broken down into Notifications, Pinned Stats, and To-Dos. The user here has control around what they see, but also is made aware of the most pressing information on a given page. For example, in the My Health Page, users are able to filter between data coming from Wearable and Environmental sensors. From that page, they are able to select a Health Metric to go deeper into. In the detailed view, they have information broken down again. The same is true for notifications, which allow users to go a level deeper into suggestions and more data if they wish.

Strategic Development of Data Visualization. Like for the patient portal, Apple Health's visualization served as a strong reference not only in the design of a high-level overview page, but also in the detailed view of how to see the details of each health metric. *(Figure 1)* We especially incorporated the ability to visualize health over time – whether it be day, week, or month – as we knew that identifying trends was a key focus for our system. *(Figure 2)* Finally we aligned on an Apple Health-like dashboard because of its implications on the development process. The dashboard was modular in nature which meant that we could give the software team pieces to build in batches if we needed to.

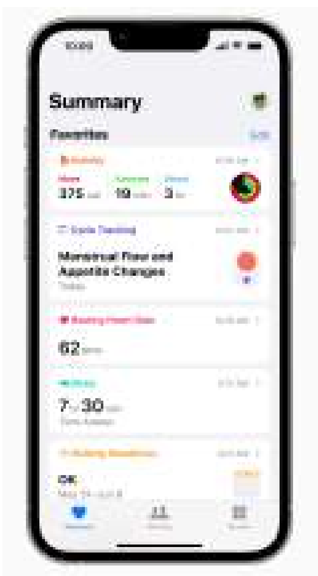


Figure 1, Figure 2 (left to right)

Prior to aligning on Apple Health's format we leveraged v0 for inspiration on visualizations. However, we decided that they were too complex given that our design principles emphasized reduced cognitive load and simplicity for mothers.

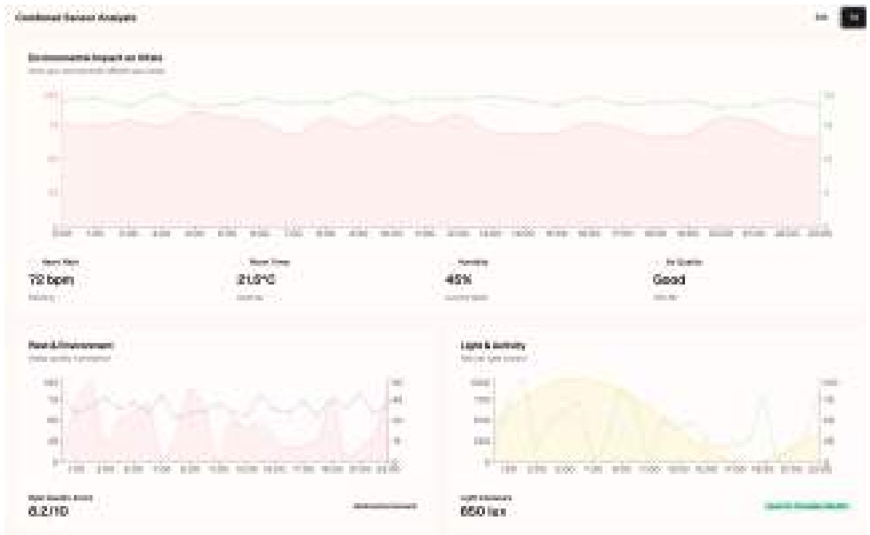


Fig. *Our initial visualization references were too complex*

Finalization. The final design represented a mix of Apple Health's modular health cards in the overview page with AI summaries to help the mother interpret all of the data she was seeing. We leveraged Intelligence's anomaly reports to bring timely attention to irregularities with gentle and actionable language so as to not overwhelm the mother. Notifications could be expanded to reveal more information around mitigation techniques based on the Mother's comprehensive medical profile. *(Fig 3, 3.1)*

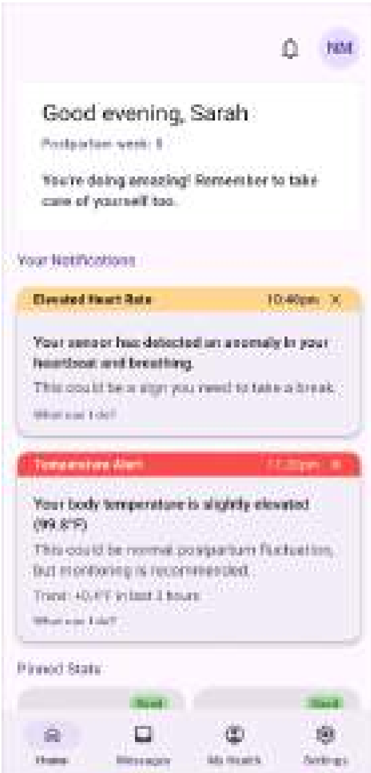

Good evening, Sarah
You're doing amazing! Remember to take care of yourself too.
Your Notifications
Elevated Heart Rate
10:40pm
Your sensor has detected an anomaly in your heartbeat and breathing.
This could be a sign you need to take a break.
What can I do?
Temperature Alert
Your body temperature is slightly elevated (99.8°F)
This could be normal postpartum fluctuation, but monitoring is recommended.
What can I do?
Pinned Stats
Home
Messages
My Health
Settings

Fig. 3

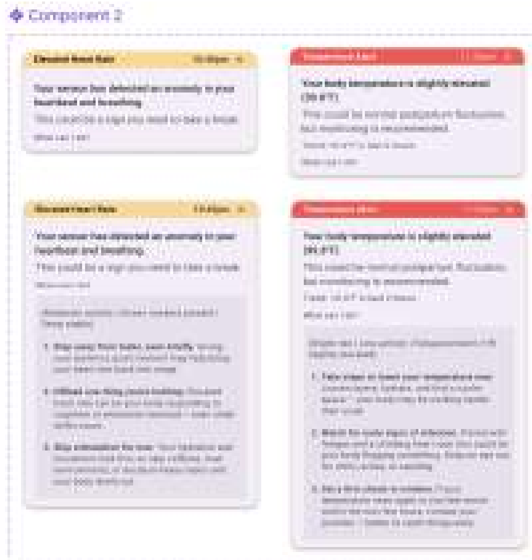

Component 2

Fig. 3.1

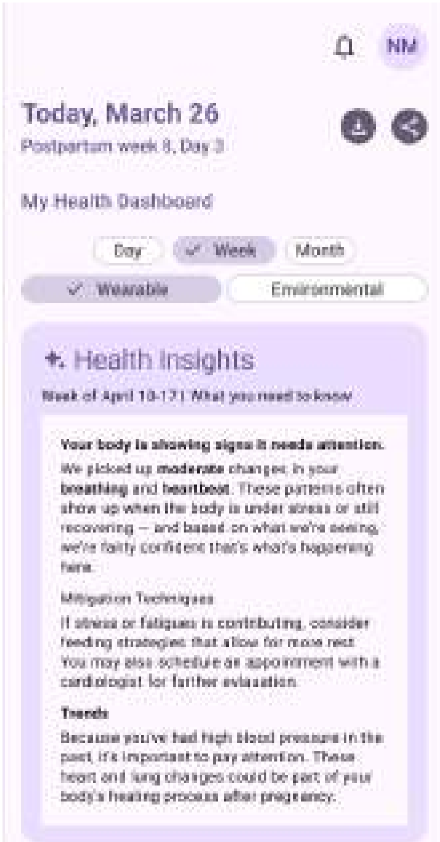


Fig.4 - Overall Health Summaries with Mitigation Techniques and Trends

The most detailed level of health insights appeared when the Mother clicked on a specific health card. Color was used to denote urgency – divided between Normal, Improving, and Attention Required. *(Fig. 5)*

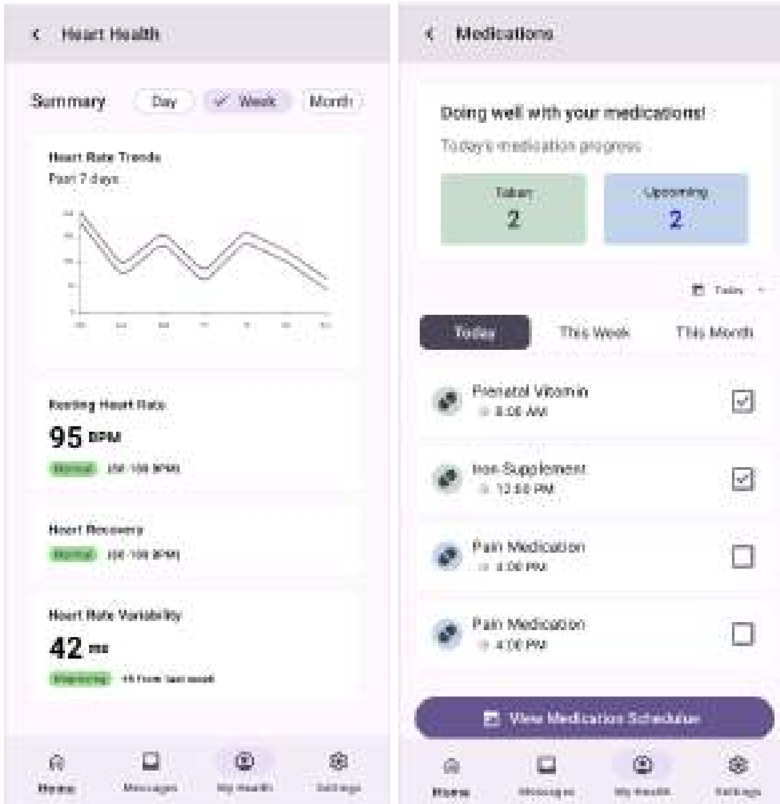


Fig. 5 *Home, My Health, and My Health (detailed)*

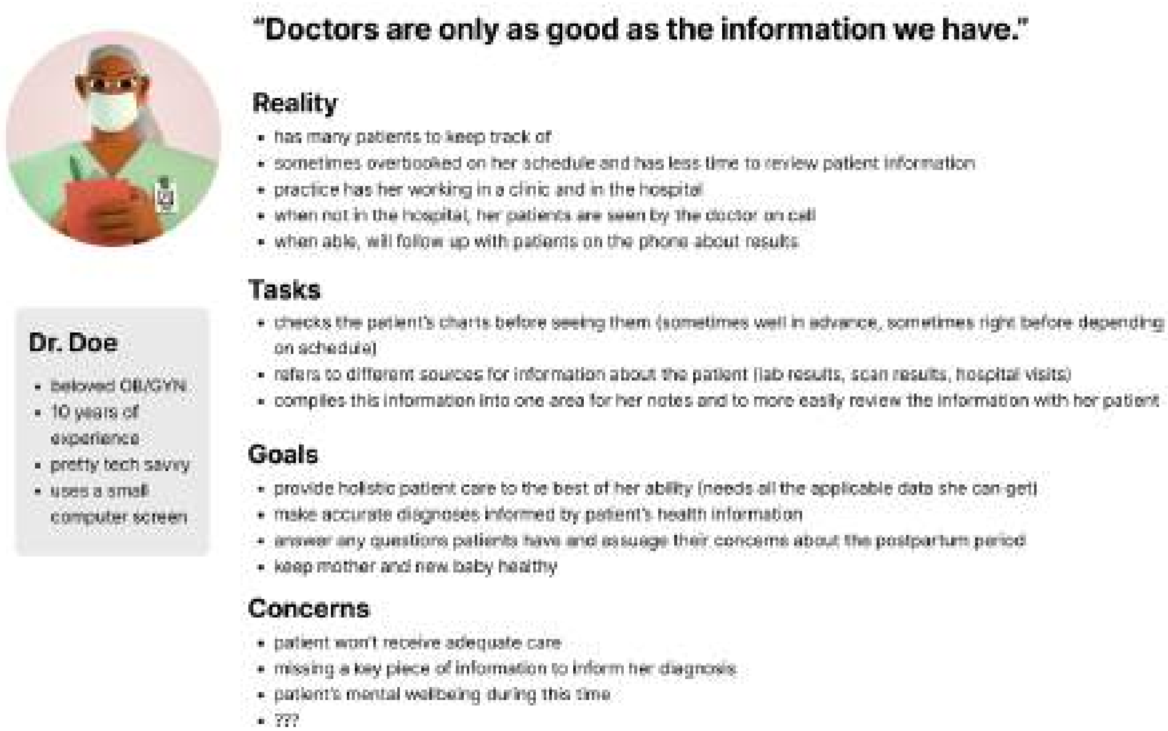
5.1.6.2 User Interface–Provider Website

Provider Research

To better understand a provider's needs, we spoke with two providers: a hematologist/oncologist and a urologist to create a provider persona. We were specifically interested in learning more about their current workflow around preparing to see a patient and any constraints they felt during that process. These conversations were not focused on which data would be important to visualize since neither were ob/gyns.

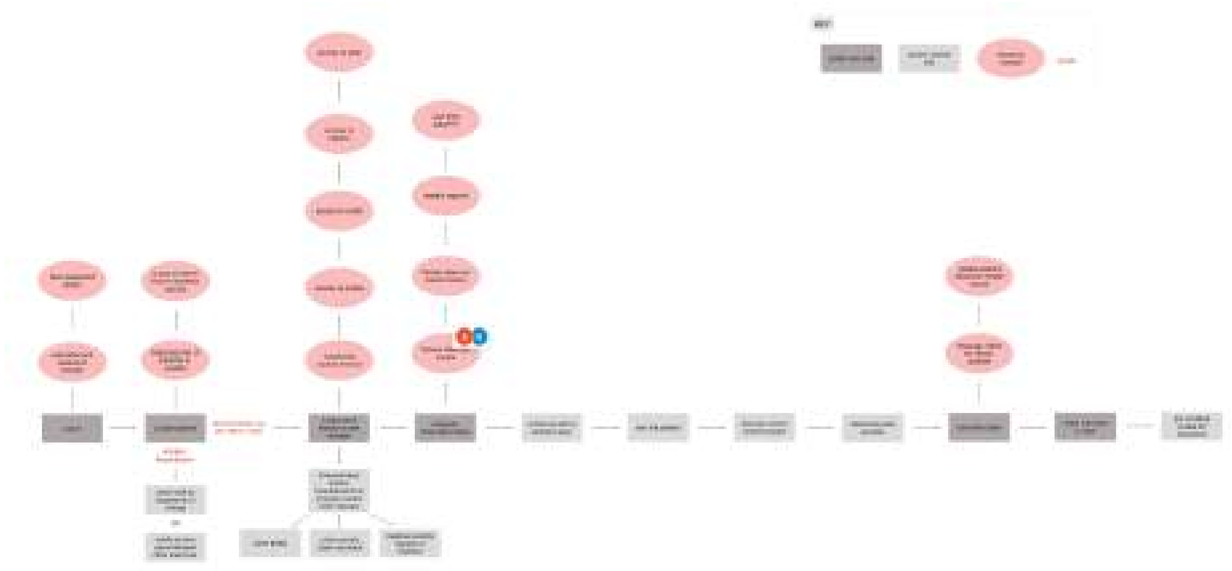
The interviews validated that the providers spent some time preparing to see their patients–although the amount of time spent varied per provider: anywhere from 24-48 hours in advance to a few hours prior. From these conversations, we learned the doctors reviewed previous notes, lab and scan results, recent hospital visits, etc. They found this information in different places: within the patient's electronic health record, in other health portals like *Labcorp*, and by reaching out to other healthcare clinics for information to be sent over.

The providers also emphasized how helpful it would be to have access to information collected in their absence, which validated our concept. The urologist had his own method of this. He had his patients fill out an objective screener that was validated by researchers as a way to collect information of a patient's status between doctor's visits. For him, the questionnaires were his "sensors" and his insight into the patient's day-to-day life. They were his way to tell what's going on. Our design would be able to add even more insight into the patient's health. Information is key. As the hematologist/oncologist said: "Doctors are only as good as the information they have." A copy of the persona is included below. You can also view it in more detail here.

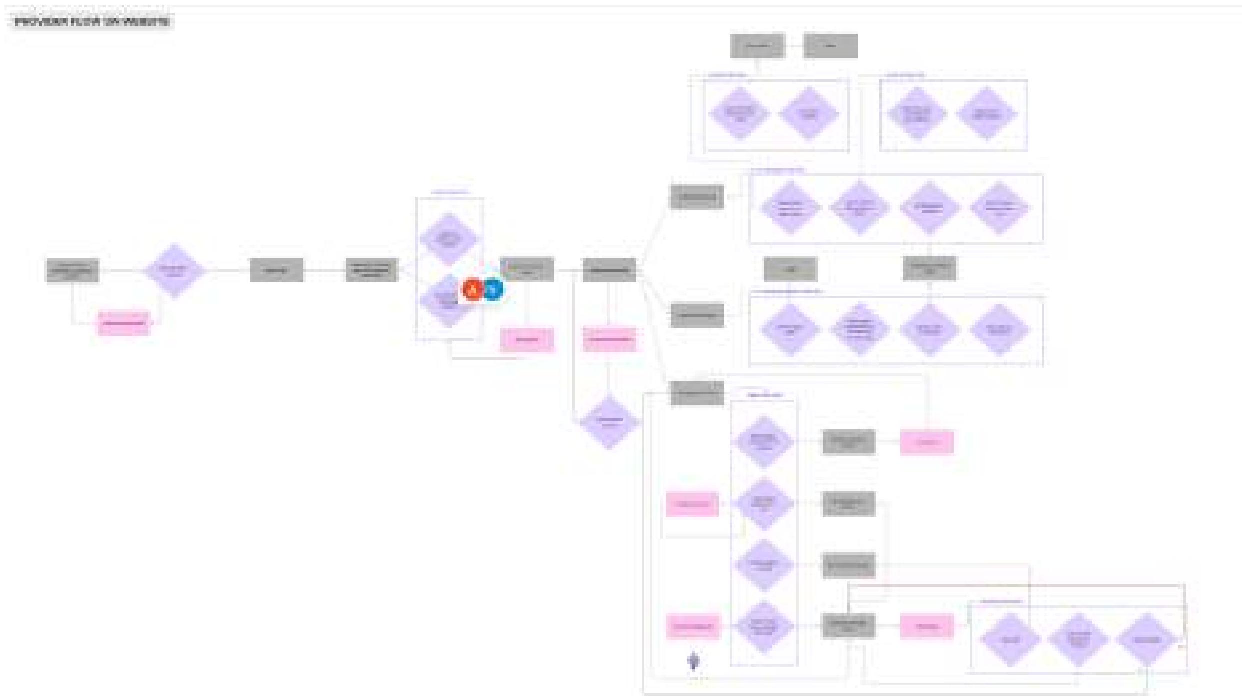


Using Research to Inform Flows

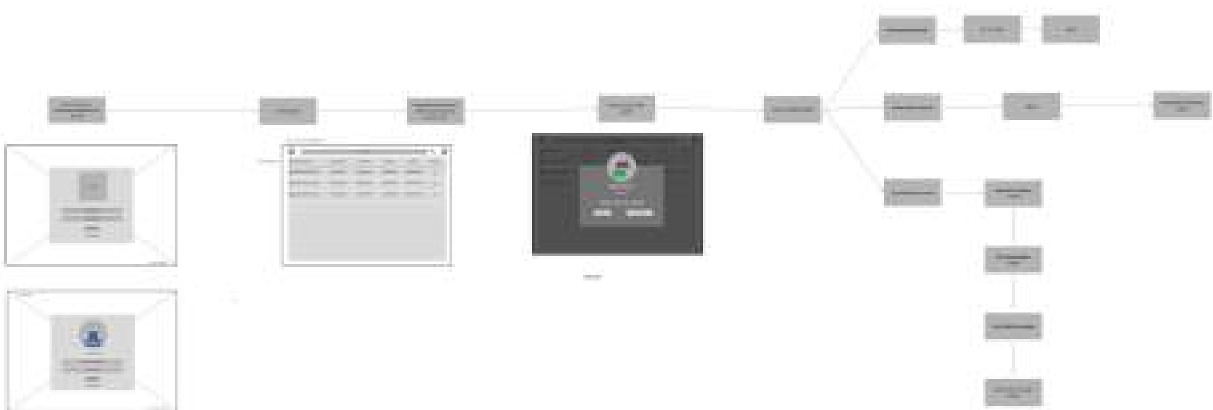
We then used the findings from the provider interviews, in addition to functional requirements from the software team, to complete a user journey. This journey started with a provider logging into a healthcare portal and ended when the provider inputted diagnosis codes for billing after seeing the patient. We mapped out the larger journey to figure out where our provider portal would fit in the current process. We are targeting the pre-planning, data analysis, and documentation phases as the provider updates their plan of care. We also wanted to identify overlaps between the functionality of our portal and the provider's current technologies to see if we could borrow existing workflows. A copy of the user journey is included below, but you can also view it in more detail here.



Continuing to build on that, we mapped out a more specific user flow that started to explore necessary screens (in grey) and the different actions providers could take (in purple). After reviewing the initial draft with the software team, we went back and added a few error flows (in pink). We used this user flow to determine which screens we would need to include in the website. A copy of the user flow is included below, but you can also view it in more detail here.

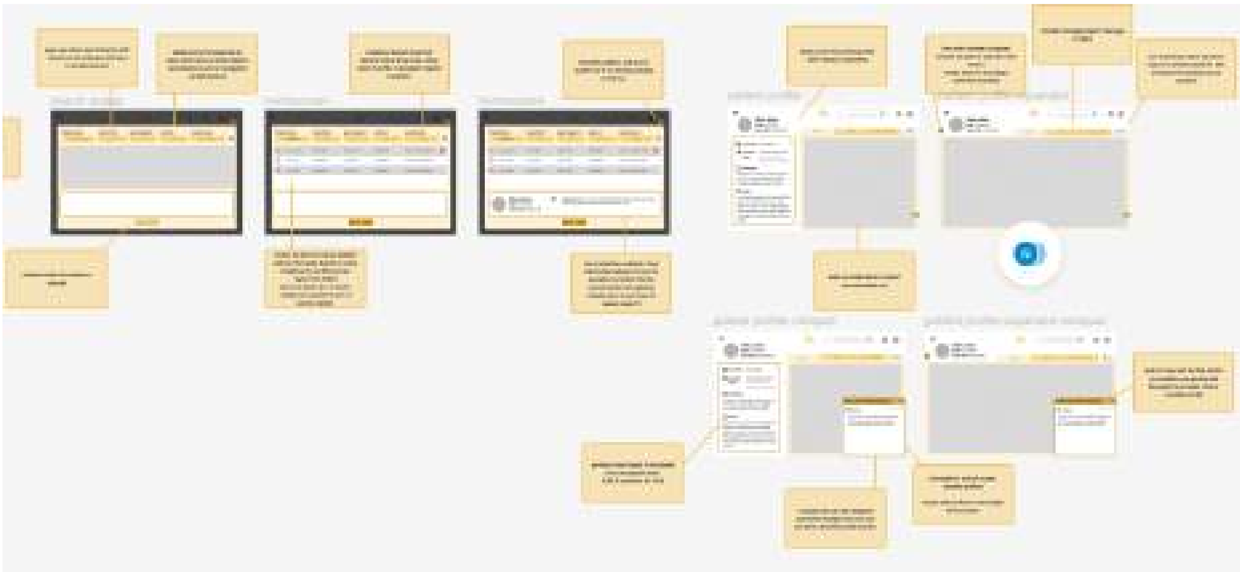


After completing the user flow, we pulled the screens (in gray) and started sketching out some rough wireframes. We decided to jump in and use the wireframing process to ideate on what what the screens should look like, drawing inspiration from current electronic medical record systems. You can view them below or in more detail here.

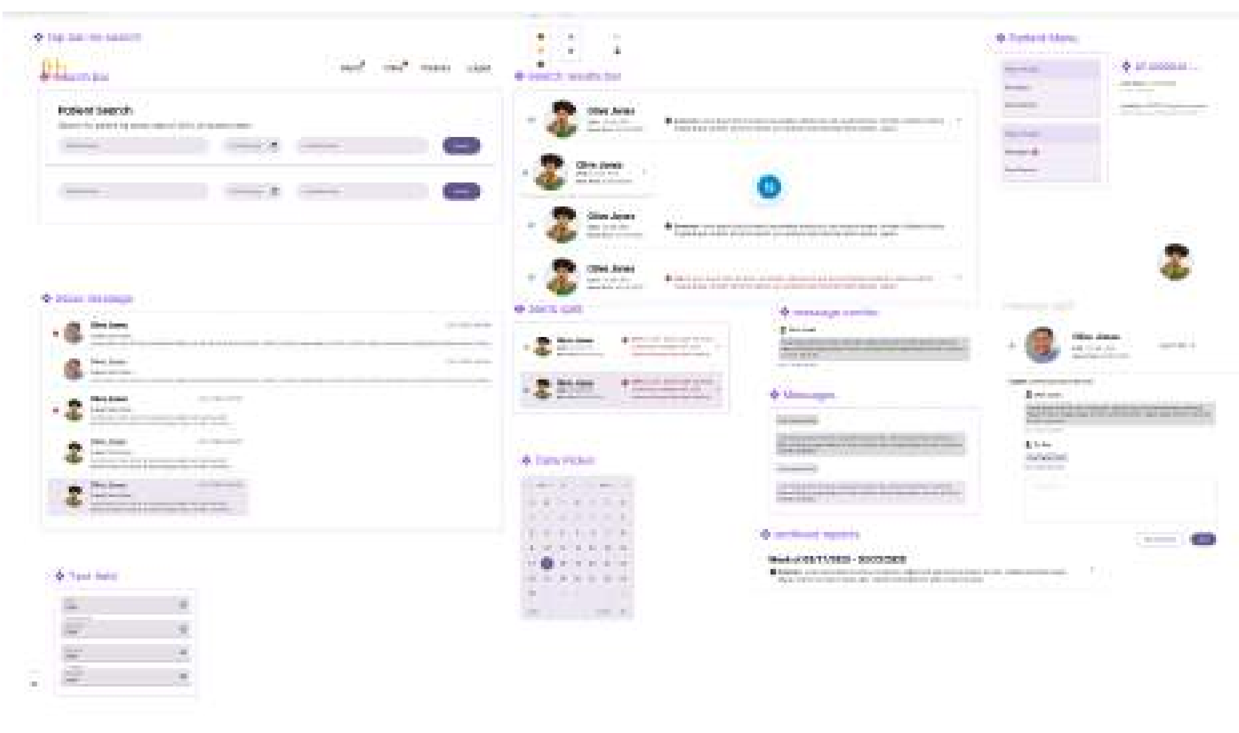


Transitioning from Lo-Fi to Hi-Fi Prototypes

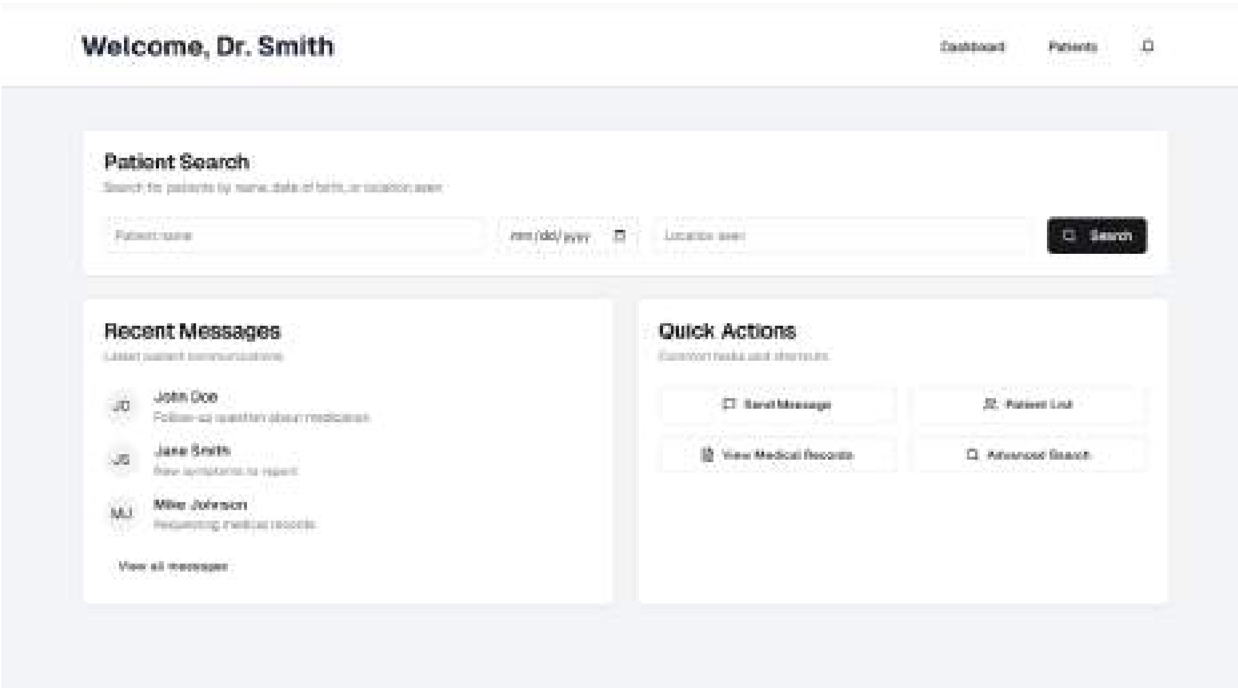
These wireframes were then developed into rough annotated wireframes that gave the software team more insight into intended functionality and interactions. For inspiration, we continued to look to existing tools that providers were familiar with like EPIC's electronic medical record system. You can view them below or in more detail here.



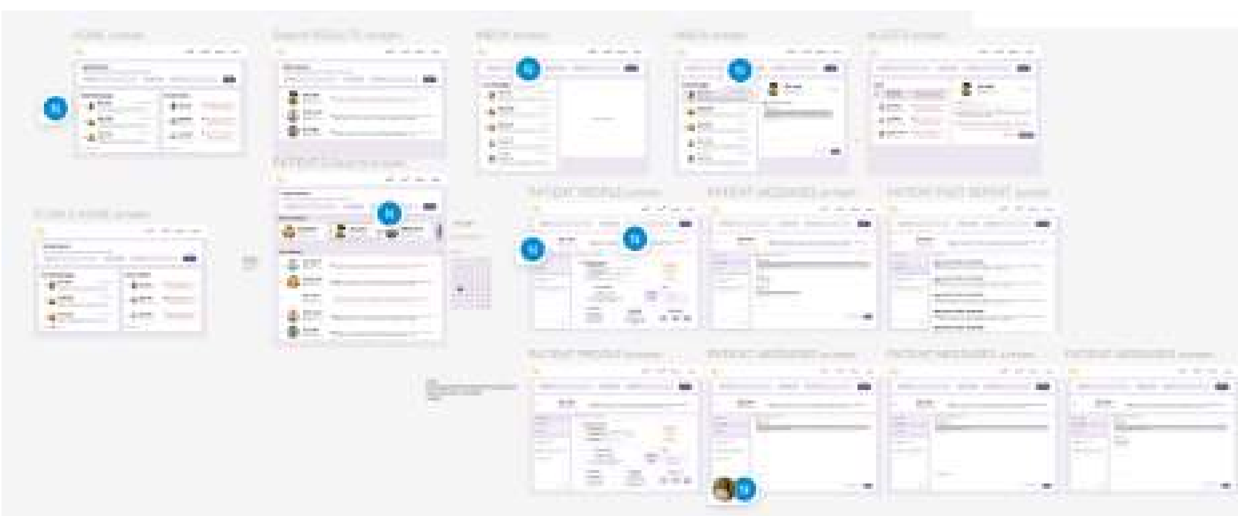
Design-wise, the User UI and Provider UI teams decided to create higher fidelity screens using the Material Design System. We created a few components with the material design assets (seen below or in more detail here).



These were used to create higher fidelity screens for the provider website–including the home, search, inbox, alert, and patient profile screens. As inspiration for the home screen, we used the tool: v0 to draft some ideas. Subsequent screens were created with the aesthetic of the home screen in mind. Please see below or in more detail [here](#).



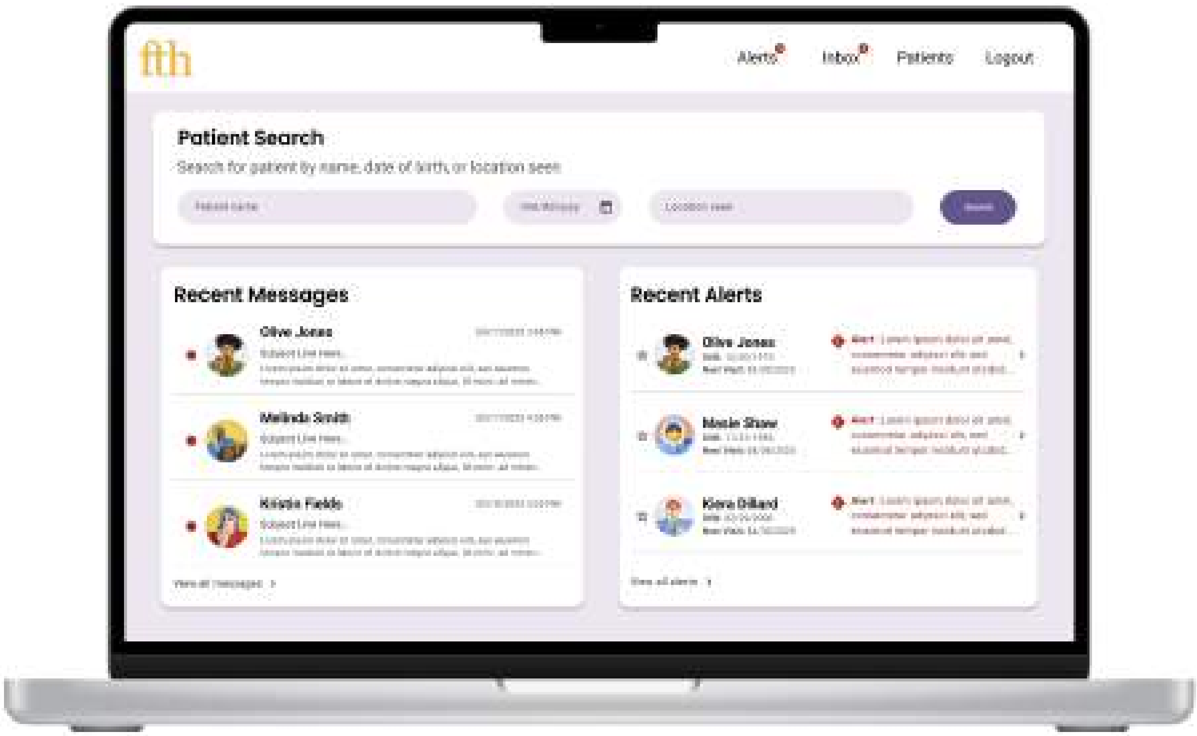
(v0 inspiration)



(designed screens)

Prototype at the Conclusion of Phase 2

We used these screens to create a user flow for the Phase 2 presentation. This flow followed a provider as they checked their messages, viewed the patient's data dashboard on their profile to inform their response, made a few notes, and copied those notes when they replied to the patient. We have included a screenshot of the prototype screen below, but you can interact with the prototype in more detail here.



5.1.7 Wearable HCI

During Phase 2, we focused on iterating on both the user interface design and the overall industrial design of the wearable.

5.1.7-1 Interaction Flow Diagram

Based on the overall visionary scenario and the wearable specific functional requirements we projected at the end of phase 1, we moved into the design phase with a interaction flow diagram that aims to reflect the following key functional requirements in sequential order:

- Continuous health signals monitoring from the wearable sensors
- Fall detection and emergency alerting
- Medication and hydration reminders

- Activity tracking
- Data collection for providers and user applications

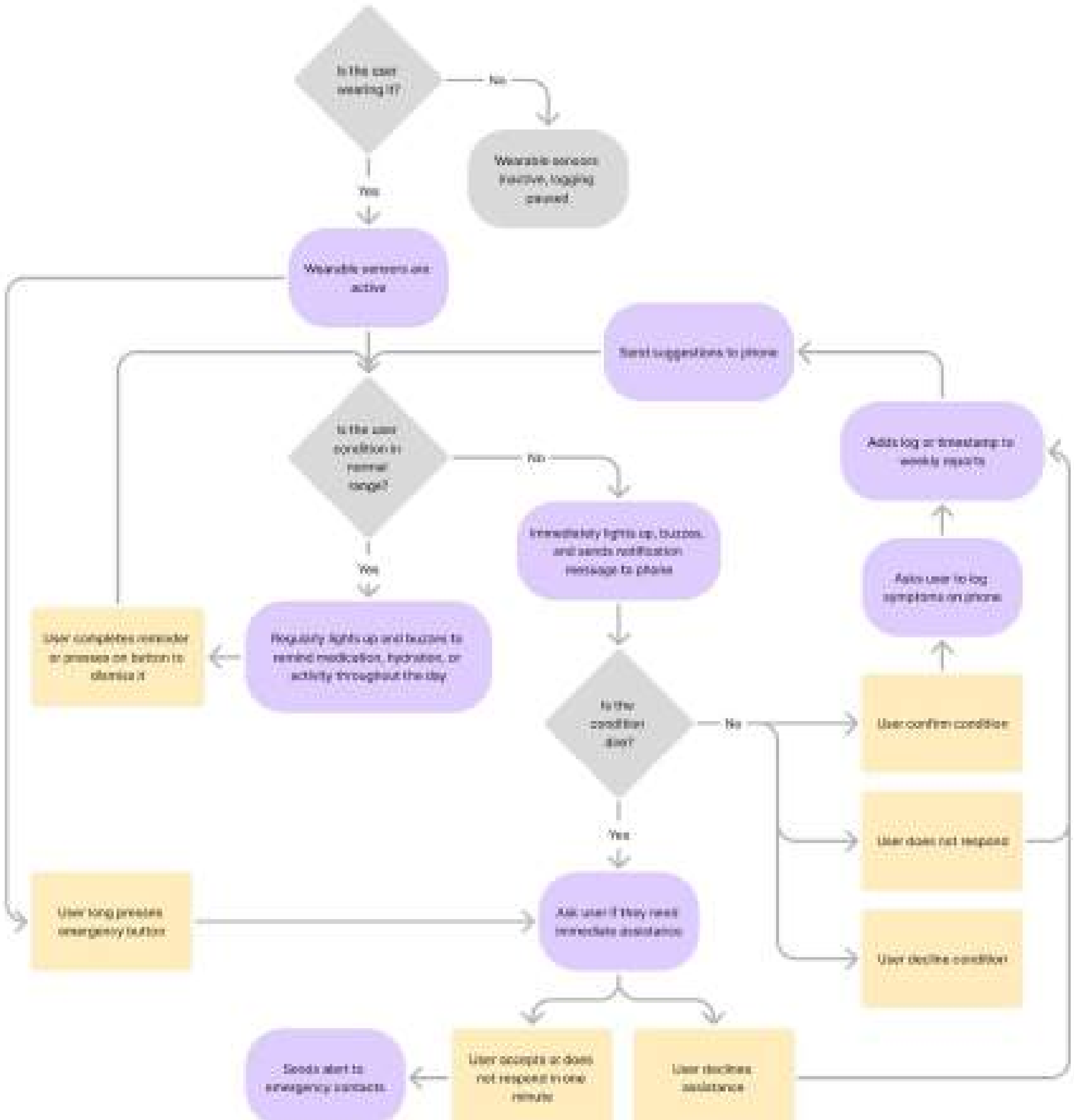


Figure: [Wearable interaction flow diagram]

5.1.7-2 Product Exterior and Material Selection

We designed the layout of the display, LED status lights, speakers, and push buttons. To ensure usability, we considered various interaction methods such as voice input and gestures, though technical complexity and limited data transmission channels required us to rely on button-based navigation rather than a touchscreen. Considering these decisions, we modeled a digital and a physical prototyping and printed out with thermoplastic polyurethane (TPU) to recreate the flexibility needed to wrap around the wrist.

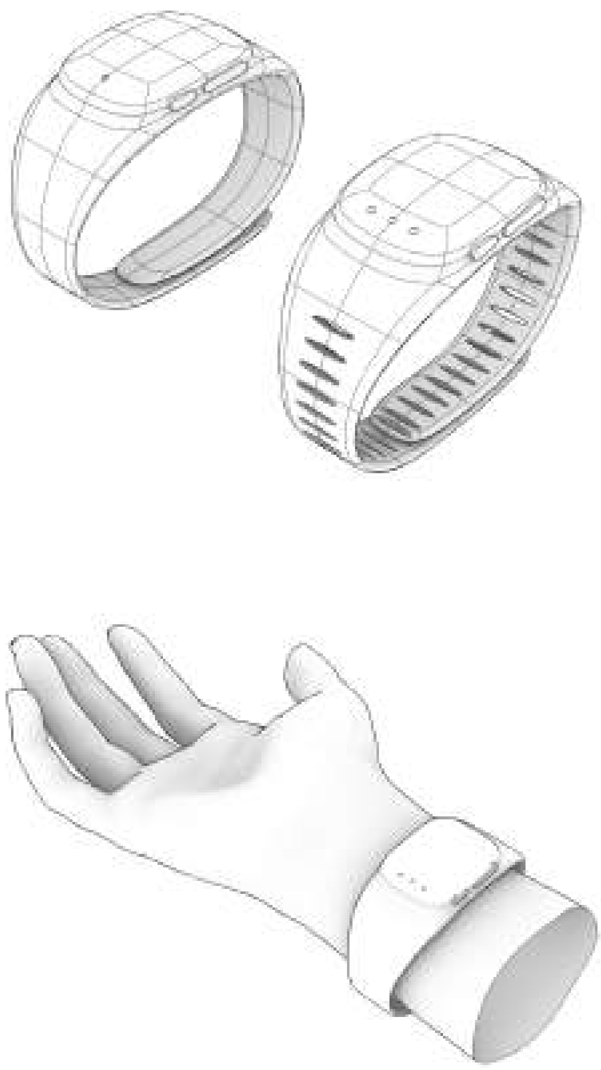


Figure: 3D model of the wearable in Phase 2

5.1.7-3 Determining User Interface Hardware

To integrate the display into our design, we evaluated different display technologies and worked with the Hardware Team to determine the most suitable option. After discussions with the wearable hardware and firmware team, we narrowed the choices down to LED or LCD, balancing factors such as power consumption, visibility, and hardware compatibility.

5.1.7-4 Designing User Interface, Cues, and Feedback Mechanism

With the display technology in mind, we designed the detailed user interface and iterated on finding the appropriate level of cues and feedback wearable. Additionally, we aligned with the intelligence team to determine thresholds for different alerts and notifications. Below is a table of all the notifications, alerts, and user input implemented on the wearable.

Section 2

Signal Name	Display	Update Frequency (non-alerting)	Alert/Reminder Condition	Haptic Alert Pattern	Dismissed Condition
Heart Rate		• Every 1 minutes at rest • Every 5-10 seconds when anomaly detected	Above/below normal range (for 10+ minutes at rest?)	Strong, long x #Risk level	Long press large button or user checks phone
Blood Oxygen		• Every 5-10 minutes • Every 1 minute when anomaly detected	Below normal range	Strong, long x #Risk level	Long press large button or user checks phone
Blood Pressure		Twice a day (or however many times the user measures through the external device)	Above/below normal range (and increase of ≥20/15 mmHg between measurements?)	Strong, long x #Risk level	Long press large button or user checks phone
Fall		N/A	Immediate when fall detected	Strong, short & quick x 7 (+ audio + 30 sec countdown before escalating)	Long press large button to dismiss, long press small button to escalate. Auto escalates after 30 sec of inactivity
Body Temperature		• Every 30 minutes • Every 10 minutes when anomaly detected	Above/below normal range (and increase of >1°F (0.6°C) within 1 hour?)	Strong, long x #Risk level	Long press large button or user checks phone
Hydration		N/A	Every 2 hours, adjustable	Weak, short x 2	Press any button or goes away after 5 sec
Activity		N/A	Two hours of inactivity while not asleep	Weak, short x 2	Press any button or goes away after 5 sec
Medication		N/A	1 hour after prescribed time if meds are still not taken	Strong, long x 2	Press any button
App Notification		N/A	New notification from system or provider	Weak, short x 2	Press any button or goes away after 5 sec
Battery		N/A	Battery < 20%	Weak, short x 2	Press any button or goes away after 5 sec
Meal		Always available to accept input	Everyday before bed?	Weak, short x 1 every increment by user	Long press any button to start input; press large button for increment, small button for reduction; long press any button to confirm

Figure: [Notification pattern designed for the wearable in Phase 2]

5.1.7-5 Scenario Simulation and User Feedback Mechanisms

To make sure users can engage with the cues and feedback patterns above, we designed three types of interaction that allow users to switch views, receive notifications/alerts, and input likert scale value on the wearable using simple button presses. Click on the links below to view animated interactions.

Figure: Interaction for [switching between views]

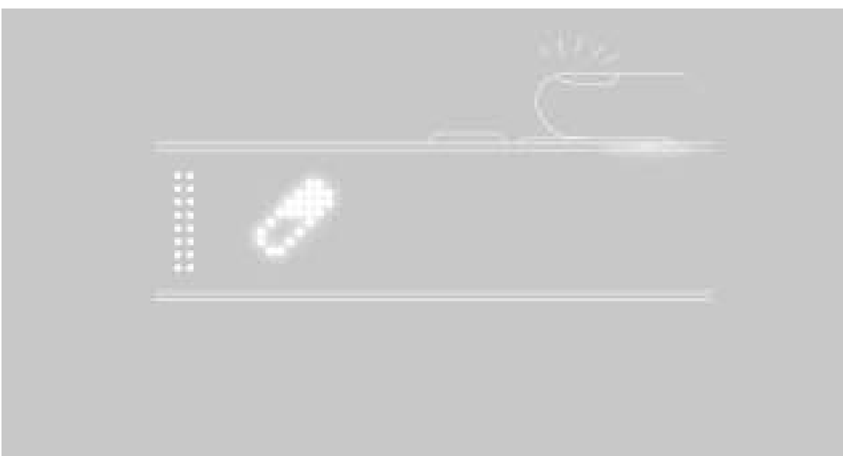


Figure: Interaction for [receiving notifications]

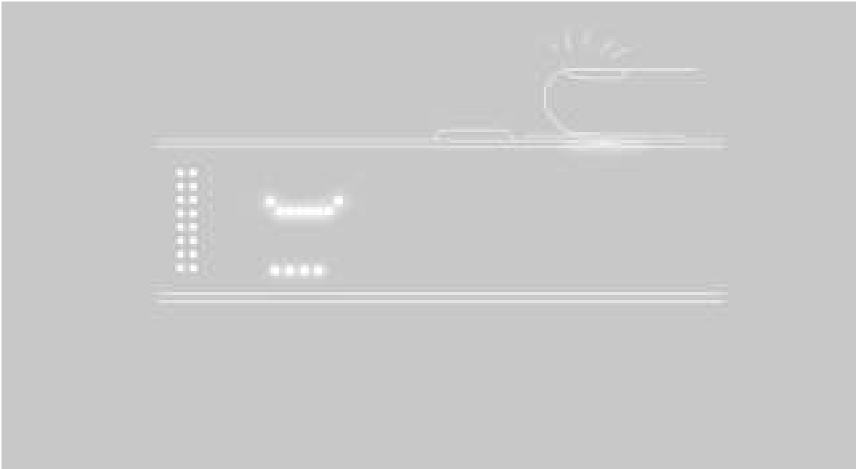


Figure: Interaction for [inputting mood changes]

5.1.7-6 Prototype Video Production

To effectively communicate our progress, we filmed and edited a prototype video showcasing common scenarios including fall detection, mood monitoring, and medication reminders. This video helped illustrate how users would interact with the device and how it would respond to different situations, making our design concepts more tangible.

Figure: a screenshot from the wearable [demo video]

5.1.7-6 Final Prototype design

Based on our evaluation and testing of the prototype in phase 2, we decided that making the interface a horizontal screen that is perpendicular to the wrist provides more space for displaying information and for the user to see the screen more conveniently when they are holding the baby. We designed new interfaces with clearer information and affordance, with important vital signs such as heart rights displayed on the right, icons displayed on the left, and lights signifying state of health underneath.

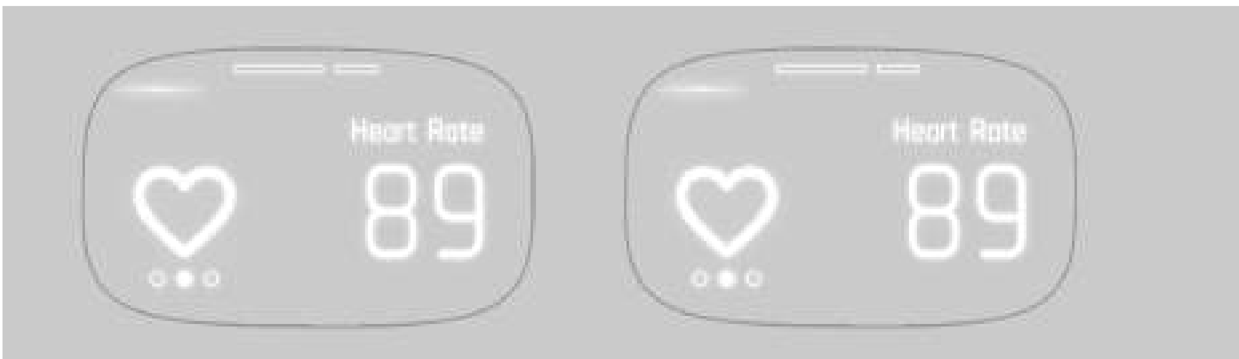


Figure: New interface

After sketching out the final design, we modelled the prototype in Rhino. The main changes made were stretching the screen to a 1:2 ratio to fit with the PCB hardware (after the hardware team showed us the printed PCB board), and turning the prototype into a two-piece object, with different materials are wristband and screen. In order to fit the wristband into the screen component, we used the dimensions of the wristband to cut out appropriate slots on the sides of the screen.

Figure: Digital model of wearable device

5.1.7-8 Final Prototype printing and assembly

We tested three different printing materials: PLA, White resin, and clear resin.
We preferred resin over PLA for the appearance modell, because it has a smoother finish and more refined edges compared to filament printing. However, the PLA prints came out to be more sturdy and resistant to compact, while the resin prints were very brittle and broke easily under pressure, especially when we pushed the wristband through the slots of the print.
Finally, we assembled the 3D prints with the wristbands and pasted the printed interface for display.

Figure: Assembled prototype

Figure: Final prototype and comparison to PCB

5.1.8 Environment HCI

5.1.8.1 Research

We researched the market and found that smart monitoring tools are still underdeveloped. We see a demand for environmental sensing and medication management features in postpartum.

With the background information and functional requirements from the Environment Hardware/Firmware team, we started by researching existing market products and commercial designs.

Here are a few existing products we looked at:

1. **Medical-Grade Philips Monitors:** Accurate but complex to use; not friendly to the average user who has no medical background
2. **Xiaomi Smart Home:** User-friendly and fits our overall vision of the home station; lacks some functions and content
3. **Smart Home Google Nest:** Strong ecosystem integration but does not have health or postpartum-specific features

From these existing designs, we concluded key features that we would like to incorporate or avoid in our iterative process. By reviewing and researching existing products, we noticed that a gap exists in the integration of environmental, health, and medication tracking.

We would also like to exclude potential harm that may be brought to postpartum moms when they are using the product. Therefore, we looked at design guiding for a postpartum physical interaction device and came up with three key risks to address:

1. **Physical safety:** Rounded edges and safe materials
2. **Information overload:** Simplified interface with fewer text and info
3. **Psychological stress:** Warm-toned alerts to reduce anxiety

In addition, we talked to one mom to understand her needs:

1. **Unobtrusive monitoring**
2. **Clear, quick information**
3. **Error-proof design**
4. **No harm design**

Our interviewee was managing a newborn while taking care of other children. She mentioned that she wants devices that blend into her daily routines, and display clear, glanceable information, and has error-proof designs to account for mistakes, which physically prevents potential harm to her and her kids.

5.1.8.2 User Flow

The Environment HCI team created a user flow to ensure that all parts of the system - the physical base station, the base station interface, and the smart pill box - work together smoothly. The user flow helped us visualize how users interact with each component, without any parts of the design being disjointed or disconnected.

The three components of design each take on a unique position:

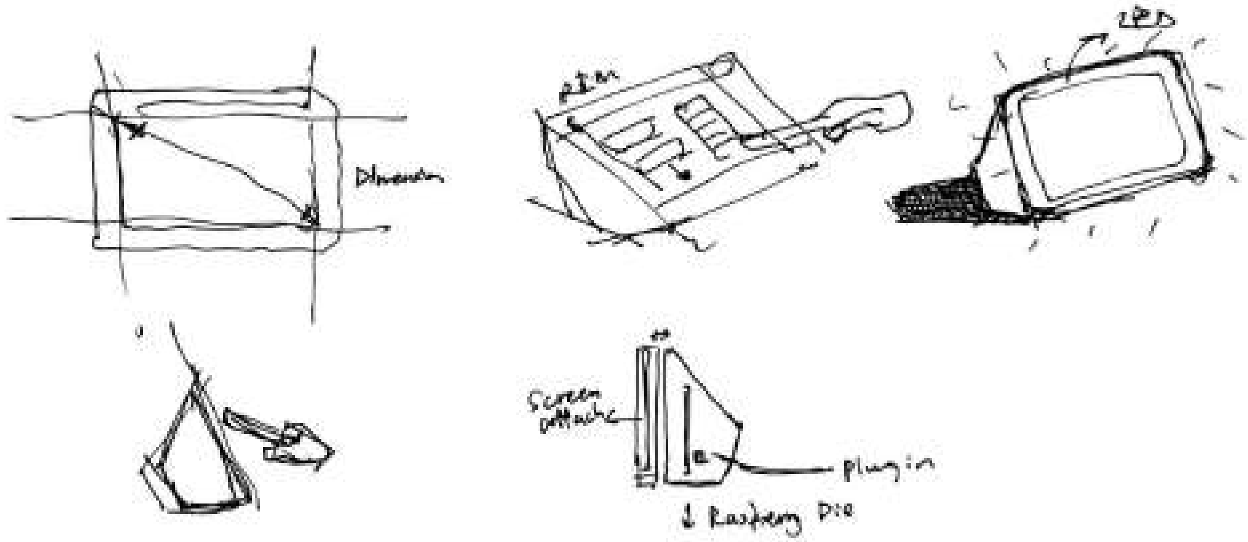
1. **Base Station (Physical):** guides the user to touch or interact with the device; uses LED light to indicate the user's state of being
2. **Base Station (Interface):** guides and helps the user to set up all the physical devices for postpartum monitoring; displays environment information for the user to view; pushes notifications to remind the user
3. **Smart Pill Box:** tracks the user's medication intake status; syncs with the overall system to manage medication and push reminders

In the user flow, before giving birth, the user would come home after a doctor's appointment and bring home all the physical devices for postpartum monitoring. The user would plug in all stations as instructed. Then, the user would be prompted by the base station interface to set up the environment. After giving birth, the user would use the screen interface to view information and receive notifications, the base station LED to understand their state of well-being, and the pill box to take medication.

5.1.4.3 Base Station Physical Design

Ideation & Concept

Based on our research, the Environment HCI team created rough idea sketches to set the initial direction and form for the physical base station. The base station's physical design prioritizes intuitive, stress-free interaction for postpartum users. We imagine it built with a minimalist aesthetic, features a lightweight, smooth casing, and child-safe rounded edges to blend into bedroom environments or living spaces.



Iteration 1 (Phase 2)

For iteration 1, the base station features a slanted form to ensure stability, at the same time creating natural alignment for easy access during feeding or rest periods. It can be flexibly placed on nightstands, dressers, or mounted surfaces without dominating the room. The back of the base station is made to fit the hardware components.

Another key design is the responsive LED ring surrounding the screen. It communicates the well-being status of the user through color and rhythm: a calming white light indicates stable vitals, yellow light indicates need for attention, and a red pulsating light indicates alerts for urgent issues.

Iteration 2 (Phase 2)

For iteration 2, we made a few adjustments. Firstly, we changed the tilting angle to 12 degrees to ensure better stability and usability. This follows the ergonomics and makes the base station responsive for postpartum moms. The affordance/visual cues align with users' needs for glanceable feedback, avoiding overwhelming interfaces. We also adjusted the depth of the back of the station for both better stability and accommodation for hardware components. In addition, we modeled the component that the screen and the LED light will be mounted onto.

Final Iteration (Phase 3)

For the final iteration of the base station, we kept the 12-degree tilting angle for ease of use. We further enlarged the back half of the station to make it more stable and fit the hardware components.

After communicating with the hardware team, we learned that a pogo pin and the thermal camera needed to be added. We placed the pogo pin on top of the base station case ledge so that users could stably place and easily plug in the substations for Bluetooth connection. We also explored how to best integrate the thermal camera. Two main factors played into our decision:

1. The thermal camera is connected to the Raspberry Pi (which is inside the base station case) through a USB cable
2. The photo-taking process needs to be triggered from the base station screen (not a physical button on the camera)

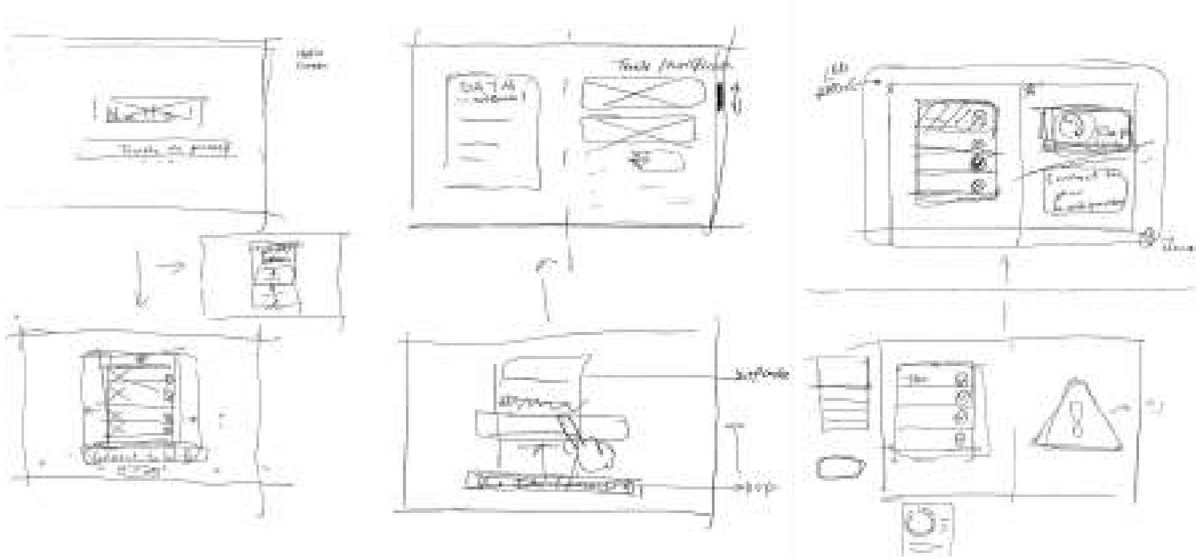
Therefore, it did not make sense for the camera to be a completely separate substation standing on a tripod that the user needs to plug in every time they have to take a photo. To resolve this, we created a slot in the back of the base station where the thermal camera can be kept when it is not in use. When the user needs to take a photo, they can easily prop the camera up in the slot and follow the on-screen instructions.

Final Iteration 3D Print

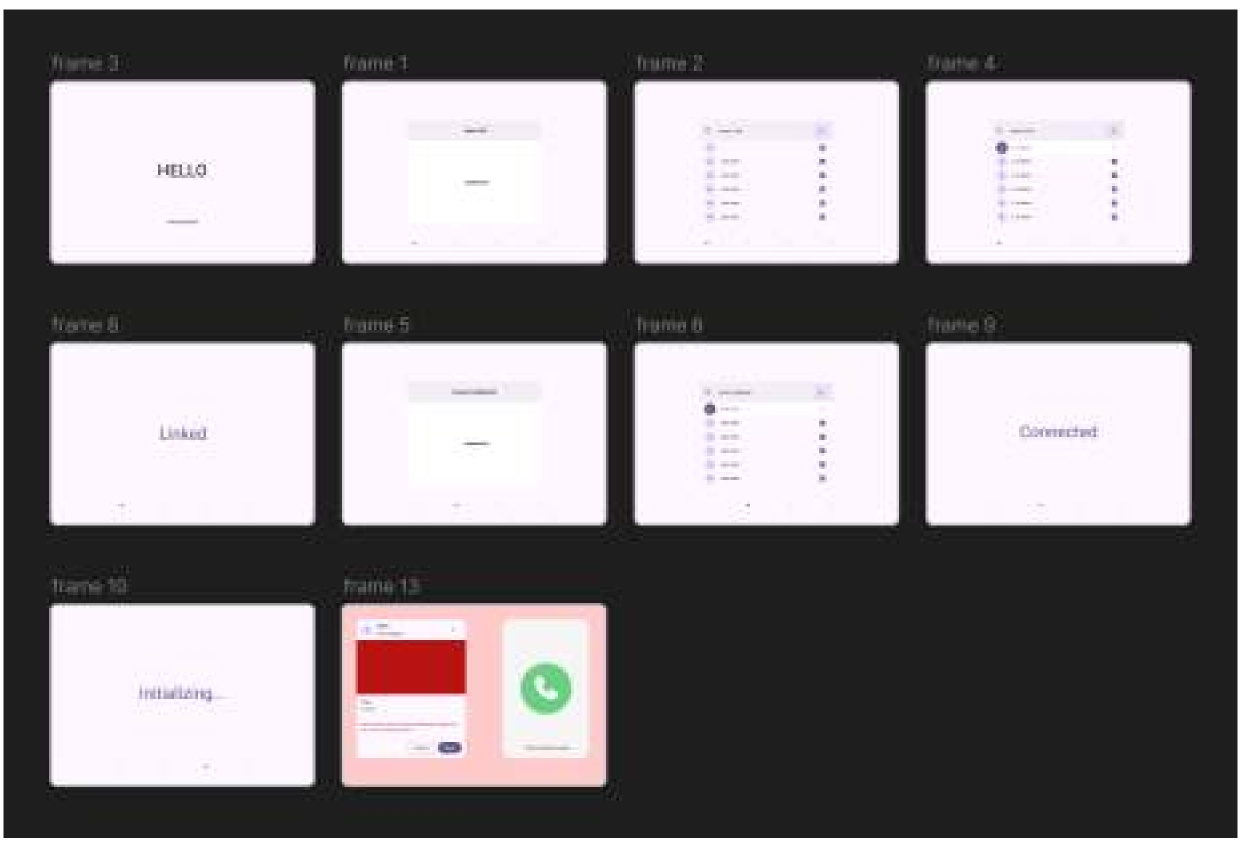
5.1.8.4 Base Station Interface Design

Wireframe

Based on the initial user flow, we first created a set of wireframes. The goal of the wireframes is to design the rough layout of all screens. The three main sets of screens include the welcome screens, the setup screens, and the info and notification screens. The sketched wireframe focuses on the placement of visual components that would be intuitive for users. The Figma wireframe, on the other hand, focuses on adopting the *Material Design Kit* components into our vision so that the HCI designs would be coherent across the project.



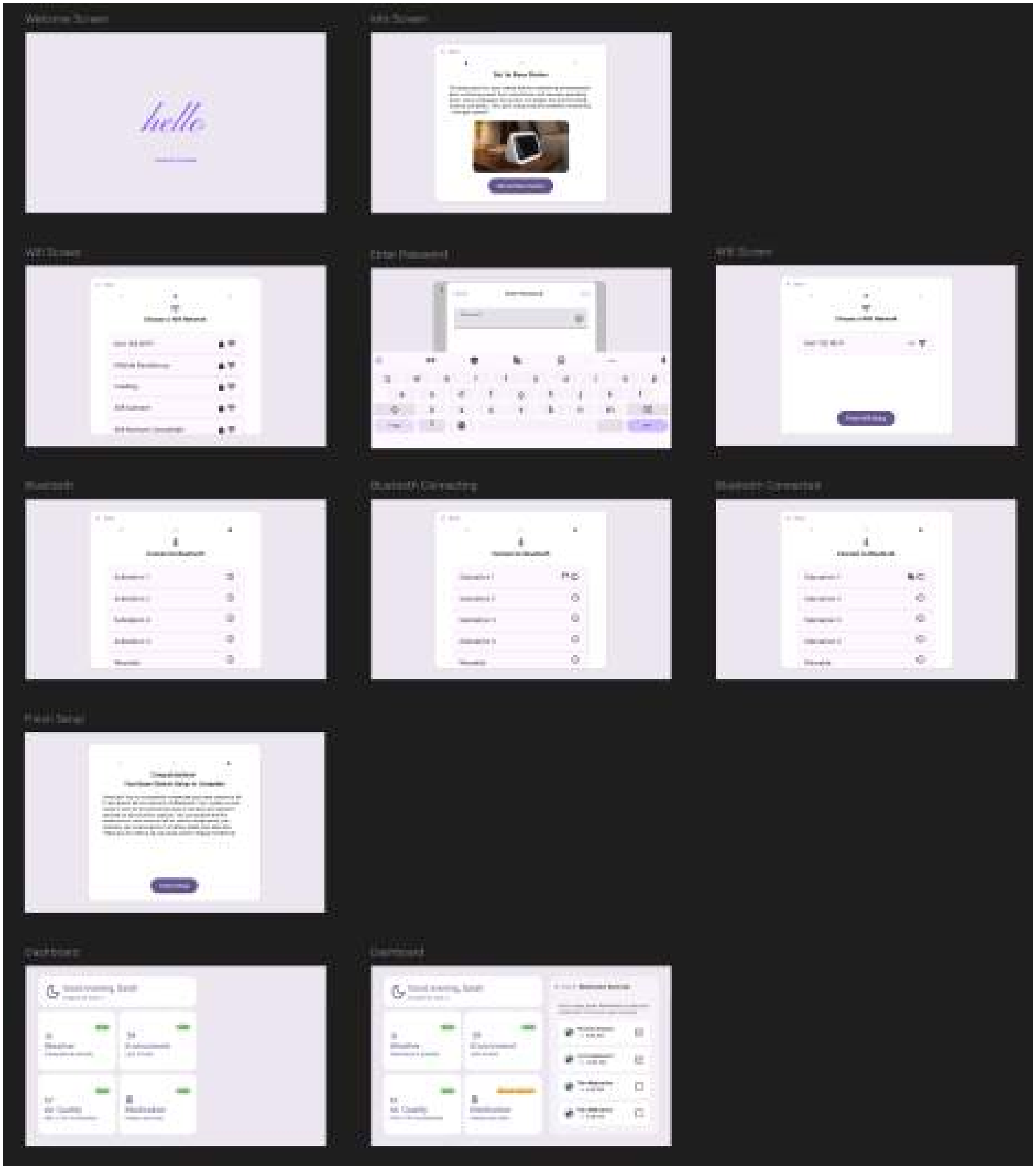
Sketch Wireframe



Figma Wireframe

Iteration 1 (Phase 2)

Based on the wireframes, we refined this interaction of the high-fidelity mockup. The screen welcomes the user, introduces the setup process, guides the user to set up WiFi and Bluetooth, completes setup, displays environment info, and pushes notifications. We incorporated the design system and theme colors. Components like buttons, icons, and data visualization were aligned with the other HCI teams' designs. We also created subtle animations for a better user experience.

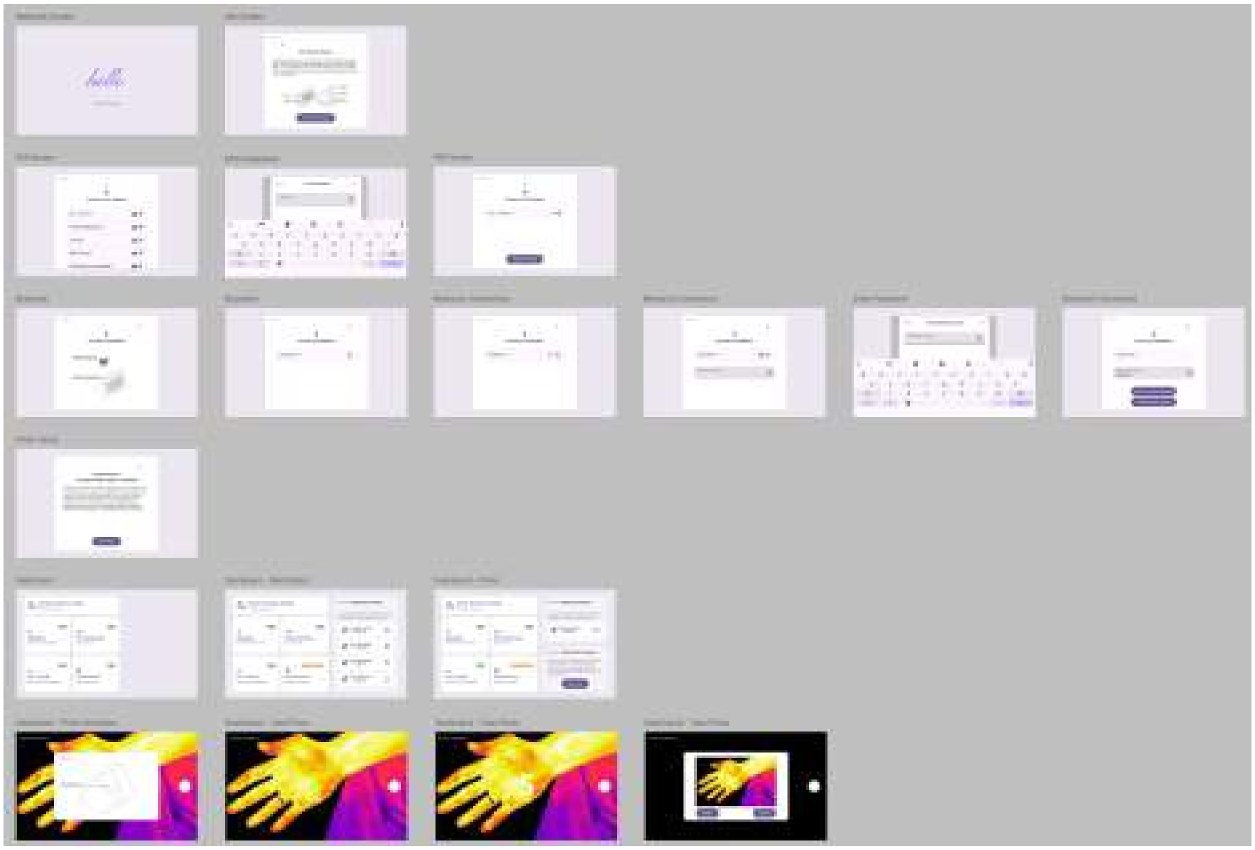


Figma Mockup

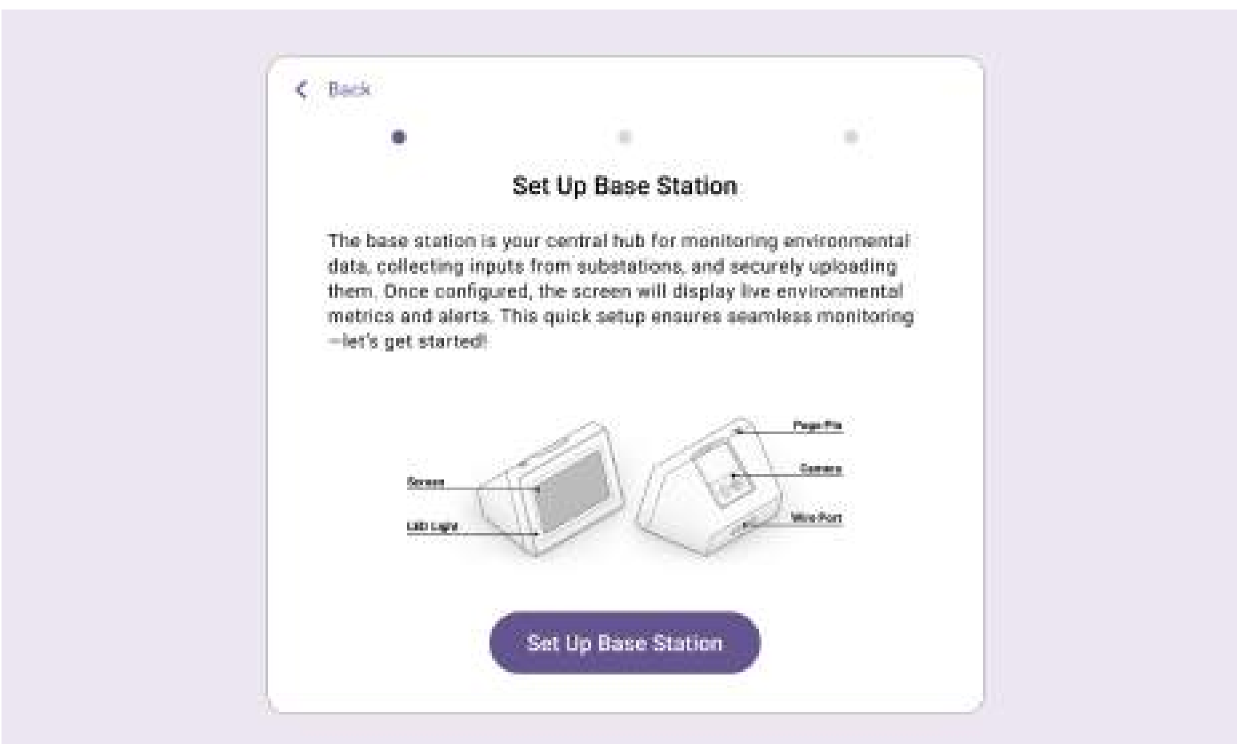
Demo

Final Iteration (Phase 3)

For the final iteration of the base station interface, the overall flow stayed mostly the same. We added new screens to support the pogo pin and the thermal camera features. Instruction photos and animations were created to guide the users.

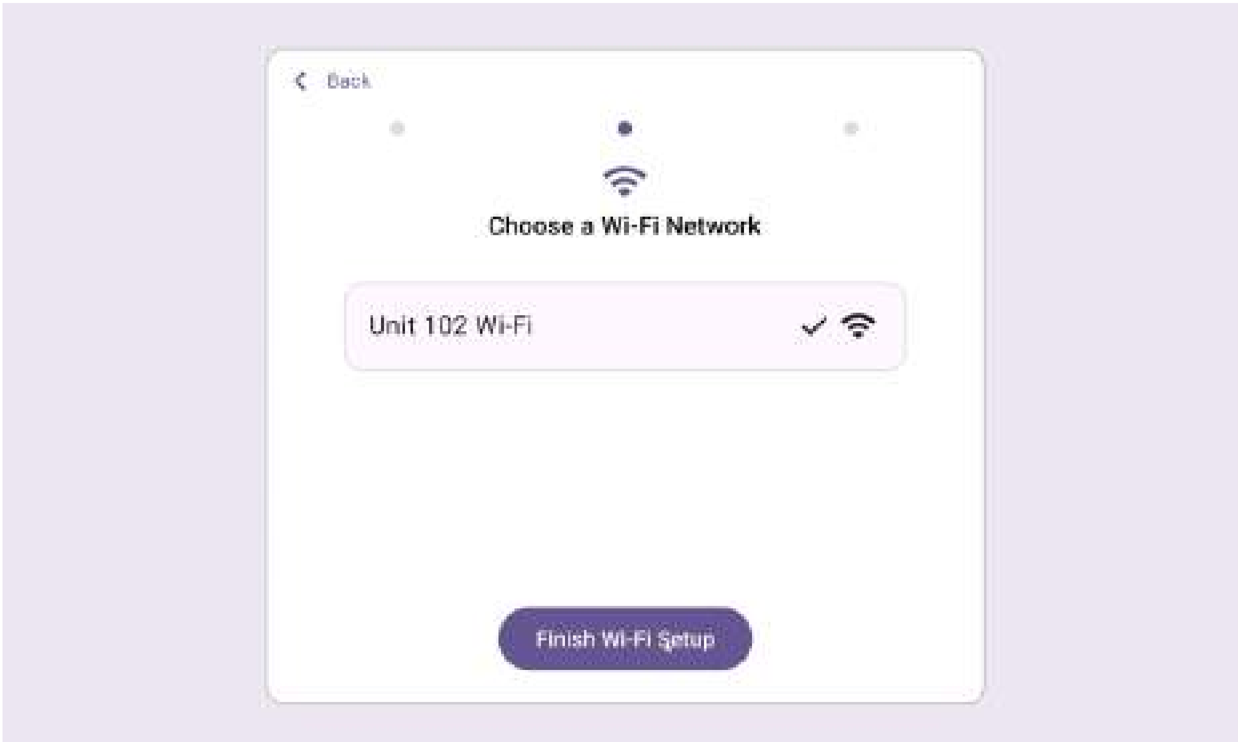


Figma Mockup



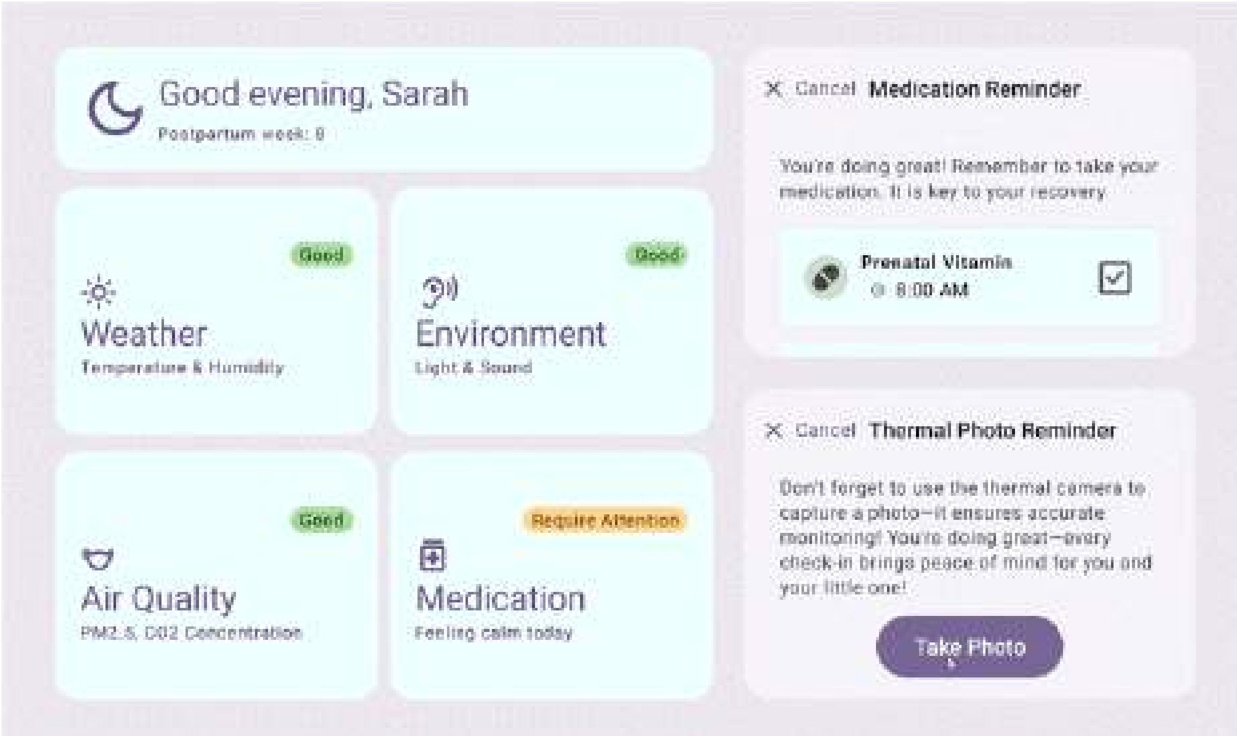
Initial Info Screen

One of the animations shows how to plug in the substation to connect to Bluetooth.



Bluetooth Connection

The other animation shows how to set up and use the thermal camera. We brainstormed different ways to trigger the thermal camera. Ideas included using the phone app as a trigger, using a pose of hand gesture as the trigger, or using a simple countdown. We chose the countdown method because it is the most intuitive and does not require any extra devices. Users can just follow the instructions on the screen and take the thermal photo easily.



Thermal Camera

5.1.8.5 Pill Box Design

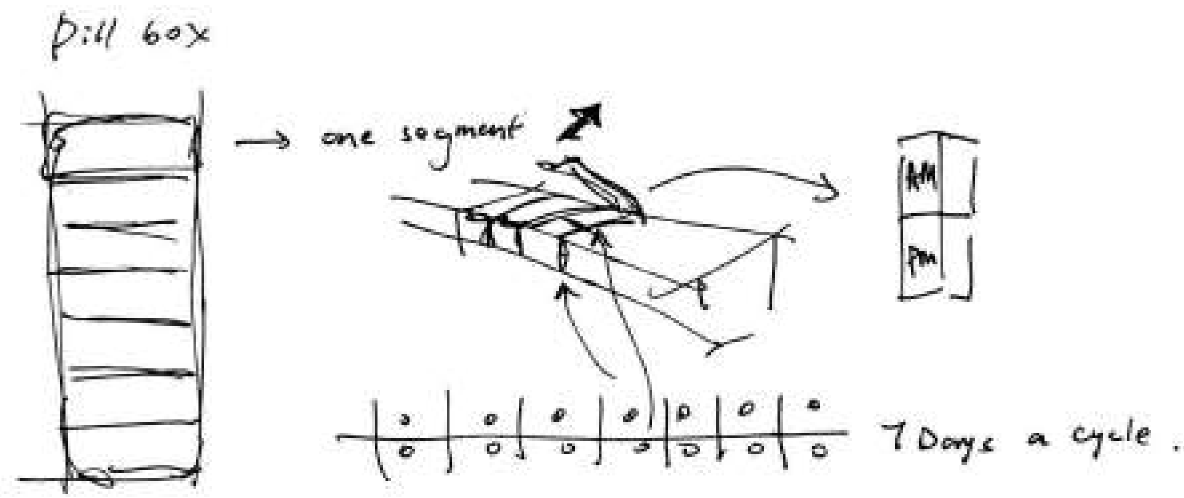
Ideation & Concept

We wanted to create a smart pill box that is functional, simple, safe, and usable. A few key considerations include:

- **Simplistic Physical Design:** The pill box should minimize complexity and be easy to fit into a bag
- **Time-Based Pill Slots:** The slots are divided into morning and afternoon to align with a common medication schedule to reduce confusion
- **Clear Labeling:** There should be labeling of the day and time
- **Space for Sensor Integration:** The lid of each pill slot should have a flat area that is large enough to attach the sensor
- **Ease of Use with Safety:** The pill box should be easy to open (e.g., not a screw cap) but prevent children from using

The last consideration emerged from our interview with a mother of multiple young children. When she is taking care of her newborn, she has limited energy to monitor her other children.

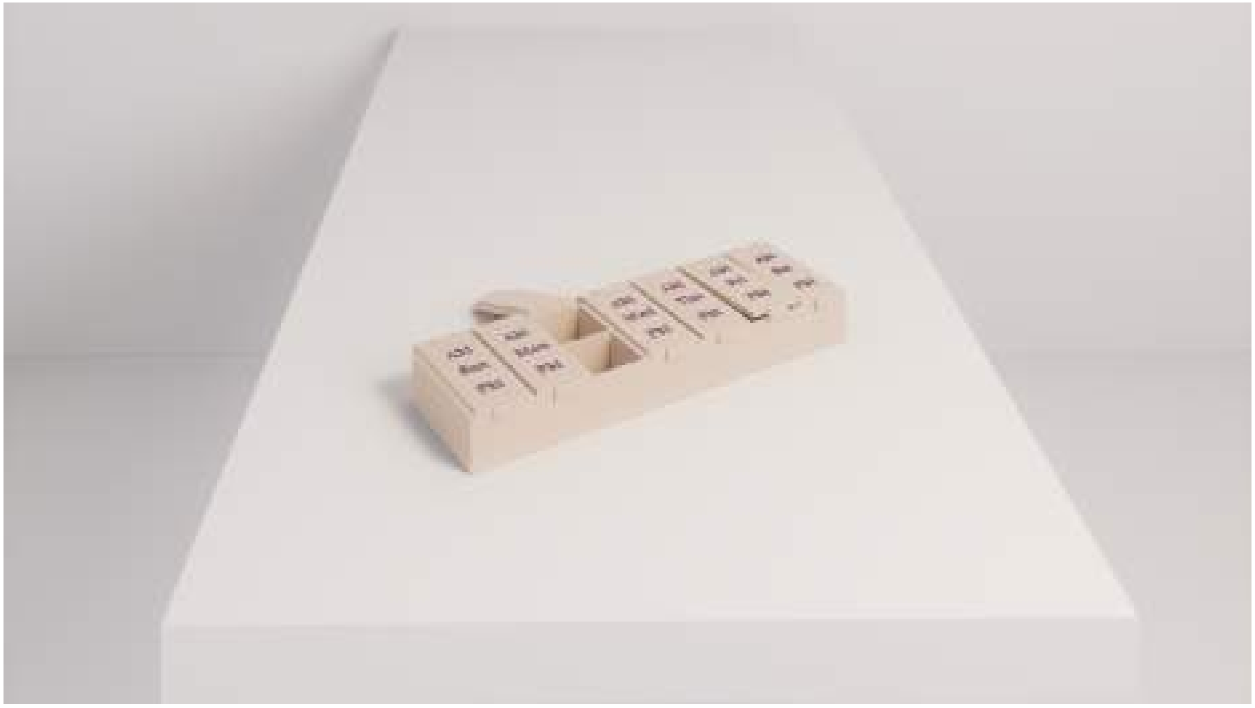
She expressed concerns with her other children “messing with things they’re not supposed to”. Medication would fit within that category of concern.



Iteration 1 (Phase 2)

Based on our ideation, we created this iteration of the pill box design. The pill box has a simplistic, compartmentalized design. Each slot is clearly labeled with the day and the time in large, raised text. The lid clips to the pill box, making it harder for young children to open.

The next step would be to test alternative lid types and see which design offers the best combination of usability and safety.



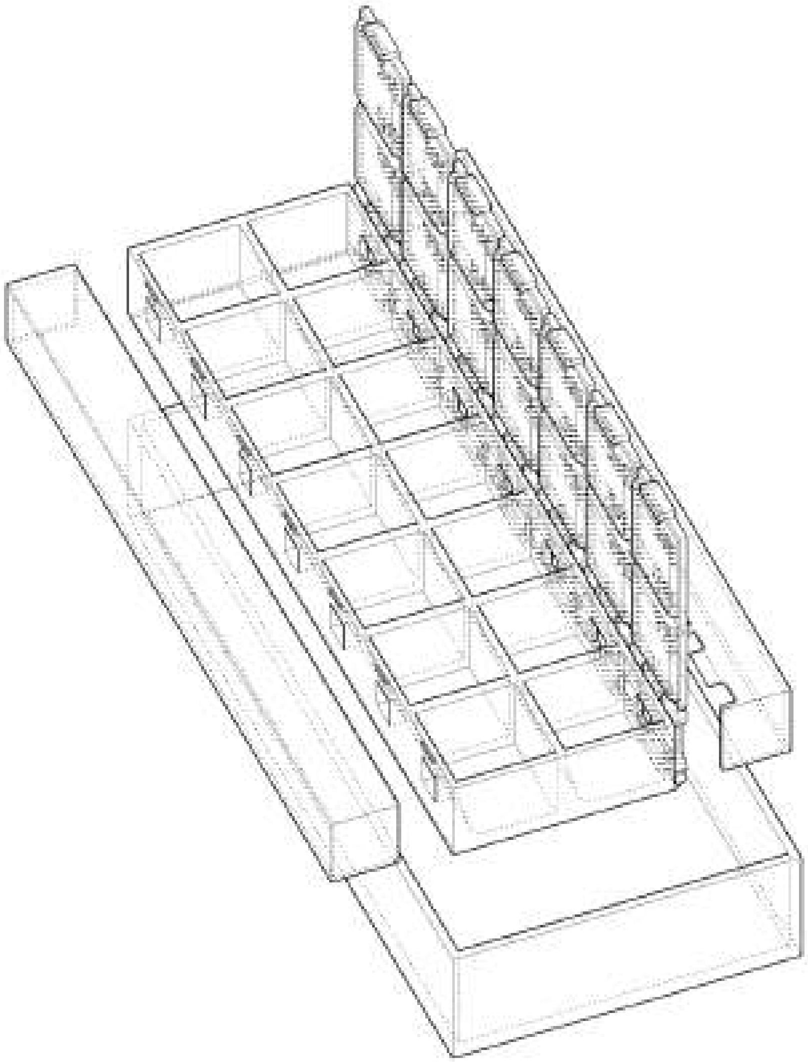
First stage Render

Final Iteration (Phase 3)

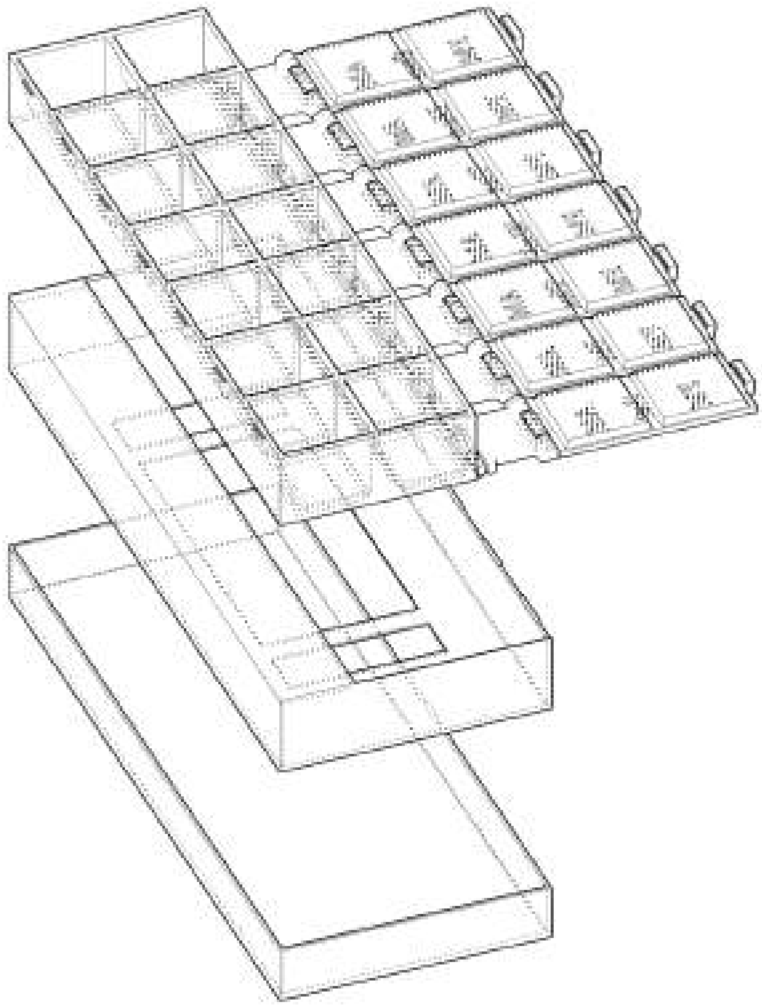
- Accommodate the hardware needs
  We are given more details and scenario-specific feature requirements after assessing the environmental team members' needs and feedback. The pill box, after linking to data and clouds, accesses the medicine amounts at each cell. The sensor uses ultraviolet light to monitor the cell space and pill position in the space. Once it detects it, the sensor will recognize the pill and update this cell's information to the cloud.

  To accommodate the environment hardware team's needs, we chose to add an addition to the original phase 2 pill box, and we added sensors, a Raspberry pie chip for connection, and a power bank for power supply. They are all inside the one minimalistic design casing, which connects to the pill box.

- Sensor on the side VS sensor on the bottom
  We tested two different design iterations, one of which is that all sensors and the chip are under the bottom of the pill box, with cells opening to the bottom. Another iteration is that sensors reside at the side, with the chip and power bank at the bottom.



Sensor on the side of the pill box iteration



Sensor at the bottom of the pill box iteration

We chose the Sensor on the side iteration. Here are the benefits in comparison to the Sensor at the bottom of the pill box iteration.

- First, the Sensor on the side iteration has smaller space use and utilizes the blind area well, allowing efficient sensor wiring connections to the chip.

- Secondly, a Sensor attached on the side of the cell can have a higher detection rate, making the sensing process fast and accurate, compared to the low rate of sensors at the bottom iteration, which saves time for adjusting and repair.

- Finally, the material cost is lower, making the profitable and fast for production.

5.1.8.6 Final Rendering

Phase 2

*All physical parts are 3D printed and handed to the Environment Hardware team for testing

5.1.8.7 Reflection

Throughout the project, our Ambient HCI team identified possible guardianship does and environmental gaps brought about by postpartum environmental perception. We responded to this need and thus designed a seamless system combining a base station, a smart pill box, and an intuitive interface that interconnected with other teams.

Through market research, interviews with postpartum users, and iterative testing, we prioritized safety, simplicity, and unobtrusive interactivity.

The physical base station was designed with ergonomics, LED feedback, pogo pin integration and thermal camera photography in mind. We designed the base station's interactive interface to provide clear guidance to help women recover from childbirth.

In addition, we designed a complementary smart pillbox. By combining child safety, clear labeling, and real-time pill tracking with efficient sensor locations we made the smart box efficient and easy to use.

Through our analysis and back-and-forth with the Environmental firmware team and iterative design, we arrived at a well-thought-out system that supports postpartum care through convenient, easy-to-use, simple and effective technology.

5.2 Provider/User Software

In Phase 1, we proposed a postpartum health monitoring platform consisting of a Flutter-based mobile app for patients and a React-based web dashboard for providers, supported by a shared Node.js backend hosted on AWS with DynamoDB. The mobile app focuses on mood tracking, real-time alerts, and wearable integration, while the provider dashboard enables proactive intervention through data visualization and alerts.

To meet the distinct needs of users and providers, we divided the system into two tailored solutions: a mobile app offering patients simple, accessible tools, and a web app providing medical professionals with more advanced functionality for data analysis and patient monitoring.

5.2.1 Initial Research (Product/Feature Matrix)

A major part of Phase 1 for our team was researching possible options for frameworks and/or libraries to aid in our design of the system. We considered a number of factors, including ease of use, familiarity of the tool within our team, potential for complexity and scalability, security, and community support. Following is a condensed summary of the findings of our research, including multiple backend and frontend options for both website and mobile application development.

5.2.1.1 Website Development

We researched five different options for our *website frontend*, summarized below based on feature and base language(s).

	React	**Bootstrap**	**Vue.js**	**Angular**	**Semantic UI**	**Foundation**
Community Support / Documentation	Yes	–	–	–	No	Yes

Extensive Libraries	Yes	–	No	–	–	–
Ease of Learning	No	Yes	Yes	–	No	No
Responsive / Grid System	–	Yes	–	–	Yes	Yes
Component Based	Yes	Yes	Yes	–	Yes	–
Theming	–	Yes	–	–	Yes	–
Two-way Binding	–	–	Yes	Yes	–	–
JavaScript	Yes	No	Yes	No	No	Yes
CSS	No	Yes	No	No	No	Yes
TypeScript	No	No	No	Yes	No	No
HTML	No	No	No	No	Yes	Yes

Table 5.2.1.1.A Frontend Web Development Options

We also researched four different options for our *website backend*, summarized below based on feature and base languages(s).

	Django	**Node.js**	**Flask**	**Ruby on Rails**
Community Support / Documentation	–	–	Yes	Yes
Built-in Security Tools	Yes	–	–	Yes
Ease of Learning	No	–	Yes	Yes
Flexible	–	–	Yes	No
Scalable	No	Yes	–	–
Fast	–	Yes	No	No
Python	Yes	No	Yes	No
JavaScript	No	Yes	No	Yes

HTML/CSS	No	No	No	Yes

Table 5.2.1.1.B Backend Web Development Options

The final part of our web development research was three different options for a *web host*. Our findings are summarized below.

	AWS	GCP	Heroku
Ideal Project Size	Larger / Complex	Larger / Complex	Smaller / Rapid Dev
Complex / High Pricing	Yes	No	Yes
Wide Range of Add-ons	–	No	Yes
Scalability	Yes	Yes	No

Table 5.2.1.1.C Web Host Options

5.2.1.2 Mobile App Development

We researched options for four options for *frontend mobile development*, summarized below by pros/cons and the platform (iOS vs Android) it supports.

	Android SDK	React Native	React	JS/TS
Pros	Higher performance.	Can publish on the web, not the app store. Quicker dev process. Could share codebase with the website.	See Table 5.2.1.1.A	Could share codebase with the website.
Cons	Higher maintenance cost.	Slower performance than Android SDK.		Limited access to platform-specific features. Slower performance than Swift for iOS.
iOS	No	No	Yes	Yes

Android	Yes	Yes 	No	Yes

Table 5.2.1.2.A Mobile App Frontend Options

We researched two options for the *backend mobile development*, summarized below.

	Java	Kotlin
Pros	Good security features. Wide range of libraries (including ML), and tools.	Compatible w/ Java. Includes KotlinDL, a deep learning API.
Cons	–	Learning curve.
iOS	Yes	Yes
Android	Yes	No

Table 5.2.1.2.B Mobile App Backend Options

We also researched two options with support for *both frontend and backend mobile application* development.

	Swift/SwiftUI	Flutter
Pros	Fast and simplified UI dev. Cross-platform compatibility for Apple devices. Safety features. Integrates with Xcode dev tools. Performs better/less storage than Flutter.	Can develop Android and iOS in a single codebase. Targeted for mobile apps. Has many UI packages & libraries. Compatible with VSCode.
Cons	Limited compatibility with pre-iOS 13. Lacks advanced UI features.	Testing requires device emulation, which is complex. Compatibility issues with iOS (minor).

		Delay in accessing new iOS features.
iOS	Yes	Yes
Android	No	Yes

Table 5.2.1.2.C Mobile App Frontend/Backend Options

5.2.2 Final Design Decisions

After deliberation and discussion, we decided on Flutter for the mobile app, React and Vite with TypeScript for the website frontend, NodeJS for the backend, and AWS (with DynamoDB) for the host. The layout and communication between these sub-subsystems are displayed in Figure 5.2.2.A. This section will discuss our reasoning behind these design decisions.

After corresponding with some other subteams, we decided that the mobile app would collect wearable sensor data via Bluetooth, while both the mobile and web apps would communicate with APIs to manage user and provider data. These communication paradigms in some part informed our design decisions, as well as our previous research.

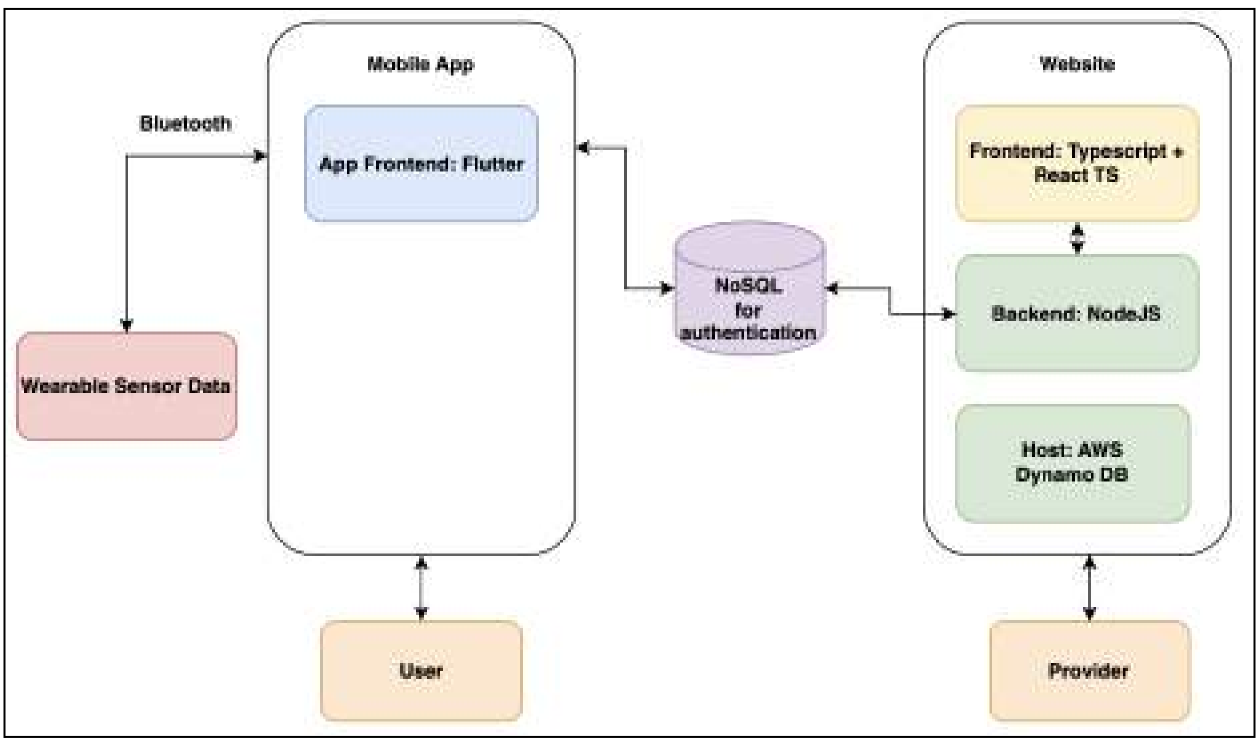


Figure 5.2.2.A Software System Architecture

We choose to use TypeScript as our base language, because it is suitable for large-scale applications and its static typing allows for error detection early in the development process.

There are multiple frameworks we explored which integrate with TypeScript, so that factored partially into our final decisions for those.

5.2.2.1 Backend Framework: NodeJS and TypeScript

Although we researched backend options for both the website and the mobile applications, we decided to have one coherent backend to ensure the website and mobile application have cohesive access to backend data and communication.

After discovering that the cloud subteam has decided to use DynamoDB for our project's database, we decided to use NodeJS as our backend framework. Its built-in library support for communication with DynamoDB will make API calls and transferring user data significantly easier. Additionally, NodeJS works with TypeScript.

NodeJS is asynchronous and event-driven, meaning it creates scalable applications and allows multiple tasks to run at once, which could be helpful for real-time communication between patients and providers. It also provides a rich package ecosystem and long-term corporate support, ensuring stability for our system.

5.2.2.2 App Frontend: Flutter

The HCI subteam is interested in using device accessories such as the microphone and the wearable hardware subteam wants to communicate with the mobile device via bluetooth. Both of these requirements need a language with easier to access device systems, which eliminates Javascript and React Native.

Another requirement our subteam has identified is being able to test on both iOS and Android, for wider access, which eliminates Swift, Android SDK, and Kotlin.

In the end, we decided on Flutter, which has many pre-built libraries that make rapid prototyping easier.

5.2.2.3 Website Frontend: React with TypeScript

React has a lot of community support[5.2.a], is very flexible for website applications, and is interactive, making it ideal for our dynamic content and complex interactions. Furthermore, our team is very familiar with React and TypeScript.

5.2.3 Expected Functionality

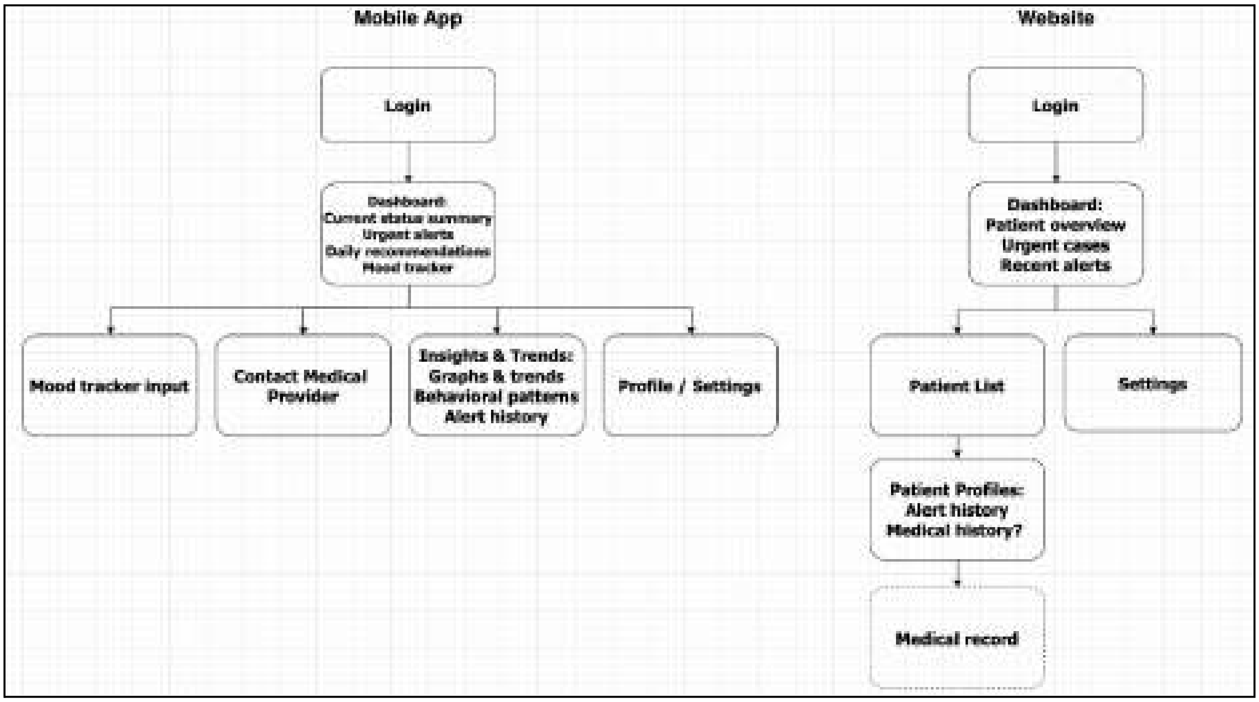


Figure 5.2.3.A: Information Architecture for Software Systems

Currently we made several decisions on platforms' core functionalities with HCI team: on the mobile app, patients can access a health status summary, send real-time urgent alerts, and use daily features like the mood tracker. Meanwhile, providers can monitor patient profiles and status through the web dashboard, including recent alerts & potentially the medical history.

These features ensure that both users and healthcare providers have a reliable way to track and respond to critical health data efficiently.

5.2.3.1 Development Outline

We plan to create the initial design for the website and app with Figma, which allows users to collaborate together to design user interfaces. After communicating with the HCI and hardware subteams to finalize software features, we plan to use Framer or Anima to create the website frontend, both of which would allow us to transform visual designs to actual code. After the cloud subteams start hosting our website, we can build the backend and test it with APIs calls. Most importantly, we will leave some slack time in case of deployment and integration issues.

5.2.4 Phase 2 Implementation Status

In this section, we will discuss the progress towards our design goal that we achieved during Phase 2.

5.2.4.1 Task Outline

During Phase 2, we met with the HCI team to determine the necessary functionality of the provider website and prioritize these tasks. Part of this interaction was defining which features were important to have completely functional end-to-end, and which could be implemented purely on the frontend and left open to future development. Following is an overview of the outcome of this discussion. It is not an exhaustive list of every task we are considering implementing, but gives us important information about what to prioritize for the MVP and what can be considered a reach goal.

Main Task	**Subtask**	**Robustness Goal**
Account Management	Create an Account	Functional
	Email Confirm to Verify User	Non-Function
	Log In	Functional
	Reset Password	Non-Functional
	Timeout on Period of Inactivity	Non-Functional
	Update Account Info	Non-Functional
Patient Search (Provider Side)	Load Patient Data for Viewing	Functional
	Search by Patient Name	Functional
	Search by DOB	Middle-ish
	Search by Date Seen	Non-Functional
	Search by Location Seen	Middle-ish
	Search by Multiple Parameters	Middle-ish
Chat/Messaging	Search Messages by Name	Functional
	Search Messages by Text	Middle-ish
	Search Message by Date	Middle-ish
	Message One Person	Functional
	Message Multiple People	Middle-ish

	Send Text	Functional
	Send Images	Functional
	Send Files	Middle-ish
	Send Alerts	Functional
Alert Tasks (User Side)	Pop Up Alert on Anomaly	Functional
	Send to Provider	Functional
	Send to Emergency Contacts	Middle-ish
Alerts (Provider Side)	Search Alerts	Functional
	View Report from Alert	Functional
Patient Profile (Provider Side)	Static View	Functional
	Dynamic View (Filter by Date)	Functional
	View Demographic Info	Functional
	Type Notes for Patient	Functional
	Save Notes for Patient	Middle-ish

5.2.4.2 Mobile App Frontend Progress

Our goal for the mobile app is for it to be able to provide mood tracking, analyze symptom surveys, provide notifications, reminders, and emergency alerts, and to have Bluetooth integration with wearable health-tracking devices. These functionalities rely mostly on data created by the intelligence team that is stored in the Cloud, along with raw data provided by the wearable hardware/firmware team's devices.

The app is being developed with Flutter, with the following screens having already been completed: a Login/Registration screen, a Home screen (in which the most recent notifications, reminders, and important health metrics can be viewed), a Messages screen (with which to contact providers), and a Profile screen (to view personal health documents).

Login Screen

Home Screen

Messages Screen

Profile Screen

In Phase 3, these screens will be slightly altered, with the “Sensors” page becoming a consolidated hub to view all details of important health metrics, and the “Profile” page becoming a “Settings” page through which users can adjust personal preferences and information.

Ideally, our app will be able to handle smooth daily input processes, local storage with cloud synchronization, and real-time form validation. In Phase 3, we will focus on completing the UI components of the frontend, form handling/logging, and API integration with external health-sensing devices. We plan to perform user testing for accessibility and ease of use on the mobile app to verify the robustness of our implementation. Currently, for the sake of rapid development, we use an emulator built into Android Studio to test initial prototypes. The emulator has a hot reload feature which allows users to see changes to their code without having to recompile everything. Ultimately, we hope to have our app locally running on a phone for demo purposes. The major challenge will be figuring out how to do so on an iPhone, which requires additional security verifications.

5.2.4.3 Website Frontend Progress

Our primary functionality goals for the website frontend are provider login, patient search, patient overview dashboard, alert logs, and messaging. The overarching programming tasks for these are dynamic filtering, searching, and sorting.

We have developed a locally running version of the website programmed and built with TypeScript and React. Currently in development are the home screen and the patient search screen, with the patient profile and message/alert screens pending development.

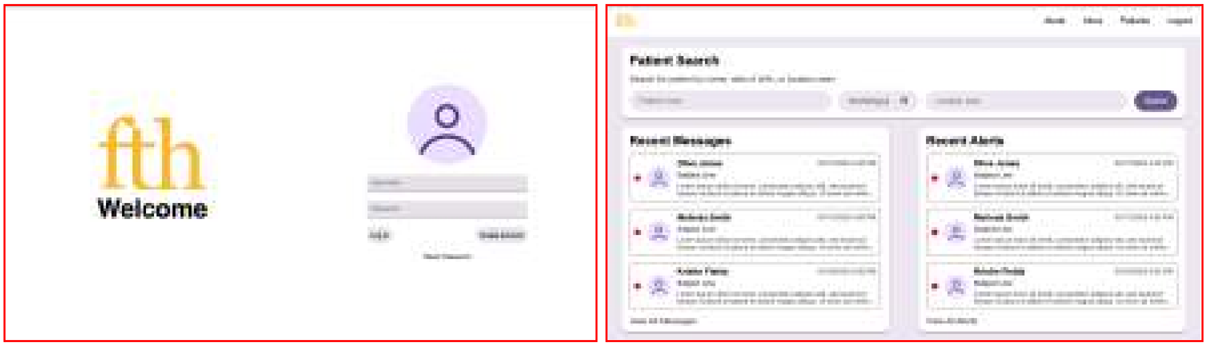


Fig. 5.2.4.3.1 In-Progress Home and Patient Search Screens, developed with TS and React.

Our first goal was to get some of these screens up and running with fake data and without the backend connection because the coding of the screens would not be trivial. The progress of the actual functionality on the backend will be described in the next section.

To streamline frontend development and maintain visual consistency, we plan to use Anima, an AI-powered Figma plugin that converts designs into responsive React code. This helps reduce manual coding and speeds up prototyping for layout-heavy pages like the home screen and patient search view, aligning the frontend closely with our intended user experience.

5.2.4.4 Backend Progress

In Phase 2, we started development on the backend, written in JavaScript with Express. Using this, we were able to implement authorization with a JWT token, which allows the user to log in and then stores an authentication token in their browser for a period of time which would grant them access to all the pages which require authentication. This was implemented with API endpoints, e.g., POST /auth/register and POST /auth/login.

This functionality required a database connection to store and get user data like passwords and usernames. For very initial development, we used PostgreSQL on our local machines, but with the help of the cloud team, we have integrated to local DynamoDB. Using a local instance of the same database which we will eventually integrate with in development will make integration much easier.

Our anticipated database schema is as follows:

- Users Table (id, role, name, username, passwords, contacts...)
- Logs Table (user_id, timestamp, type, value)

- Alerts Table (user_id, timestamp, description, status)
- Message Table (sender_id, reciever_id, conversation_id, timestamp, text, subject_line)

Now that the local database connection has been set up, it will be much more straightforward to test and develop our messaging, data visualization, alerts, and search functionalities with fake data before integration. These are the next steps on the backend.

Since there was some of a learning curve for these technologies, ChatGPT was used as a helpful tool during development for information on React, Typescript, and the rest of our tech stack. Reference [5.2.h] is a ChatGPT conversation from one of our team members, which demonstrates how it was used to make our programming more efficient and streamlined while still supporting our project's unique use case.

5.2.5 Final Implementation

In Phase 3, we successfully completed the majority of our planned development for both the mobile app and the website. Our efforts focused on building fully functioning frontends for patients and providers, integrating backend APIs, implementing Bluetooth functionality, and connecting both frontends to a shared cloud-hosted backend infrastructure. Although a few features, such as advanced data visualizations, were deferred due to time constraints, the core system was completed and demonstrates strong potential. Key functionalities—including user registration, login, patient search, mood tracking, and data retrieval—were fully implemented and are operational.

5.2.5.1 Task Outline

In Phase 3, we aimed to implement a range of functional and non-functional features for account management, patient search, messaging, alert handling, and patient profiles. Below is a summary of the major tasks and their status:

For Account Management, we implemented key features including account creation and login functionality. Timeout handling on periods of inactivity, email confirmation for verification, password reset, and account information updates remains flagged as a non-functional feature for future enhancement.

In the Patient Search module for providers, we successfully enabled loading and viewing patient data, as well as searching by patient name. Searching by date of birth and location seen was also implemented, while searching by date seen and more complex filtering (such as nearby, not exact dates) marked for future refinement.

For Chat and Messaging, our functionality was frontend-only, as messaging APIs were designated as taking too much time, and also being a solved problem. These messaging screens can be seen in section 5.2.5.2, Mobile App Frontend Progress. In this frontend-only functionality, we set up one-on-one messaging with the provider, including sending anomaly reports. Other than the backend API to enable real messaging, additional features such as searching messages by text or date, messaging multiple people, and sending files are planned for future development.

In the Alert Tasks (user side), we enabled anomaly-based pop-up alerts and allowed patients to send alerts to providers. Sending alerts to emergency contacts was partially implemented and remains a future goal.

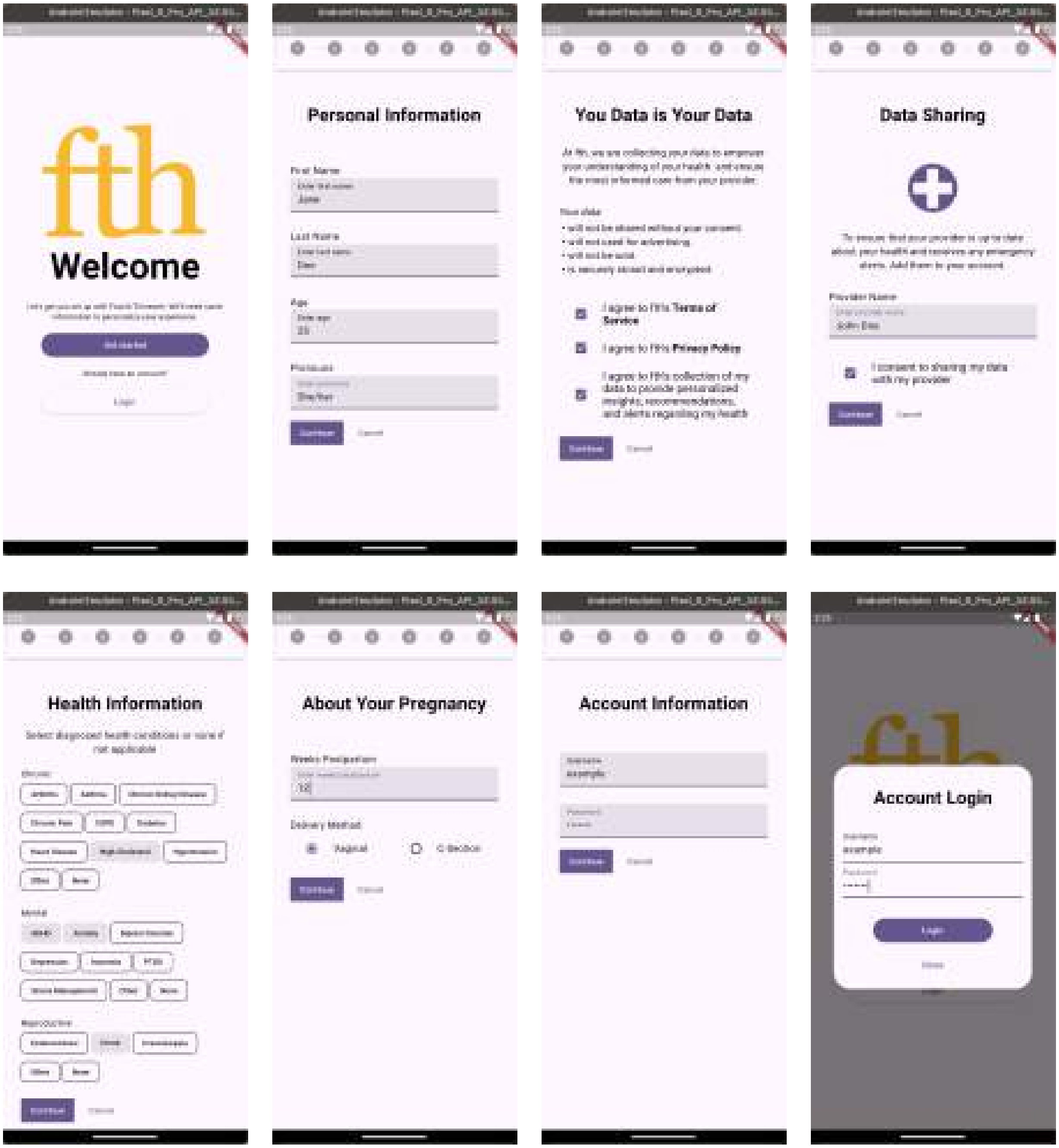
On the Provider Side Alerts, we implemented alert listing functionality and viewing detailed reports triggered by alerts.

Lastly, for the Patient Profile (provider side), we successfully implemented loading a patient profile and reports, but left the data visualization and messaging features for further development. Providers can also type and copy notes on the patient from this page, but saving those notes was not implemented.

Overall, we prioritized the core functional tasks necessary to achieve a robust minimum viable product, while leaving certain "middle-ish" or non-functional robustness goals for future development as time permitted.

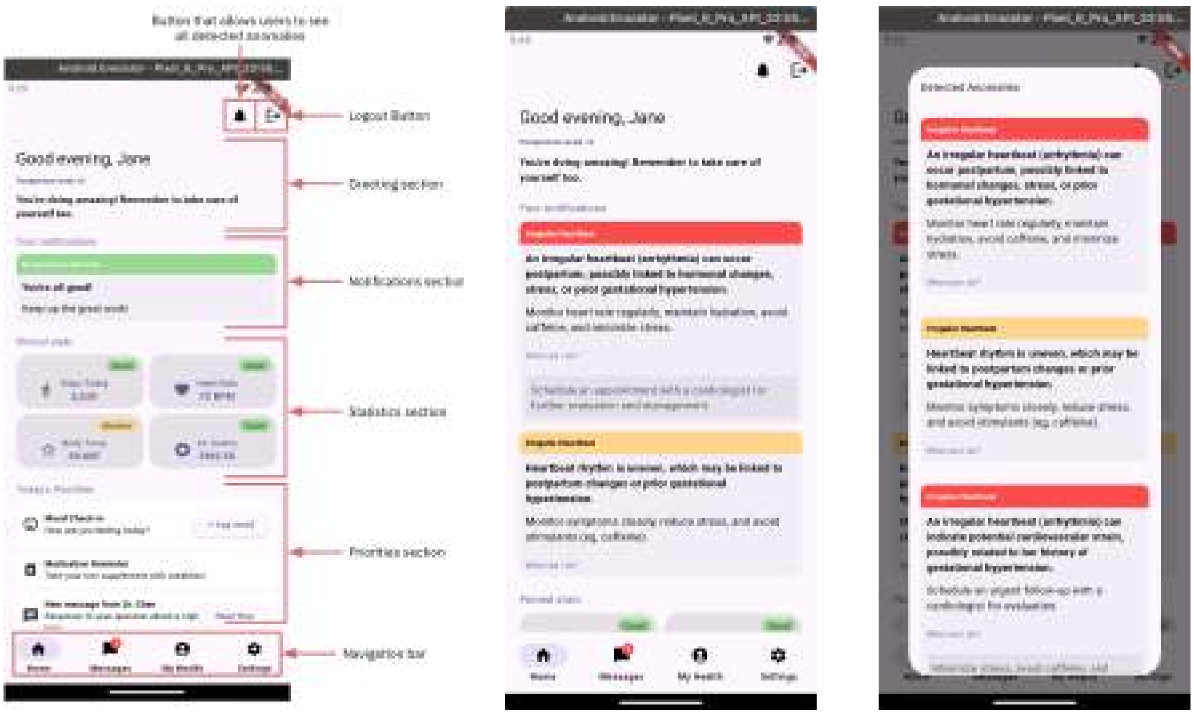
5.2.5.2 Mobile App Frontend Progress

- Registration/Login (Alice): Implemented basic login and registration screens for patients and providers, allowing authenticated access to app features. The registration screens allowed for patients to input vital information that is used in the ML models when detecting health trends and anomalies. The registration screens also provide full transparency of the data it will collect and what it is used for.



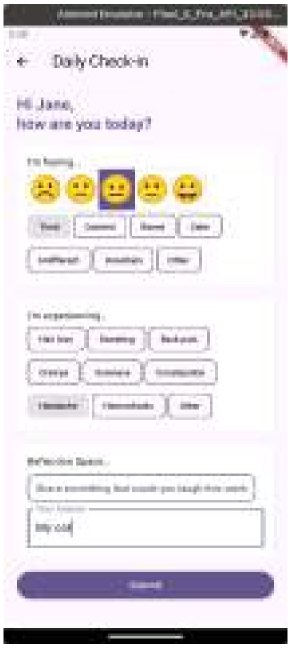
Registration Screens

- Landing Screen (Alice): The landing screen is composed of 4 main sections: a greeting section that welcomes and uplifts the user, a notifications section that contains the two most recently detected anomalies, a stats section that shows the patients current health measurements, and a priorities section that has the most important to-dos.



Landin Screens

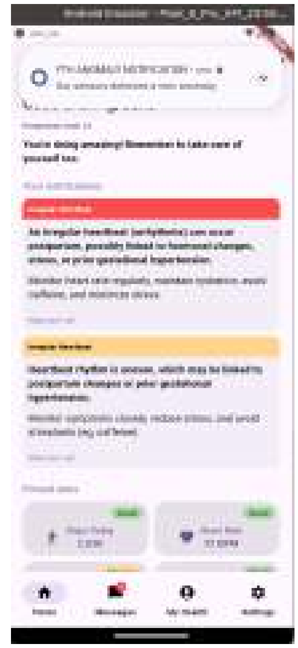
- Mood Screen (Yiyun): Built the mood reporting feature, where patients can track and log their emotional states. The mood reporting is meant to be simple and open-ended, aiming to relieve the stress of having to do extensive self-evaluations every day.
- Bluetooth Screen + Functionality (Alice): Developed a settings page where users can configure Bluetooth connections. Basic pairing functionality implemented (details in Section 5.2.5.4).
- Notifications (Alice): Added basic notification triggers for user alerts (details in Section 5.2.5.4).
- Profile (Alice): Implemented a user profile screen displaying personal information.



Mood Screens

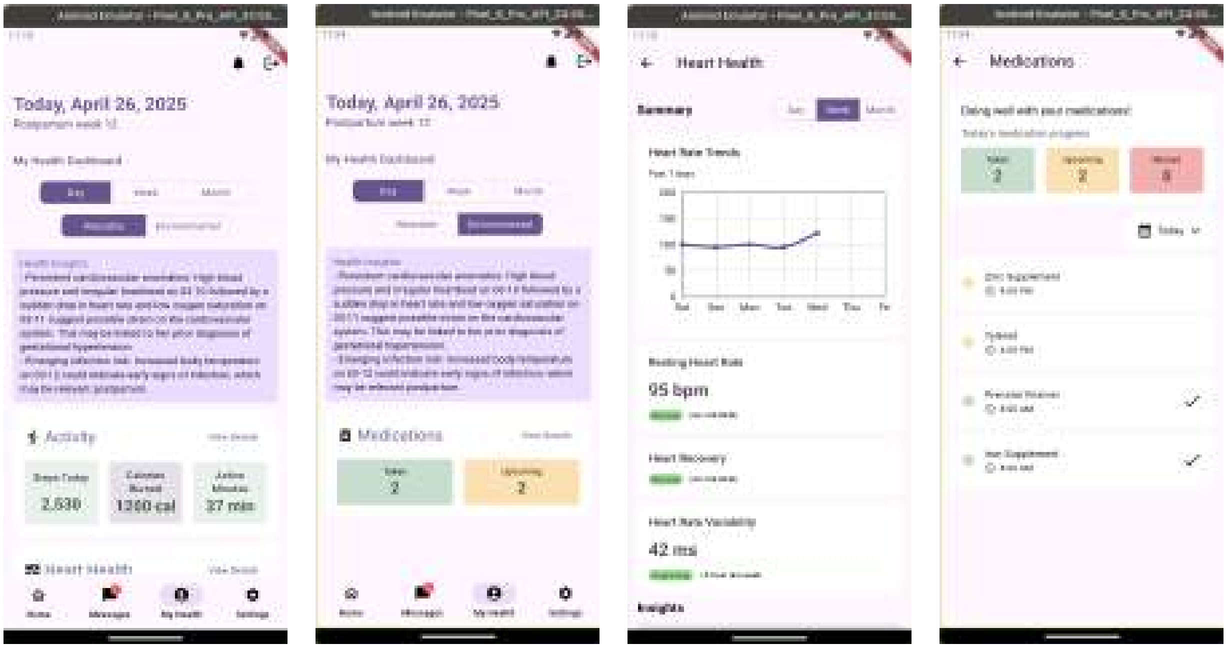
Setting Screens

Bluetooth Screens



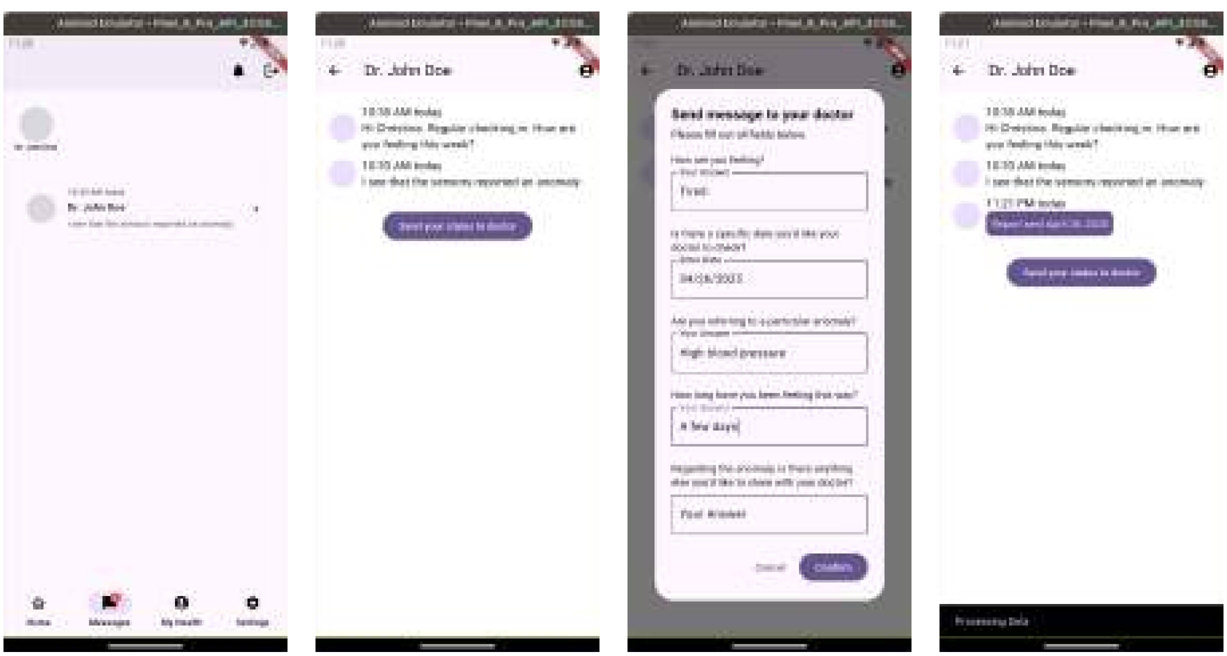
Notification Example

- My Health and Subscreens (Yiyun): Built the "My Health" hub showing health summaries, wearable data, and weekly reports. Health summaries and insights are shown through bullet points, analyzing health trends and providing information regarding possible causes or related diagnoses. Wearable data is displayed through many different formats: activity includes steps walked, calories burned, and active minutes; heart data includes current and trends in heart rate, with a line graph for visualization; body temperature includes present and past body temperature data, also displaying a line graph for visualization. The environmental data consists of medication tracking, displaying the number of medications taken, upcoming, and missed, along with details such as the name of the medication and the time designated for it to be taken.



My Health Screens

- Messages (Yiyun): Created a messaging interface for communication between patients and providers. The messaging is form-based, providing prompts such as “How are you feeling?” and “Are you referring to a particular anomaly?” to make it easier for patients to describe their symptoms, and for providers to more easily understand how their patients are feeling.



Messaging Screens

5.2.5.4 App Integrations

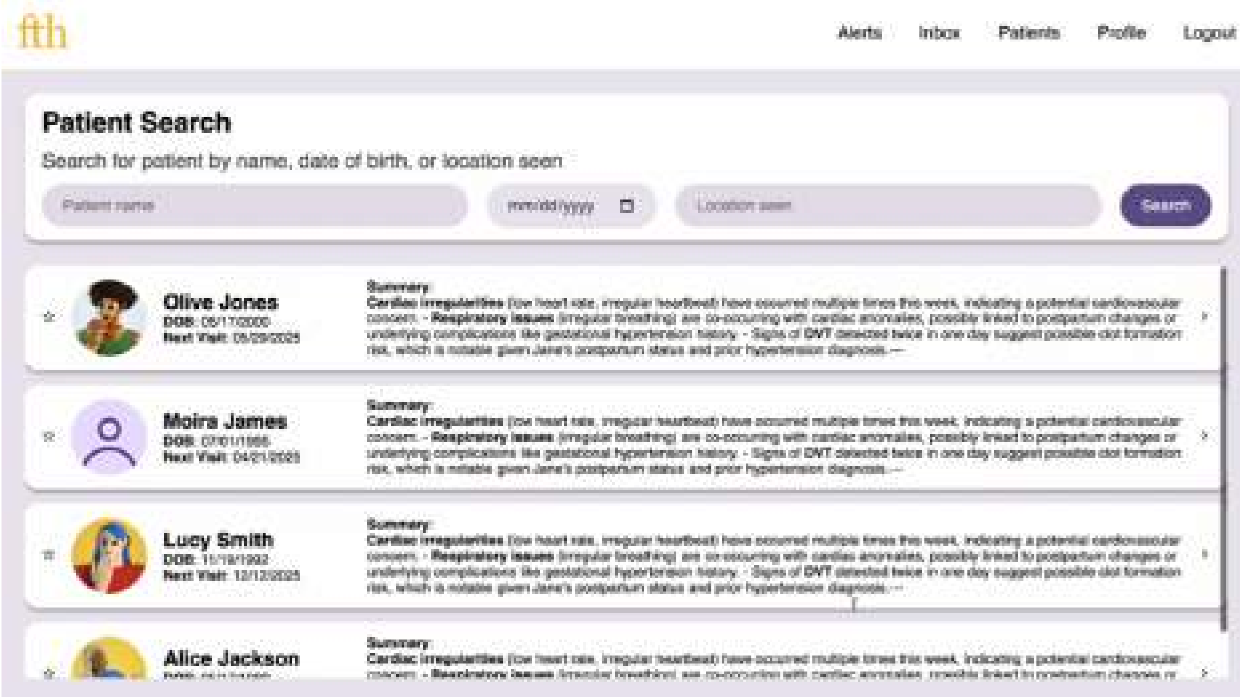
- App API Calls to Backend (Alice): Connected app frontend components to backend API endpoints using HTTP requests, allowing dynamic fetching and updating of user and patient data.

API Request Example

- Bluetooth Screen + Functionality (Alice): Integrated basic Bluetooth pairing capability, enabling device connection and reading of data from the wearable device. The bluetooth functionality was implemented using the flutter_blue_plus plugin and starter code.
- Notifications (Alice): Initial notification logic built for app alerts tied to backend events like new messages or anomalies using the flutter_local_notifications package.

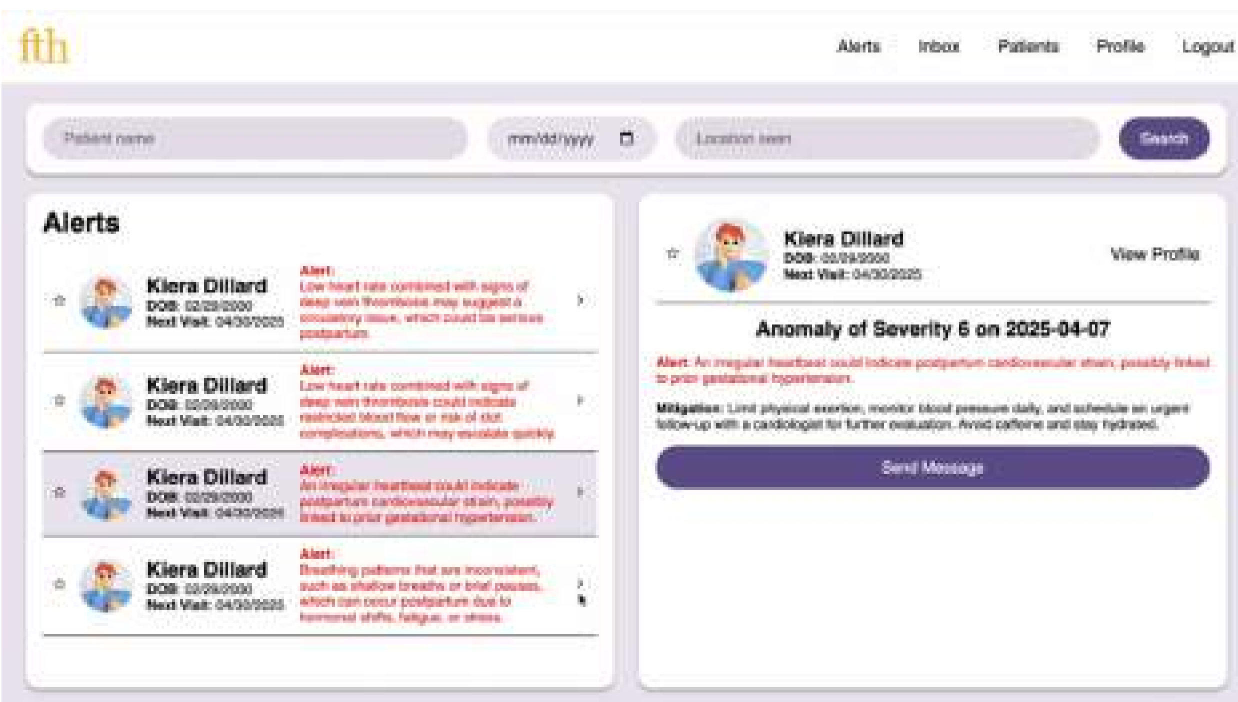
5.2.5.5 Website Frontend Progress

- Patient Search (Fiona): A functional patient search feature that allows providers to quickly find patients by name. The search results display essential information such as the patient’s demographic details, upcoming visits, and recent activities. From this page, the provider can click on a patient to navigate to the patient profile.



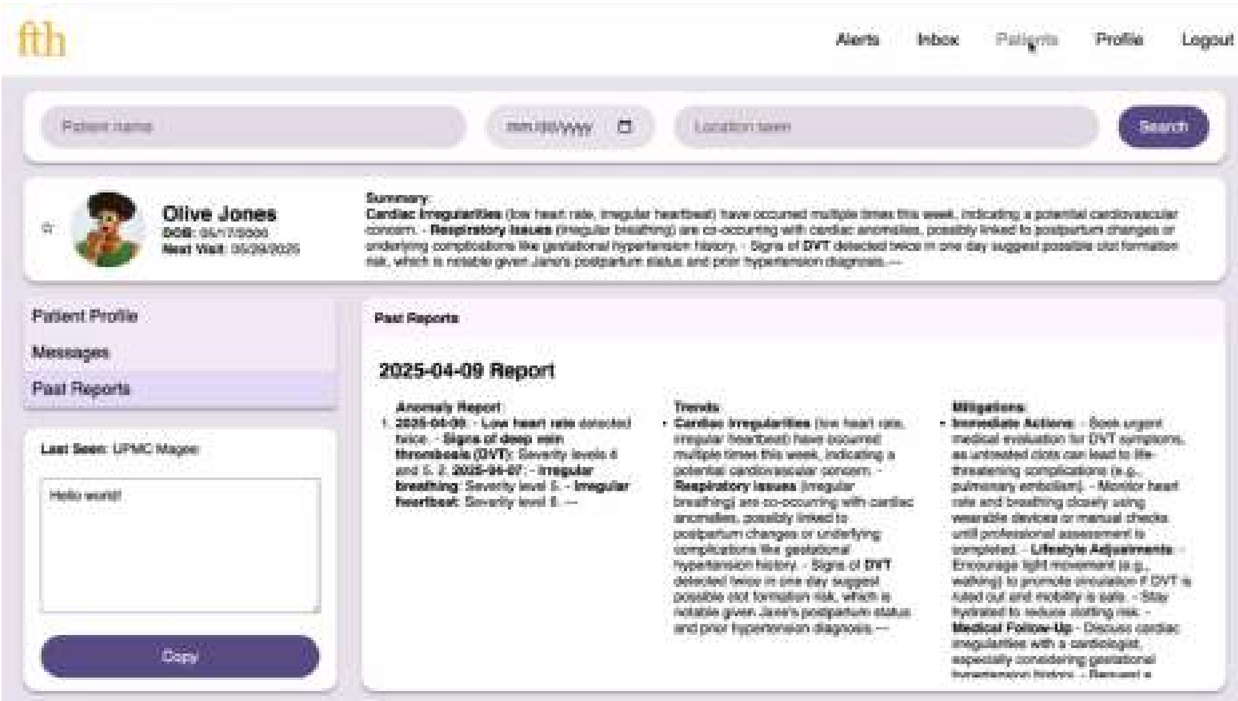
Patient Search Screen

- Alert Page (Fiona): An alert box component where providers can view anomalies loaded to the database by the intelligence team's software. Clicking on an alert will display mitigation techniques, date, and severity.



Alert Page Screen

- Past Report (Fiona): Providers can select and review previous weekly health reports, which logged anomalies, trends, and mitigations throughout the week.



Past Report Screen

5.2.5.6 Backend

- User Login and Registration Authentication (Yuchen):
  We implemented secure user login and registration endpoints for both patients and providers. Authentication is handled with password hashing and token-based sessions, ensuring secure access control. This system allows users to register accounts, verify their identity, and log in to access their personalized data securely.
- Database Interaction with the Cloud (Fiona):
  We designed and built the API routes to fetch patient data, weekly reports, and alerts from the cloud-hosted database. These APIs allow the frontends (both app and website) to request and display live patient data in real-time. Core functionalities include retrieving individual patient profiles, pulling weekly health reports and fetching anomaly reports. We initially tested these database interactions with local instances before transitioning successfully to the production cloud environment.

```
const router = Router();

router.post('/register_provider', registerProvider);
router.post('/login_provider', loginProvider);
router.post('/register_patient', registerPatient);
router.post('/login_patient', loginPatient);

export default router;
```

```
const router = Router();

router.post('/search', authenticateJWT, searchPatient);
router.post('/recent_report', authenticateJWT, findMostRecentReport);
router.post('/reports', authenticateJWT, findReports);
router.post('/get', authenticateJWT, getPatient);
router.post('/anomalies', authenticateJWT, getAnomalies);
router.post('/medication_adherance', authenticateJWT, getMedicationAdherance);
router.post('/push_sensor_data', authenticateJWT, pushSensorData);
router.post('/get_sensor_data', authenticateJWT, getSensorData);

export default router;
```

Existing Routes

5.3 Cloud/Data/Communication Systems

5.3.1 Functional Requirements

Our cloud, data, and communication team is focused on building a scalable, secure, and resilient platform to support dynamic workloads and seamless collaboration. By integrating different teams into a unified cloud environment, we ensure flexibility and scalability for evolving business needs. Security and compliance are key priorities, with robust access controls and protections against network threats. Additionally, we emphasize reliability and resilience by implementing data backup strategies and efficient recovery mechanisms to maintain system integrity even in the event of failures. Together, these efforts create a robust cloud infrastructure that enhances performance, security, and operational continuity.

5.3.2 System Architecture

Our Cloud Data & Communication System is organized into three core subsystems—Compute, Storage & Metadata, and Notifications—extending seamlessly from edge-IoT devices up through mobile and web clients (as shown in Figure 5.3.3.A). At the heart of the Compute layer sits an EC2 instance, running our frontend and backend application. It handles incoming requests from both the User App (which aggregates data from wearable sensors over Bluetooth) and the Provider App (a web-based frontend/backend), synchronizes with object storage, and delivers real-time responses with high availability and horizontal scalability.

All raw sensor streams, whether thermal-camera frames or environmental readings relayed from the Base Station, are persisted into Amazon S3. DynamoDB maintains our structured metadata, including user profiles, inference outputs, and system configuration. An automated S3-sync process keeps each Agent's local cache up to date, minimizing latency for time-sensitive operations while centralizing long-term archives in S3.

Event-driven workflows are handled by our Notifications subsystem. Changes in DynamoDB streams or new S3 uploads emit signals that invoke AWS Lambda functions, which generate anomaly alerts and data-update notifications. These alerts travel back through the Compute layer to the User App and can also surface through the Provider App for administrative visibility.

On the edge, the Environment Hardware team's sensors forward measurements to a local Base Station, which bundles and pushes them to our cloud ingestion API. Simultaneously, the Wearable Hardware team's devices communicate bi-directionally with the User App, enabling both data capture and alert delivery on the device. Downstream, the Intelligence team pulls raw data and weekly reports from S3 and DynamoDB to train and validate machine-learning models, then deploys updated models into the EC2-based Agents for continuous, real-time inference—completing a reliable, end-to-end feedback loop.

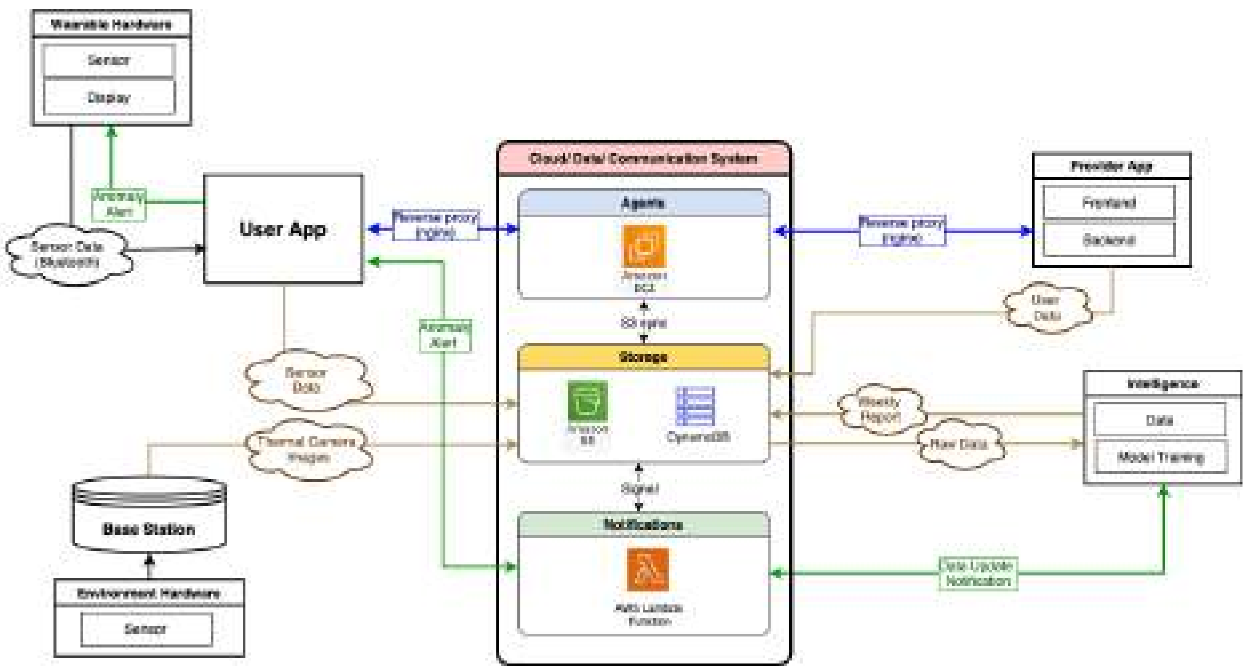


Figure *5.3.3.A: System Architecture*

5.3.3 Services

5.3.3.1 EC2

For our project in Phase 3, Amazon EC2 is used to host the backend services for both the Provider software and the User software, offering a centralized, scalable, and reliable environment for development, testing, and production deployment. To efficiently manage incoming requests and distribute load across different backend components, we configured a reverse proxy using Nginx(as shown in Fig. 5.3.3.B) on the EC2 instance. This setup allows

Nginx to perform load balancing, improving service responsiveness, availability, and fault tolerance.

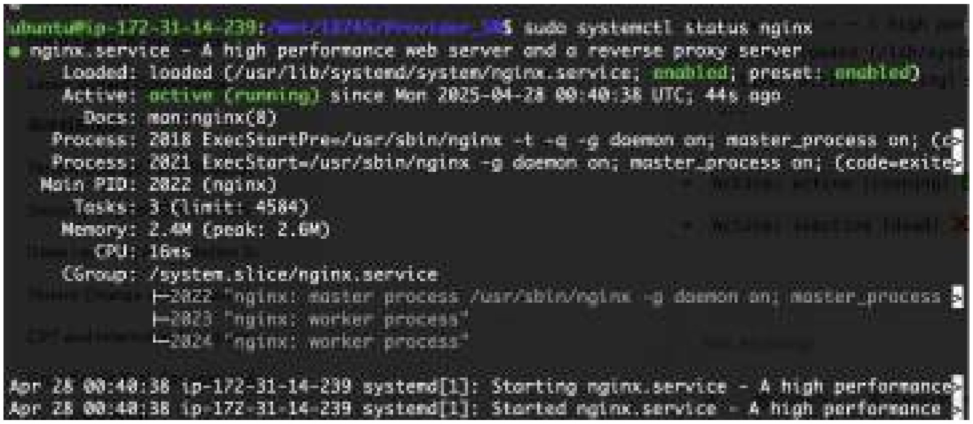


Figure 5.3.3.B: Nginx status on EC2

To make our services accessible over the internet, we carefully edited the EC2 instance's security group settings through the AWS Management Console, allowing inbound traffic on necessary ports at port 3000 for frontend and 5001 for backend(as shown in Fig. 5.3.3.C). These configurations ensure that both internal teams and external users can reliably access the software.

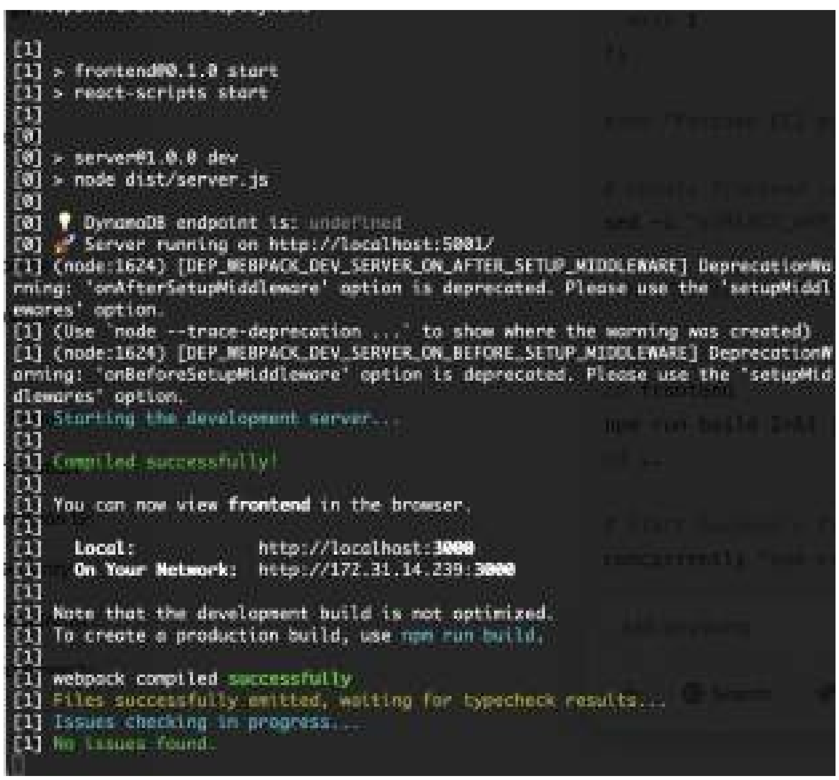


Figure 5.3.3.C: Service port Running

In addition, we developed an automated cloud deployment solution using a Bash script. This script handles the entire deployment process(as shown in Fig. 5.3.3.D), including pulling the latest software version from the repository, installing dependencies, restarting services, and verifying deployment status. By automating these steps, the script significantly reduces manual workload, minimizes human error, and speeds up the update cycle for both Provider and User backends. This automation ensures consistent and efficient software rollout, supporting agile development practices across the teams.

Figure 5.3.3.D: Bash Script for Software Deployment

5.3.3.2 Database

As for a final deliverable, we completed the setup of the database system and kept a continuous maintenance of the database to ensure smooth access for each subteams. This includes setting up automated backup processes, fine-tuning the indexing keys of each table, and resolving access problems from other teams in real-time through online chats and in class collaboration.

The number of teams supported by us has increased from a single team to all 4 teams that requested the use of a database during the prototyping phase. As such, the total number of tables that are hosted on our cloud system has greatly increased in its scale. The following screenshot captures some of the tables that exist on our system and are actively used by the teams.

Figure 5.3.3.E: DynamoDB tables

The original python scripts from phase 2 includes an upload function, a data-accessing function, and a delete function, with easy-to-use function design. Users can upload a piece of data by putting the data in a dictionary format and pass it to the function, or they can get a piece of data based on its key, or delete one based on the key as well. During phase 3, we have reassessed the needs from teams and decided to add more functions to the script. For example, we have added a function for teams to retrieve data based on some attributes from the table in order to expand the old get-data-with-key functionality. We also included a method for manipulating images from and to our database, which elevated our database to more than just interacting with plain text.

```
def get_item(dynamodb, table_name, key_data):
    table = dynamodb.Table(table_name)

    try:
        response = table.get_item(Key=key_data)
        print("GetItem response:", response.get('Item'))
    except Exception as e:
        print("Error getting data:", e)
```

```
def put_item (dynamodb, table_name, item_data):
    table = dynamodb.Table(table_name)

    try:
        response = table.put_item(Item=item_data)
        print("PutItem response:", response)
    except Exception as e:
        print("Error inserting data:", e)
```

```
def delete_item(dynamodb, table_name, key_data):
    table = dynamodb.Table(table_name)

    try:
        table.delete_item(Key=key_data)
    except Exception as e:
        print("Error deleting data:", e)
```

Figure 5.3.3.F: Some Parts of the Python scripts to manipulate data in the database

By maintaining a strong operating focus, we ensured that the database consistently supported all subteams' requirements, allowing the overall project to move forward without bottlenecks.

5.3.3.3 Lambda Function

For the notification, we use AWS Lambda function for sending text messages and emails to the users. Lambda function is a serverless computer service. When an event arrives, such as S3 upload, API Gateway call, DynamoDB table insert, lambda executes the handler function and returns the response.

Figure 5.3.3.G: Lambda function

We decided to use DynamoDB as the trigger. When an anomaly occurs, a record is sent to the anomaly_report table in DynamoDB. Each time a new item is added to the table, it indicates that we need to notify the user of the anomaly. The notification must include three pieces of information: the anomaly report ID, the anomaly type and the anomaly severity level. Upon receiving the alert, the user will either call for help immediately or open the app to view more details like recommended action to cope with the anomaly and root cause of the anomaly.

5.3.3.4 S3 Bucket

Figure 5.3.3.G shows the high-level workflow of our S3 service. In our pipeline, the primary purpose of the S3 service is to act as a highly durable, scalable repository for all large binary assets, particularly images that would otherwise strain our transactional datastore. The application begins by authenticating against AWS and connecting to the designated S3 bucket, then uploads each image into a chosen prefix or “folder” namespace. This organization not only keeps assets discoverable, but also leverages S3’s virtually unlimited capacity, built-in redundancy across multiple Availability Zones, and lifecycle policies for efficient archival and cost management.

After an upload completes, the workflow transitions to the metadata tier which is DynamoDB in our case, where a lightweight index record is created for each object. Each entry uses a unique key, optional descriptive fields, and the S3 object's URL or a pre-signed link. By splitting heavy-weight blob storage (in S3) from fast, fine-grained querying (in DynamoDB), we ensure rapid lookups, precise access controls, and a clean separation of concerns between media handling and application data management.

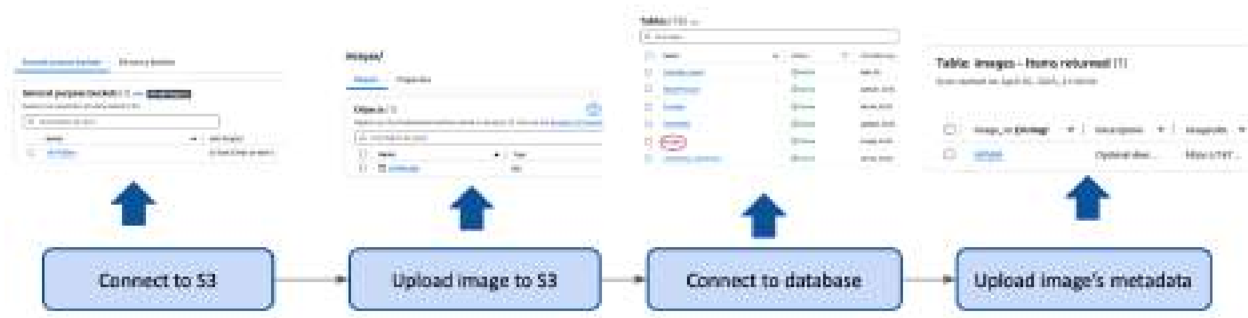


Figure 5.3.3.H: High-level pipeline of S3 Services

5.4 Intelligence

The intelligence team was tasked with doing the anomaly detection and subsequent inferencing for mitigation techniques and harmful trends related to these detected anomalies.

Our detailed functional requirements are below:

- Detect anomalies given wearable and environmental sensor data
- Allow the ML model to be retrained based on new patient information
- Create human readable reports and anomaly notifications
    - Include a severity score for each anomaly notification

In order to achieve these requirements, we've created a pipeline with three major parts:

1. Data pre- and post-processing: The sensor data we receive will likely not be adequate as direct inputs to our machine learning model. So, we will need to preprocess the data before feeding it through our model. Similarly, the output of the ML model might not be human-readable. So, we need some postprocessing.
2. ML model: In order to detect anomalous data points from sensor outputs, we decided to implement an aggregate machine learning model with two major parts — classification and thresholding. The model will output the data points that were most anomalous and will be sent to post-processing.

3. LLM: In order to use the output of the ML model effectively, we will feed in its processed output to OpenAI's GPT-4.0 LLM. The LLM will be able to take in user information, past anomalies, current anomalies output by the ML model, and other contextual information and output a report of concerning trends, mitigation techniques, and a severity score, all in human readable language. This will be sent to the front-end teams to be displayed on the application.

We discuss our reasoning for these design choices and a more thorough exploration of how we will achieve them below.

5.4.1 Data Processing

5.4.1.1 Data Formatting

Our data will be collected from environmental and wearable sensors, and interviews, which will need conversion into data formats that will allow data preprocessing prior to making proper machine learning models.

Raw Data Format and Parameters

For this current phase in the data preprocessing, while we have not received our sensor data, we have generated the following sample data to test our system and process. Currently, we have generated random samples to perform the data preprocessing pipeline. The above tables show the proper ranges and units for each data generated, given their generation times and frequencies.

1. Oximeter/Heart Rate Data

Parameter	Description	Normal range / Normal wave
Oximeter	Blood oxygen saturation level (SpO_2)	95% - 100%
Heart Rate (BPM)	Heart rate in beats per minute	70 - 80 BPM Resting HR (Adults): 60–100 BPM
Temperature	Body temperature	Mean = 98.6°F SD = 0.3
Blood Pressure Waveform	Simulated as a sine wave representing pulsatile BP pattern	Baseline = 100 mmHg Amplitude = 30 mmHg → Range: [70, 130]
Time (Unit: ms)	Time stamp	—

(1) SpO2 (%): In a healthy individual, normal SpO blood oxygen saturation level is around 95-100%. The values are constrained to the range within this healthy level.
(2) HR (BPM): Normal resting heart rate is around 60-100 BPM. The values are set between 70-80 BPM to simulate a resting but active state.
(3) Temperature: Body temperature is a measurement simulated in degrees fahrenheit. The mean body temperature is at 98.6 degrees fahrenheit, with a standard deviation of 0.3.
(4) Blood pressure waveform: Blood pressure is measured in waveforms, which has been simulated as a sine wave to represent the pulsatile blood pressure patterns. The baseline for this simulated data is 100m mmHg, with an amplitude of around 30 mm
(5) Time (ms): Sampled at 100Hz (10ms interval), which is a common rate for real-time heart rate monitoring.

Below are the examples of the four generated datasets:

1. Oximeter Data & Heart rate (BPM) Data

	Time (ms)	SpO2 (%)
0	2024-01-01 00:00:00.000	98
1	2024-01-01 00:00:00.010	99
2	2024-01-01 00:00:00.020	97
3	2024-01-01 00:00:00.030	99
4	2024-01-01 00:00:00.040	99
5	2024-01-01 00:00:00.050	96
6	2024-01-01 00:00:00.060	97
7	2024-01-01 00:00:00.070	97
8	2024-01-01 00:00:00.080	97
9	2024-01-01 00:00:00.090	99
10	2024-01-01 00:00:00.100	98
11	2024-01-01 00:00:00.110	97
12	2024-01-01 00:00:00.120	99
13	2024-01-01 00:00:00.130	96
14	2024-01-01 00:00:00.140	98
15	2024-01-01 00:00:00.150	96
16	2024-01-01 00:00:00.160	98
17	2024-01-01 00:00:00.170	99
18	2024-01-01 00:00:00.180	95
19	2024-01-01 00:00:00.190	98

	Time (ms)	HR (BPM)
0	2024-01-01 00:00:00.000	72
1	2024-01-01 00:00:00.010	78
2	2024-01-01 00:00:00.020	73
3	2024-01-01 00:00:00.030	72
4	2024-01-01 00:00:00.040	71
5	2024-01-01 00:00:00.050	79
6	2024-01-01 00:00:00.060	75
7	2024-01-01 00:00:00.070	70
8	2024-01-01 00:00:00.080	78
9	2024-01-01 00:00:00.090	73
10	2024-01-01 00:00:00.100	79
11	2024-01-01 00:00:00.110	72
12	2024-01-01 00:00:00.120	77
13	2024-01-01 00:00:00.130	79
14	2024-01-01 00:00:00.140	79
15	2024-01-01 00:00:00.150	71
16	2024-01-01 00:00:00.160	73
17	2024-01-01 00:00:00.170	75
18	2024-01-01 00:00:00.180	73
19	2024-01-01 00:00:00.190	78

2. Temperature & Blood Pressure

	timestamp	sensor_type	value	label
0	2025-04-16 14:00:00.000	bp_sensor_500Hz	100.0	0
1	2025-04-16 14:00:00.002	bp_sensor_500Hz	103.2	0
2	2025-04-16 14:00:00.004	bp_sensor_500Hz	106.3	0
3	2025-04-16 14:00:00.006	bp_sensor_500Hz	109.4	0
4	2025-04-16 14:00:00.008	bp_sensor_500Hz	112.4	0
5	2025-04-16 14:00:00.010	bp_sensor_500Hz	115.2	0
6	2025-04-16 14:00:00.012	bp_sensor_500Hz	117.9	0
7	2025-04-16 14:00:00.014	bp_sensor_500Hz	120.3	0
8	2025-04-16 14:00:00.016	bp_sensor_500Hz	122.6	0
9	2025-04-16 14:00:00.018	bp_sensor_500Hz	124.5	0
10	2025-04-16 14:00:00.020	bp_sensor_500Hz	126.2	0
11	2025-04-16 14:00:00.022	bp_sensor_500Hz	127.6	0
12	2025-04-16 14:00:00.024	bp_sensor_500Hz	128.7	0
13	2025-04-16 14:00:00.026	bp_sensor_500Hz	129.5	0
14	2025-04-16 14:00:00.028	bp_sensor_500Hz	129.9	0
15	2025-04-16 14:00:00.030	bp_sensor_500Hz	130.0	0
16	2025-04-16 14:00:00.032	bp_sensor_500Hz	129.7	0
17	2025-04-16 14:00:00.034	bp_sensor_500Hz	129.1	0
18	2025-04-16 14:00:00.036	bp_sensor_500Hz	128.2	0
19	2025-04-16 14:00:00.038	bp_sensor_500Hz	127.0	0

	timestamp	sensor_type	value	label
0	2025-04-16 14:00:00	temp_sensor_1Hz	98.32	0
1	2025-04-16 14:00:01	temp_sensor_1Hz	98.79	0
2	2025-04-16 14:00:02	temp_sensor_1Hz	98.97	0
3	2025-04-16 14:00:03	temp_sensor_1Hz	98.68	0
4	2025-04-16 14:00:04	temp_sensor_1Hz	98.52	0
5	2025-04-16 14:00:05	temp_sensor_1Hz	99.17	0
6	2025-04-16 14:00:06	temp_sensor_1Hz	98.81	0
7	2025-04-16 14:00:07	temp_sensor_1Hz	98.74	0
8	2025-04-16 14:00:08	temp_sensor_1Hz	98.54	0
9	2025-04-16 14:00:09	temp_sensor_1Hz	97.98	0
10	2025-04-16 14:00:10	temp_sensor_1Hz	98.64	0
11	2025-04-16 14:00:11	temp_sensor_1Hz	98.11	0
12	2025-04-16 14:00:12	temp_sensor_1Hz	98.58	0
13	2025-04-16 14:00:13	temp_sensor_1Hz	98.35	0
14	2025-04-16 14:00:14	temp_sensor_1Hz	98.17	0
15	2025-04-16 14:00:15	temp_sensor_1Hz	99.03	0
16	2025-04-16 14:00:16	temp_sensor_1Hz	98.60	0
17	2025-04-16 14:00:17	temp_sensor_1Hz	98.64	0
18	2025-04-16 14:00:18	temp_sensor_1Hz	98.87	0
19	2025-04-16 14:00:19	temp_sensor_1Hz	98.21	0

Processed Data Format

After preprocessing, the data undergoes multiple transformations to enhance reliability and usability in our ML model, especially has the features of integrated time stamps and features in a single large dataset. Below is an example of processed sensor data that can be used for ML models:

	Time (ms)	SpO2 (%)	HR (BPM)	Body Temp	Blood Pressure
1	0	0.7062691860203876	1.5771891519049583	0.14498415750423999	6.02497850652678
2	200	1.412842591145758	2.1047141252209265	-0.18047697224655046	7.354109662109304
3	400	-0.00030421910498291905	2.6264221646961103	-0.3016737946225174	8.649345007061816
4	600	1.412842591145758	3.132070689232334	-0.22132983372174542	9.907172065306916
5	800	1.412842591145758	3.6119333750419766	0.7931828929071946	11.124268067366062
6	1000	-0.7068776242303535	4.057137724785225	2.0010658305178075	12.297518319517874
7	1200	-0.00030421910498291905	4.459961100104997	1.0274059653638592	13.42403330788839
8	1400	-0.00030421910498291905	4.814075196529165	-0.7306288401098634	14.50116438600374
9	1600	-0.00030421910498291905	5.114731175751491	-0.10558006954255689	15.526517904665043
10	1800	1.412842591145758	5.358880744085351	0.7414359350389168	16.497967654946788
11	2000	0.7062691860203876	5.545230900291942	0.0782578170948384	17.413665508563273
12	2200	-0.00030421910498291905	5.674231234887748	-0.8559109486331941	18.272050154664573
13	2400	1.412842591145758	5.747993855739922	-0.4078912344574961	19.071853848167716
14	2600	-0.7068776242303535	5.770148104556593	1.1240910708546956	19.812107101839498
15	2800	0.7062691860203876	5.745634843533245	1.1587733600346892	20.492141272351525
16	3000	-0.7068776242303535	5.6804484592815605	0.430954187829047	21.11158900924558
17	3200	0.7062691860203876	5.581339683293392	-1.5926242172321223	21.67038255498565
18	3400	1.412842591145758	5.45549760473156	-0.9471490059273199	22.16874990383084
19	3600	-1.4134510293557239	5.310231235445873	-0.3615913247857936	22.607208846929137
20	3800	0.7062691860203876	5.152667644563121	1.02332067921636	22.986558950562297

5.4.1.2 Abnormal Values

Potential anomalies include out-of-range readings and extreme fluctuations, which require further processing, attention or identification.

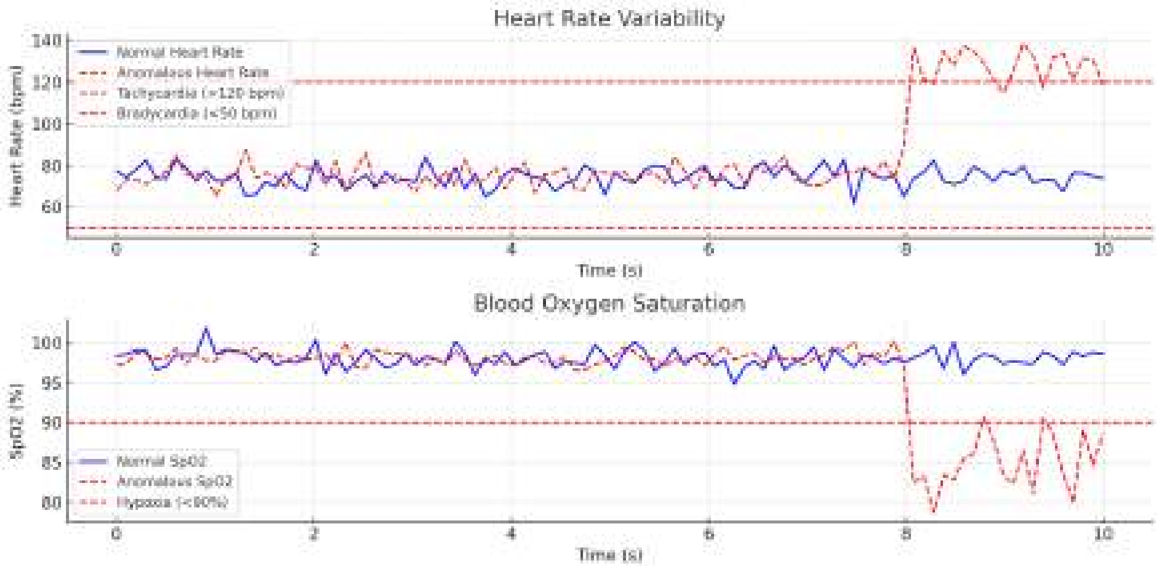
Out-of-range values include extreme high values and extreme low values, which are displayed below. In this case, measurements that are below detection threshold are not taken into account.

Here are abnormal values of the concerned factors:

Factor	**Abnormal range**
Body Temperature	< 95°F: Hypothermia > 99.5°F: Fever > 102.2°F: High Fever > 106.7°F: Hyperpyrexia
Blood Oxygen (SpO_2)	90–94%: Mild hypoxemia 85–89%: Moderate hypoxemia < 85%: Severe hypoxemia

Heart Rate (BPM)	< 60 BPM: Bradycardia > 100 BPM: Tachycardia > 120 BPM: Severe tachycardia
Blood Pressure	120–129/<80: Elevated 130–139/80–89: Stage 1 HTN > 140/90: Stage 2 HTN >180/120: Crisis
Humidity	< 30%: Too Dry (irritation) > 60%: Too Humid (mold, discomfort)
CO_2 Concentration	1,000–2,000 ppm: Drowsiness 2,000–5,000 ppm: Headaches > 5,000 ppm: Hazardous
PM2.5	16–55 µg/m³: Moderate > 55 µg/m³: Unhealthy
PM10	46–150 µg/m³: Moderate > 150 µg/m³: Unhealthy
Environmental Temperature	< 64°F: Cold stress > 78–86°F: Heat stress (especially with high humidity)

Specifically, the extreme fluctuations in heart rhythm (ECG) shows the features of arrhythmias and missing peaks, away from the regular PQRST waveform. And the abnormal fluctuations in blood oxygen saturation are suggested by the sudden drop of oxygen saturation of arterial blood.



5.4.1.3 Data Preprocessing Workflow

Generated datasets

Currently, in our preprocessing, we generated sample datasets with the corresponding hertz for 1 hour to validate that our data preprocessing workflow is functioning correctly. This generated dataset allows us to test our system efficiently while conserving resources during the sample usage phase.

To ensure our data considers situational patient outcomes, we have distributed the testing data into 80% normal values and 20% anomalies. We designed this distribution to mimic real-world sensor readings by including both normal and anomalous behavior.

1. 80% Regular Fluctuations
This portion represents normal, everyday sensor readings that fall within expected ranges. It serves as our baseline to confirm that our data preprocessing workflow correctly handles routine data. These “regular fluctuations” follow the ranges as shown in the tables above to follow ranges of regular activity.

2. 20% of our generated data are anomalies to mimic possible anomalies in our data collection. This distribution is intentional. By including both normal and anomalous data in these proportions, we can test our data preprocessing process to ensure it accurately identifies and handles irregularities, ultimately enhancing the reliability of our system while efficiently using resources during testing.

Data Preprocessing Workflow

Our data workflow for preprocessing is designed as an end-to-end pipeline that ensures raw sensor data is transformed into a clean, synchronized, and standardized format ready for machine learning tasks.

1. Loading Data
We begin by reading raw sensor files (e.g., CSV), including heart rate, blood oxygen, blood pressure and body temperature. These files contain timestamped readings and other metadata, serving as the basis for preprocessing.

2. Data Cleaning and Timestamp Parsing
The first thing we do is clean the data and make sure the timestamps are in a consistent format. That means we drop any missing or incomplete rows, and we convert all the time columns to a standard datetime format.

3. Signal Filtering to Reduce Noise
Next, we apply filtering to reduce noise in the signals. Heart rate and blood pressure data are especially prone to motion artifacts and high-frequency noise, so we use a bandpass filter to isolate the meaningful frequency range. For body temperature, which changes slowly, we use a Savitzky-Golay filter to smooth the data without flattening it. The goal here is to clean up the signals so the model doesn't get distracted by irrelevant fluctuations. It helps the model focus on real physiological trends, instead of on noises.

4. Standardization
Standardizing the signals ensures that all features contribute equally to the model and prevents high-magnitude variables from dominating. We fit the standardization parameters (mean and standard deviation) on the first 80% of each factor, which represents normal baseline behavior. By doing this, the variations in the final 20% anomaly data will stand out during learning or inference. Standardization also speeds up model convergence and improves numerical stability.

5. Pattern Labeling to Simulate Anomalies
To train and evaluate our model effectively, we label the data with synthetic patterns. This gives the model some structure to learn from, and later, it helps us measure how well the model is detecting abnormal patterns. This is only for our use to make sure that the abnormal patterns are preserved after processing.

6. Time Synchronization Using a Synthetic Timeline
Since our signals come from different sensors — each with its own clock and sampling rate — we build a synthetic timeline using evenly spaced intervals, such as 0 milliseconds, 200 milliseconds, 400 milliseconds, etc. We then align all the signals to this timeline. This ensures that every row represents the same point in time across all signals. It makes the dataset consistent and model-friendly, especially for time-series models.

7. Combining Features and Structuring the Dataset
After everything is preprocessed and aligned, we put all the signals together into one dataset. Each row now contains all the factors, including SpO_2, heart rate, body temperature, and blood

pressure, along with their pattern labels.

8. Final Export for Modeling
Finally, we export the dataset as a CSV file. It's now fully cleaned, aligned, labeled, and ready to go into a machine learning model.

By completing this end-to-end pipeline for preprocessing workflow, we've ensured that our sample data is not only clean and reliable but also representative of the operational conditions for real-time analysis, with the abnormal patterns being preserved. This preparation is critical for validating our system and enhancing the overall performance of subsequent ML models.

5.4.1.4 Public Datasets

We have explored several publicly available datasets to support our project's data-driven modeling simulations.

1. The Human Activity Recognition (HAR) dataset
This dataset provides wearable sensor data capturing human activities such as walking, running, and sitting, which can be useful for movement-based monitoring and fall detection systems.
2. The PhysioNet database
This database offers large-scale physiological and clinical datasets, including ECG, EEG, pulse oximetry, and respiration data, which are valuable for health monitoring and predictive modeling.
3. The UK Biobank dataset
This dataset contains comprehensive genetic, lifestyle, and health-related information, enabling deeper insights into health patterns and disease risks.
4. Air Quality Dataset
For environmental analysis, the Air Quality Dataset from the EPA provides crucial data on outdoor air pollution levels, supporting studies on air quality and public health.
5. Global Health Observatory (GHO)
This dataset from the WHO presents global health statistics, disease prevalence, and key risk factors, facilitating epidemiological research and health policy development. These datasets collectively cover a broad spectrum of health, physiological, environmental, and activity-based data, forming a robust foundation for our analysis.

5.4.1.5 Planned Data Post Processing (if applicable)

Evaluating ML Model Predictions

Machine learning models for sensor data, including for regression, classification and anomaly detection models can be evaluated using standard metrics.

In regression models, where the goal is to predict continuous values such as temperature or pollution levels, metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and the R^2 score help with assessing the accuracy and variance explained by the model. In classification models, such as categorizing physical conditions or detecting anomalies, evaluation relies on accuracy, precision, recall, F1-score and ROC curve (along with AUC, specificity and sensitivity) to determine the model's effectiveness, particularly in imbalanced datasets. In unsupervised learning scenarios, such as anomaly detection or clustering, performance can be measured using the Silhouette Score for cluster validation and precision-recall analysis for identifying rare physical conditions. In addition, explainability techniques like feature importance analysis help with interpreting the model's returns, ensuring the interpretability, transparency and trustworthiness of the models for being applied in real-world problems.

Evaluating LLM Model Responses

For LLM models, the evaluation focuses on the accuracy, consistency and relevance of the generated text-based insights. Since LLMs interpret and summarize sensor data rather than producing structured numerical predictions, their outputs require qualitative assessment through expert validation and semantic similarity metrics like cosine similarity, bilingual evaluation understudy (BLEU), and recall-oriented understudy for gisting evaluation (ROUGE). Fact-checking against historical sensor trends ensures reliability, while robustness testing, such as out-of-distribution (OOD) analysis, helps assess how well the model handles unexpected sensor readings. Bias detection may also be applied to prevent misleading or exaggerated claims based on training data biases. Unlike traditional ML models, which rely on structured evaluation metrics, LLM models require a combination of automated and human review to ensure their outputs are meaningful, reliable, and actionable in environmental monitoring applications.

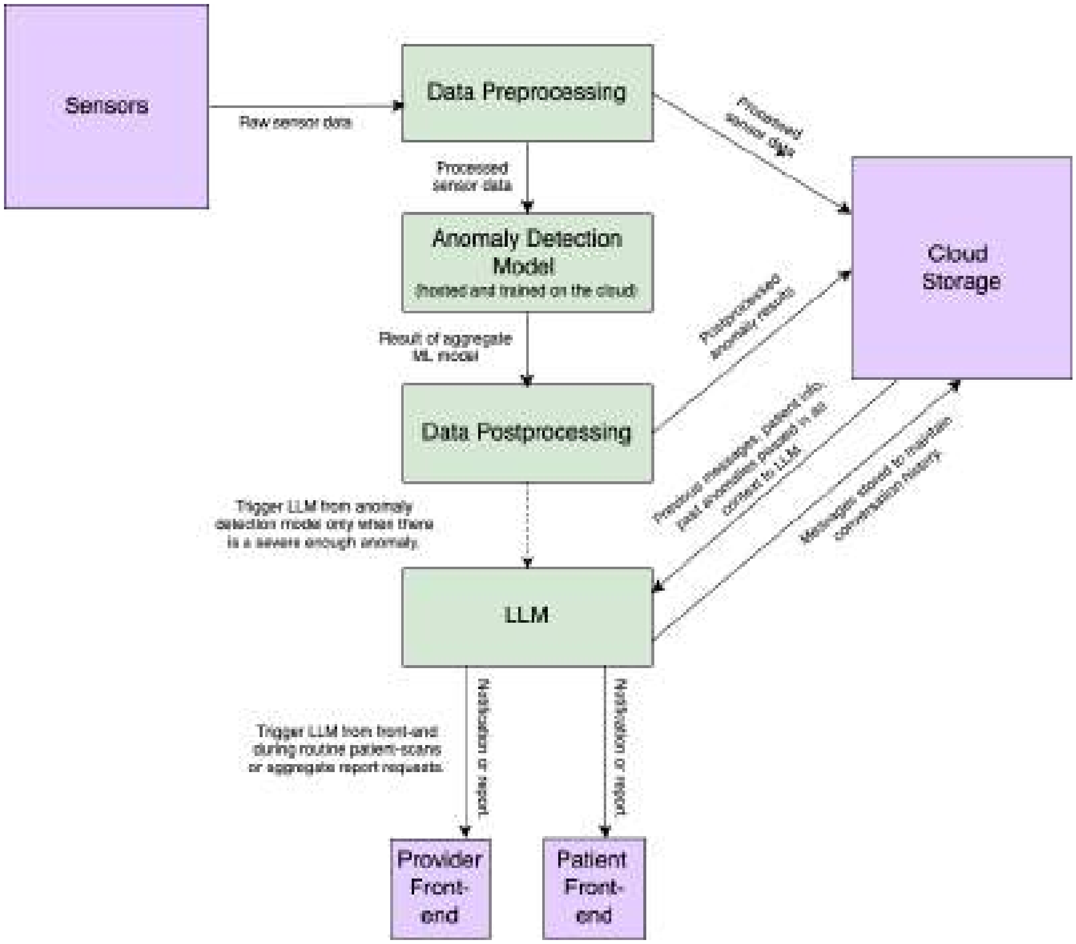
5.4.2 ML Models

The ML model is the central component of our intelligence subsystem. It processes continuous streams of sensor data from wearable and environmental devices and converts cleaned, pre-processed data into natural language anomaly strings that are subsequently used by a large language model (LLM) to generate patient-friendly notifications.

5.4.2.1 Data Pipeline Overview

Raw data is ingested from multiple sources, including wearable sensors (e.g., heart rate, movement, ECG, PPG) and environmental sensors (e.g., ambient temperature, humidity, CO_2 concentration), along with auxiliary inputs such as user-entered metrics.

Pre-processing—comprising noise filtering, outlier removal, and signal normalization—is applied to ensure high-quality data before it enters the ML pipeline.



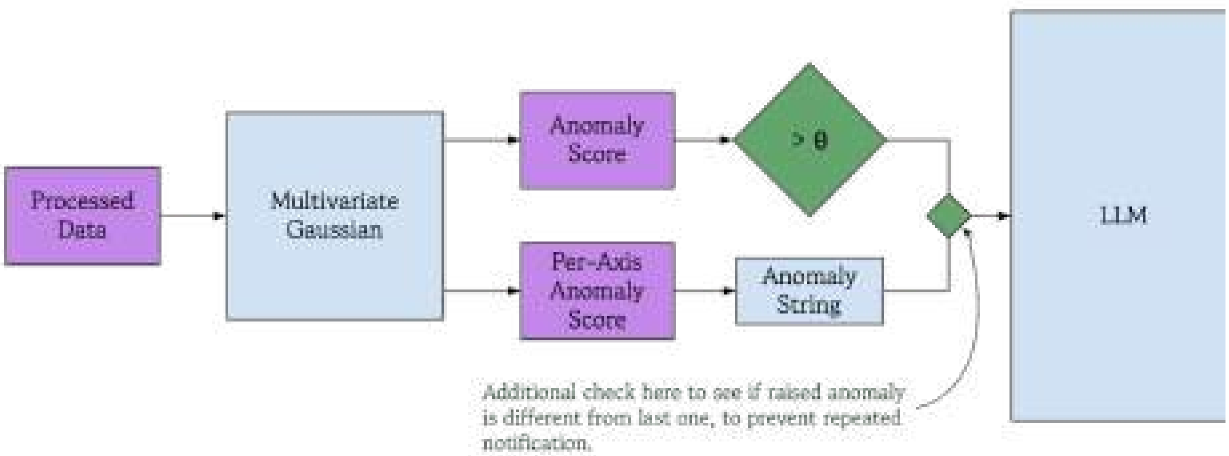
5.4.2.2 Anomaly Detection Framework

Our ML model employs a unified anomaly detection approach based on a Multivariate Gaussian framework, which captures the joint behavior of all sensor signals. This approach is designed to be both statistically robust and interpretable:

- **Holistic Signal Analysis:** The model evaluates the joint distribution of all pre-processed sensor data. By analyzing both the mean values and typical variations—along with inter-signal correlations—it is capable of detecting anomalies that might be missed when evaluating individual signals in isolation.
- **Statistically Defined Thresholds:** Anomaly thresholds are formulated in terms of standard deviations (sigmas) from normal behavior. The model computes both an

overall anomaly score and individual scores for each biometric axis, providing a clear, quantitative measure of deviation from expected patterns.

- **Unsupervised and Adaptive Training:** Trained using historical sensor data in an unsupervised manner, the model learns the normal behavioral patterns without reliance on extensive labeled data. As new sensor data is continuously ingested, the model adapts to evolving, patient-specific baselines, ensuring both sensitivity and specificity in anomaly detection.
- **Notification Filtering:** To avoid redundant alerts and reduce potential alert fatigue, the system compares each detected anomaly with the previous one sent to the LLM. Only when it is different is the new anomaly flagged. The resulting natural language anomaly string (for example, “High Heart Rate, Low Movement”) is then passed to the LLM.

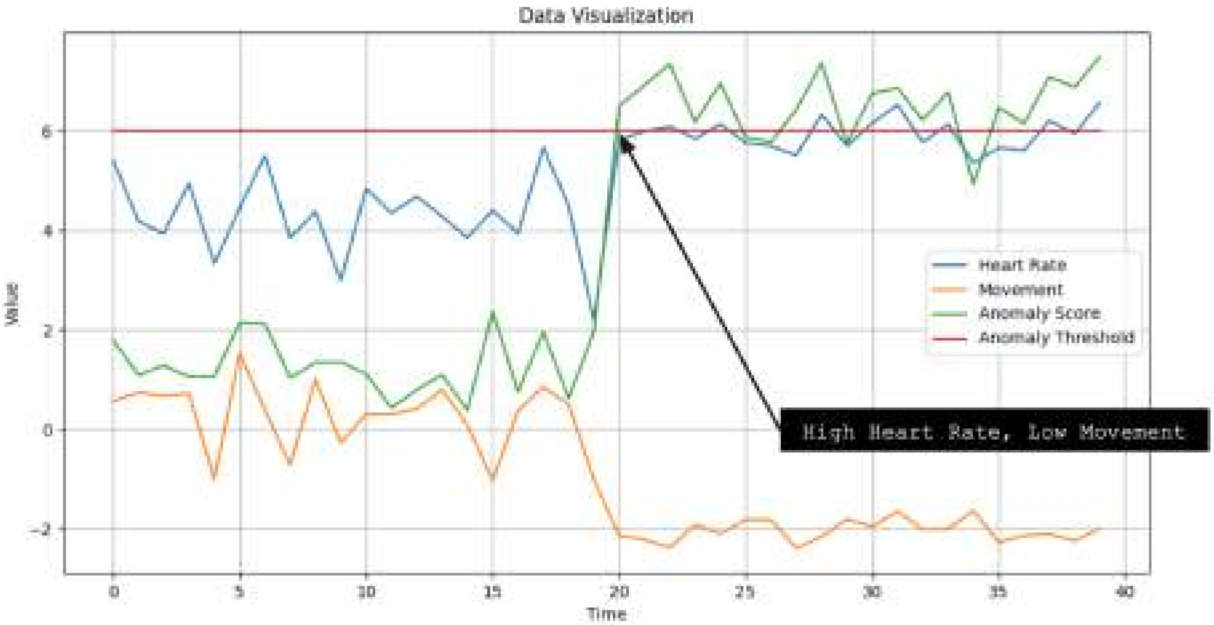


5.4.2.3 Implementation and Operational Results

In operation, the unified model has proven effective in identifying critical patterns. For instance, it reliably flags situations characterized by “High Heart Rate, Low Movement.” In such cases, the model:

- Computes a global anomaly score alongside individual scores for each sensor input.
- Evaluates these scores against statistically derived sigma thresholds.
- Outputs a natural language anomaly string when a significant deviation from the patient’s recent baseline is detected.

This anomaly string is stored in the cloud as part of a comprehensive patient history. The LLM later retrieves these stored anomalies to provide context when generating detailed, patient-friendly insights and notifications.



5.4.2.4 Integration with the Overall System Architecture

The ML model is tightly integrated within our system architecture. Cleaned data from upstream pre-processing is fed directly into the ML model, which outputs a natural language anomaly string that encapsulates the detected irregularity. This string is then forwarded to the LLM, which, using the stored record of past anomalies as patient history, generates comprehensive notifications and insights.

- **Local Edge-Based Processing:** Critical components of the ML pipeline are deployed on local base stations, enabling rapid retraining and real-time anomaly detection directly from the latest sensor inputs.
- **Cloud Storage for Patient History:** While the ML model is trained on sensor data, the cloud is utilized to store a record of all detected anomalies. This historical repository is essential for the LLM to reference past events and generate context-aware insights.

5.4.2.5 Further Improvement of Anomaly Detection Algorithm (Phase 3)

In Phase 1 and Phase 2, our focus was on detecting anomalies at the single-point level. Each sensor reading (e.g., heart rate, body temperature) was treated independently, and the ML model evaluated whether the value at a given time was anomalous without considering temporal context.

In Phase 3, we introduced a more robust approach by detecting anomalies based on short time intervals rather than individual points. For each interval, we computed statistical features such as the mean and median of sensor readings. These aggregated values were then used as inputs to the anomaly detection model, allowing it to identify deviations in the overall behavior rather than isolated spikes.
Additionally, we incorporated temporal variation analysis. Each sensor reading was compared to its preceding and following values to assess the degree of fluctuation. A significant increase in variation, especially over a short period, was treated as a signal for potential anomaly and was also fed into the model for decision-making.

5.4.3 LLM

Large-language models (LLM) can turn the output of the ML model into a human-readable notification and anomaly report. The LLM would also be able to infer both potential causes of the anomalies and mitigation techniques/remedies. This is useful because it would already be trained on a vast amount of medical information and could produce these notifications with ease, which allows us to focus on creating a more accurate ML model rather than adding an extra natural language processing (NLP) layer on top of our current work.

Out of all of the readily-available LLMs we considered (Gemini 1.5, Claude 3.5, Llama 2, GPT 4.0), we chose to use OpenAI's ChatGPT-4.0 model, primarily because It performs well on medical-related tasks and can efficiently convert detected anomalies into easy-to-understand notifications and reports. In addition, ChatGPT-4.0 has strong contextual understanding capabilities and can combine sensor data and medical knowledge to provide reasonable mitigation measures.

Model	OpenAI GPT-4	Gemini 1.5 Pro	Claude 3.5 Haiku	Llama 2
Hardware Requirements	Cloud-hosted	Cloud-hosted	Cloud-hosted	GPUs required (for local hosting)
Context Length	Up to 128k tokens	Up to 128k tokens	Up to 200k tokens	Configurable (4-32k tokens)
Max Response Length	4096 tokens	8192 tokens	8192 tokens	4096 tokens
Input Cost	$2.50 / 1M input tokens	$1.25 / 1 million tokens	$0.80 / MTok input	Free to use model when self-hosted (but incurs hosting cost)
Output Cost	$10.00 / 1M output tokens	$5.00 / 1 million tokens	$4 / MTok output	Free to use model when self-hosted (but incurs hosting cost)
Availability	Now	Now	Now	Now

Compared with other LLMs (such as Gemini 1.5, Claude 3.5, and Llama 2), ChatGPT-4.0 performs better in medical reasoning, knowledge coverage, API availability, and reasoning speed, while its ecosystem support is strong and easy to integrate into our products. We present a comparison of relevant features in the above table. Llama 2 would be difficult to set-up and host because it would require a GPU for local hosting. In comparison, GPT-4 is cloud hosted and can be accessed through Azure. While GPT-4 incurs the highest cost ($10/1 million output tokens), we asked GPT to generate weekly reports and anomaly notifications (such as the ones presented below) and ran them through OpenAI's tokenizer tool: We found that each report was, on average, ~1000 tokens and each notification was ~70 tokens. If we conservatively estimate that a patient receives 5 anomaly notifications a day, and requests 2 reports per week, they will only incur an output token cost of 0.0445 cents/week per patient. As for input tokens, we estimate that each time the API is invoked, we will pass an input context of ~1500 tokens (including patient info, instructions to the model, output of the ML model, previous anomalies). Keeping with the 5 anomaly notifications and 2 reports per week example, the patient will only incur an input token cost of 0.0175 cents/week per patient. Overall, using GPT-4.0 will cost us only 0.062 cents/week per patient.

5.4.3.1 Prompt Engineering Framework

Based on the research on Prompt Engineering, we selected several prompt word frameworks for exploration and finally chose COSTAR Prompt Engineering as our design framework to understand how the framework can play a role in the context of medical and sensor data processing. The COSTAR framework emphasizes solving specific problems through modular system construction, in which the tasks and functions of each module are carefully designed to ensure efficient and accurate information flow.

Introduction to COSTAR Framework

- **Context:** Providing context for the task
  Providing detailed context to the LLM helps it understand the context of the discussion and ensures that the feedback it provides is relevant.
- **Objective:** Clarify the task you ask the LLM to complete
  Clearly defining the task objective allows the LLM to more focusedly adjust its response to achieve this specific goal.
- **Style:** Clarify the writing style you expect
  You can specify the writing style of a specific famous person or industry expert, such as a business analyst or CEO. This will guide the LLM to respond in a manner and vocabulary choice that matches your needs.
- **Tone:** Set the emotional tone of the response
  Set an appropriate tone to ensure that the LLM's response is in harmony with the expected emotional or sentimental context. Possible tones include formal, humorous, sympathetic, etc.
- **Audience:** Identify the target audience
  Tailoring the LLM's response to a specific audience, whether it is an expert in the field, a

beginner, or a child, ensures that the content is appropriate and easy to understand in a specific context.

- **Response:** Specify the output format
  The purpose of determining the output format is to ensure that the large language model outputs according to your specific needs and facilitate downstream tasks. Common formats include lists, JSON formatted data, professional reports, etc. For most applications that need to programmatically process the output of the large language model, the JSON format is an ideal choice.

5.4.3.2 Research

To explore how GPT-4 can be effectively integrated into our system, we first focused on connecting the model with relevant medical and sensor-based data sources. By designing structured prompts using the COSTAR framework, we aimed to test the model's ability to understand and respond to diverse types of data inputs. These include real-time environmental data (e.g., noise level, temperature), user-specific physiological data (e.g., heart rate, movement), and historical user interaction records.

In parallel, we developed a preliminary notification system. The design of different types of notifications is based on context-aware analysis: for instance, urgent physiological changes may trigger immediate alerts, while mild environmental discomfort may only generate gentle suggestions. We are currently categorizing notification types into three levels: Critical, Advisory, and Informational.

To support adaptive responses and reduce information overload, we introduced a scoring system that evaluates both the urgency and the relevance of incoming data. The score determines whether the data should trigger a notification, and if so, how it should be phrased and prioritized in the LLM's response. This mechanism also helps mitigate unnecessary alerts, ensuring users only receive meaningful and timely information.

For testing, we simulated various user scenarios by feeding GPT-4 with combined inputs consisting of environment readings, user biometric data, and behavior history. We observed how the model adapts its tone, style, and structure depending on the prompt structure and the COSTAR elements in use. This experiment helps us refine our prompts and determine which input formats and combinations yield the most appropriate and accurate responses.

5.4.3.3 LLM Scripts

The files that we created to interface with the LLM will be explained below, but are listed here for reference: anomaly_notification.py and weekly_report.py. All of the .py files were designed and initially developed by us, but refined and refactored with ChatGPT. We dive deeper into each file below.

Anomaly_notification.py

This script creates a notification for a given anomaly by interfacing with OpenAI 4.0, given a prompt, context window, and an anomaly.

This is the context window we use:

Unset
```
    You are a medical assistant. Please follow these rules:
    - Provide a severity score that is a **single integer** between 0 and 5.
    - Here is an example calibration of a severity score, given an anomaly related to "low blood oxygen saturation":
        Severity Score 0: 95-100% Blood Oxygen
        Severity Score 1: 92-95% Blood Oxygen
        Severity Score 2: 90-92% Blood Oxygen
        Severity Score 3: 88-90% Blood Oxygen
        Severity Score 4: 80-88% Blood Oxygen
        Severity Score 5: <80% Blood Oxygen
    - Describe the anomaly in simple terms.
    - Suggest a clear mitigation technique.
    - DO NOT diagnose but can reference past diagnoses made by doctors.
    - Keep responses concise and structured.

    ### **Patient Information**:
    - Name: {self.patient_info.get('name', 'Unknown')}
    - Pronouns: {self.patient_info.get('pronouns', 'Unknown')}
    - Age: {self.patient_info.get('age', 'Unknown')}
    - Time after birth: {self.patient_info.get('time_after_birth', 'Unknown')}
    - Prior diagnoses: {self.patient_info.get('diagnoses', 'None')}
    - Allergies: {self.patient_info.get('allergies', 'None')}
```

```
    - Breastfeeding Status: {self.patient_info.get('breastfeeding_status', 'Unknown')}
    """

def generate_prompt(self, anomaly):
    return f"""
    Anomaly detected: {anomaly}

    Provide:
    1. **Severity Score**: [A number from 0-5]
    2. **Anomaly Description**: [Short medical description]
    3. **Mitigation Technique**: [Steps to mitigate the issue]

    General Severity Score Guidance:
```

Note that we include an example calibration of severity score for blood oxygen-related anomalies. We do this so that the LLM will accurately provide severity score regardless of the anomaly that was detected, because earlier testing showed that different types of anomalies produced differently calibrated severity scores.

This is the prompt we use:

```
Unset
f"""
Anomaly detected: {anomaly}

Provide:
1. **Severity Score**: [A number from 0-10]
2. **Anomaly Description**: [Short medical description]
3. **Mitigation Technique**: [Steps to mitigate the issue]
"""
```

For example, when we receive the following anomaly: “Irregular heartbeat detected” from the anomaly detection model, we output the following notification:

```
Unset
{
	"date": "2025-03-17",
	"anomaly": "Irregular heartbeat",
	"severity_score": 6,
	"mitigation_technique": "Monitor heart rate regularly, reduce caffeine intake, stay hydrated, and prioritize rest. Schedule an appointment with a cardiologist to rule out underlying concerns."
}
```

Weekly_report.py

This script creates an aggregate report given all of the anomalies in the past week.

This is the context window we use:

```
Unset
You are a medical assistant. Please follow these rules:
- Infer possible causes of anomalies based on sensor data, ML model output, past diagnoses, and patient info.
- Suggest mitigations based on these data points.
- Prefer concise responses for readability and to limit token usage.
- Be explicit about which data points were anomalous for transparency.
- DO NOT diagnose but can reference past diagnoses made by doctors.
- Do not mention all previous anomalies, but can reference them as needed.

### **Patient Information**:
- Name: {patient_info['name']}
- Pronouns: {patient_info['pronouns']}
- Age: {patient_info['age']}
- Time after birth: {patient_info['time_after_birth']}
- Prior diagnoses: {patient_info['diagnoses']}
- Allergies: {patient_info['allergies']}
- Breastfeeding Status: {patient_info['breastfeeding_status']}

---

### **Weekly Anomaly Report**
Summarize the anomalies from the past week in a structured format.
```

```
### **Trends**
Identify potential concerning patterns based on the anomalies.

### **Mitigation Techniques**
Provide a **bulleted list** of recommended mitigation techniques.
```

This is the prompt we use:

```
Unset
Here are the anomalies detected over the past week:

{anomaly_text}

Generate a detailed report including:
1. **Weekly Anomaly Report** summarizing the anomalies.
2. **Trends** based on observed patterns.
3. **Mitigation Techniques** as a **bulleted list**.
```

For example, when we receive the following anomaly: “Irregular heartbeat detected” from the anomaly detection model, we output the following notification:

```
Unset
        {
        "date": "2024-03-12",
        "anomaly report": "- **2024-03-10**:\n- High blood pressure (Severity: 7).\n- Possible cause: Related to prior gestational hypertension diagnosis.\n- Irregular heartbeat (Severity: 8).\n- Possible cause: Postpartum cardiovascular stress or electrolyte imbalance.\n- **2024-03-11**:\n- Low oxygen saturation (Severity: 9).\n- Possible cause: Respiratory issue or cardiovascular instability.\n- Sudden drop in heart rate (Severity: 6).\n- Possible cause: Possible vagal response, dehydration, or cardiovascular issue.\n- **2024-03-12**:\n- Increased body temperature (Severity: 5).\n- Possible cause: Potential postpartum infection or inflammation.",
        "trends": "- **Cardiovascular instability**: Repeated occurrences of high blood pressure, irregular heartbeat, and heart rate anomalies may indicate ongoing postpartum cardiovascular stress, potentially exacerbated by her history of gestational hypertension.\n- **Respiratory or oxygenation concerns**: Low oxygen saturation on 2024-03-11 is an outlier but concerning, especially if linked to cardiovascular irregularities.\n- **Infection risk**: Elevated body temperature on 2024-03-12 could indicate an early-stage postpartum infection.",
```

"mitigation": "- **Monitor cardiovascular health**:\n- Record blood pressure and heart rate daily.\n- Schedule a follow-up with her primary care provider or cardiologist to rule out postpartum preeclampsia.\n- **Address oxygenation issues**:\n- Conduct a pulse oximetry test if symptoms like shortness of breath recur.\n- Stay hydrated and avoid overexertion.\n- **Infection prevention**:\n- Check for localized signs of infection (e.g., redness, swelling, or discharge from the C-section site or vaginal area).\n- Consult a physician promptly if fever persists or worsens.\n- **Lifestyle adjustments**:\n- Ensure adequate rest and minimize stress.\n- Maintain breastfeeding if feasible to support immune function and recovery."
}

5.4.3.4 Anomaly Notification Language Design

Collaboration with HCI Team

The phrasing system and severity mappings were developed collaboratively with the HCI team, whose expertise ensured that the language model outputs are medically appropriate, emotionally considerate, and accessible to users undergoing postpartum recovery. This collaboration bridges the gap between technical anomaly detection and effective, empathetic communication with end users.

The notification phrasing is based on three key principles:

- Consistency: Standardized sentence structures make notifications predictable and easy to understand.
- Severity Awareness: Language reflects the risk level associated with each anomaly, ranging from low-risk monitoring to critical emergency escalation.
- Actionability: Each notification includes a direct, appropriate next step for the user to take.

Notification Template

The general template for phrasing anomaly notifications is as follows:

```
Unset
It looks like your [anomaly subject] is [severity level] [anomaly
adjective]. Try [mitigation technique].
```

This format allows the LLM to dynamically insert relevant anomaly information, severity levels, and suggested mitigation actions based on the incoming sensor data.

Severity Score System
Each detected anomaly is assigned a severity score from 0 to 5, which determines the tone and urgency of the notification:

Severity Score	General Meaning	Example Language
0	No Risk (no notification sent)	N/A
1	Slightly elevated, monitor recommended	"Your blood pressure is slightly elevated. Try monitoring regularly."
2	Somewhat elevated, slight adjustment recommended	"Your blood pressure is somewhat elevated. Try reducing sodium intake."
3	Moderately elevated, behavioral change advised	"Your blood pressure is moderately elevated. Please monitor regularly and adjust your lifestyle."
4	Very elevated, immediate attention needed	"Your blood pressure is very elevated. Your care team has been notified. Please monitor symptoms and seek medical care if necessary."
5	Critically elevated, emergency action triggered	"Your blood pressure is critically elevated. 911 has been called. Seek immediate medical assistance."

5.4.3.5 Input/Output Cloud Data

The patient data, which is passed in as context to the LLM, and the output data like the anomaly notifications and weekly reports, will all be stored in DynamoDB tables in the cloud. We list the data format and contents of these below, but the tables are discussed in more detail in Section 5.3.3.2.

Anomaly Data Table

This table holds the data of all anomalies that have been detected. For each anomaly, we output the following data:

1. Date
2. Anomaly
3. Severity Score

4. Mitigation Technique

Weekly Report Data Table

This table holds the data of all weekly reports that we have generated. For each report, we output the following data:

1. Date
2. Anomaly Report
3. Trends
4. Mitigation

Patient Data Table

The patient data table holds the relevant information about each patient, which will be passed into the LLM as context. It is structured into seven major sections, with the associated justification for including each:

1. Name
   - This is so that the LLM is able to refer directly to the patient with their preferred name. Personalized interaction helps build trust, which is especially important during the postpartum period when individuals may be particularly vulnerable or overwhelmed.
2. Age
   - This is because certain diagnoses and trends are more common at different ages. For example, postpartum hypertension and gestational diabetes are more prevalent in patients over 35, while younger patients may face increased risks for postpartum mood disorders. Age also influences how someone recovers physically, how their metabolism responds, or how sleep disruption affects them—all things that might show up in sensor data.
3. Pronouns
   - This is so that the LLM is able to use the patient's preferred pronouns. One consideration we made was that not all people who give birth self-identify as women or as mothers. Thus, we need the LLM to be more sensitive to the patient's preferred method of address, ensuring respectful and inclusive communication throughout their postpartum care.
4. Time after birth
   - This is included because some symptoms are more common when someone has just given birth. In the first few days, things like heavy bleeding or intense

emotional swings are more common, whereas issues like ongoing fatigue, joint pain, or anxiety might show up later. Wearables can track subtle changes in things like heart rate, sleep, or movement, but the context of time since delivery is key to making sense of those changes.

5. Diagnoses
    - This is because certain trends and anomalies might be caused by underlying diagnoses made by medical professionals. If someone has been diagnosed with a medical condition, that can shape what we expect to see in their sensor data. Since we've decided the language model won't be used to make new diagnoses, it's important that it has access to known diagnoses when interpreting data.
6. Allergies
    - We want to make sure that any recommendations made by the LLM do not conflict with underlying conditions, including allergies. Allergies may also contribute to anomalies. For example, if someone has a pollen allergy, outdoor environmental data could explain sudden changes in their physical state. Including allergy information helps avoid misinterpretation and ensures any suggestions made don't conflict with their known sensitivities.
7. Breastfeeding status
    - This is important because breastfeeding affects hormones, energy levels, hydration, and sleep, all of which are relevant to what sensors are tracking. It is also relevant to the treatments and the medications, especially whether the medications are safe. Having this context helps the system make more tailored, appropriate suggestions and interpretations.

5.5 Wearable Hardware/Firmware

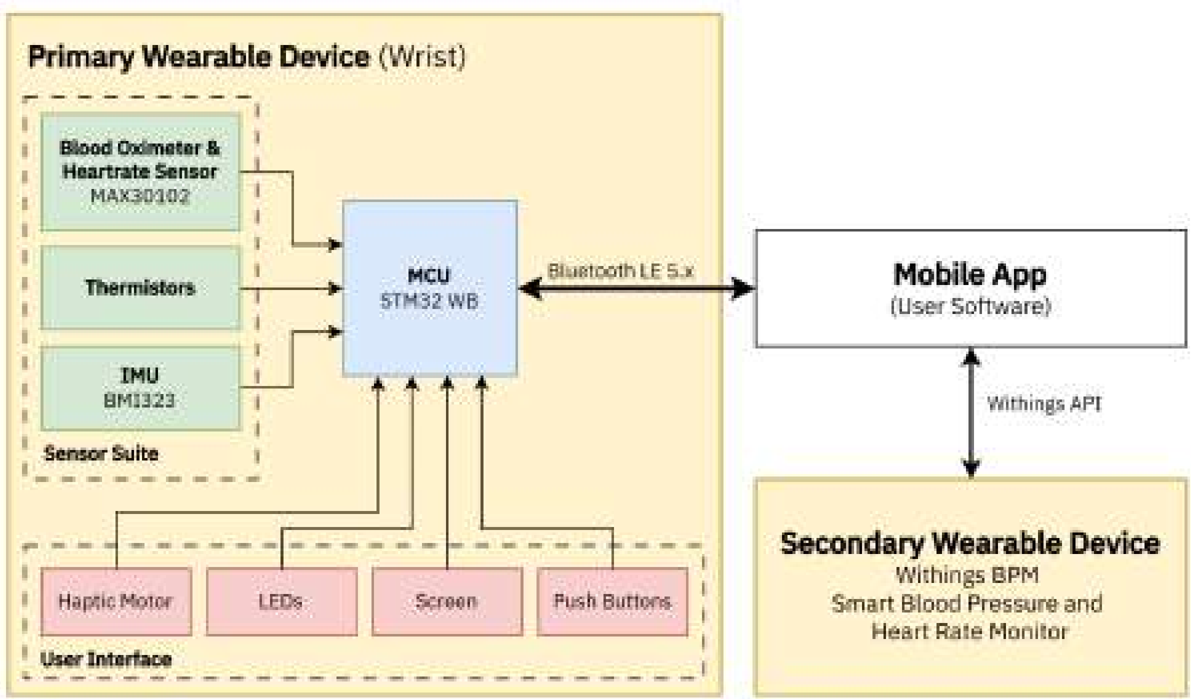
The wearable device plays a crucial role in the overall system by serving as the primary source of physiological data collection and providing real-time feedback to the user.

It is equipped with various sensors to monitor key health metrics, including heart rate, heart rate variability, heart rhythm, arrhythmias, blood oxygen saturation, body temperature, and motion data through an inertial measurement unit (IMU). These sensors continuously gather data, which is processed by the onboard microcontroller unit (MCU) and transmitted via Bluetooth to the patient-facing mobile application.

The wearable device also includes actuators, such as vibration motors, LEDs, or speakers, to provide immediate feedback based on the system's analysis or user interaction. The collected data is shared with the software platform, where it is further analyzed and integrated with environmental sensor data and processed by the data processing model. The processed data is then sent to the cloud for anomaly detection, historical storage, and reporting. Insights from this data are accessible to both the patient and healthcare providers via the patient app and provider dashboard.

As before, the design of the wearable device is driven by three core goals: ease of use, noninvasiveness, and responsiveness. The device is designed to be intuitive, requiring minimal effort to learn and operate. This product would be used by patients who have just been released from the hospital. Ensuring that the product doesn't add any unnecessary complications to the patient's routine is important in making sure that the patient continues to use the device. This post-discharge is considered a vulnerable period for patients thus making sure that the wearable is responsive to any health anomalies and can effectively communicate this to providers is an important focus of the device's design.

5.5.1 System Architecture



The Wearable Hardware/Firmware team's system architecture diagram at the end of phase 2.

The above diagram depicts the finalized system architecture for our subsystem as well as its relation to the mobile app. In addition to the primary wearable device which incorporates custom hardware, we opted to include a secondary wearable device to measure blood pressure. Further, we have finalized the constituent components for the user interface to accommodate the desired user flows and interactions by the HCI teams.

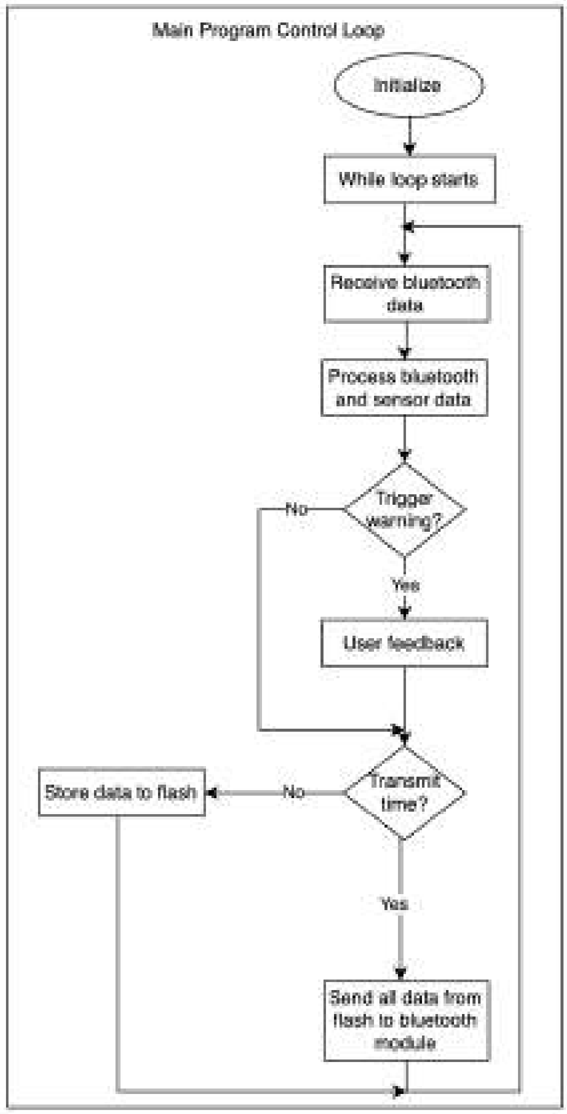
This diagram also represents the components that will be present on the first iteration of the custom PCB; the first iteration will be ordered between the end of phase 2 and the start of phase 3 to allow sufficient time to test and integrate the system onto the board.

Our implementation revolves around several components: the sensor suite, user interface, communication both within the wearable system and to the rest of the overarching system via the mobile app, and the firmware that enables these processes. We further detail these three aspects of the wearable in the following sections.

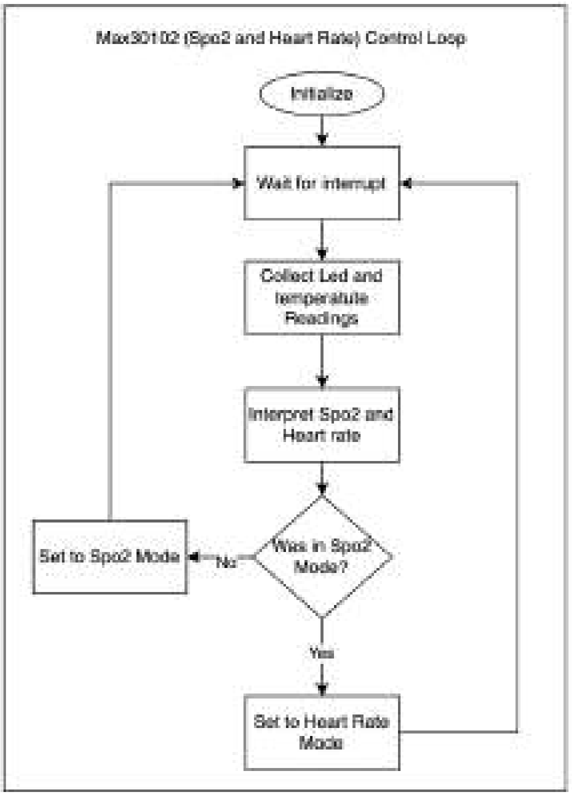
5.5.2 Microcontroller and Firmware Architecture

As stated in the phase 1 report, we decided to use a STM32WB series MCU for our application as it most closely met our needs for a builtin bluetooth module, low power consumption, sufficient I/O, and ease of development. For development, we used the STM32Cube IDE. We chose to abstain from using a RTOS to keep the application lightweight and we found timer interrupts to be sufficient for our needs.

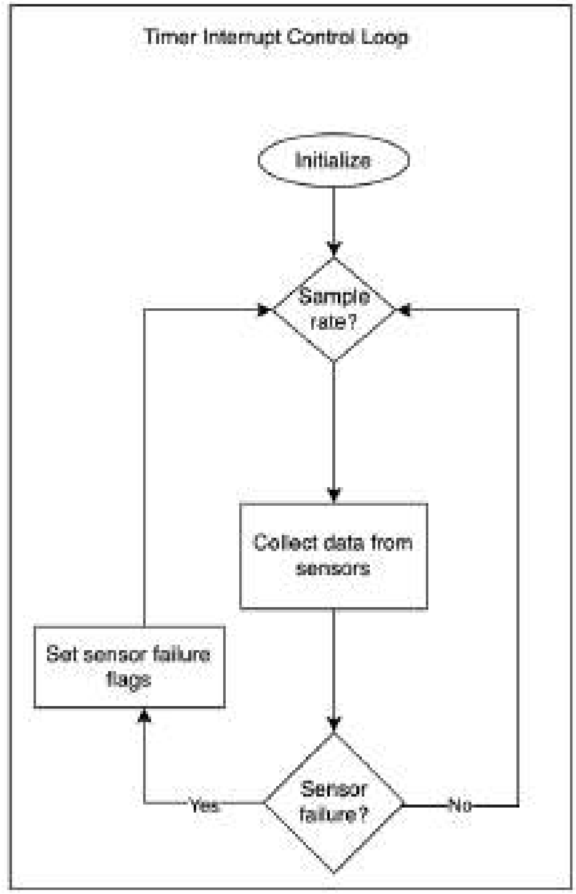
The following diagrams illustrate the general firmware flow of our program:



Main program control loop.



Example sensor interrupt-triggered data collection loop.



Example timer interrupt-triggered control loop.

5.5.3 Sensors

As previously mentioned, one of the primary purposes of the wearable is to collect pertinent biometric data from the user. To do this, we provide support for a variety of metrics through our sensor suite which spans both the primary and secondary wearable. We provide a general overview of these sensors in the following table:

Wearable	**Sensor**	**Raw Biometrics**
Primary	MAX30102 Blood Oximeter & Heart Rate Sensor	Blood Oxygen Heart Rate
Primary	Thermistors	Temperature
Primary	BMI323 IMU	Inertial Measurements (3-D accelerometric and gyroscopic,

		utilized for fall detection)
Secondary	Withings BPM Smart Blood Pressure	Blood Pressure Heart Rate

Manifest of sensors implemented across the wearable system and associated biometric data.

5.5.3.1 Primary Wearable Sensor Suite

As mentioned previously, part of phase 2 was determining the feasibility of implementing the variety of sensors we proposed in phase 1. From this, we chose to implement the following sensors, some of which consolidate several sensors into one (for example, the blood oximeter also provides a means of obtaining an estimated body temperature).

5.5.3.1.1 Pulse Oximeter

Pulse Oximeters are sensors that measure a patient's heart rate and blood-oxygen (SpO2) level. These two factors are standard measurements in health care as they can reveal signs of heart or respiratory problems. Heart rate data can be analyzed to determine respiration rate and heart rate variability.

The raw data from the blood oximeter is used to derive and approximate blood oxygen (Spo2).

5.5.3.1.2 Thermistors

Thermistors are included as a means of estimating the body temperature of the user. Since body temperature is difficult to accurately measure on the wrist, we opt for a simpler approach via thermistors to provide a rough estimate as is common across wrist-worn health devices.

5.5.3.1.3 IMU

The IMU measures accelerometric and gyroscopic data along 3 axes. The primary purpose of this sensor is to collect inertial data and ascertain via sharp changes in motion or spikes if a fall has occurred.

5.5.3.2 Secondary Wearable

One biometric that we were unable to accommodate on the primary wearable that was stressed as crucial to collect is blood pressure. Since it is closely associated and indicative of many risk factors related to postpartum, we still needed some way to obtain this data, even if not done passively. However, blood pressure remains a significant challenge to implement in a wearable and we decidedly lack the technical expertise and time to attempt to implement one either on or as a secondary wearable. Thus, we chose to utilize a third-party smart health device to measure blood pressure and send blood pressure data to the mobile device separately.

After comparing various models, we selected the Withings BPM Smart Blood Pressure for its accessible API, clinical validity, and ostensibly clean design.

5.5.4 User Interface

The user interface provides the most immediate and direct way for the user to interact with the wearable device. The design and implementation of this interface involved collaborating closely with the Wearable HCI team to balance tradeoffs in design complexity and user experience (as discussed in section 5.1.7). We detail the final constituent components below.

5.5.4.1 Haptic Motor

A haptic motor is used to provide means of notifying the user through vibration. Haptic feedback is particularly useful in situations where visual or auditory cues may go unnoticed, ensuring timely awareness without being intrusive.

Implementation-wise, we opted to use a linear resonant motor as these are what are typically used to create haptic feedback on smart watches. The haptic motor is controlled by a PWM signal from the MCU.

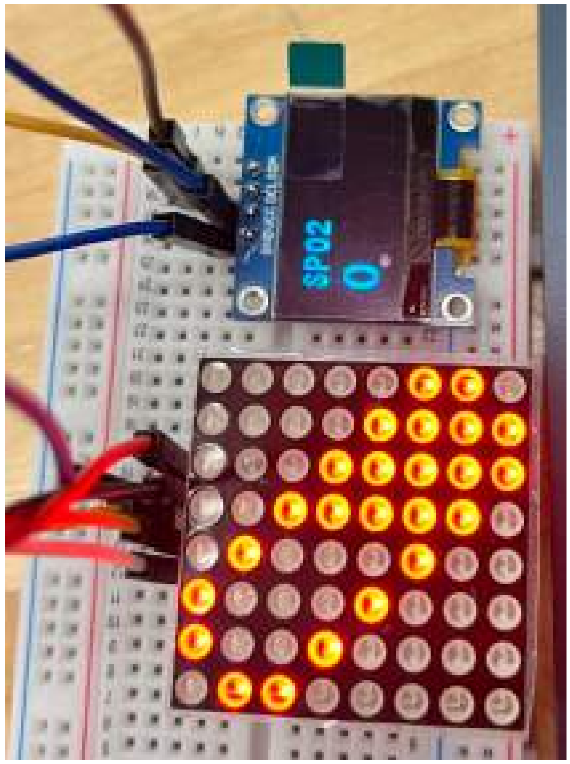
5.5.4.2 LEDs

In order to quickly communicate a general status to the user, we make use of several LEDs to indicate if an anomaly has occurred and a rough guide of severity. The comparatively lower fidelity compared to the screen display also allows for quicker understanding of the user's detected status. Design-wise, these are also easy to implement in both hardware and firmware and can be controlled easily via GPIO pins from the MCU.

5.5.4.3 Screen

Pushback during phase 1 caused us to more seriously consider a higher fidelity display option ala a screen to communicate with the user. After exploring several higher fidelity display options—including an LCD display, simple alphanumeric character displays, and an 8x8 LED matrix—we settled on using a 0.96 Inch OLED screen with a resolution of 128x64 pixels. This allows for us to display text in a wide variety of forms as well as individual pixel control. Since it is not limited to only alphanumeric characters, we can incorporate various icons to more efficiently communicate messages to the user.

The popularity of this type of module means it is well documented and we have the option of using various open-source libraries to streamline development; this also means we will be able to allocate more time on the display quality. We communicate with the module via I2C.



Two of the considered display options: the OLED display (top) and 8x8 LED matrix (bottom).

5.5.4.4 Push Buttons

In Phase 3, we implemented an intuitive push button interface that significantly enhances user interaction with the wearable device. Given the compact form factor and project constraints, we opted for physical buttons rather than a touch screen, resulting in a more accessible and reliable interface suitable for users of all technical abilities. The system supports multiple interaction modes: short presses allow users to cycle through different sensor data displays, while long presses provide functionality for canceling alerts such as fall detection warnings. We extended this interaction model to enable mood input through various button press combinations. The interface provides immediate visual feedback through the LED or screen, confirming user actions and improving the overall user experience. The technical implementation employs efficient hardware interrupts and a state machine architecture to manage the various system states and transitions, ensuring responsive performance while maintaining power efficiency—a critical consideration for wearable devices intended for continuous use outside the home environment.

5.5.6 Communication

5.5.6.1 Primary Wearable to Phone

As discussed in the phase 1 report, the primary wearable is unique in that it has no direct connection to the internet; this is intentional since the wearable is intended to be functional outside the home. During phase 2, we were able to reliably transmit data over bluetooth from the development board to the STBLEToolbox mobile platform. This bodes well for future development with the P/U Software team.

For Phase 3, we successfully implemented a robust Bluetooth communication framework between the primary wearable and the user's mobile application. Building upon our Phase 2 achievements, we enhanced the data transmission protocol to efficiently encode and transmit critical information including real-time sensor data, fall detection warning alerts, and precise timestamps. The fall detection algorithm now promptly communicates potential emergency situations to the mobile application with minimal latency, allowing users to receive immediate notifications when anomalous movement patterns are detected. Timestamps accompany each data packet, ensuring accurate chronological tracking of events and facilitating comprehensive data analysis. This implementation maintains the wearable's functionality outside the home environment while providing users with continuous monitoring and alert capabilities through their mobile devices, all without requiring direct internet connectivity from the wearable itself.

5.5.6.2 Secondary Wearable to Phone

We plan to utilize the existing API to send blood pressure data from the secondary wearable device to the phone. This was selected as opposed to sending data to the wearable since the mobile device will have more bandwidth than the wearable, and there is already an existing API specifically for sending data from the Withings device to a mobile platform. This link is still a work in progress.

5.5.7 Printed Circuit Board

To integrate all of our systems together, we designed a Printed Circuit Board with our Bluetooth MCU. We fabricated this board, and did initial testing, but were unable to get the final PCB working, leading us to go with a perfboard setup that integrated all of our sensors. However, this was a valuable experience in system design and thus, we document it below.

5.5.7.1 Schematics

Our schematics described the circuitry for each of the sensors and systems on our PCB. We have included screenshots of our schematics below.

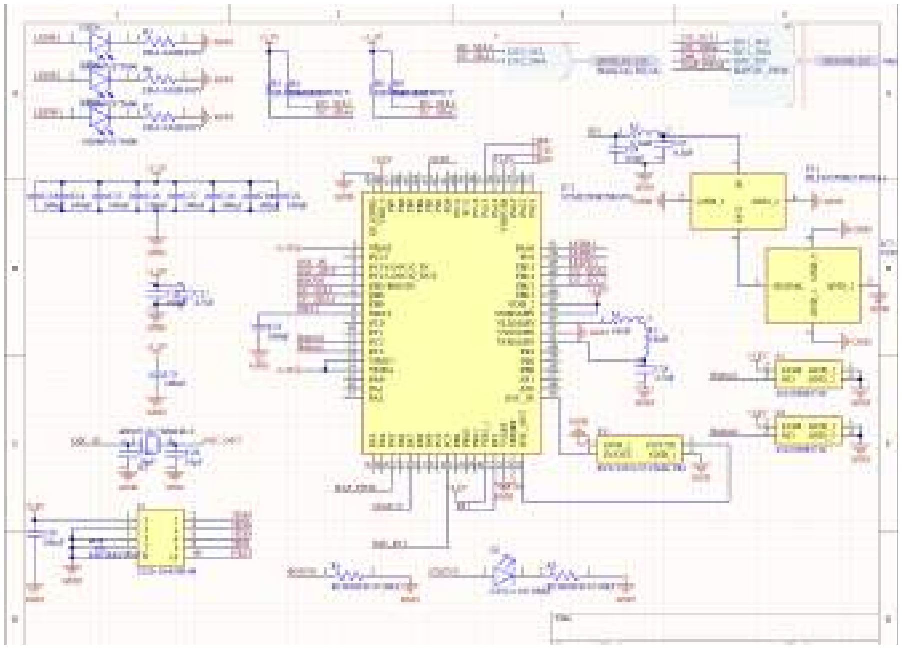


Figure 5.7.1.A: Microcontroller Unit Schematics

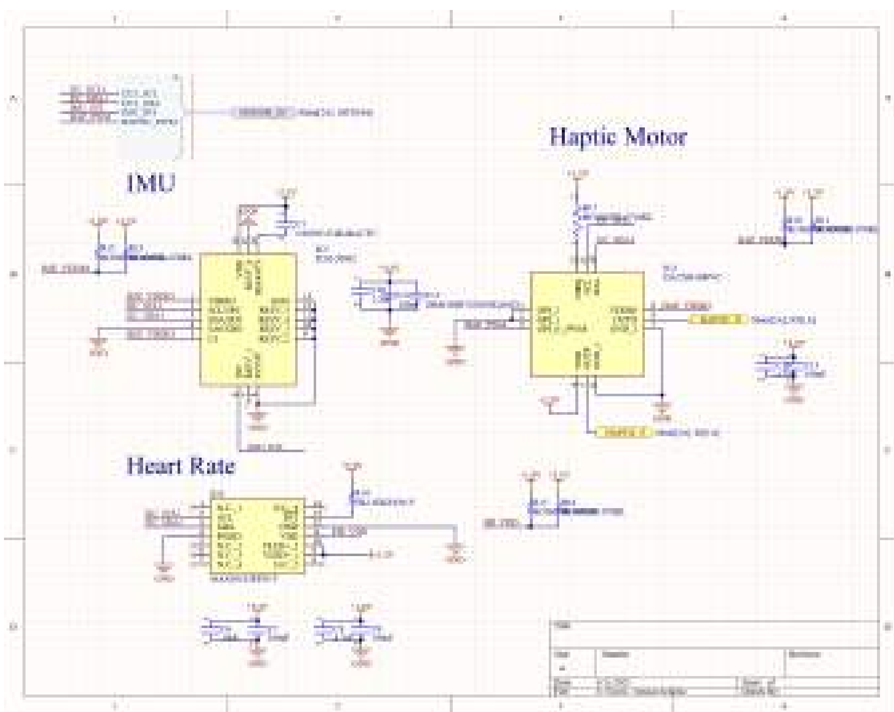


Figure 5.7.1.B: Sensor Schematics

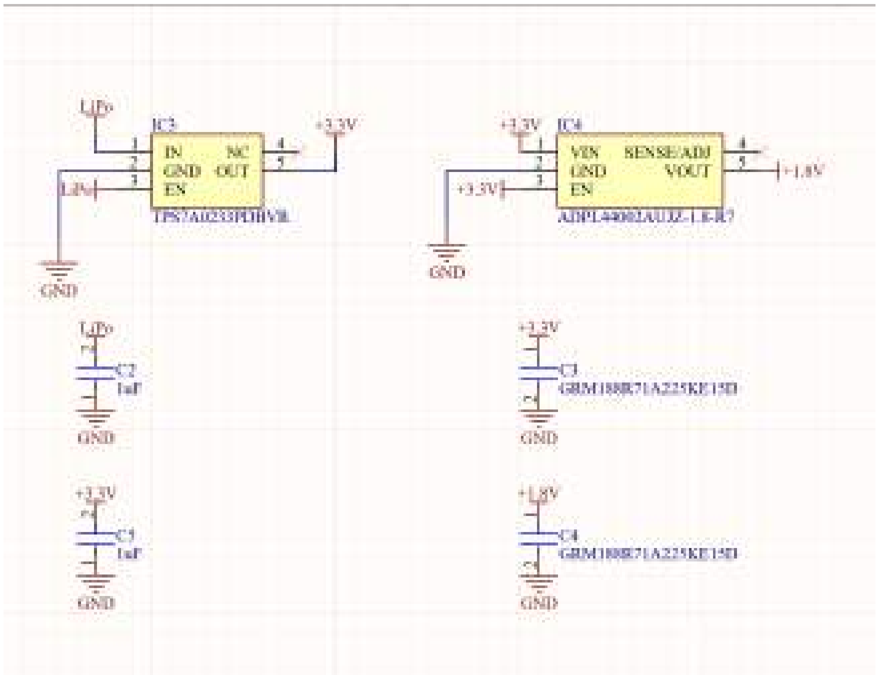


Figure 5.7.1.C: Power Schematics

5.5.7.2 Layout

After creating the schematics for our sensors and microcontroller unit, we then proceeded to the layout portion of our design. This involved making the electrical connections between all of our various components.

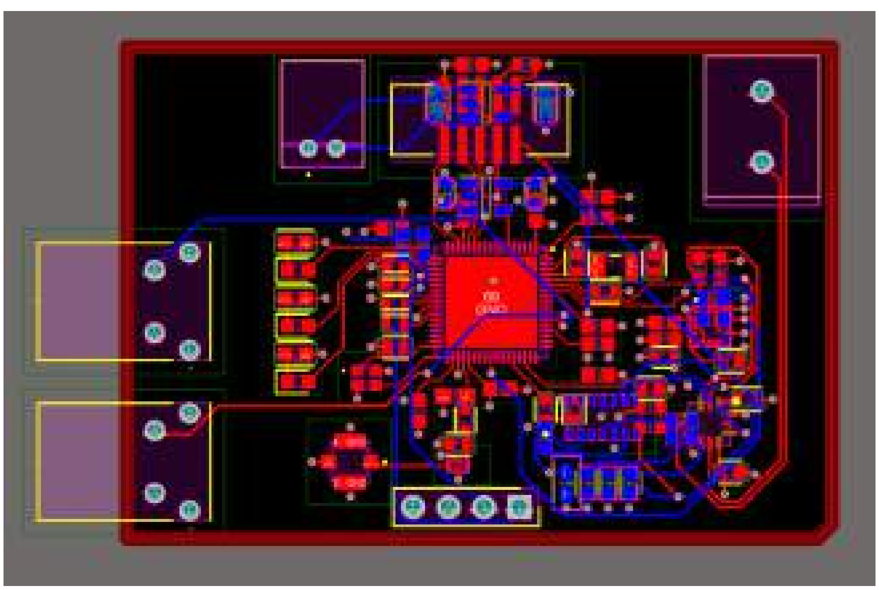


Figure 5.7.2.A: 2D Snapshot of Layout

5.5.7.3 Final Circuit Board

After the layout of our circuit board, we sent it off to be manufactured, and after it arrived, we reflowed the board and hardware/software tested it. During this testing process, we found an error in our circuitry, and we couldn't fix it in time for our final demo. However, the circuit board was a valuable design and learning experience.

Figure 5.7.3.A: Final Manufactured Circuit Board

5.6 Environmental Hardware/Firmware

Our team is focused on sensors that are placed in the user's environment to monitor the user or the environment in which they live. The solution must be modular so that it is easily customizable to the user's needs. Moreover, the solution must seamlessly integrate into the user's home and be easy to use. Various factors must be measured including air quality, ambient conditions, medication adherence, and thermal imaging. This information must be stored and sent to the cloud so it can be used in conjunction with other user data to monitor the user's health.

5.6.1 System Architecture

Our system uses a star architecture with a central unit, the base station, that is responsible for cloud communication of data from the sensor substations. The sensor substations will collect data from the rooms they are in and send that data to the base station via Bluetooth.

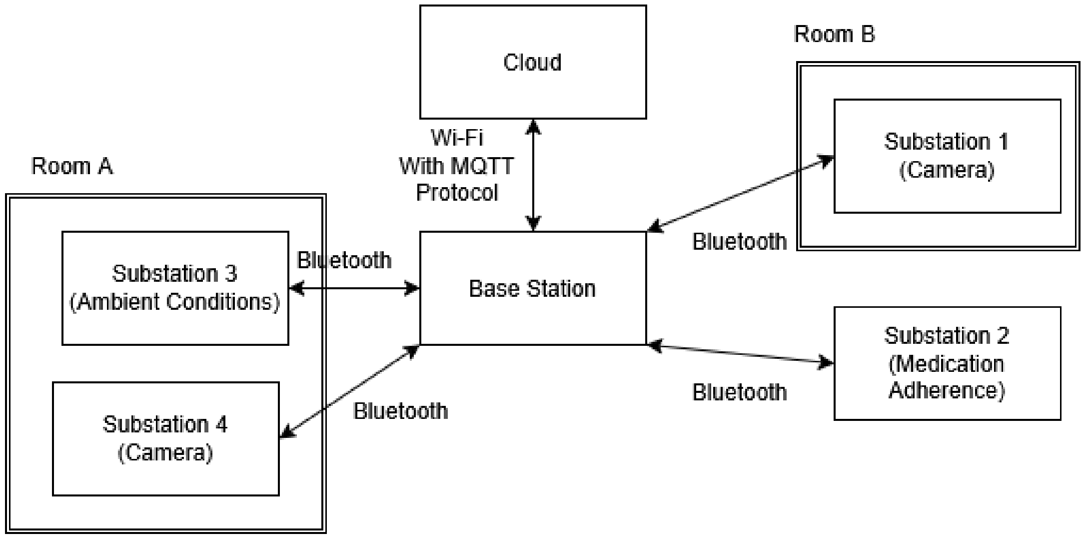


Figure 5.6.1.A: Environmental Subsystem Architecture

To further break it down the base station will have a more powerful processing unit so it can take care of some of the data computation. It will be able to display to the user and will be using both Wi-Fi and Bluetooth wireless connections. Due to the complexity of the base station it will be plugged into the wall for power.The substation will be a lower-powered device using a simple microcontroller to read data to send to the base station.

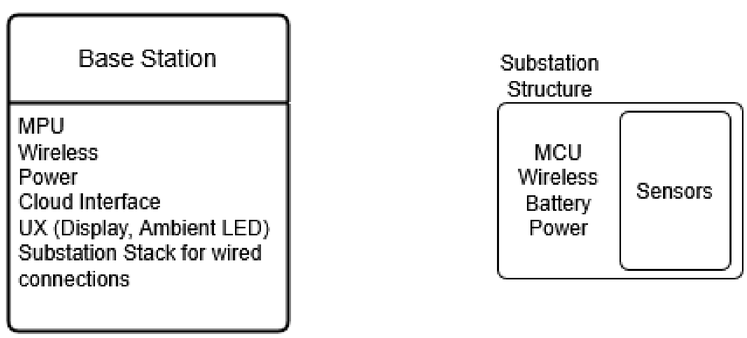


Figure 5.6.1.B: Base station and substation design

5.6.2 Initial Research

5.6.2.1 Main Computer

We are building a multi-nodal system with at least one base station and multiple sub-stations. And it should be powerful enough to handle communication with other substations and process their data before sending it to the cloud. It’s preferred to have built-in wireless connections and ample computing power, while keeping a small footprint so we are more flexible with the design. The main computer may not use active cooling to provide better noise performance to the target user. Other less important factors include cost, ease of usage, and the amount of (community) support.

5.6.2.2 Micro-Controllers

Each submodule in our monitoring network requires a microcontroller capable of interfacing with sensors and managing wireless communication. The microcontroller in each unit is responsible for reading sensor data, processing the measurements, and transmitting this information to the base station using either WiFi or Bluetooth connectivity, depending on the implementation. To select the most suitable microcontroller for our application, we compared several options across key parameters including wireless capabilities, power consumption, ADC features, and cost. We focused our research on the following microcontrollers: Arduino UNO R4, STM32-WB55RG, Raspberry Pico and ESP32.

	Arduino UNO R4	**STM32 WB55RG**	**Raspberry Pico**	**ESP32**
Built-in WiFi	Yes	No	Yes	Yes
Built-in Bluetooth	Yes	Yes	Yes	Yes
Price	$27.50	$23	$6	$10
Memory (RAM)	32KB	256KB	264KB	512KB
Form Factor	Medium	Small	Small	Small
OS Supported	No	FreeRTOS	No	No
Number of ADC	6	30	4	16

Number of GPIO	14	58	28	34
SPI	Yes	Yes	Yes	Yes
I2C	Yes	Yes	Yes	Yes

*Table 5.6.2.A: **Micro-Controller** Options Product/Feature Matrix*

5.6.2.3 Wireless Protocols

In designing the environmental subsystem's communication architecture, we conducted an analysis of both WiFi and Bluetooth technologies for two critical communication paths: the connection between sub-modules and the base station, and the link between the base station and cloud servers. WiFi offers high bandwidth data transmission with reliable wall penetration and a range of up to 100 meters, though it requires significant power consumption - a characteristic that requires consideration for battery-operated devices. Bluetooth, while providing a more moderate bandwidth of 2 Mbps, delivers comparable wall penetration and range to WiFi while operating with notably higher energy efficiency.

5.6.2.4 Air Quality Sensor

Some of the main relevant factors for determining the quality of air are volatile organic compounds (VOCs), noxious chemicals, and particulate matter (PM). When it comes to general ambient conditions that could affect the user there is also humidity, lighting, and noise level. Once there were specific metrics assigned to us for monitoring postpartum issues, we researched sensor packs that could measure multiple of the metrics we were interested in.

The air quality sensors will need to be placed in such a way that they are open to the room, which means they may not have access to a wall outlet. Therefore, we researched light-weight microcontrollers that could be powered off battery. Below is the product feature matrix of some of the sensors we explored.

	MQ-135	**Grove Air Quality Sensor v1.3**	**Enviro+ Air Quality for Raspberry Pi**	**Adafruit PMSA003I Air Quality Breakout**	**SparkFun Air Quality Breakout**

Precision	10-100 ppm	Mid	Mid	Mid	Ppm, ppb
Characteristic	NH_3, CO_2, NOx	Toxic Gas (Carbon Monoxide, Formaldehyde, Alcohol, Acetone, Thinner, etc.)	Mixed(Temperature, Pressure, Light, Humidity, Gas, MEMs Microphone, & Noise level	Particulate Matter (PM1.0, PM2.5, PM10.0)	TVOC, equivalent CO_2
VOCs?	Yes	Yes	Yes	No	Yes
Noxious Chemicals?	No	Yes	Yes	No	No
Particulate Matter?	No	No	No	Yes	No
Humidity?	No	No	Yes	No	No
Ambient Temperature?	No	No	Yes	No	No
Pressure?	No	No	Yes	No	No
Calibration	Required	Not Required	Not Required	Not Required	Not Required

*Table 5.6.2.B: **Air Quality** Sensor Options Product/Feature Matrix*

5.6.2.5 Camera

To enhance the detection and monitoring of Deep Vein Thrombosis (DVT) in postpartum patients, we have integrated an infrared thermal imaging system into the environmental subsystem. Specifically, we have chosen the **Seek Scan Thermal Camera** (https://shop.thermal.com/Seek-Scan) for its high accuracy and suitability for medical applications. This camera features an accuracy of **±0.3°C**, ensuring reliable thermal measurements that align with clinical needs for detecting temperature asymmetry between the legs—one of the key indicators of DVT.

	Seek Scan
Resolution	206 × 156 pixels

Temperature Accuracy	±0.3°C (with blackbody), ±0.5°C (without)
Field of View (FOV)	30° x 22°
Thermal Sensitivity (NETD)	<50 mK
Interface	USB
Power Supply	USB-powered
Frame Rate	9 Hz
Output Format	MJPEG, YUY

Table 5.6.2.C: Characteristics of ***Seek Scan Thermal Camera***

Rationale for Using Thermal Imaging

DVT diagnosis is critical, as delays can lead to severe complications such as pulmonary embolism. Traditional diagnostic methods like Color Doppler Ultrasound (CDUSG) require trained personnel and specialized equipment, making them less accessible for continuous monitoring. Recent studies, such as those outlined in *The Use of Infrared Thermal Imaging in the Diagnosis of Deep Vein Thrombosis* (Kacmaz et al., 2017), highlight the potential of thermal imaging as a **pre-screening tool**. Research findings indicate that the mean temperature of a DVT-affected leg is significantly higher compared to the non-affected leg, with a mean difference of approximately **1.69°C** in DVT patients, compared to an insignificant **0.13°C** difference in healthy controls (Kacmaz et al., 2017).

5.6.2.6 Medication Adherence

Post-partum patients may be prescribed various medications by their doctor. Some medications must be taken at certain times or after certain activities. Typically patients may have a pill organizer or reminders on their phone reminding them to take their pills.

We want to remind users to take their medications while also tracking whether or not they have actually taken them. The small size and lightweight nature of pills makes them difficult to detect using automated methods. A high level of precision is required. We investigated three different methods of detecting pill presence: weight sensors, pressure sensors, and infrared emitter-receivers.

Weight and pressure sensors have the same issue when it comes to detecting something as light as pills. Pills typically weigh less than a gram so an appropriate sensor would need to detect differences of at least half grams in order to be effective. There are not any good options for sensors this precise that are affordable and would fit within a pill organizer design.

To better suit our form factor and budget goal we decided to go with a reflective infrared sensor. It is an emitter receiver pair that detects the amount of infrared reflected back to the receiver. We expect this to work since it can detect small changes in the infrared which will be affected by the presence of the pill that the infrared is reflecting off.

5.6.3 Final Design Decisions

5.6.3.1 Base Station

The Base Station consists of a Raspberry Pi 5 and its peripherals, as shown in Figure *5.6.3.A*. It has a UI system including some user input (buttons or touch screen) and a display for the user to set up the substations or show the current status. This could be further extended to some entertainment or calming system. The Base Station also contains an RGB light strip to tell the user a certain important status quickly. Since the thermal camera is designed to be used relatively infrequently and requires more power, we integrate it into the base station. Lastly, there will be a set of pins or contacts on the external of the casing for the Raspberry Pi to communicate with sub-stations during the setup stage.

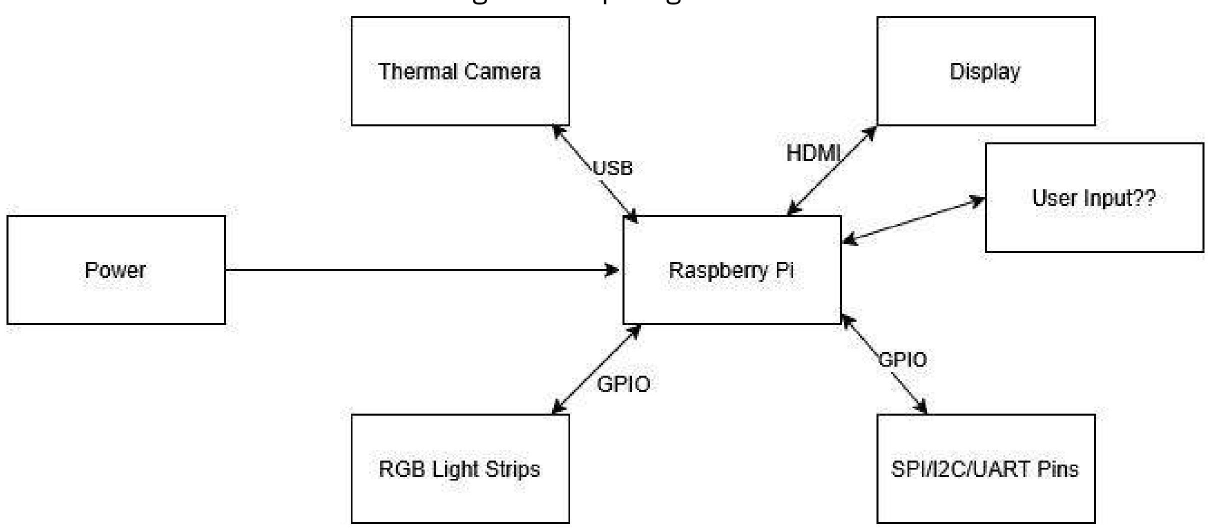


Figure 5.6.3.A: Components of the Base Station

5.6.3.2 Ambient Condition Subsystem

The Ambient Condition Subsystem will come in two types based on the sensors we selected, general ambient conditions, and air quality General ambient conditions will include temperature, pressure, light, humidity, and noise level. Air quality will include PM1.0, PM2.5, PM10.0, and CO2.

The general ambient conditions will be measured by the Enviro+ breakout board connected to a Raspberry Pi Zero 2W. More details about this configuration can be found in Figure 5.6.3.B. It will be battery powered so it can remotely operate and communicate its data back to the base station via Wi-Fi. The Enviro+ is made such that its pins align with the pins on the Raspberry Pi Zero 2W and the two boards press fit together.

This subsystem will be running Micropython firmware and will use the Pimoroni Envrio+ python library to read and interpret the data from the various sensors on the Enviro+ breakout board. The Enviro+ will measure temperature, pressure, humidity, light, proximity, noise, NH3, and reducing and oxidising gases. All of these data points will be displayed on the mini LCD on the Enviro+ and the user can cycle through the displays by hovering their finger over the proximity sensor. The temperature, humidity, light, and noise data will be sent back to the base station for analysis.

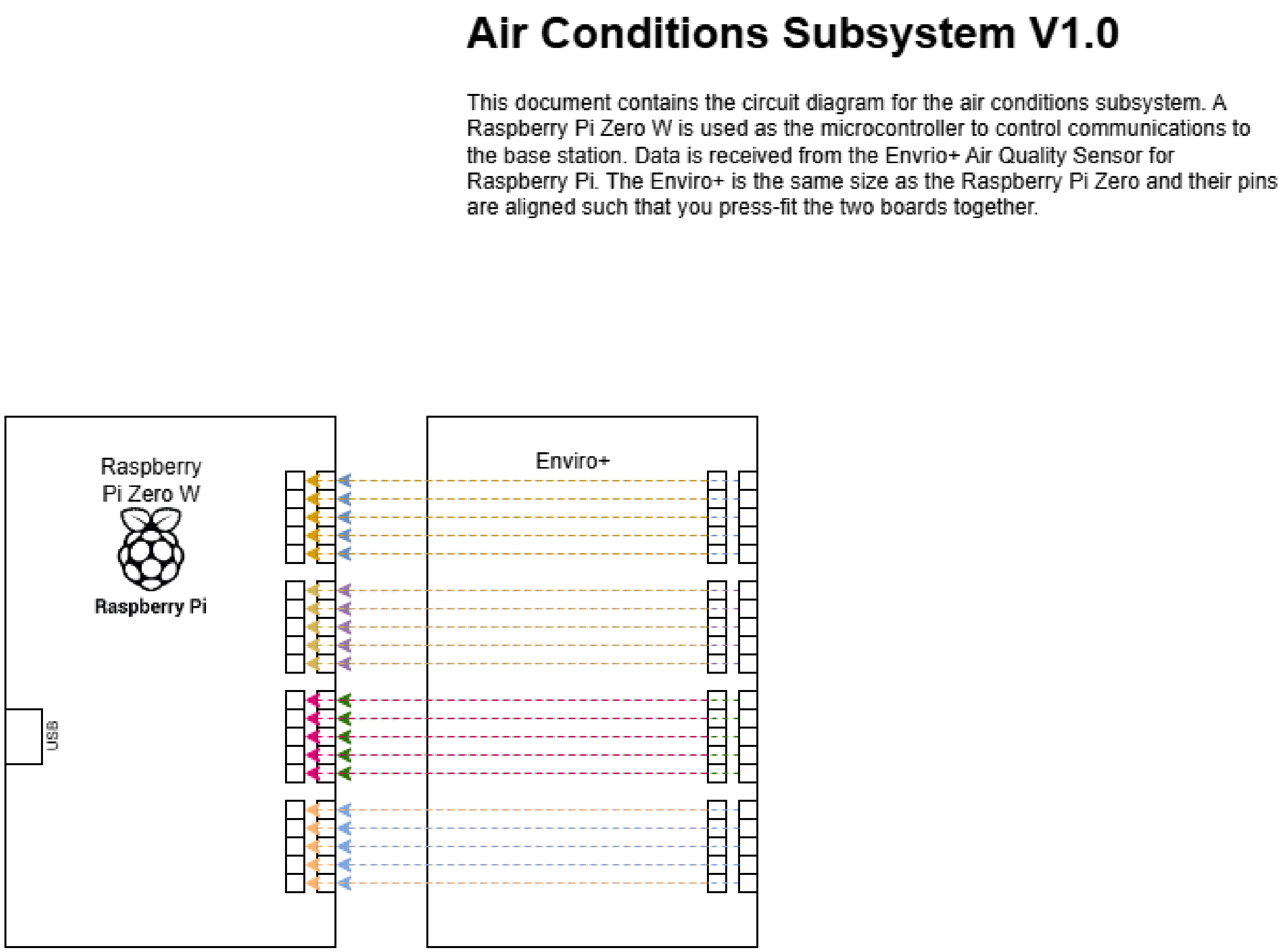


Figure 5.6.3.B: ***Air Conditions Subsystem*** *Electrical Diagram*

The air quality will be measured by an Adafruit PMSA003I and a MQ-135 connected to a Raspberry Pi Pico 2W that is powered by a battery. The subsystem will be using CircuitPython firmware. The PMSA003I measures particulate matter (PM1.0, PM2.5, PM10.0) and the MQ-135 measures CO2 concentration 5 times every second. Data will be communicated back to the base station via Wi-Fi. More details about the electrical configuration of the air quality system can be found in Figure 5.6.3.C.

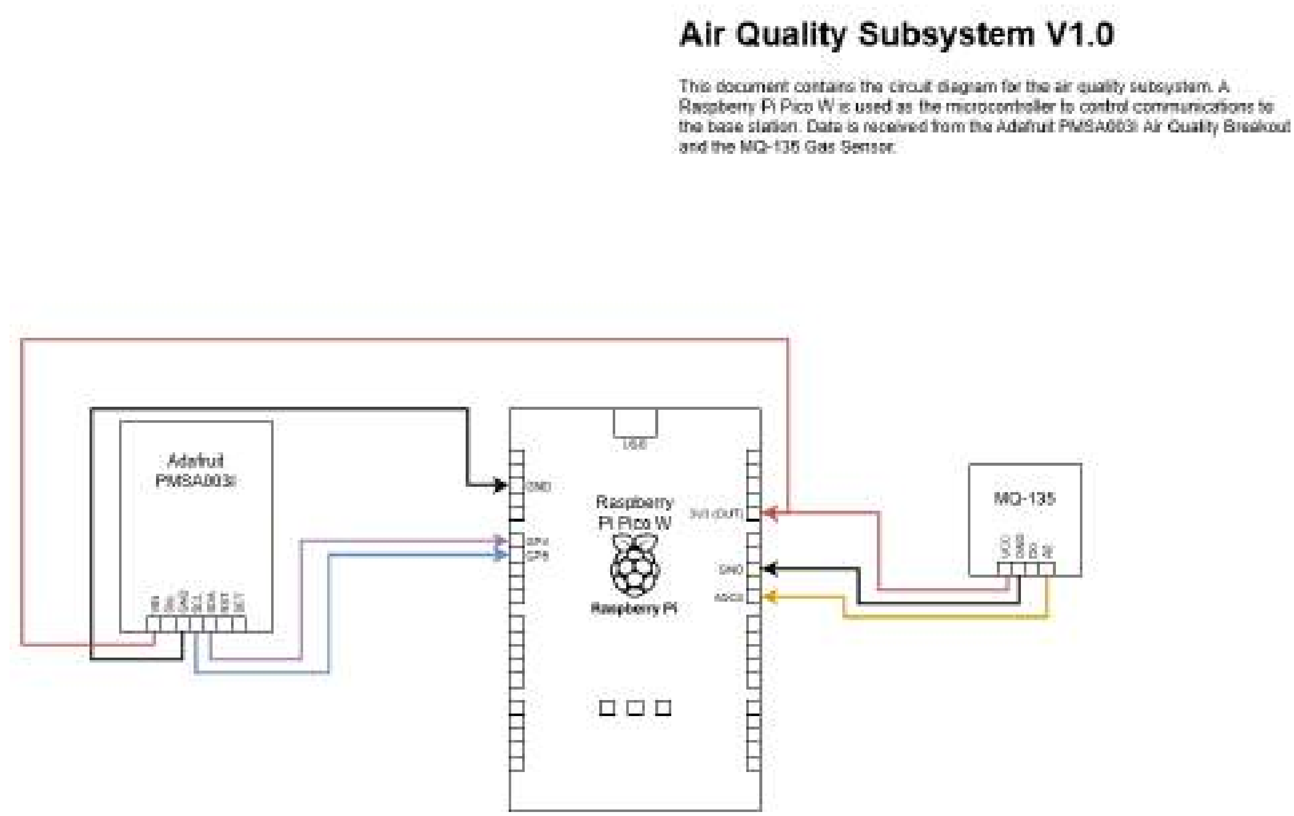


*Figure 5.6.3.C: **Air Quality Subsystem** Electrical Diagram*

5.6.3.3 Camera Subsystem

The **Seek Scan Thermal Camera** will be installed within the base station, where it continuously captures thermal images of the patient's lower limbs. The system will compare the thermal data against a baseline to identify temperature asymmetries indicative of potential DVT. **Key features include:**

- **Automated temperature differentiation:** Detects variations between the left and right legs.
- **Data integration with cloud services:** Enables remote monitoring and early intervention.
- **User alerts:** Sends notifications if abnormal temperature differences are detected, prompting further clinical assessment.

The captured thermal images will be analyzed using an algorithm that calculates **mean leg temperature differences**. According to Kacmaz et al. (2017), a **sigma threshold of 0.33** is an optimal indicator for distinguishing between healthy individuals and DVT patients. If a significant **temperature difference exceeding this threshold** is detected, the system will flag the data for medical review.

Supporting Data and Figures:

Group	Leg temperature with high temperature (°C)	Leg temperature with low temperature (°C)	P[a]
Control Group	32.74±0.86	32.61±0.87	0.5903
Study Group	33.62±1.22	31.93±1.44	<0.0001

Table 5.6.3.D: Mean temperature values for DVT patients and controls (Kacmaz et al., 2017).

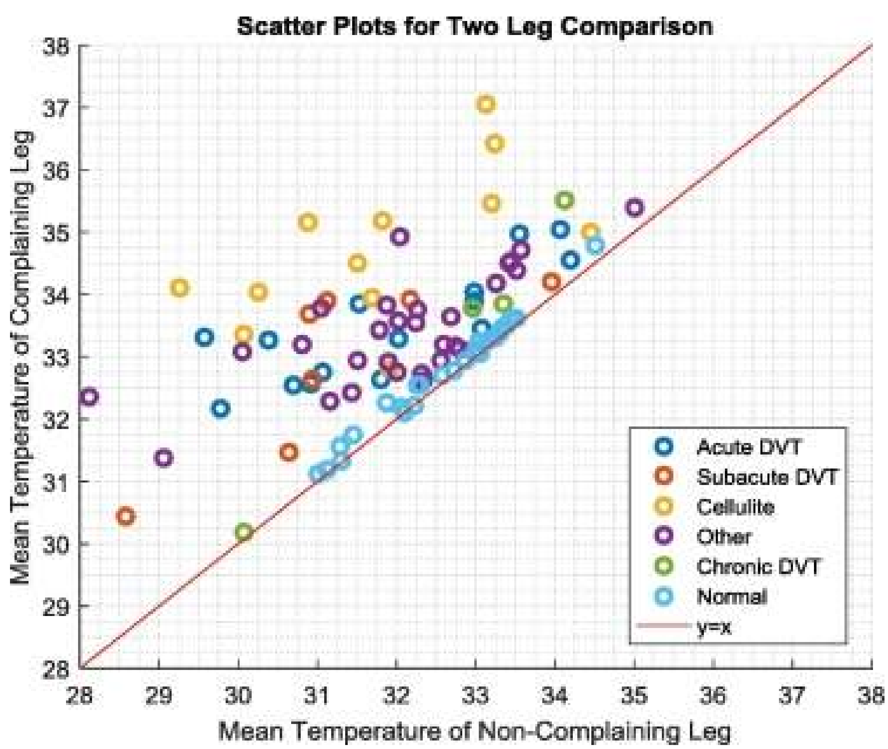


Figure 5.6.3.E: Scatter plot comparing mean leg temperatures of DVT and non-DVT cases, showing temperature asymmetry (Kacmaz et al., 2017).

Integrating the Seek Scan Thermal Camera into the postpartum care monitoring system provides a **non-invasive, real-time method for detecting DVT symptoms**, reducing reliance on costly and time-consuming ultrasound screenings. This subsystem enhances patient safety by facilitating **early detection** and **timely medical intervention**.

5.6.3.4 Medication Adherence Subsystem

The medication adherence subsystem will be a smart pill organizer that consists of an array of reflective infrared sensors and a microcontroller that connects to the base station via bluetooth. These sensors continuously scan the compartment, detecting the presence or absence of medications by measuring infrared light reflection off the pills' surfaces.

The smart pill organizer's core processing is handled by an STM32-WB55RG microcontroller, specifically chosen for its integrated Bluetooth functionality which eliminates the need for a separate wireless module. Each infrared sensor's analog output is routed to a dedicated ADC channel on the microcontroller, allowing for continuous monitoring of pill presence across all compartments. The microcontroller samples these ADC inputs at regular intervals, converting the analog reflection readings into digital values that indicate whether medications are present or have been removed.

The microcontroller processes this raw sensor data and comes up with a status for each of the sensors that indicates the presence or absence of the pills. We are currently testing different placements of the IR sensor to improve the detection of the medication inside the compartments of the organizer.

The current status is packaged and transmitted from the STM32 to the base station via Bluetooth Low Energy (BLE) radio every minute. The base station is able to connect to the microcontroller by the use of the Python Bleak library. The base station then forwards a message to the cloud that indicates whether or not the patient has taken the medication.

5.6.4 Final Implementation

For the final implementation of our prototypes we integrated our hardware with 3D printed casing for most of our substations and the base station. We connected our substations to the base station and demonstrated live data collection and transmission to the base station. You can view our code here.

5.6.4.1 Base Station

The base station serves as the central hub for all connected environmental sensor modules (substations), a thermal camera, and the user interface. Its primary responsibilities include processing data from sensors and the camera, displaying information locally, and forwarding data to the cloud for further analysis.

At the heart of the system is a Raspberry Pi 5, which acts as the main processor for sensor communication, data handling, and UI rendering. Paired with a 7-inch touchscreen, the Pi enables users to configure network settings, manage substations, and view real-time environmental metrics. To streamline the setup process, the base station features a pogo-pin interface for quick pairing and initialization of new substations. Additionally, a WS2812 addressable LED strip encircles the display, visually indicating system status or other basic information. The integrated Seek Scan thermal camera can be activated from the dashboard and displayed in a separate overlay window, while data from all sensors is rendered live using tile-based UI components.

The graphical interface is built using PyQt 6, offering a responsive and visually cohesive user experience. The base station also functions as an MQTT broker, facilitating wireless communication with substations over Wi-Fi.

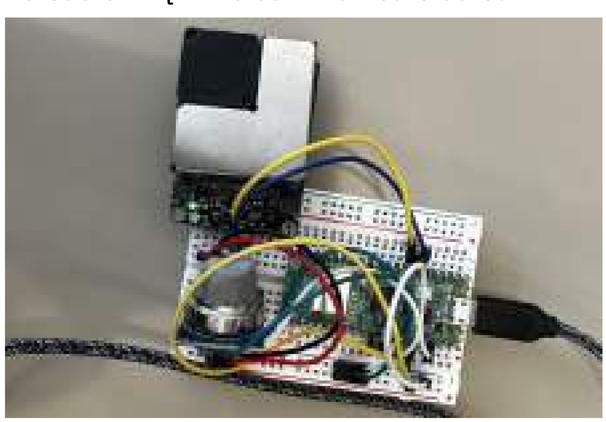
When a substation is connected via UART, and the user initiates setup, the system performs a handshake protocol. Upon successful pairing, the base station configures the substation with appropriate settings (e.g. Wi-Fi credentials) and begins subscribing to its data stream.

5.6.4.2 Ambient Conditions and Air Quality Substations

Our substation for measuring ambient conditions through the Enviro+ was integrated with a battery pack to fit into a 3D printed casing. There is a port to plug in the battery pack to recharge it. The case has pegs that slot into the screw holes of the Enviro+ breakout board and the Raspberry Pi Zero 2W board. The lid and the back of the case pop on and off for easy access to the battery pack and hardware components.

When closed, this casing allows for the screen to be viewed and operated as well as allowing the sensors to have access to the air. It connects to the base station over the Wi-Fi network and sends json formatted data using MQTT.

For measuring the quality of the air regarding C02 content and particulate matter we made another substation using a Raspberry Pi pico 2W. We prototyped this substation on a breadboard. It also connects to the base station over Wi-Fi, but unlike the ambient condition substation it uses HTTP instead of MQTT to communicate data.



5.6.4.3 Medication Adherence Substation

The final version of the medication adherence module consists of a 3D-printed pill organizer with fourteen separate compartments, two designated for each day of the week. Each compartment features a hole on one side housing an Infrared (IR) reflective sensor. Wiring from the sensors is concealed within the organizer's casing and connects to the microcontroller's Analog-to-Digital Converter (ADC) pins. The microcontroller samples these ADC inputs at a frequency of 10 Hz.

Bluetooth connectivity enables the microcontroller to transmit sensor data to a base station (a Raspberry Pi). Client software running continuously on the Raspberry Pi requests the latest sensor readings from the microcontroller, also at a rate of 10 Hz.

To establish baseline values for empty compartments, a calibration sequence runs when the client software first connects to the microcontroller via Bluetooth. This sequence requires all compartments to be empty and closed. It involves requesting and receiving sensor data for 20 seconds, then calculating and storing the average reading for each individual IR sensor.

These calibration averages define the sensor values corresponding to an empty compartment. During normal operation, if a sensor's reading consistently exceeds a predetermined threshold above its calibrated average, the system concludes that medication is present in that specific compartment.

The base station averages the ten readings received *for each sensor* every second. This resulting averaged value for each compartment is used to determine the presence or absence of medication. This status data is then forwarded to a cloud platform for storage. This stored information can later be used to determine if medication reminders should be sent to the patient.

6 Design Process

Summary Work Logs

HCI

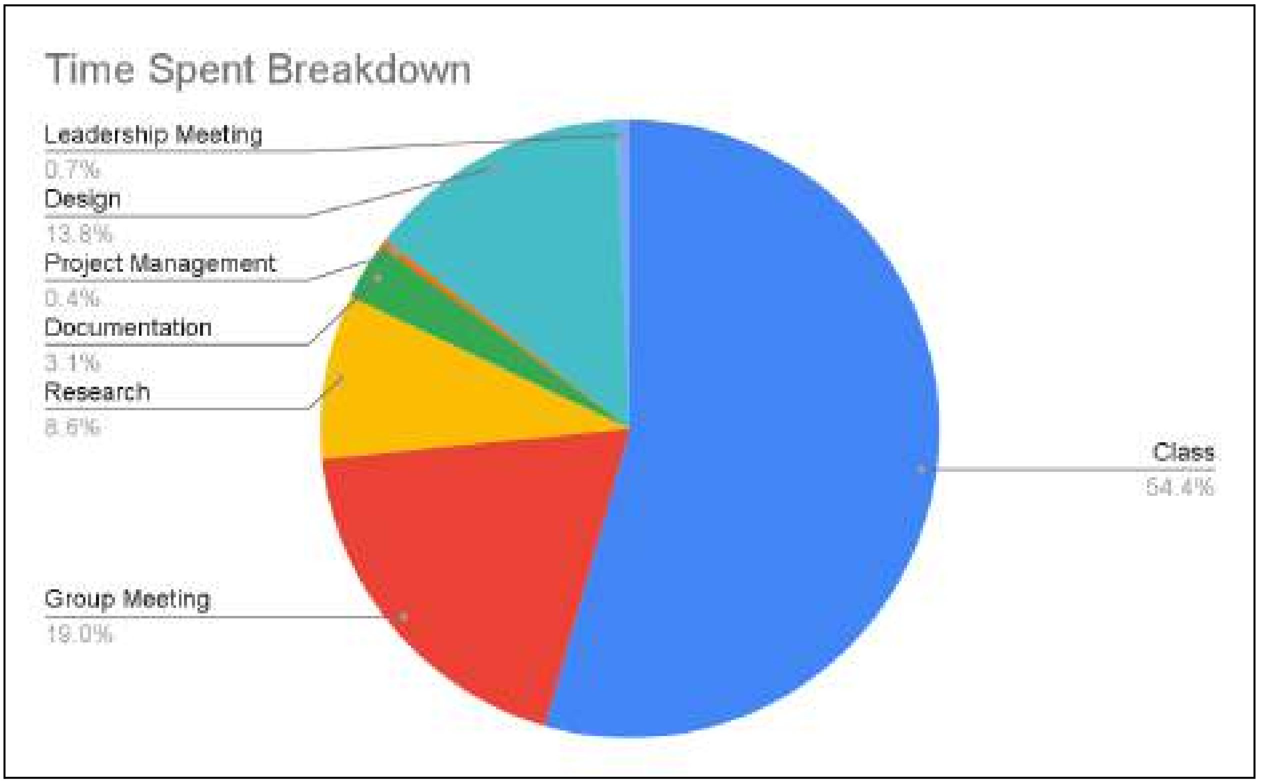
Team member	Contribution
Service Design	
Brenna Lindblad	•
Alyssa Ogle	• Co-designed the onboarding component of the app • Researched into data privacy protection and ethical data collection by mHealth applications ○ Finding out what is the gold standard applying that to the design of our app
Zoe So	•
	•
	•
	•
	•
	•
	•
Wearable HCI	
Ray Xia	Phase 1: • Researched existing wearables related to health monitoring

	- Collaborated on producing materials for the draft visionary scenario - Drafted system architecture focusing user interaction with front-end - Designed functional requirements and interaction flow of the wearable Phase 2: - Determined hardware interface needed on the works-like prototype - Ideated different forms of looks-like prototype - Designed user interaction cues and feedback, including visual and haptic patterns and button presses - Worked in close liaison with Wearable Hardware Team to align on system output and interaction design - Directed and edited video presentation
Chenyue Shen	Phase 1 - Researched wearable medical technologies, focusing on feasibility and user needs. - Brainstormed interaction flows and device functions with cross-team input. - Attended leadership meetings and shared team progress. - Sketched potential device designs and discussed key features. - Evaluated feasibility of sensors, battery life, and display options. Phase 2 - Researched wearable medical devices and identified user needs and feasibility constraints. - Evaluated different display technologies and material options in collaboration with the Wearable Hardware team - Sketched initial design concepts and iteratively refined product appearance based on team input. - Modeled the product exterior using Autodesk 360 to visualize form factor and usability. - Designed logos and constructed a rough product mockup to represent functional modules.

	● Developed UI prototypes in Figma and explored various device functionalities. ● Defined core messaging and feedback mechanisms for the device.
Leyi Han	Phase 1 ● Brainstorming wearable devices for post partum care ● Investigated sensors and technologies available for wearable health devices ● Sketching out wearable prototypes ● Creating interaction flow of wearable device based on different health conditions and user responses Phase 2 ● Sketching designs for wearable device ● Digital modelling of wearable prototype using Rhino, creating four design iterations ● Researching materials and prototyping methods ● 3D printing 1st wearable prototype with TPU material ● Prototype testing and filming ● Evaluating the functionality, ergonomics and material selection of the prototype based on user testing Phase 3 ● Sketching designs for the final prototype based on previous feedback and evaluation ● Modelling the prototype in Rhino, creating three iterations for the final design ● Selection of readymade wristband and research of prototype components ● Modelling connection joints to PCB for working prototype ● Creating 3D files for filament and resin 3D printing in stl ● Refining 3D prints and assembling different parts for the final prototype

Note: During Phase 1, HCI was divided up into 2 subteams, Provider/User HCI and Wearable HCI. Given the multiple directions of work in parallel anticipated in the upcoming phases, there will be 5 subteams under HCI moving forward: Environmental Design, UI/Visual Design, Data Visualization, Service Design, and Wearable HCI.

Hourly Task Breakdown



Provider/User Software

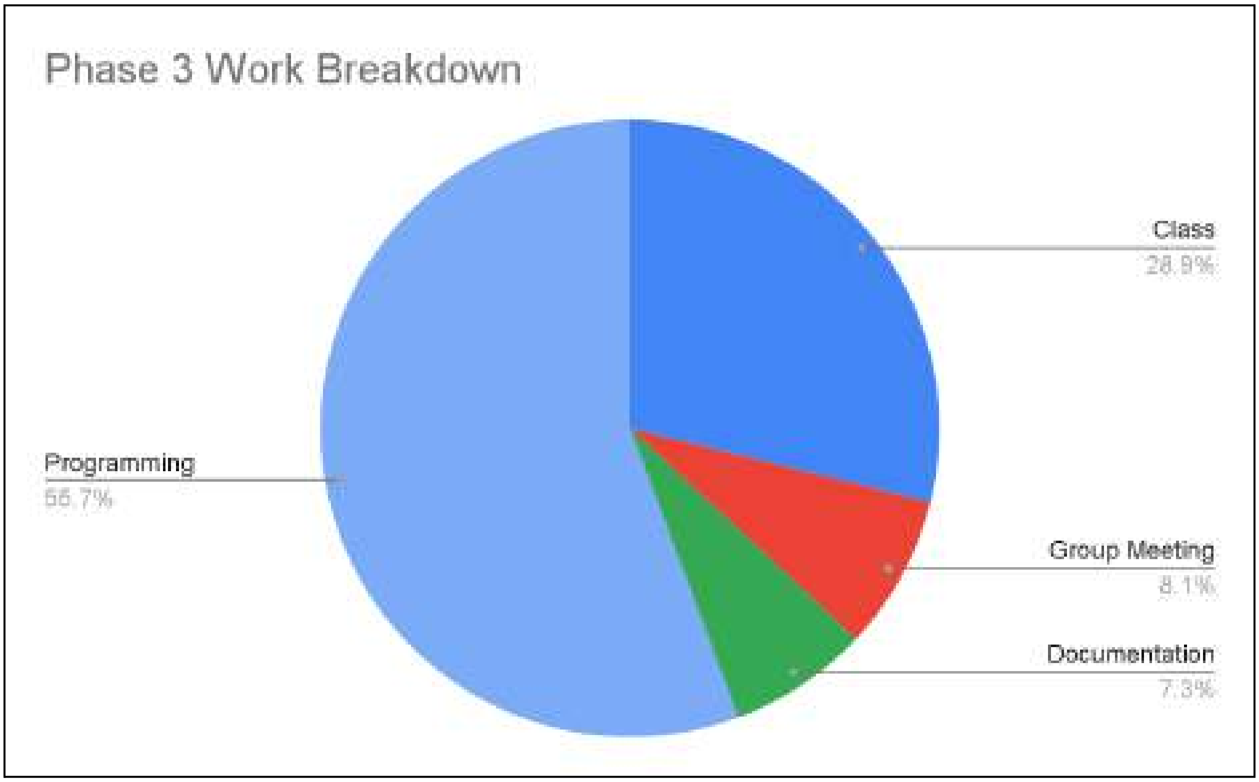
Team member	Role	Contribution
Website Application		
Fiona Fisher	Team Lead	Phase 1 • Research on web backend/frontends for product features/matrix. • Made NodeJS slide for Phase 1 presentation. • Condensed product/feature matrix research for Phase 1 report. • Calculated hour breakdown for Phase 1 report. • Consolidated teammates' thoughts into readable text for the report. • Formatting of Phase 1 report and product features/matrix. Phase 2 • Attended the meeting with Honda reps on behalf of the

		software team. Prepared a short presentation for them on our progress. ● Prepared part of the frontend framework for the website and demoed it at the Phase 2 demo. ● Prepared slides and presented the provider (website) side of the software for the Phase 2 presentation. ● Programmed some dummy frontend web frames (home page, patient search page) based on the HCI wireframes. Phase 3 ● Built core website frontend pages: Patient Search, Alert Box, and Past Report view. ● Handled database interactions for the cloud backend, including fetching and displaying patient information and reports. ● Contributed on backend API design and helped transition from local testing databases to the real cloud database. ● Attended team lead meetings to coordinate with other teams and define final deliverable expectations. ● Created slides for the final presentation and presented them to Honda reps.
Tassy Chen		Phase 1 ● Researched and consolidated mobile backend/frontends for product features/matrix. ● Made Database/Cloud slide for Phase 1 presentation. ● Presented at Phase 1 representation for the team. ● Wrote the presentation script as used in the report
Yuchen Dai	Editor	Phase 1 ● Researched web app frontend and backend and host choices, and listed their pros and cons ● Contributed to the slides on system architecture diagram ● Researched the possibility of using Vercel for our project and contributed to questions to ask other teams ● Set up the web app development environment and assisted teammates to set up everything Phase 2

		• Programmed the frontend and backend for the user authentication • Documented current endpoints and wrote a README file for teammates to collaborate and catch up with the progress • Helped teammates set up databases on the local end and solve dependency issues • Communicated with the cloud team to settle the backend part • Asked for reimbursement of Anima • Drafted and improved the phase 2 report Phase 3 • Coordinated and supported cloud backend deployment efforts • Developed a patient search screen (was cut due to old design) • Wrote the phase 3 report, consolidating team inputs and documenting project progress.
Mobile Application		
Alice Tang	Presenter	Phase 1 • Attended team lead meetings to help narrow down initial design ideas. • Facilitated inter-team communication. • Researched app frontends for product feature matrix. • Made Flutter slide for Phase 1 presentation. • Organized and led team meetings. • Set up the development environment for the flutter app. Phase 2 • Configured set up for app • Integrated authentication from the backend of the website • Programmed screens with Flutter according to figma prototypes given by the design team Phase 3 • Developed major mobile app frontend components including Registration/Login, Landing page, Bluetooth setup screen, Notifications, and Profile page. • Implemented Bluetooth functionality and integrated API calls for backend communication.

		Assisted in testing mobile app features on Android emulator and real devices.
Yiyun Wei		Phase 1 - Researched mobile app development languages for product features/matrices - Researched and set up Figma to code helper tools - Made Typescript/React slide for Phase 1 presentation - Communicated with other subgroups regarding project specifications Phase 2 - Attended weekly leaders meetings - Created screens in app with Flutter according to specifications given by the HCI team - Discussed with Cloud + Wearable Hardware subteams regarding integration between software and other components of the system Phase 3 - Built mobile app screens for Mood tracking, "My Health" dashboard (and its subpages), and Messages. - Helped define the app's purpose, technology stack, and key functionalities for final documentation. - Contributed to system design and app user experience improvements.

Hourly Task Breakdown



Cloud/Data/Communication Systems

Team member	Role	Contributions
Ariel Wu		• Developed EC2 automation tools to manage cloud resource • Conducted in-depth research on AWS EC2, DynamoDB, S3 buckets, and AWS IAM usage to optimize cloud operations • Joined in writing Python scripts to parse DynamoDB for data processing and integration • Integrated MQTT through AWS IoT Core for real-time communication between devices and the cloud • Collaborated with frontend and backend APIs on the cloud, ensuring public access for external users. • Collaborated with the Provider Software Deployment

<table>
<tr><td></td><td></td><td>team to create a script for easy EC2 connection and app startup.
Prepared for Phase 2 demo presentation, highlighting key developments and progress
Wrote DynamoDB scripts for automated data upload and download
Built S3 scripts for uploading and retrieving large sensor files
Managed DynamoDB and S3 resources for clean and efficient storage
Refined scripts for general team usage and easier system integration
Ensured AWS infrastructure scalability and high availability
Developed Lambda functions for health anomaly detection and real-time SMS alerts
Prepared and contributed to the Phase 3 final presentation and report</td></tr>
<tr><td>Enze Yuan</td><td>Editor</td><td>Participate in class
Align with other User Software, Env Intelligence and hardware team for their cloud needs and align system design
Do final testing on connection between cloud and Env hardware, Software, Intelligence team
Attend group meetings to discuss weekly progress
Assign weekly tasks to team members
Communicate with Software team and Hardware team on integration and the cloud service connection to their product
Migrate EC2, DynamoDB as well as S3 to another account with teammates due to cost limit
Set up cloud Environment
Prepare EC2 environment, configure packages and necessary tools
Write auto deployment bash script for EC2
Configure DynamoDB on Index configuration</td></tr>
</table>

		and test them with requests from Provider software web app • Connect Env Hardware team to Cloud ○ Collaborate with teammates on setting up IAM roles and policies of DynamoDB and S3 ○ Configure the connection between hardware team and DynamoDB ○ Configure the IOT core for routes and policies • Integrate Cloud with Software ○ Debugging Provider Software and User Software on backend development and troubleshooting ○ Deploy Provider Software and User Software to EC2 ○ Migrate Provider Software backend code from postgresql to DynamoDB ○ Configured EC2 environment for deployment ○ Deploy both frontend and backend to cloud that allows public access • Build data notification system for intelligence team ○ Set up an event-driven lambda function using python, automatically triggered by r updates to entries in DynamoDB. ○ Notifications are dispatched to the Intelligence team through multiple communication channels, including email alerts via Amazon SES,and SMS messages simulated using Twilio's messaging service. • Prepared and presented Phase 3 system architecture ○ Prepared presentation slides ○ Record the demo for software deployment on cloud ○ Record the demo for connection from Env hardware to DynamoDB ○ Showcased EC2 tasks we accomplished and future considerations
Fu-Cheng Pan	Presenter	• Participated in class and team meetings ○ Identified AWS credit shortage, researched

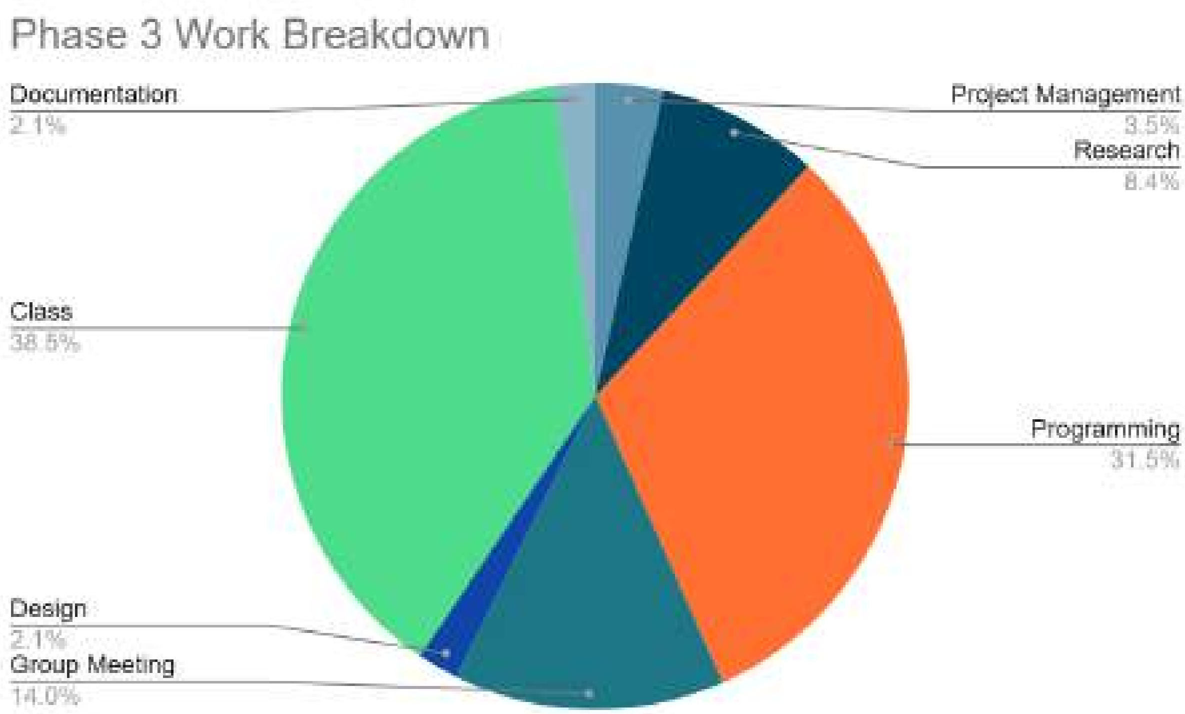
		service costs, and transferred existing infrastructure into a teammate's account. ○ Created and configured cross-account IAM roles to grant teammates access to our AWS environment. ○ Aligned on Phase 3 objectives and timeline during group discussions. ● Built and enhanced our data ingestion pipelines ○ Discussed subteams' requirements for image storage, then authored a Python prototype to upload pictures into S3 and record the resulting URLs in DynamoDB. ○ Extended the environmental-data script to push sensor readings into DynamoDB, and verified end-to-end uploads with the hardware team's Raspberry Pi devices. ● Designed and tested an event-driven notifications system ○ Researched AWS Lambda integrations, then wired DynamoDB stream events to invoke serverless functions on new anomaly entries. ● Prepared and delivered Phase 3 deliverables ○ Drafted the final system-architecture diagram and detailed the S3 bucket integration slide. ○ Presented our Phase 3 demo focusing on S3 usage and notification flows to the Honda team.
Meihui Liu	Demo Lead	● Participated in class ● Researched strategies for migrating AWS data across multiple accounts due to credit limitations. ○ Alongside, researched the database migration ● Migrated the database to a new AWS account; created IAM roles for the intelligence team access and updated access scripts accordingly ● Researched in cloud image storage solution ● Facilitated communication between various subteams to clarify the general data flow from and to the cloud database; clarified technical dependency among each

		team to ensure a better collaboration
		● Participated the leaders meeting as a demo lead and representative of the cloud team
		  ○ Took notes about the progress of each teams
		  ○ Participated in the discussion about the expectation for the deliverables of each team
		  ○ Participated in the discussion about the progress of the cloud team
		  ○ Participated in the discussion about the final presentation contents and presentation flow
		● Contributed to presentation preparation by designing slides and participated in dry run
		● Delivered the final presentation (database portion)

Wei Liu	Team Leader	● Participated in class and team meetings
		  ○ Set up the AWS cloud environment to host webapps from the software subteam.
		  ○ Prepared phase 3 slides and presentation with group members.
		  ○ Interacted with other subteam members to advance the testing process to ensure our collaborative features are implemented correctly.
		● Participated in leader meetings
		  ○ Monitored the progress of subteams and discussed ways to enhance cooperation between groups.
		  ○ Assisted in integrating team scripts with other subteams' needs to ensure smooth collaboration.
		  ○ Summarized requirements from other subteams and assigned tasks to our group members.
		● Built up data transmission
		  ○ Accomplished sending text messages to users by using lambda function.
		  ○ Established direct connections from physical devices to DynamoDB to enable seamless data transfer.

		○ Set up S3 service to receive and store large images from users to the cloud

Hourly Task Breakdown



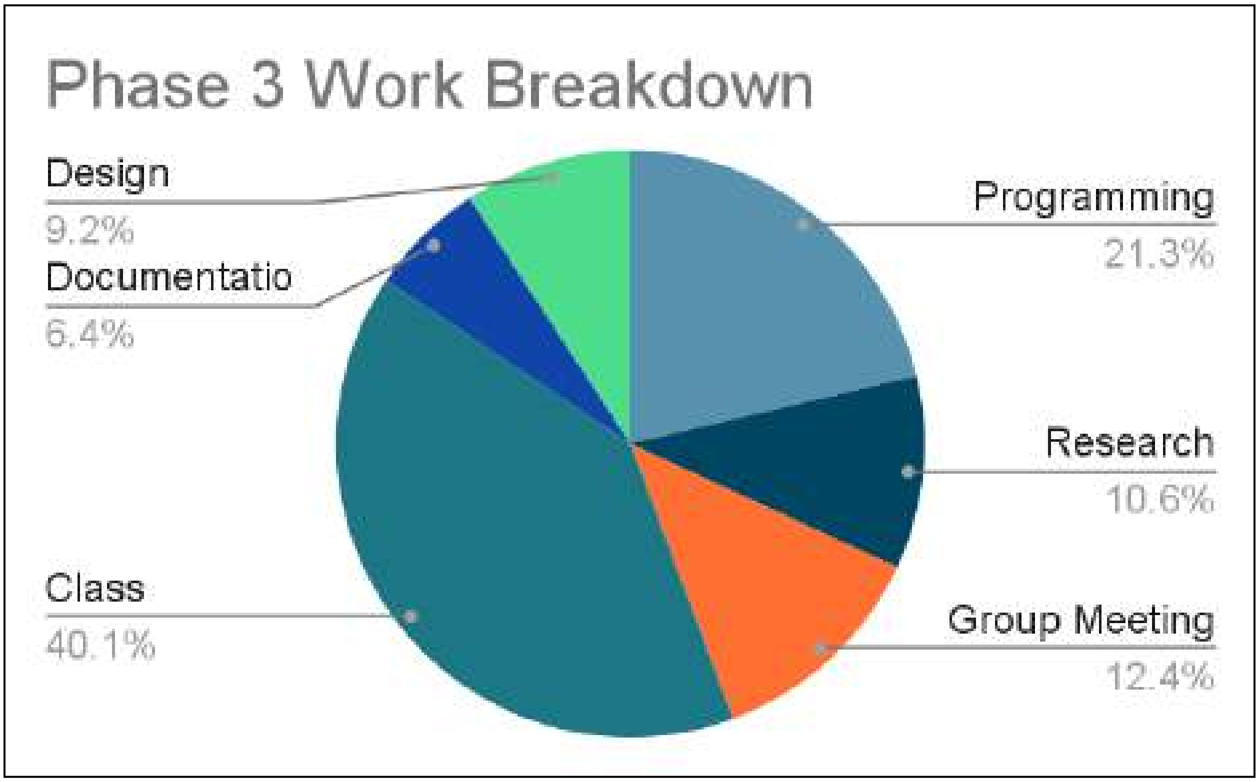
Intelligence

Team member	Role	Contribution
Aditi Narasimhan	Demo Lead	● Facilitated across all teams to make sure demo went smoothly ○ Helped user software teams and cloud teams interface, worked with hardware teams, etc. ● Added severity score structured context window using HCI's examples ● Reformatted output parsing due to modified context window ● Refactored all code into easily usable functions for

		better integration within our subteam
Ifeanyi Ene		● Helped work to integrate subteam's code with cloud team's database ● Worked on ML model (anomaly detection) code ● Worked on demo code ● Designed slides for Phase 3 presentation ● Presented at Phase 3 demo ● Wrote part of ML model (anomaly detection) section of the report
Linqi Zhang	Editor	● Designed, implemented and refined the whole data processing pipeline; ● Generated separate sample datasets of body temperature, blood oxygen, heart rate and blood pressure with certain patterns of anomalies, and wrote dataset descriptions; ● Communicated with hardware, software and cloud teams for discussing the sensor data types and the data points flow between teams; ● Cooperated with my teammates for any team work required; ● Attended classes and group meetings; ● Wrote the documentations including worklogs, reports, presentation scripts and the backup documents on my own end; ● Made the presentation deck and presented it for the presentations.
Xiang Chen	● Editor	●
Rory Cai	Leader	● Designed interval-based anomaly detection by aggregating sensor data over short time windows to improve stability and reduce noise. ● Implemented an LSTM Autoencoder to detect anomalies based on temporal patterns and variance in sensor sequences. ● Collaborated with the cloud team to successfully

		retrieve and store sensor data using DynamoDB. ● Developed a demo script that reads data from the cloud, runs ML-based anomaly detection, and writes notifications back to the database.
Phyllis	Editor	● Helped to design, implement and refine the whole data processing pipeline ● Generated separate sample datasets of body temperature, blood oxygen, heart rate and blood pressure with certain patterns of anomalies, and wrote dataset descriptions ● Collaborated with cloud team to understand process of automating data processing scripts and help in integration between Intelligence and cloud team for script automation ● Collaborated with hardware sensor team to finalize and confirm sensors, formulate proper data ranges, and communicate updates between hardware and Intelligence team ● Helped resolve version control issues and maintain Github pushes and merges during data processing ● Create sample data tables to present during showcase of generated sample data ● Research on sensor types and possible values such as measurements and ranges, to determine anomaly values in data

Hourly Task Breakdown

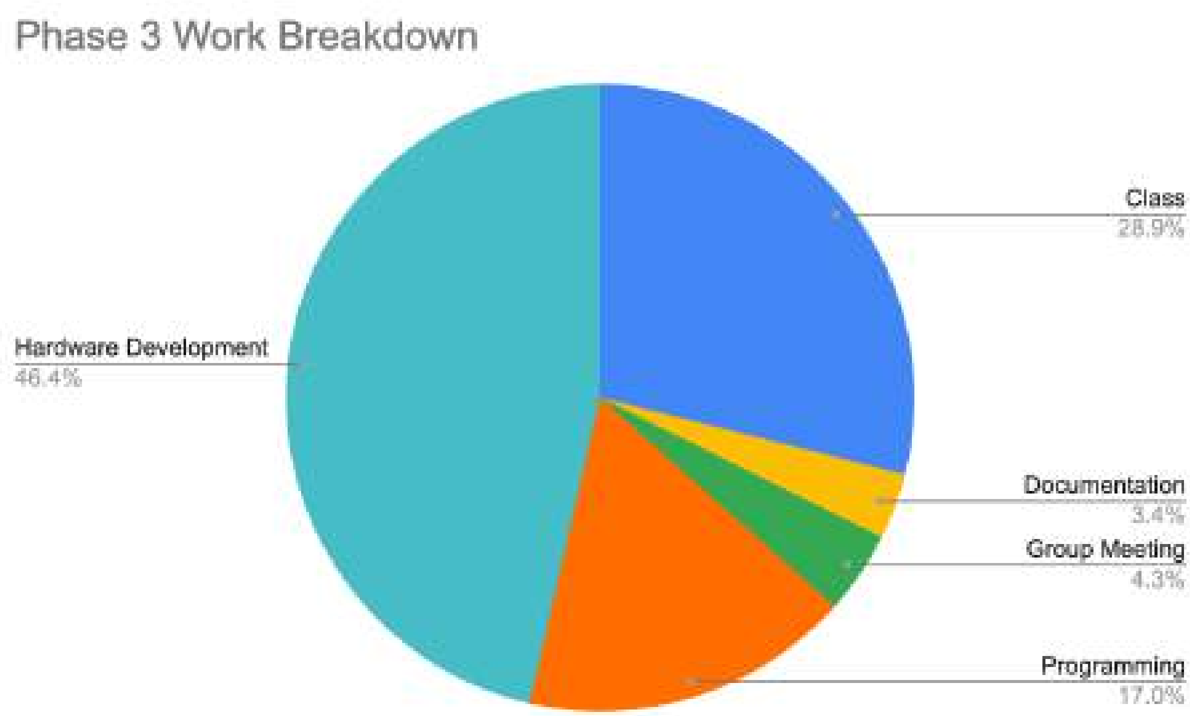


Wearable Hardware/Firmware

Team member	Role	Contribution
Ching-Han	Team Leader	• Attended weekly team lead meetings and led subteam meetings • Coordinated with other teams for discussion surrounding bluetooth connection, sensor reading, and user interfaces • Implemented User Interface: screen, buttons, haptic • Implemented IMU for fall detection
Nadia Palar	Editor	

Sid		• Designed, helped reflow, and debugged the final Printed Circuit Board that integrated all of our components together • Worked with Ching-Han on implementing the IMU for fall detection
Devi	Presenter	• Acted a primary presenter for subteam in final presentation • Implemented and optimized heart rate and SpO2 readings

Hourly Task Breakdown



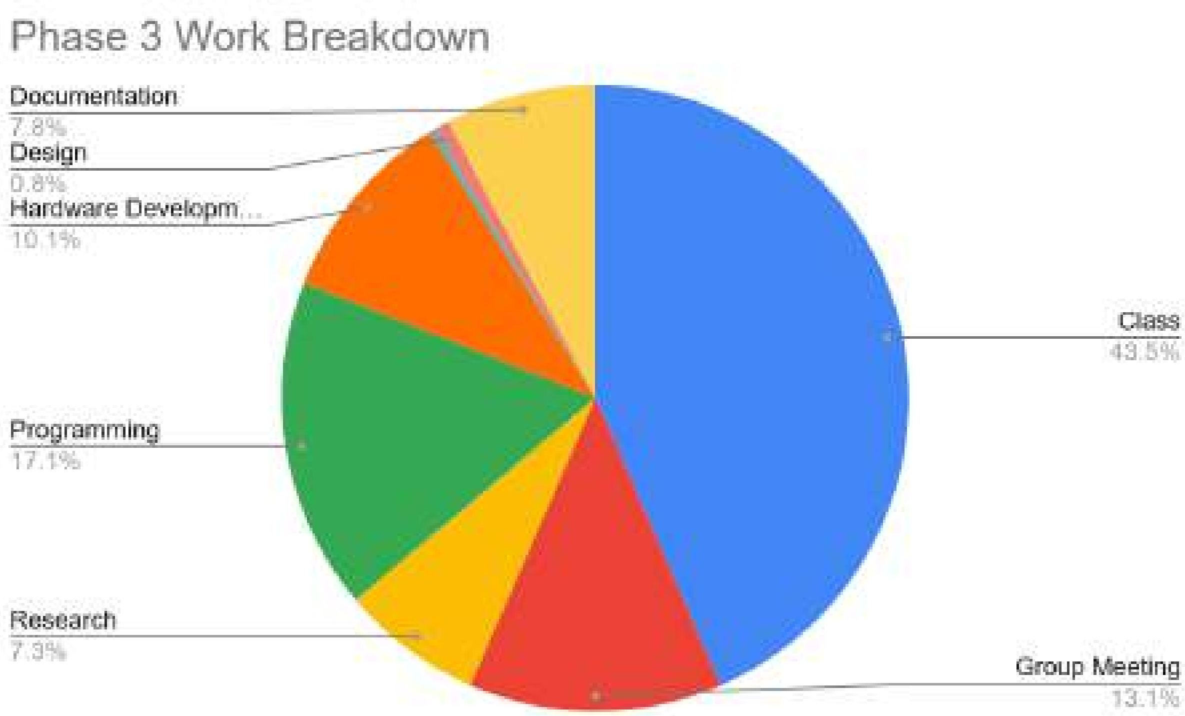
Environmental Hardware/Firmware

Team member	Role	Contribution
Shao-Ju Wang	Editor	• Assembled the base station with the

		case made by Environmental HCI. • Made the GUI frontend • Built the communication backend with MQTT and flexible architecture supporting Bluetooth and Nordic's ESB protocol (not tested yet).
Adwoa Asare	Presenter	• Wrote code to collect data from Enviro+ sensors • Wrote code to send air quality data over MQTT • Wrote shell script and system command to auto run main code on startup • CAD modeled and 3D printed case for Enviro+ substation • Coordinated with other teams to iron out details for our overall architecture and packaging.
Tianyi He	Leader	• Attended Team leads meetings and communicated with other teams leads, coordinating team goals and tasks • Set up Wifi connection between air quality sensors (rpi pico 2w) with rpi 5 • Setup cloud communication for air quality sensors (PM and CO2) • Programmed the frontend of chat app functionality for SW team
Adrian Cantu	Developer	• Setup STM32 microcontroller to communicate with the base station via Bluetooth Low Energy (BLE). • Read multiple ITR20001 infrared reflective sensors with the microcontrollers ADC ports using DMA. • Designed and tested multiple configurations of where the IR sensor could be placed in order to better detect the presence of medication inside the pill organizer.

		• Attended Team Lead meetings and the meeting with Honda representatives • Wrote code to be run on the Raspberry Pi to communicate with the microcontroller via BLE.

Hourly Task Breakdown



Task Dependencies (view in [Google Sheet](#))

Month	February								
Week	**5**			**6**			**7**		
Date	10	12	14	17	19	21	24	26	28
Everyone		**Pre**	**Team Lead Meeting**			**Team Lead Meeting**			**Team Lead Meeting**

		sentation Phase 1							
P/U HCI			Establish entire user experienc e			Wireframing			Mockups & data visualization of preliminary data
Wearable HCI			Determin e types of input and output from and to the users			Determine action flow			Sketch/wirefra me based on action flow (need to consult user software on how to translate decisions to explainable messages)
P/U Software			Get repos and dev environm ents set up			Getting familiar with languages and frameworks			Have some screens implemented (HCI: contact persons=Yiyun Wei for App, Yuchen Dai for website)
W/Env Intelligen ce			Look at existing data, see what we can infer from it.	Finalize ML model structure and fully justified		Figure out how to structure LLM requests will work with the API.		Be done with initial data analysis.	Do a first pass of integration with OpenAI API and have simple thresholding ML model implemented + full integration

W Hardware /Firmware			Determine MCU and required sensors			Determine firmware architecture, messaging structure			Set up development board I/O and communication ports (BLE)
Env Hardware /Firmware			Determine required parts			Start Base-Station prototype. Raspberry Pi to Pico W communication distance test.			Finish wiring schematics
Cloud/Data/Communication Systems			Apply for AWS credits			Getting familiar with different services on AWS			Set up EC2, S3 and database

Month	**March**											
Week	**9**			**10**			**11**			**12**		
Date	10	12	14	17	19	21	24	26	28	31	2	4
Everyone	**Hello World Communication Demo**		**Team Lead Meeting**			**Team Lead Meeting**		**Presentation Phase 2**	**Team Lead Meeting**			
P/U HCI												

Wearable HCI		Model design + component design		Model design + component design			Model Mockup					
P/U Software												
W/Env Intelligence		Interfact with cloud team about our first pass of integration + ML models.		Finish integration with OpenAI API for LLM.		Have ML model implemented	Test LLM integration, do a study on costs incurred, if there's anything that needs to be improved, etc.					
W Hardware/Firmware		Develop bluetooth firmware and start PCB design.		Design bluetooth serialization protocol and setup heart rate sensor.			Start PCB design and user interface firmware development.					
Env Hardware/Firmware			Finish base station interior prototype			Finish substation interior prototype						

						(ambient)						
Cloud/Data/Communication Systems												

7 Reflections

Phase 1

Provider/User Software

We successfully evaluated different frameworks and languages, and considered the needs of the overall project, the needs of the other subteams, and our group's expertise in the final decisions. Throughout Phase 1, we collaborated with the HCI team in exploring and designing user-centered features and requirements for both the patients and the medical providers. We successfully defined some essential functionalities of our subsystems, and our next steps are to create prototypes of the mobile and web applications based on these functionalities. These prototypes will be necessary to validate the technical assumptions we have made in this phase.

Cloud/Data/Communication Systems

Throughout Phase 1, we evaluated cloud platforms, data frameworks, and communication protocols to ensure scalability, security, and efficiency. Guided by project needs and team expertise, we prioritized seamless integration. Collaboration with the software, hardware, and intelligence teams aligned our data infrastructure with user requirements, resulting in a well-defined cloud architecture, storage solutions, and backend pipelines. Next, we will develop a backend prototype, integrate real-time communication, and validate assumptions on data synchronization, security, scalability, and fault tolerance to ensure a seamless mobile and web experience.

Provider/User HCI

We've realized that the role of HCI in this project is to truly set a "pie in the sky" vision and that there is a difference between demanding a vision be brought to life vs. providing a clear direction to work from. In the beginning, we tried our best to democratize the vision setting to ensure all the teams were on board, excited about the idea, and could start collaborating with us on feasibility. We may have leaned too heavily into this, as we learned that what was most really desired and needed was direction and clarity in the vision in order to move forward on negotiating feasibility more effectively.

We also developed a greater respect for the negotiation process, particularly in our collaboration with the wearables hardware/firmware team. Letting go of certain sensors we initially thought would be innovative and applicable due to feasibility concerns was tough but at times necessary. Just because a sensor exists doesn't mean it will be the perfect fit for what we are looking to track and for the larger system. Something that was additionally valuable to us was working with the intelligence team to better understand that while we can't collect everything, there are ways to make inferences based on the data we do have.

When making decisions about which conditions and sensors to prioritize in our vision, we had to weigh likelihood against severity. Some conditions are rare, but their potential consequences are extremely deadly and preventable, making them critical to include.

Wearable HCI

During the first phase, our team made progress in evaluating and selecting wearable medical devices, technologies, and sensors tailored for postpartum care. Through collaboration with the Wearable Hardware/Software subteams, we explored the feasibility of different solutions, ensuring they align with both user needs and technological capabilities.

A major focus of our work was designing the interaction flow for a wearable device, considering various health conditions and user responses. We brainstormed potential features and sketched out initial prototype concepts to refine the user experience. Additionally, our discussions with the User/Provider HCI team helped us align the device's features and services within the broader postpartum care ecosystem.

We faced challenges, particularly regarding feasibility constraints raised by the hardware team. These discussions underscored the importance of balancing innovation with practical implementation. Moving forward, improving communication and establishing clearer responsibility delegation within our subteam will be key to enhancing efficiency and collaboration. By refining our internal processes, we can ensure smoother development in the next phase.

Overall, this phase provided valuable insights into interdisciplinary teamwork, design iteration, and the complexities of developing a user-centric wearable device. Our learnings will serve as a foundation for future advancements in product development.

Environmental Hardware/Firmware

We researched and determined what factors we wanted to detect and track. Once we knew what we wanted to measure, we evaluated various sensors that could potentially work for our purposes. We broke down the different things we wanted to sense into separate sub-stations based on what type of health factor they were detecting. The substations communicate their data back to the base station for basic processing and cloud communications.

In order to make the overall system more customizable to the user, we decided to make the subsystems modular such that multiple substations can be connected and disconnected from the base station based on the user's needs. We collaborated with HCI to discuss sensing requirements. We also collaborated with the cloud and user software teams to determine the optimal architecture. Lastly, we collaborated with intelligence to determine what quality of camera would be required.

Intelligence

Our work began with focused brainstorming sessions where we explored integrating sensor data, machine learning, and language models to support postpartum care. We identified key technical challenges and quickly learned that balancing innovation with practicality was essential—simplicity and clarity were critical in a healthcare setting. After receiving a detailed project description from the HCI team, our role became clear: develop intelligent data analysis methods that detect anomalies in both physiological and environmental data and create human-readable outputs for the frontend, effectively bridging sensor inputs and user interfaces.

Collaboration with the sensor, frontend, and cloud teams allowed us to gather essential technical specifications and understand the system's data flow, underscoring the importance of clear cross-team communication. We designed a high-level system architecture that is both modular and scalable, which led to the development of a dual-model anomaly detection system using simple thresholding for basic metrics and a more complex classification model for nuanced signals. Setting clear procedures for integrating large language models further emphasized the need for transparent, rule-based outputs. The lessons learned—from effective teamwork to flexible design approaches—will guide us as we move into deeper design and implementation phases.

Wearable Hardware/Firmware

During Phase 1, our team explored hardware solutions and collaborated closely with the software and HCI teams to identify components that would enhance the device's functionality while ensuring user comfort. Our primary focus was selecting sensors that provide accurate physiological data while balancing power consumption and data transmission efficiency.

One of the key challenges we encountered was finding suitable sensors that could collect the desired metrics while being technologically feasible for us to implement, maintaining a low profile, and minimizing power consumption. Selecting feasible sensors that produce reliable and usable data is also a challenge. To address these challenges, we worked closely with the HCI team to evaluate functionalities and adjust the requirements based on the wearable device's constraints. One specific example is how we collect blood pressure data, a metric that is of high importance for PPD monitoring but not implementable on a wrist wearable. Since the position and technical requirements to implement a blood pressure sensing device made it infeasible to implement on our primary wrist wearable, we found a compromise with the HCI teams by using an additional smart blood pressure sensor to still have a way to accurately collect this key metric.

Phase 1 provides clarity on the development direction of the device and helps us define key requirements for hardware choices for the wearable device. We will continue to collaborate closely with other subteams in future phases as we anticipate more challenges when it comes to acquiring and processing sensor data, establishing integration with user software, and optimizing user feedback mechanisms.

Phase 2

Provider/User Software

One of the biggest takeaways from this phase was just how important it is to keep in close communication across all subteams. Syncing our data between the mobile app, the provider dashboard, and the backend wasn't as straightforward as we thought—every small change in structure or naming had to be double-checked to avoid breaking things down the line. We learned that having regular check-ins and clear documentation helps a lot in keeping everyone on the same page.

Another lesson was that we realized how valuable it is to think about scalability from the start. Even though our system is still in the early stages, making smart choices now—like organizing our APIs cleanly and designing flexible database schemas—will save us a lot of technical headaches later. It's tempting to rush into coding features, but planning for the future early on definitely pays off.

Cloud/Data/Communication Systems

In Phase 2, we focused on automating and optimizing cloud infrastructure to support backend and frontend development as well as Provider Software Deployment team. We developed tools for EC2 automation, enabling efficient cloud resource management, and used Python script to parse DynamoDB for data processing. We integrated MQTT through AWS IoT Core for real-time communication and set up EC2 to host both backend and frontend components, with GPU support for machine learning tasks when needed. In cooperation with the Provider Software Deployment team, we wrote a script to simplify EC2 connections and app startup, built an environment for Node.js and React for cloud deployment, and used DynamoDB for enhanced scalability. Both frontend and backend APIs were successfully deployed on the cloud with public access, laying a strong foundation for seamless integration and efficient cloud operations while setting the stage for future improvements.

Service HCI

When dividing the user/provider HCI group into further subgroups, we knew we had to define what service design meant for this project. Knowing we couldn't tackle every experience from pre-birth to 1 year postpartum on a service blueprint with great detail, we decided to strategically narrow our focus on notifications and emotional user input– two areas both rich in potential for intentionality and integral to the system. Other areas we considered but ultimately decided against is 1.) an extensive and informative onboarding that adequately prepares mothers for what postpartum might be like and 2.) tapping into crowdsourcing and community by exploring ways to connect mothers using the system. We learned a lot during phase 2 from secondary research, continuing interviews with both mothers and providers, and from prototyping with mothers to elicit their opinions on notification tone, content, and attitudes around surveillance and data privacy. We also extremely enjoyed the experience and creativity of prototyping and how it moved us from theory to building. While we wish there were more time to iterate upon these prototypes and test the effectiveness of different mediums of emotional input, our priority going into phase 3 is to be respectful of other teams' time and bandwidth by providing support where it is needed rather than introducing more work.

Data Visualization HCI

User Interface HCI

During Phase 2, HCI UI's main priority was to deliver screens to the Software team as quickly as possible. We felt the pressure of being a bottleneck for that team and worked hard to provide something they could at least comment on. In doing so, we learned the valuable lesson of ‘moving fast and breaking things.’ Making our thoughts tangible led to constructive feedback from the Software team, greatly benefiting our subteam. We were able to identify issues earlier in the process by sharing our designs, even at a lower fidelity. Eventually, we decided to adopt the Material Design System, which enabled us to create new wireframes more rapidly, benefiting both our team and the Software team.

One major takeaway from this phase was the importance of collaboration with the Software team. We had to maintain constant communication to continuously redefine the scope of our ideas and ensure we were aligned with our plans moving forward. We also needed to coordinate with the Cloud team to ensure that all three teams (HCI UI, Software, and Cloud) were on the same page.

Wearable HCI

One major challenge we encountered during the design phase was ensuring up to date communication between our team and the rest of the teams. Some designed functions could not be promptly tested on the hardware side, leading to delays in implementation. Additionally, integrating the physical prototype with the functional prototype proved challenging, requiring us to present two separate models—one demonstrating appearance and another showcasing functionality. UI usability was another area that needed improvement, as technical limitations prevented us from implementing touchscreen navigation, forcing users to rely on button-based switching, which sometimes felt redundant. Moreover, designing feedback mechanisms for certain features was particularly complex. For instance, heart rate monitoring had to account for exercise-induced variability or situations where the device was not being worn, to avoid incorrect readings of zero heart rate.

Overall, Phase 2 made considerable progress for our wearable device’s design and interaction framework. We successfully created a user-centered interface, explored different material choices, and identified key areas for improvement. As we move into Phase 3, our primary focus will be to refine our prototyping process, working towards a unified model that effectively combines appearance and function. Usability improvements will also be a priority, as we seek to optimize navigation and interaction methods, making the device more intuitive. Finally, we will continue to refine feedback mechanisms for complex functions like heart rate monitoring, ensuring greater accuracy and reliability across different user scenarios.

Environment HCI

The postpartum environmental design was developed through teamwork, research, and testing.

We began with market research to study existing products and their strengths and flaws. These findings guided our design goals to create a system for safe, stress-free interactions with postpartum mothers and real-time health monitoring.

One of the key moments during phase 2 was our conversation with a new mother. She shared her experience of constant multitasking and exhaustion. Her experience inspired us to refocus our work on "invisible support". Instead of taking a traditional approach in healthcare interface, which tends to be overwhelming and daunting, we brainstormed for subtle indicators and physical interactions instead of overwhelming screens or alarms for the base station. Then, we took an interactive approach to translate these ideas into the physical and interface design.

We had a few challenges, including understanding hardware integration, 3D printing, adjusting component sizes, and delivering our progress to the class and sponsors. We had a hard time transferring an idea into the real product several times. For example, adding LED lights to the screen and creating realistic product renders took multiple tries. Communication with the firmware team was fast-paced and sometimes slightly stressful. It was tough to translate design language into what engineers would understand, but we delivered a physical model, wireframes, and final renders that fulfill our vision.

Environmental Hardware/Firmware

In this phase we were focused on implementation. We set up all the firmware and operating system requirements for our hardware and implemented code to read from our different sensors. We also started testing different communication protocols between our subsystems and the base station. Additionally, we started setting up the user interface (UI) for the base station set up process. We collaborated with HCI to determine optimal packaging of our components. Moreover, we started giving finished hardware components to intelligence so they could start testing and started testing our cloud integration.

Our next steps will be finalizing and integrating communication protocols to make all of the data accessible to the intelligence team and the cloud. Moreover, we will integrate our hardware with 3D printed packaging and finalize our UI to give our devices a more finished look.

Intelligence

In phase 2, we were able to build out the full ML and LLM systems, and integrate all of our pieces together within the subteam. We were able to build a working demo, and saw real anomalies get turned into real reports. For our next phase, we must focus on integration with other teams. For example, the anomalies and output of the notifications should be stored in the cloud, so we must collaborate with the cloud team. For the ML aspect, we have a simple system for flagging an abnormal set of current sensor readings. Going forward, we would need a more complex system for tracking trends over time. For the data processing aspect, our future work involves the integration of automation (as data points come) and finishing the data post-processing pipeline for model evaluations.

Wearable Hardware/Firmware

Phase 2 was significantly more hands-on than phase 1 since we immediately started with some preliminary hardware after placing our first parts order at the end of phase 1 prior to spring break. We were able to make significant progress and fully design and implement nearly all aspects of our subsystem and a basic prototype level, including reliable bluetooth communication from our board to a mobile device; communicating with our chosen sensors and obtaining reasonable data from them; implementing a preliminary user interface as a proof of concept; and, arguably most excitingly, designing a custom PCB incorporating our finalized components.

Next steps for phase 3 include ordering (and potentially fabricating) the custom PCB, integrating with the other teams (specifically P/U Software as they remain our only direct link to the rest of the overarching system), refining user interactions with Wearable HCI, and continuing to make our wearable as robust as possible.

Phase 3

Provider/User Software

Throughout Phase 3, we encountered several challenges that helped us grow as a team. One notable difficulty was a miscommunication where two teammates unknowingly worked on the same page, with one teammate referencing an outdated design. This overlap caused some confusion and redundancy in our workflow. However, the issue was quickly resolved through rapid clarification of the latest designs, reinforcing the importance of staying aligned during development.

In addition, integrating multiple technologies—such as Bluetooth connections on the app side and cloud database interactions for both app and website—pushed us to deepen our understanding of cross-platform data handling, API communication, and real-time notifications. The experience also highlighted the value of frequent testing and communication across different parts of the system to catch and fix issues early.

Lastly, one challenge we faced in previous phases really came to head in this one: the limited team size. Originally at the beginning of the class, the provider/user software team was meant to have two sub-teams (one for the app and one for the website) each with 4 members. However, by the middle of phase 2, we were down to only 2 people each. While this made communication easier, the workload for each individual became a lot more. We had to really prioritize features to have a functioning product in the end.

Cloud/Data/Communication Systems

Throughout Phase 3, we faced several challenges that helped us grow as a team. One major challenge was setting up stable communication between the wearable/environmental hardware, user/provider apps, and intelligence modules, while making sure the cloud could handle large volumes of data reliably.

When setting up health alerts for abnormal patient metrics like heart rate and blood pressure, we initially ran into delays because AWS required a lengthy approval process for SMS notifications. To stay on schedule and keep alerts responsive, we switched to Twilio, which allowed faster integration through its free trial service.

We also learned to manage cloud resources like DynamoDB and S3 while keeping the system lightweight and scalable. Debugging across EC2, Lambda functions, and Twilio workflows showed us that building real systems requires more than just coding — it also demands designing reliable infrastructure and monitoring everything end-to-end.

Service HCI

This phase reemphasized the critical importance of communication. We had to clearly pause the design and research work for all HCI teams and shift our focus to supporting other teams by helping them prioritize features for integration and making documentation easy to use, especially for the software team.

In this phase we reflected a lot on how much breadth we were asking for from the engineers and able to accomplish as a class. When reflecting, we wondered if we were doing too much at a broad level, and had too much breadth as opposed to finding niche interactions that were innovative in themselves rather than being in an innovative topic space such as postpartum health. However we realized that both have merit, and we felt this more “niche” space of postpartum health was overall very grounded in real needs and had high feasibility. We also reflected upon how important it is to work in niche spaces to be able to dig deep and give it the correct contextual justice. Even within niche spaces, there will always be analogous domains that parts of solutions can transfer over to (e.g. how Honda gave us feedback post demo that parts of our solution are very applicable to elderly care).

Data Visualization HCI

This phase of the project emphasized how critical cross-team communication is to designing systems that are both beautiful and buildable. We learned to tailor our visual decisions to the needs of both users and engineers, and to use tools like Figma not just for design, but for dialogue. Wizard of Oz prototyping gave us the flexibility to move forward without getting blocked, helping us validate key interactions even before the system was complete.

User Interface HCI

This Phase taught us a lot about how to support other teams. The previous phase was very design-focused with a notable shift to implementation for phase 3. This meant watching the Software Team kick things into overdrive as they began bringing our designs to life.

As we got closer to our Demo Day, we started seriously considering how we could support the Software Team and decided to wizard-of-oz some of the functionality in Figma to lessen Software’s load.

Another lesson arose from not communicating enough with the Intelligence Team. At one point, we spoke with the Software Team to decide on what functions should be prioritized. We told them that it should be Searching and Messaging, not realizing that the Intelligence Team wanted the Weekly Report to be prioritized instead. Figuring this out ahead of time may have saved the Software Team some stress and was a good lesson learned moving forward.

Wearable HCI

In Phase 3, our Wearable HCI team focused on finalizing both the visual and interactive aspects of our wearable device while continuing to adapt to the evolving requirements from the Hardware Team. One of our main tasks was adjusting the product’s user interface (UI) to match the updated screen size specifications provided by the Hardware Team. This involved redesigning several UI elements and carefully recreating a full set of 32x32 pixel icons that could be used effectively within the constraints of the functional prototype.

At the same time, we completed the 3D modeling for the appearance prototype. This model was then used for 3D printing, through which we experimented with different materials, colors, and levels of transparency to assess their impact on usability and realism. Based on the printed models, we made further refinements to the design to enhance user comfort and experience.

To support our iterative design process, we also conducted user research, gathering feedback to guide final adjustments and ensure the prototype was as intuitive and accessible as possible. Toward the end of the phase, we synthesized outcomes from all three phases, integrating our efforts into a cohesive final presentation and demonstration.

Despite these accomplishments, we encountered several challenges. One major obstacle was the limited availability of 3D printing resources, which made scheduling and iteration difficult. Communicating precise adjustments between modeling and printing required highly accurate coordination and often several rounds of revision. Additionally, material limitations meant that certain ideas—such as using semi-transparent surfaces or embedded LED indicators—could not be fully realized in the final appearance prototype. Communication across teams also became more difficult during this period, especially with approaching final deadlines and end-of-semester pressures, making project coordination more demanding.

Phase 3 marked the culmination of our design and development efforts. We were able to deliver both a functional and appearance prototype, each informed by user feedback, technical constraints, and collaborative decision-making. Our work in this stage brought together the visual, structural, and interactive elements developed throughout the semester into a unified product vision.

Although the project has reached the end, looking forward, we hope to explore more advanced materials and features—such as integrated lighting or flexible displays—to better simulate real-world use cases and increase product realism. Strengthening UI usability and expanding user testing will also continue to be a priority as we refine the device beyond the classroom context.

Environment HCI

Environmental Hardware/Firmware

For Phase 3 we finalized our data formats, integrated the base station with all of its substations, and integrated with other teams. Some of our components were leant out to the intelligence team for integration since they were running some models locally on the base station. One challenge was coordinating who had which hardware components at which time. It would have been helpful to order more duplicates to share with the teams we were integrating with.

We were using a variety of communication protocols for the various substations, since it would give us the chance to learn and practice different implementations. This increased our testing scope since we had to verify all the different communications. We each worked separately on our substations up until this point, so we had to work together to combine everything and integrate it with the base station code. One challenge we had with the communication was getting everything connected on the same network at school and knowing each device's IP address. We tried using a hotspot as well as the school provided Wi-Fi. The IP addresses at school are randomly assigned and can change when you are in a different location on campus so we had to do all our testing close to the classroom since that was where the demo would be. Overall, doing a lot of testing was a key part of our process both individually and together to achieve the optimal demo set up.

Intelligence

In phase 3, we finished up all integration with other teams. There were some reconfigurations that we had to make with the cloud integration, but we were able to get fully integrated. We also wanted to take into consideration the HCI team's ideas of "calibrating" the severity score, so we added that to the LLM and conducted more testing to make sure the LLM was still outputting the correct text, even with this change in context.

In Phase 3, we changed our machine learning approach from detecting anomalies at single points to analyzing trends over short time intervals. Instead of just looking at one sensor value at a time, we used the mean and median of values (like heart rate and body temperature) within short time windows. This helped reduce noise and gave us a clearer picture of the body's condition. We also added an LSTM Autoencoder to detect unusual patterns in how sensor values change over time. The LSTM model was trained to learn normal patterns and try to reconstruct them. If the reconstruction error is high, it means something unusual happened—like a sudden jump in heart rate—which is flagged as an anomaly.

For the data processing side, an initial challenge we faced was syncing on the generated sample data. While we were generating sample data, we had taken previous metrics and values from the hardware team without complete knowledge of updated metrics. We made many updates after discussions with understanding the sensors that were feasible with the hardware teams and generating the proper data in the correct and consolidated the formatting for each of them. Additionally, we were unfamiliar with automating the data preprocessing scripts initially to be triggered, but after collaboration with the cloud team we were able to successfully run automated scripts for data preprocessing through AWS.

Generally, the final subteam integration went smoothly, as all we had to do was pull data from the cloud and push data to the cloud, which we were able to get set up easily.

Wearable Hardware/Firmware

Phase 3 was the most challenging stage, as it focused heavily on integration across different subteams. We invested significant effort in communication and collaboration. In particular, we worked closely with the wearable HCI team to finalize icon and button designs and collaborated with the user software team to establish reliable Bluetooth data transmission. Additionally, we made substantial progress on our custom PCB design, finalizing it and sending it out for production.

Although we dedicated a lot of time to designing and assembling our custom PCB, time constraints prevented us from getting it fully operational before the presentation. While we successfully reflowed the board, we were unable to debug all issues in time to have it function correctly. As a workaround, we used a perfboard setup to demonstrate our firmware design and presented the custom PCB to illustrate the intended final product. The demonstration went smoothly, and we were able to effectively showcase all the functionalities and design considerations of our wearable hardware.

Blog Posts

Provider/User Software

Purpose of App:
The app consolidates wearable sensor data into an easily accessible space for users, reducing the burden of self-monitoring by providing real-time alerts and health information when sudden changes in health metrics occur. It also enables efficient communication with healthcare providers during emergencies.

Technologies:
The app is developed using Flutter, with various Flutter packages such as `flutter_blue_plus` for Bluetooth functionality, `shared_preferences` for local storage, and `flutter_local_notifications` for managing real-time alerts. Android Studio was used for testing, deployment, and emulation during development.

Functionalities:

- Real-time notifications for anomaly detection (e.g., heart rate spikes, falls)
- Displaying wearable sensor data and visualizing trends in health metrics
- Offering advice and health information based on detected patterns
- Enabling direct messaging between users and providers

Purpose of Website:
The website serves as a provider-facing portal that enables healthcare professionals to efficiently monitor patient information, review reports, and quickly identify potential health concerns flagged by the wearable devices. It provides an organized and centralized view of multiple patients' data without needing to access the app directly.

Technologies:
The website is built using React for the frontend and communicates with the shared backend API via HTTP requests. The backend is built using Node.js with Express (in TypeScript) and is hosted on a cloud server.

Functionalities:

- Patient search by name, date of birth, or location
- Viewing past anomaly reports and mitigations

- Viewing alerts triggered by patient anomalies
- Access to demographic information of patients and the ability to write notes

Cloud/Data/Communication Systems

Our team's primary goal was to build a reliable, scalable, and secure cloud pipeline connecting all components of the Fourth Trimester Health system. We enabled seamless communication between wearable and environmental hardware, mobile and provider applications, and machine learning intelligence modules.

The cloud system supports three critical functions:

- **Data Storage**: Collecting and storing patient and sensor data.
- **Synchronization**: Maintaining real-time data flow across user and provider platforms.
- **Real-time Notifications**: Alerting users when anomaly metrics, such as heart rate or blood pressure, fall outside safe thresholds.

Our architecture was designed for real-time health monitoring and immediate anomaly detection. Throughout this project, we collaborated closely with other teams to ensure data flows were robust, efficient, and secure.

Key technologies and services we used include:

Amazon EC2 + NGINX
We hosted user/provider applications and backend services on Amazon EC2, using NGINX as a reverse proxy for improving system reliability, load balancing, and scalability.

Amazon S3
Sensor data, such as thermal camera images, is stored in Amazon S3 for durable, scalable storage, with metadata linked to DynamoDB.

AWS DynamoDB
DynamoDB stores all structured data, including patient profiles and sensor data, providing low-latency, real-time updates to apps and machine learning modules.

AWS Lambda + Twilio
AWS Lambda functions monitor DynamoDB for anomalies and trigger SMS alerts via Twilio to notify users, such as patients of critical health risks.

Data Agents and Synchronization
Backend agents synchronize sensor data between devices, cloud storage, and apps, while also feeding raw data to the Intelligence system for anomaly detection.

With all these technologies working together, we made sure Fourth Trimester Health could track, process, and communicate important health information after delivery — helping close the care gap with a system that's reliable, responsive, and scalable.

Service HCI

Purpose
The Service Design team was formed as an evolution of the HCI department during Phase 2 of our project, with the objective of situating the prototyped system within a larger context as a product service system (PSS). This approach combines tangible products—the wearable and environmental sensors—with intangible services to deliver a holistic solution meeting the needs of new mothers during the postpartum period.

We firmly believe this systems-level thinking is essential for both the business and social viability of our product. As service designer Andy Polaine notes, "Service design is about designing with people, not for them." Even in a prototyping-focused project, considering the entire service ecology—the value exchange between customers and enterprise—provides crucial insights. Services are ultimately value co-produced by multiple stakeholders, including new mothers/parents, healthcare providers, and Fourth Trimester Health.

Learning from Stakeholders
To develop a user-centered solution, we conducted in-depth interviews with seven mothers and five providers, including OB/GYNs and doulas. These conversations revealed the importance of addressing the very real and common risk of postpartum mood disorders, particularly depression and anxiety. The mothers we spoke with shared powerful insights:

- "[Once I had delivered the baby,] I was just left out there, alone."
- "Postpartum depression is rooted in a sense of isolation, loss of autonomy... [It's] a radical change to your life."
- "I went from having a very happy postpartum experience to being highly anxious, having feelings of despair, and not knowing why... Everything was overwhelming."

These testimonials reaffirmed our decision to focus on the postpartum period and redirected our efforts to emphasize emotional well-being alongside biometric markers, recognizing that these elements are deeply interconnected.

Three Main Components
Based on our research and stakeholder feedback, we focused on developing three essential aspects of the system:

1. Emotional Input Tracking
Working alongside the patient UI team, we developed this feature as a part of the user-facing mobile app. After nuanced discussions with other departments about the feasibility of incorporating self-reported mood into the system's ML model and the challenges this would present for accurate outputs, we decided to keep this separate from the ML model.

Instead, we designed it as a tool for personal tracking, helping users become more mindful and aware of their emotional states. This also supports self-advocacy in conversations with healthcare providers. The final implementation is a daily check-in flow where users can:
- Input their mood
- Add notes about specific issues
- Access an open space for personal reflection

2. Provider/User Onboarding
This component was crucial for contextualizing our system and bridging the gap between the technical aspects of our product and the human needs of providers and patients. Our development process involved extensive research, user conversations, and discussions about critical issues including:
- Data privacy (determining appropriate data-sharing boundaries)
- Medical history integration (understanding how existing health conditions might impact model recommendations)

The resulting onboarding process creates a thoughtful introduction to our system for both providers and postpartum patients, establishing appropriate expectations and context for the system.

3. Severity Score Rating
The ML model, developed by the intelligence team, can reliably detect anomalies and suggest mitigation strategies based on collected data. However, these outputs can be difficult to understand from a human perspective.

To improve human-AI interaction and make the ML outputs more meaningful, we developed a severity score rating system that effectively translates technical findings into understandable terms. Based on background research on various postpartum conditions and biometric indicators, we created a scale that normalizes severity scores:

- Level 0: No Risk - No notification sent, no anomaly detected
- Level 1: Low Risk - "Slightly" concerning - Action: monitor/watch out
- Level 2: Low-Medium Risk - "Somewhat" concerning - Action: slight adjustment recommended
- Level 3: Medium Risk - "Moderately" concerning - Action: change behavior, watch for other symptoms
- Level 4: High Risk - "Very" concerning - Action: change course immediately, watch for symptoms (requires immediate action, notifies care team, alerts the user)
- Level 5: Critical Risk - "Critically" concerning - Action: go to hospital (alerts user and caregiver before calling emergency services)

The Broader Context

The Service Design team's work aimed to embody Herbert Simon's fundamental definition of design as "changing existing situations into preferred ones." For postpartum mothers, the existing situation is often one of isolation and insufficient support during a crucial transition period. Through thoughtful service design, we aim to transform this experience into one of connection, awareness, and timely intervention when needed.

The Service Design team's work helped situate Fourth Trimester Health within the context of comprehensive postpartum care. By considering the human factors alongside technical capabilities, we've attempted to create a compelling business and user case for our product as it fits into a new parent's life.

Provider Data Visualization HCI

Designing Clarity in Postpartum Health Dashboards

In the Fourth Trimester Health project, we focused on creating simple, clear visualizations to help providers quickly understand postpartum health data. Our goal was to support clinical decision-making without adding extra complexity.

Research-Informed Design

We interviewed seven mothers and five providers to learn about their experiences. Using affinity diagramming, we identified common themes and developed core design principles. The most important was minimizing cognitive load. Our visuals were designed to be calm, non-intrusive, and easy to interpret.

We used a tile-based layout inspired by Apple Health to organize key metrics into categories such as vitals, environment, mental health, and medication. Each tile gave providers a quick overview, with the option to view more detailed trends if needed.

Building with the Software Team
We worked closely with the software team to ensure our designs matched what was technically possible. For features still in progress, like graph views, we used Wizard of Oz methods to simulate functionality and test interactions.

Conclusion
By combining user research with cross-functional collaboration, we created a dashboard experience that reduces mental load and improves clarity. This approach can support better care not just in postpartum health, but across healthcare more broadly.
(Drafted with ChatGPT and edited for accuracy)

Patient Data Visualization HCI

User Interface HCI

Enhancing Provider Workflows: Key Insights from Our Research
Fourth Trimester Health strives to enhance healthcare delivery by understanding the needs of our providers. We conducted research with several practicing doctors to gain insights into their workflows and challenges.

Understanding Provider Needs
Our conversations revealed that providers spend varying amounts of time preparing for patient visits, reviewing notes from other doctors or hospitals, and test results across multiple platforms. These providers emphasized the importance of having access to patient information collected in their absence.

Informing Design with Research
Using insights from these interviews and functional requirements prepared by the software team, we mapped out a provider's journey from logging into a healthcare portal to inputting diagnosis codes after visits. We focused on integrating our portal into existing workflows, particularly in pre-planning, data analysis, and documentation phases.

Developing User Flows and Wireframes

We created a detailed user flow that outlines the necessary screens and actions, incorporating error flows for added robustness. These guided our transition from rough sketches to annotated wireframes, inspired by existing electronic medical record systems, such as Epic.

From Lo-Fi to Hi-Fi Prototypes
Our design team developed high-fidelity screens using the Material Design System, including home, search, inbox, alert, and patient profile pages, maintaining a cohesive aesthetic.

Prototype Presentation
During our final Phase 3 Demo, we presented a provider checking messages, viewing patient data dashboards, making notes, and responding to patients, showcasing the potential of our provider portal.

Conclusion
Our research and design process aims to provide doctors with access to patient information and streamline their workflows, thereby enhancing the quality of care they offer. While we designed this product to operate within the postpartum space, we see many implications for other healthcare use cases in the future.

(Drafted with Microsoft Copilot and edited for accuracy)

Wearable HCI

The purpose of the Wearable HCI subteam is to design the interfaces and interactions of the wearable system. The team consists of an industrial designer, an interface designer, and a visual artist. Our goal is to support mothers' daily health monitoring at home by displaying essential sensor data and alerts, serving as a primary touchpoint when their hands are occupied with childcare or when they're away from their phone. Throughout the course semester, we learned to design with ease of use, comfort, and meaningful feedback in mind.

Ease of Use
We designed the interface with a simple screen with two buttons. Through a combination of short and long press gestures, the mother can use the wearable for signal browsing, alert confirming, emergency dialing, and emotion logging. We used a layered information architecture that allows users to browse essential health data with a single, short, button press, while emergency and logging actions are conditionally available through long press.

Comfort

The Wearable HCI team ensures mothers' comfort at both physical and emotional level when engaging with the wearable system. A layer of hypoallergenic silicon is designed to cover the wearable device (including the screen) and wrist band with round edges to retain flexibility while reducing skin irritation, making it safe for both the mother and the newborn when there is incidental contact during feeding and care. For the screen interface, we created calm, non-alarming icons and animations to inform levels of urgency without causing panic.

Meaningful Feedback

The interface incorporates colored LEDs and haptic motors to deliver tiered notifications that distinguish between informational updates and actionable alerts. We designed various vibration and status light patterns that aim to provide clear acknowledgment when the mother takes action or when the system detects significant events.

Team Reflection

The wearable interface design emerged through consistent collaboration with other subteams and could be implemented without them. Our most critical partnership was with the Wearable Hardware/Firmware team, where we discussed and determined the physical form factor, co-designed button placement and tactile feedback, and aligned display capabilities and detailed content. We also worked closely with the User Interface HCI team and the Service HCI team to ensure the experience we designed is valuable to the mothers and remain consistent across platforms. It was through the tight communication we understood user expectations, technical constraints, and finding the balance between them through experimentation and integration with the overall system.

Environment HCI

The goal of the Environment HCI team is to design an intuitive and supportive system that helps postpartum mothers monitor their environment and medication at home. We wanted the system to be safe, easy to use, and to merge naturally into the mother’s daily life. To achieve this, we directed our design efforts to three main components: the physical base station, the base station interface, and the physical pill box. The physical base station, with a soft LED light ring around the screen that indicates the users’ state of well-being, is a hub for displaying information at a glance with easy affordance. The interface for the base station helps the users set up their devices (substations, wearable, pillbox, etc) and view environment information in a clear, gentle way. The smart pill box tracks medication use, sends information to the cloud through the base station, and helps mothers stay on schedule without stress by pushing reminders. These three main design components make health monitoring at home feel natural and stress-free, helping mothers stay safe and confident during a vulnerable time.

We are proud that we created a system that feels supportive rather than overwhelming. Designing for postpartum mothers made us think about the importance of clear feedback, minimal cognitive load, and emotional sensitivity. We were able to reach our goals by working closely with the Environment Hardware/Firmware team and revising our designs based on feedback. This helped us learn how to work across specialties and bridge physical and digital experience.

Environmental Hardware/Firmware

Our Environmental Hardware/Firmware team aimed to create an environmental monitoring system that is robust, modular, and designed to blend seamlessly into the home environment. We envisioned more than just a data-gathering device; we wanted a system that allows the patient to interact with it and stay informed. We understand that different people have different needs, so we separated the processing and sensor into a base station and multiple substations; this enables users to customize their setup by selecting various sensors and placing them strategically in different areas.

Below are the components of our design, we have included two substations to show our flexible design:

- Base Station
    - The base station is powered by a Raspberry Pi 5, as it's flexible, and includes both hardware and software communication interfaces. It uses its hardware interface (UART) to communicate with the substations to configure them and set them up for wireless communication (for example, MQTT via Wi-Fi), then the user can place them anywhere (within range). It will gather environment information and tile-based display them and send them to the cloud for further processing.
- Substation 1 (Ambient Conditions)
    - This ambient substation uses an Enviro+ and a Raspberry Pi Zero 2W to gather temperature, humidity, pressure, light, and noise. It has a small built-in display to show different information, as well as a battery.
- Substation 2 (Medical Adherence)
    - This is a smart pill organizer that tracks the mother's medicine intake. It consists of a 3D printed case with an array of IR sensors and STM32 microcontroller that reads the sensor data and forwards it using bluetooth.

Intelligence

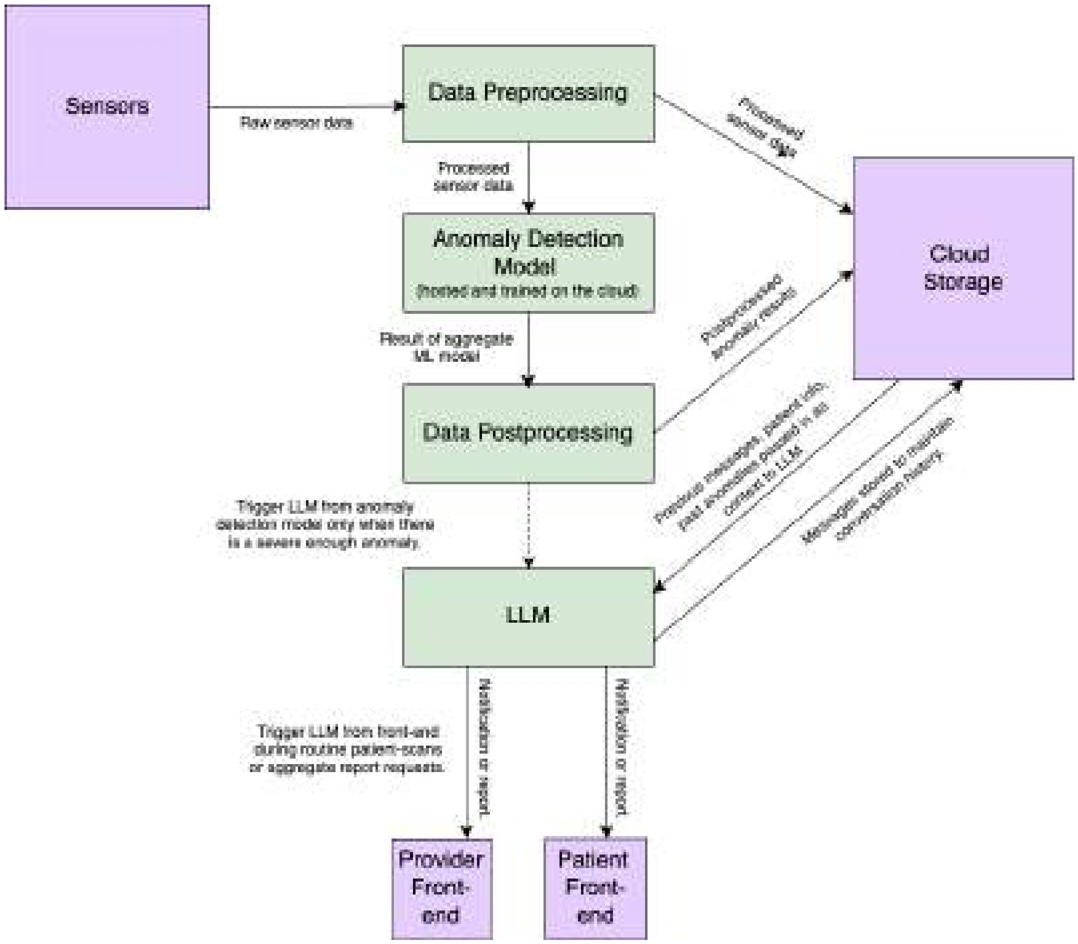
The purpose of the Intelligence system is to act as a notification and trend report generator, which can be passed on to the front-end systems to create user-facing notifications and reports. Given raw sensor data, we utilize ML modeling techniques to detect anomalies in the patient. The output of the ML model is passed in as context to our LLM, which generates a human readable notification. These notifications and reports are uploaded to the cloud, where the front-end teams can read from and parse.

Our detailed functional requirements are below:

1. Data Pre and Post processing: Raw sensor data received from our hardware teams first undergo preprocessing. This ensures our data is cleaned, normalized, and correctly formatted for further analysis and can allow accurate model predictions.
2. Anomaly Detection Model: Our aggregate machine learning model hosted on the cloud leverages classification and thresholding methods to detect anomalies in sensor outputs.
3. LLM Integration: The processed anomalies supplemented with historical context and patient data, are inputted together into OpenAI's GPT-4 LLM. This model will provide a resource to evaluate anomalies and help generate reports to indicate trends, any severity levels, and mitigation strategies for the patient.

Our integrated pipeline provides clarity and usability for the user, taking consideration of sensor data and necessary precautions for notifying users of anomalies.

Our subsystem architecture is represented in the following:

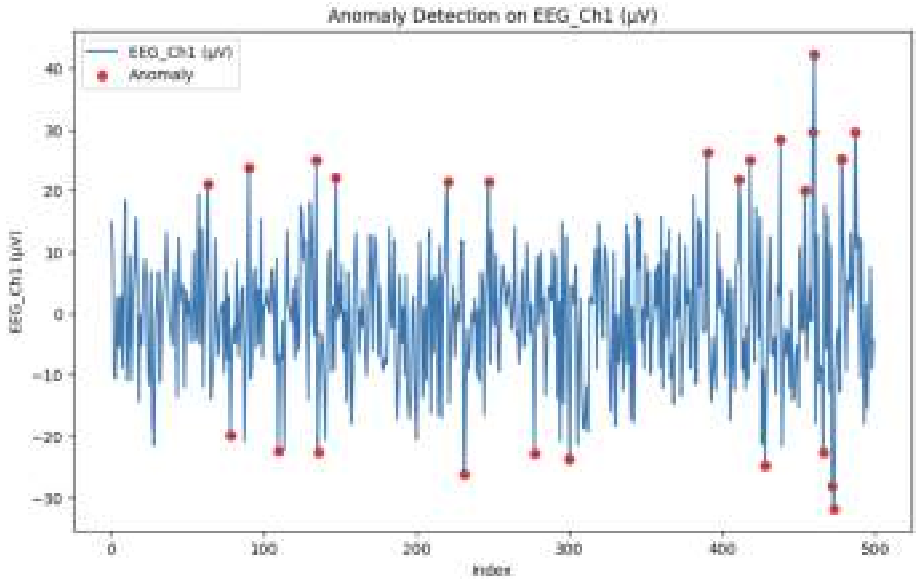


The data preprocessing workflow focuses on loading raw sensor data and working through handling missing values, sensor-specific filtering techniques, standardization of data to ensure uniform feature scales, and synchronization to resample sensors to fixed frequencies. This data pipeline prepares the collected sensor data from its raw form to processed data ready for ML and LLM processes.

Our system employs two distinct machine learning models targeting different types of anomalies:

1. Isolation Forest for Single-Feature Anomaly Detection
   To identify anomalies in individual sensor features such as heart rate, body temperature, and oximeter readings, we use the Isolation Forest algorithm. For each type of sensor data, a separate Isolation Forest model is trained using a dataset of normal physiological values. This training process allows the algorithm to learn a threshold that separates normal from anomalous readings.

During real-time prediction, each incoming sensor value is independently evaluated by its corresponding Isolation Forest model. If a value falls outside the learned threshold (indicating it is "isolated" from typical values), it is flagged as anomalous. This approach is well-suited for detecting extreme outliers in individual physiological signals.



2. LSTM Autoencoder for Temporal Variance-Based Detection
   To detect anomalies caused by abnormal variance patterns—such as sudden spikes or drops across time—we utilize a Long Short-Term Memory (LSTM) Autoencoder. This neural network model is trained to reconstruct short windows of sequential raw data from multiple sensors.

   The model learns the typical temporal patterns during training. During inference, it attempts to reconstruct new sequences. When a sensor signal exhibits an unexpected or abrupt change in variance (e.g., sudden acceleration in heart rate), the reconstruction error becomes significantly higher. These high reconstruction errors are treated as indicators of anomalous behavior.

By combining point-based anomaly detection (via Isolation Forest) with sequence-based anomaly detection (via LSTM Autoencoder), our system can detect both isolated anomalies and evolving abnormal trends in sensor data.

More on LLM
To generate human-readable outputs from the machine learning model, we integrated OpenAI's GPT-4.0 as the core of our notification and reporting system. GPT-4.0 was selected for its strong medical reasoning performance, rich knowledge base, and reliable cloud API access, which allowed us to seamlessly connect it with our backend systems.

We designed structured prompts using the COSTAR Prompt Engineering framework, ensuring that the LLM responses were not only medically accurate but also sensitive to the needs of postpartum mothers. To evaluate the effectiveness of our approach, we simulated user testing by simulating realistic patient data, feeding it into our system, and observing the clarity, tone, and actionability of the generated outputs.

Specifically, we created two types of outputs:

- Anomaly Notifications: Brief, supportive alerts based on detected anomalies.

Notifications are calibrated based on a severity scoring system to adjust the tone and urgency.

Unset


```
"date": "2024-03-10",
        "anomaly": "High blood pressure",
        "severity_score": 7,
        "mitigation_technique": "Monitor regularly, reduce sodium intake"
 "date": "2024-03-10",
        "anomaly": "Irregular heartbeat",
        "severity_score": 8,
        "mitigation_technique": "Schedule an ECG, reduce stress"
"date": "2024-03-11",
        "anomaly": "Low oxygen saturation",
        "severity_score": 9,
        "mitigation_technique": "Administer oxygen, check for respiratory
distress"
```


- Weekly Reports: Summarized health updates that aggregate the anomalies detected throughout the week. These reports provide trend insights and suggested mitigation actions, helping mothers maintain awareness of their overall well-being without feeling overwhelmed.

Unset


```
"date": "2024-03-12",
        "anomaly report": "- **2024-03-10**:\n- High blood pressure (Severity:
7): Elevated readings detected by sensors, consistent with prior gestational
hypertension diagnosis.\n- Irregular heartbeat (Severity: 8): Detected abnormal
rhythm, potentially linked to stress or cardiovascular strain.\n\n-
**2024-03-11**:\n- Low oxygen saturation (Severity: 9): SpO2 dropped below 90%,
indicating potential respiratory or circulatory issues.\n- Sudden drop in heart
rate (Severity: 6): Brief episode of bradycardia detected, possibly linked to
oxygen level fluctuations.\n\n- **2024-03-12**:\n- Increased body temperature
```

```
(Severity: 5): Fever detected, likely indicating inflammation or
infection.\n\n---",
        "trends": "- **Cardiovascular strain**: Persistent irregular heartbeat
and sudden drops in heart rate, coupled with prior gestational hypertension,
suggest ongoing circulatory challenges.\n- **Respiratory concerns**: Low oxygen
saturation is a significant anomaly, potentially compounding cardiovascular
strain.\n- **Systemic inflammation**: Increased body temperature could indicate
underlying infection or stress response.\n\n---",
        "mitigation": "- **Monitor blood pressure**: Increase frequency of
readings to track trends and report sustained high values to a physician.\n-
**Consult a cardiologist**: Irregular heartbeat and bradycardia warrant further
investigation.\n- **Evaluate oxygen saturation**: Use supplemental oxygen if
needed and consult a pulmonologist if drops persist.\n- **Hydration and rest**:
Support recovery from fever and potential systemic inflammation.\n- **Track
postpartum recovery**: Ensure adequate nutrition, stress management, and avoid
overexertion.\n- **Seek urgent care**: If oxygen saturation falls below 88% or
fever exceeds 101\u00b0F, prioritize immediate medical attention."
```

Wearable Hardware/Firmware

For patients recently discharged from the hospital, the transition period can be particularly vulnerable. The Wearable Hardware/Firmware team has developed a comprehensive wearable health monitoring system designed specifically to address this critical need, providing continuous health tracking without adding complexity to patients' recovery routines.

Our primary wearable features a suite of sensors and interface components:

- Equipped with multiple sensors for continuous monitoring of vital signs:
  - Heart rate
  - Blood oxygen saturation
  - Motion tracking for fall detection
- Intuitive user interface featuring:
  - Vibration alerts for time-sensitive notifications
  - LED indicators for status updates
  - OLED display for detailed information
  - Simple button controls designed for ease of use

The architecture balances sophisticated health monitoring capabilities with user-friendly design. Our custom-designed firmware runs on an STM32WB microcontroller, chosen for its Bluetooth connectivity, power efficiency, and sufficient processing capabilities.
The system architecture prioritizes three core principles:

- Ease of use: Intuitive interface requiring minimal learning
- Non-invasiveness: Comfortable design for continuous wear
- Responsiveness: Real-time anomaly detection and alerts

Sensor Suite:
The sensor suite in our primary wearable represents a careful balance between comprehensive monitoring and practical limitations. The MAX30102 sensor serves double duty, tracking both blood oxygen levels and heart rate. This data not only provides immediate health status but can also be analyzed for deeper insights through deep learning.

Our implementation of fall detection utilizes the BMI323 Inertial Measurement Unit (IMU) to access accelerometer readings. When a potential fall is detected, our system initiates a critical safety protocol. A 30-second countdown begins on the wearable device, giving the user an opportunity to cancel the alert in case of a false alarm. This can be done with a simple long press on the device button. If no cancellation occurs within the 30-second window—suggesting the user may be incapacitated or unable to respond—the wearable automatically signals the mobile application to initiate emergency protocols, which can include contacting emergency services or designated caregivers. This automated safety net provides peace of mind for both users and their families, ensuring help arrives even when the user cannot manually call for assistance.

User Interface Design:
During our development process, we discovered that the interface design was just as critical as the sensing technology. Our push-button interface was engineered to support multiple interaction modes:

- Quick status checks: Short presses cycle through different data displays
- Alert management: Long presses cancel false alarms (such as misidentified falls)
- Mood tracking: Combination presses allow simple emotional state input

The 0.96-inch OLED display with 128x64 pixel resolution balances between information density and power efficiency. Our UI design leverages this by using intuitive icons to convey essential health information at a glance, while still enabling users to access more detailed data when required.

Connectivity Architecture:
The primary wearable communicates via Bluetooth with the user's smartphone, which serves as the gateway to cloud services. This approach delivers several benefits:

- Extended mobility: The wearable functions anywhere within Bluetooth range of the user's phone
- Power efficiency: Bluetooth Low Energy protocols maximize battery life

Each data packet from the wearable includes precise timestamps to ensure accurate chronological tracking of health events, essential for both immediate alerting and longitudinal analysis by healthcare providers.

References

Section 1 Overview and Problem Definition

Cho, Sylvia et al. “Factors Affecting the Quality of Person-Generated Wearable Device Data and Associated Challenges: Rapid Systematic Review.” JMIR mHealth and uHealth vol. 9,3 e20738. 19 Mar. 2021, doi:10.2196/20738’

Eyal, Maytal, and Bridget Freihart. "The Sorry State of Postpartum Care in America." Time, 19 Apr. 2024, time.com/6967372/postpartum-care-america-essay/. Accessed 19 Feb. 2025.

Gunja, Munira Z., et al. Insights into the U.S. Maternal Mortality Crisis: An International Comparison. The Commonwealth Fund, June 2024, www.commonwealthfund.org/publications/issue-briefs/2024/jun/insights-us-maternal-mortality-crisis-international-comparison. Accessed 19 Feb. 2025.

Matthews, Jared et al. “Cloud-Integrated Smart Nanomembrane Wearables for Remote Wireless Continuous Health Monitoring of Postpartum Women.” Advanced science (Weinheim, Baden-Wurttemberg, Germany) vol. 11,13 (2024): e2307609. doi:10.1002/advs.202307609

Nazarpour, Sima et al. “The Relationship between Air Pollution and Infant Mortality Rate.” Iranian journal of public health vol. 52,6 (2023): 1278-1288. doi:10.18502/ijph.v52i6.12994

Paladine, Heather L et al. “Postpartum Care: An Approach to the Fourth Trimester.” American family physician vol. 100,8 (2019): 485-491.

Sudhanthar, Sathyanarayan et al. “Postpartum depression screening: are we doing a competent job?.” BMJ open quality vol. 8,4 e000616. 13 Oct. 2019, doi:10.1136/bmjoq-2018-000616

Sun Y, Headon KS, Jiao A, Slezak JM, Avila CC, Chiu VY, Sacks DA, Molitor J, Benmarhnia T, Chen JC, Getahun D, Wu J. Association of Antepartum and Postpartum Air Pollution Exposure With Postpartum Depression in Southern California. JAMA Netw Open. 2023 Oct 2;6(10):e2338315. doi: 10.1001/jamanetworkopen.2023.38315. PMID: 37851440; PMCID: PMC10585409.

Section 5.2 Provider/User Software

[5.2.a] https://www.geeksforgeeks.org/top-front-end-frameworks/
[5.2.b] https://www.geeksforgeeks.org/frameworks-for-backend-development/
[5.2.c]https://www.computer.org/publications/tech-news/build-your-career/top-programming-languages-for-app-development
[5.2.d] Swift.org - About Swift
[5.2.e] Figma to Code - Export React, HTML & Vue from any Figma design
[5.2.f] Figma to HTML with Framer
[5.2.g] https://www.geeksforgeeks.org/what-are-the-key-features-of-node-js/
[5.2.h] https://chatgpt.com/share/67dcc23a-43cc-8010-8994-0d1b6b99ef52
[5.2.i] https://react-icons.github.io/react-icons/
[5.2.j] https://react.dev/learn/build-a-react-app-from-scratch
[5.2.k] https://github.com/facebook/create-react-app

Intelligence

Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. *Radiology: Artificial Intelligence, 3*(3). https://doi.org/10.1148/ryai.2021200126

Hartman, V., Zhang, X., Poddar, R., et al. (2024). Developing and evaluating large language model–generated emergency medicine handoff notes. *JAMA Network Open, 7*(12), e2448723. https://doi.org/10.1001/jamanetworkopen.2024.48723

Hassan, M. M., Mollick, S., & Yasmin, F. (2022). An unsupervised cluster-based feature grouping model for early diabetes detection. *Healthcare Analytics, 2*, 100112. https://doi.org/10.1016/j.health.2022.100112

Liu, R., Li, M., Zhao, S., Chen, L., Chang, X., & Yao, L. (2024). In-context learning for zero-shot medical report generation. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)* (pp. 8721–8730). Association for Computing Machinery. https://doi.org/10.1145/3664647.3680760

Liu, Z., Xu, X., Cui, Z., Rekik, I., Ouyang, X., & Sun, K. (Eds.). (2025). VIS-MAE: An efficient self-supervised learning approach on medical image segmentation and classification. In *Machine Learning in Medical Imaging. MLMI 2024. Lecture Notes in Computer Science (Vol. 15242, pp. XXX–XXX)*. Springer, Cham. https://doi.org/10.1007/978-3-031-73290-4_10

Xu, J., Lu, L., Peng, X., Pang, J., Ding, J., Yang, L., Song, H., Li, K., Sun, X., & Zhang, S. (2024). Data set and benchmark (MedGPTEval) to evaluate responses from large language models in medicine: Evaluation development and validation. *JMIR Medical Informatics, 12*, e57674. https://doi.org/10.2196/57674

Zekaoui, N. E., Yousfi, S., Mikram, M., & Rhanoui, M. (2023). Enhancing large language models' utility for medical question-answering: A patient health question summarization approach. In *2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA)* (pp. 1–8). IEEE. https://doi.org/10.1109/SITA60746.2023.10373720

Section 5.1.4 Service Design

References

Alfawzan, Najd, et al. "Privacy, Data Sharing, and Data Security Policies of Women's mHealth Apps: Scoping Review and Content Analysis." JMIR mHealth and uHealth, vol. 10, no. 5, 2022. ProQuest, https://www.proquest.com/scholarly-journals/privacy-data-sharing-security-policies-women-s/docview/2671791199/se-2, doi:https://doi.org/10.2196/33735.

Cox, David. "'They Thought They Were Doing Good but It Made People Worse': Why Mental Health Apps Are Under Scrutiny." The Guardian, 4 Feb. 2024, www.theguardian.com/society/2024/feb/04/they-thought-they-were-doing-good-but-it-made-people-worse-why-mental-health-apps-are-under-scrutiny.

"Mobile Health Apps and HIPAA." The HIPAA E-Tool, www.thehipaaetool.com/mobile-health-apps-and-hipaa/.

Schechner, Sam, and Mark Secada. You Give Apps Sensitive Personal Information. then they Tell Facebook. Wall Street Journal Testing Reveals how the Social-Media Giant Collects a Wide Range of Private Data from Developers; 'this is a Big Mess'. Dow Jones & Company Inc, New York, N.Y., 2019. ProQuest, https://www.proquest.com/blogs-podcasts-websites/you-give-apps-sensitive-personal-information-then/docview/2184617499/se-