BRAUDE
College of Engineering, karmiel

Software Engineering Department

Braude College of Engineering

Nappi - Digital Twin Based Platform for Monitoring and Improving Baby Sleep

Project Code: 25-2-D-22

Capstone Project Phase A - 61998

June 2025

Ofer Elzara 315824409

Oran Efroni 208741934

Supervisors:

Dr. Uzi Rozen

Git repository link:

https://github.com/oferElz/Nappi.git

Table of Contents

Overview

Sleep plays a critical role in the healthy growth and development of infants. However, many parents struggle to understand why their baby wakes up during the night, often lacking the tools and insights needed to monitor and improve sleep quality. Factors such as room temperature, humidity, and noise levels can significantly affect an infant's sleep, yet these influences are rarely measured or analyzed in real time. In addition, parents have little to no visibility into how their baby's sleep is evolving over days or weeks - whether it is improving, declining, or remaining stable. As a result, identifying the root causes of disrupted sleep and recognizing long-term patterns remain a challenge, making the process of improving sleep conditions largely a matter of guesswork.

This project introduces **"Nappi"**, a smart, real-time Digital Twin-based platform designed to monitor and enhance the sleep quality of infants. By integrating various sensors, including temperature, humidity, sound, and eye's state detectors, the system creates a comprehensive digital representation of the baby's sleep environment and behavior. These sensors are connected to a Raspberry Pi controller, allowing efficient data collection and communication between hardware, backend server and database. The platform includes a mobile application that provides real-time alerts and personalized recommendations to parents, based on the system's analysis of the collected data. In addition, historical sleep trends are visualized, enabling long-term tracking and optimization. The system also lays the groundwork for future integration of machine learning algorithms, with the goal of identifying complex patterns and predicting wake-up triggers with increasing accuracy.
'Nappi' aims to transform infant sleep care by providing a reliable, accessible, and intelligent solution that empowers parents with real-time insights and actionable advice. Moving from reactive to proactive sleep management.

1. Introduction

Sleep is a cornerstone of healthy brain and body development in the first years of life. Yet an estimated 20-30 percent of infants experience significant night-time awakenings that disrupt restorative sleep cycles [1]. Chronic sleep fragmentation has been linked to slowed cognitive growth, weaker immune response, and increased parental stress [2]. Many environmental factors can nudge a baby from deep sleep to wakefulness, but these influences are rarely tracked in real time inside the home. Although an expanding market of "smart" baby monitors promises help, most current devices focus on a single signal (audio, video, or wearable pulse-oximetry) and provide limited, event-by-event alerts [3]. Parents still lack access to structured, data-driven insights that reveal how and why their child's sleep patterns and quality evolve over time. Professional sleep studies, meanwhile, require in-clinic polysomnography with bulky electrodes, long wait-lists, and high costs typically ranging from 1000$ to 3000$. For instance, “sleep resolutions” offers such services, within this price range (link). These factors represent barriers that leave many families without access to actionable guidance.

To address these gaps, we thought about a Digital Twin-based approach. By collecting real-time data over an individual infant, our program could detect correlations between environmental changes and wake-up events, and track long-term sleep trends (improvement or decline) for each baby.

Our motivation is personal: From one of us whose sister recently had another child, and another whose cousins are new parents, both have faced months of sleepless nights and trial-and-error with traditional baby monitors, whose main purpose is to ensure child safety. That frustration inspired us to create a data-driven solution aimed at addressing infant sleep issues.

We believe this platform can reshape home-based pediatric sleep care, offering families actionable insights that were previously confined to clinical laboratories.

2 Background and Related Work

In this chapter, we research and examine the importance of infant sleep, the common biological and environmental factors that influence it, and the challenges caregivers face in monitoring sleep effectively. We also review relevant real-time baby monitoring systems and sensor technologies, which inform the development of our proposed digital twin-based solution for analyzing sleep behavior and providing data-driven support to caregivers.

2.1 Sleep as a Foundation for Healthy Development

To address infant sleep challenges and support healthy development, it is essential to first understand the role of sleep in early childhood, the common factors that disrupt it, and the practical difficulties caregivers face in monitoring and managing it.

2.1.1 Importance of Sleep in Early Childhood Development

Infancy is a period of rapid brain growth and physical development, and sleep plays a foundational role in these processes. Newborns spend most of their time asleep. The National Sleep Foundation recommends about 14-17 hours of sleep per day for newborns (0-3 months), decreasing to 12-15 hours for infants 4-11 months old.

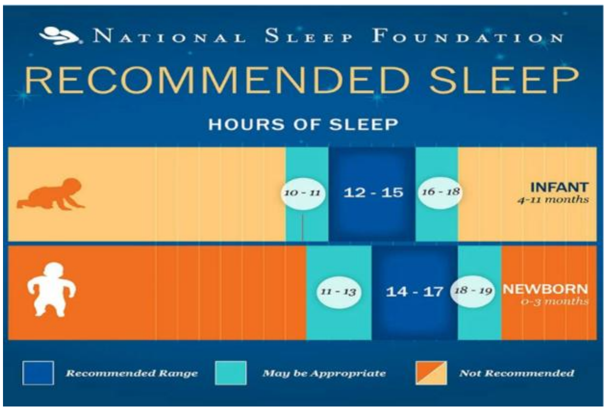


Figure 1. Recommended sleep ranges for newborns and infants

During sleep, an infant's brain processes the day's stimuli and stores new information. for example, one study found that infants who slept longer had improved working memory performance. Naps also contribute to learning - infants who napped after exposure to language patterns were better at retaining those patterns, indicating that frequent sleep periods aid long-term memory formation [2]. Adequate sleep is not only tied to cognitive gains but also to emotional and physical development. Researchers have observed that sleep quality affects infant's mood and emotional regulation, insufficient or fragmented sleep can lead to more negative emotions and difficulty in self-soothing. In fact, shorter sleep duration in the first years is associated with poorer emotional regulation in toddlers [1]. Physically, sleep contributes to healthy growth. growth hormone is primarily discharged during deep sleep, and consistent sleep has been linked to proper weight gain and lower risk of later childhood obesity [2]. In summary, ensuring infants get sufficient quality sleep is essential for their brain development, emotional well-being, and overall health during early childhood.

2.1.2 Common Sleep Disruptions and Their Causes

Despite its importance, an infant's sleep is easily disrupted by a variety of factors. In early infancy, sleep patterns are irregular by nature. Newborns do not have an established circadian rhythm, leading to day-night confusion where a baby may sleep more during the day and be awake at night [2]. Around 10-12 weeks of age, the infant's internal clock begins maturing, and sleep starts to consolidate more at night. As this biological rhythm develops, other interruptions to sleep can arise:

- **Feeding Needs:** Frequent waking is common in the first months of life due to hunger. Newborns have small stomach and require regular feeding, even at mid-night. By about 6 months of age, most infants are physiologically capable of sleeping through the night without food, yet **25-50%** continue to wake for feedings or out of habit [5]. Gradually, as their daily caloric intake increases, infants can go longer stretches at night without waking to eat.
- **Developmental Milestones and Sleep Regressions:** Infants sleep can temporarily worsen during phases of rapid development. Many parents observe "sleep regressions" at around 4 months, 8-10 months, and 12 months of age. For instance, in the first year, advances like crawling, standing, or increased awareness can disrupt sleep. Around 8-10 months, babies may experience **separation anxiety**, causing them to wake and cry for a parent's comfort. Similarly, at approximately 12 months, issues such as **teething pain** can lead to a period of increased restlessness and night waking. These regressions are usually temporary (lasting a few weeks) as the infant adjusts to new skills or overcomes the discomfort.
- **Illness or Discomfort:** Physical discomfort is a common sleep disruptor. An infant who is not feeling well may wake up more often or have trouble falling asleep. Digestive discomfort such as gas or reflux can also lead to midnight awakenings. Teething is another frequent culprit - when infants are cutting new teeth, the gum pain often spikes at night, resulting in more crying and waking. These kinds of disruptions are usually temporary and resolve once the underlying discomfort is addressed (the illness passes or the tooth emerges).

While the previously discussed causes of disrupted sleep - such as biological needs and developmental changes are natural and largely outside of our control, other factors like an over-stimulating environment (excessive light or noise at bedtime) or an inconsistent sleep routine are more manageable. Together, these influences mean that **disrupted sleep is a normal part of infancy**, though its severity can vary between children. Recognizing both the uncontrollable and controllable factors can help caregivers better anticipate and respond to sleep challenges

2.1.3 Challenges Faced by Parents in Monitoring Sleep

Frequent disruptions in infant sleep do not only affect the babies, they also take a toll on parents. Caring for an infant often means fragmented sleep for the caregiver. In fact, the sleep deprivation associated with a new baby is considered one of the most difficult aspects of early parenting

New mothers and fathers typically experience significantly reduced and irregular sleep in an infant's first year. This prolonged sleep disruption can leave parents exhausted and overwhelmed.

Such fatigue can impair parent's alertness and well-being, making it harder for them to keep up with daily responsibilities and safely care for their child.

2.2 Environmental Factors and Sleep-Related Indicators

Infants sleep patterns and quality can be significantly influenced by various external environmental conditions and the infant's own behaviors. Creating a proper sleep environment and understanding infant sleep behavior are crucial for ensuring healthy, restful sleep in early life. Factors such as room climate (temperature and humidity) and ambient noise directly impact an infant's comfort and ability to stay asleep, while behavioral cues like movement and eye activity correlate with different sleep phases.

2.2.1 Room Temperature, Humidity, and Noise Impact

Maintaining an optimal nursery temperature and humidity is **crucial for infant sleep**. Babies should sleep in a comfortably cool environment - many safe-sleep guidelines recommend a room temperature around **22-23°C** to prevent overheating [6]. Overheating during sleep is a known risk factor for **sudden infant death syndrome (SIDS)**, so keeping the room within this range helps lower that risk. Conversely, if the room is too cold or the air too dry, the infant may be uncomfortable and more likely to wake. Researchers suggest an optimal humidity level of about **35-50%** in the sleep environment, since overly dry air can irritate an infant's airways and skin - low humidity often causes coughing, congestion, or dryness that disturbs sleep [7]. On the other hand, extremely high humidity can make the room feel hotter and lead to discomfort or sweating, so a balanced moderate humidity is best.

In addition to climate, **noise levels** in the bedroom play an important role in infant sleep quality. Loud or sudden noises can startle a baby and interrupt sleep, especially during lighter sleep stages. To promote uninterrupted sleep, the ambient noise around a sleeping infant should be kept low. ideally around **50 decibels or below** - current recommended noise level for hospital nurseries [8]. Continuous noise above this threshold may fragment an infant's sleep and even pose risks to the baby's sensitive hearing.

2.2.2 Infant Movement and Sleep Phases

Infants experience distinct sleep phases, mainly divided by - **rapid eye movement (REM) sleep**, often called active sleep, and **non-REM sleep (NREM)** - often called quiet or deep sleep. These phases differ in both brain activity and observable behavior.

Newborns typically sleep 14 to 18 hours per day, with about half of that time [2], up to 9 hours- spent in REM sleep. The other half consists of NREM sleep. REM sleep is thought to support rapid brain and sensory development, while deep NREM sleep aids physical growth, tissue repair, and immune function. As infants grow, total sleep duration gradually decreases, and the proportion of REM sleep drops to around 30-35% by six months [9].

During REM sleep, infants tend to be relatively active: they may twitch or move their arms and legs, stretch, make small sucking movements, or even let out brief cries or grunts while still asleep. This corresponds to a lighter sleep state in which the infant can be more easily aroused. Breathing during REM can be irregular as well (with short pauses and faster bursts of breaths), reflecting the immature respiratory rhythm in this stage.

In contrast, during non-REM sleep - especially the deeper stages of non-REM - infants become very still. **Twitching and other movements cease** as the baby enters deep, restorative sleep, and breathing becomes steady [9]. In this quiet sleep state, the infant is much harder to awaken. These behavioral cues (active movement vs. stillness) are important because they indicate which sleep phase the infant is in at a given time.

An infant's sleep cycles are **much shorter** than average adult sleep cycle:

a newborn's cycle might last around **40-50 minutes**, cycling quickly between active (REM) and quiet (non-REM) sleep. This means that infants experience more frequent transitions and brief arousals throughout the night. Frequent movement and awakenings are a normal part of infant sleep. Recognizing this pattern can help caregivers differentiate between a baby who is momentarily active in light sleep versus one who is truly awake and in need of attention. Overall, infant movement patterns align with their sleep phases and cycles, making movement an important behavioral indicator of sleep state.

2.2.3 Eye Movement and Sleep Cycle Analysis

One defining feature of REM sleep is the presence of **rapid eye movements**, and these movements play a key role in analyzing an infant's sleep cycles. During REM (active) sleep, a baby's eyes periodically move beneath the closed eyelids - often visibly darting back and forth - whereas in non-REM sleep the eyes remain still [9]. Modern infant monitoring systems use non-invasive, camera-based IoT technology to track body and eye movements, allowing detection in real-time of both stages. In essence, the appearance of rapid eye movements serves as a biological marker signaling that the infant has entered a REM stage of sleep.

Tracking REM episodes helps map an infant's sleep cycles, which shift from frequent, short REM phases in newborns to longer non-REM intervals as they grow. Careful analysis of eye movement patterns, and thus REM and non-REM states, can reveal whether an infant's sleep cycles are developing typically for their age and help identify irregularities in sleep behavior.

2.3 Real-Time Monitoring Systems

Real-time monitoring plays a crucial role in infant care, providing continuous oversight that can detect problems as soon as they arise. Infants cannot verbalize discomfort or symptoms, so real-time sensor systems can alert caregivers to issues like irregular breathing or vital signs that may otherwise go unnoticed. By continuously tracking an infant's physiological signals and environment, these systems help ensure immediate intervention when needed and give parents peace of mind.

2.3.1 Sensor-Based Monitoring for Health Applications

Sensor-based monitoring is widely used in healthcare to continuously track patients' vital parameters in real time. Modern sensors can measure **heart rate**, **respiratory rate**, **temperature**, **oxygen saturation**, and other metrics, providing an ongoing stream of data about a patient's health status. In neonatal care, for example, hospitals use wired electrodes and probes to monitor an infant's heart rate and breathing and will sound alarms if those signals deviate from safe ranges.

There are also home-based medical devices for infants, for example, some infant could be categorized as an at-risk group for apnea - temporary cessation of breathing. A home apnea monitor is a valid solution. It uses chest electrodes or belts to sense the baby's breathing and heart rate, and it triggers an alarm if breathing stops or the heart rate falls outside preset thresholds.

Such sensor-based systems have been credited with saving lives by alerting caregivers to critical events (like apnea episodes) in real time [4]. Beyond infants, wearable health sensors such as glucose monitors or heart rhythm monitors, demonstrate how real-time data from the human body can be collected and analyzed to the benefits of the user. These examples emphasize the potential of sensor-based monitoring in health applications, where Real-Time data can prompt swift medical responses [4].

2.3.2 Integration of Real-Time Data Collection from Various Sensors

Building a real-time infant monitoring system often involves collecting data from multiple sensors simultaneously and funneling this data into a single controller. This integration is technically complex because different sensors may measure diverse parameters and communicate using different interfaces or data rates. For example, a comprehensive infant monitor might include a camera for video, a microphone for audio, a temperature/humidity sensor for the nursery climate, and a wearable pulse oximeter for the baby's vital signs. Coordinating these inputs requires careful system design to ensure that no data stream overwhelms the controller and that time-sensitive signals are captured without lag. Achieving efficient integration requires handling concurrent sensor polling without dropping critical signals. Strategies such as multi-threaded programming, input buffering and timestamping data ensure as an example, that an apnea event is not lost amid video processing. Another complexity to take in mind with Integration of Real-Time Data collection - some OS on controllers aren't truly real time, developers often add real-time extensions or delegate time-sensitive tasks to auxiliary microcontrollers. When properly implemented, data can be collected and integrated properly in real time, yielding a richer, more reliable picture of an infant's well-being.

2.3.3 Challenges in Real-Time Data Collection and Privacy

Implementing real-time data collection for infant monitoring comes with several challenges, both technical and ethical. Technical challenges include ensuring the accuracy, reliability, and immediacy of data. Sensors can produce false readings or trigger false alarms - for instance, some consumer baby monitors have been reported to alarm about low oxygen or heart rate when the infant is fine, leading to unnecessary parental anxiety. Minimizing false alarms as such requires careful use of sensors and intelligent signal processing to distinguish true danger signs from noise. Another technical hurdle is maintaining a robust real-time connection. Many modern monitors rely on Wi-Fi or Bluetooth to transmit data, any disruption in connectivity could lead to data loss or delayed alerts at a critical moment.

Beyond the technical issues, privacy and data security are major concerns in real-time infant monitoring. These systems inevitably handle sensitive data - audio and video feeds from the baby's room, biometric readings of the infant's vital signs, and possibly personal information is transferred. Real-Time Data Collection must also comply with Israel's Protection of Privacy Law, 5741-1981, which regulates the collection, storage and sharing of personal health information.

Without proper safeguards, such data could be vulnerable to unauthorized access or misuse.

To guard against these threats, strong security measures are essential. For example, using a platform like Firebase could act as a proper solution. Using Firebase Authentication for secure, multi-factor sign-in and Firebase Realtime Database with encryption protocols for data in transit and at-rest encryption.

User privacy can be further preserved by designing systems that keep ‘more safe’ data, for example- instead of a live feed camera, maybe IR sensor could get the same result while keeping full privacy of the infant. In summary, while real-time infant monitoring offers enormous benefits for safety, it must be implemented with careful attention to family’s privacy.

Each alert that a system provides in real time must be accurate and meaningful, and each bit of data collected must be treated with the same care as one would expect in a clinical setting.

2.4 Digital Twin Technology

Digital Twin has emerged as a cutting-edge approach in modern monitoring systems, creating digital twin of physical entities to enhance analysis and decision-making. This section introduces the concept of digital twins and their broad applications, discusses it’s benefits for personalized monitoring, and highlights their relevance in the context of infant sleep analysis.

2.4.1 Concept and Applications of Digital Twin Systems

A digital twin is essentially a virtual model of a physical object or system that is continuously updated with real-time data from its real-world counterpart. This dynamic linkage allows the digital twin to mirror the state of the physical system and enables simulations that can predict behavior or test changes virtually, with feedback loops informing improvements in the real system [16]. Digital twin technology originated in industries such as manufacturing and aerospace for asset monitoring and optimization, but its use has expanded to many fields. For example, healthcare has begun adopting Digital Twin for patient-specific modeling and scenario testing [10] (e.g. simulating treatments on a virtual patient).

2.4.2 Relevance of Digital Twin in Infant Sleep Analysis

Applying digital twin technology to infant sleep analysis holds considerable promise for improving monitoring and intervention in this sensitive domain. Infants have unique sleep patterns and cannot communicate discomfort or issues, which makes a real-time virtual model particularly valuable. While digital twin models are well-established for adult patients, similar models for infants have been limited until recently due to the distinct physiology of babies. New technologies now enable creating a digital twin system by combining continuous data collection from an infant’s environment (e.g. nursery sensors, wearables) with AI-based recommendations to adapt care in real time [16].

In practice, a baby's digital twin could integrate live data on the infant's sleep-wake cycles, movement, vital signs, and environmental factors (such as temperature or noise levels). Caregivers and clinicians could then simulate and analyze various conditions on this virtual infant - for instance, testing how adjustments in bedtime routines or room climate might affect the baby's sleep - By analyzing the collected data with machine learning, without disturbing the child. Researchers can even predict how environmental changes will affect them. The digital twin can thus help identify optimal conditions for safe and restful infant sleep and predict potential disturbances before they happen. Equally important, an AI-driven twin could provide timely alerts or suggestions allowing parents or healthcare providers to proactively respond. Parents can view positive or negative trends in their baby's sleep development by day/week/month, allowing them to step back from the day-to-day routine and monitor progress over time. In summary, the digital twin approach in infant sleep analysis enables personalized, preventative monitoring: it not only reflects an infant's current state in detail but also helps tailor the sleeping environment and care practices to support better sleep outcomes in the near future, for the baby.

2.5 Existing IoT Solutions for Infant Monitoring

Several commercial monitors already track aspects of a baby's sleep or vital signs. Below we outline five interesting products, summaries their capabilities, helping us to focus our product.

2.5.1 Owlet Dream Sock - Owlet

Wearable sock that uses photoplethysmography (PPG) to track pulse rate and SpO_2 in healthy infants aged 1-18 months (6-30 lb). Readings stream to the Dream App, which sends real-time alerts when values stray outside user-defined ranges and plots historical trends.

2.5.2 Nanit Pro - nanit

Wall\stand mounted 1080p camera with contact-free **Breathing Motion Monitoring**, cry detection, plus onboard temperature, and humidity-sensing.
Core features work without a subscription, but detailed sleep analytics require an "Insights" plan.

2.5.3 Miku Pro Smart Baby Monitor - miku

Using its **SensorFusion** technology to combine remote PPG and motion analysis, providing contact free respiratory-rate tracking alongside HD video, temperature, humidity, and sound data. no wearables or subscription needed.

2.5.4 Angelcare AC517 - Angelcare

Product consist of **wireless movement sensor pad** placed under the mattress plus a 5-inch touchscreen "parent unit" and video camera. Triggers alarms on prolonged stillness, monitors room temperature, and offers VOX (sound-activated) two-way audio.

2.5.5 Cubo Ai Plus - CuboAi

Bird-shaped Wi-Fi camera using computer vision to flag face-covering, rollover, "zone" exits, and true cry events. Adds temperature/humidity read-outs, contact-free breathing monitoring, auto photo capture, and sleep timeline summaries without extra fees.

2.5.6 Comparison with Our Application & Model

The comparison below aligns Nappi's evolving roadmap with how today's leading monitors currently deliver each capability:

Heart-Rate Sensing

- Other Monitors: Owlet Dream Sock uses a wearable sock to track pulse and oxygen, requiring direct skin contact.
- Nappi: Nappi will provide a non-contact pulse sensors for heart-rate monitoring and alerting - such as under-mattress sensor or camera-based heart rate detection.

Eye-State Detection

- Other Monitors: None of the major products we state above offer eyelid-open/closed detection.
- Nappi: NoIR camera + Dlib eye-aspect-ratio flags eye state in real time, enabling sleep/awake classification.

Noise Monitoring

- Other Monitors: Miku Pro and Nanit Pro include basic sound detection for crying alerts but do not log continuous noise levels for further analysis.
- Nappi: The microphone triggers alerts for cry/high noise events and records analog audio levels for detailed noise analytics and correlating environmental changes with wake-ups.

Temperature & Humidity Monitoring

- Other Monitors: Nanit Pro and Miku Pro report temperature and humidity but only issue simple threshold alerts, with no direct link to sleep disturbances.
- Nappi: Our sensor logs temp & humidity continuously and our backend, trigger alerts when necessary and correlates environmental shifts with wake-ups, highlighting specific triggers for parents.

2.6 From background work to suggested system

After reviewing infant sleep physiology, caregiver responses to our research questions, sensor technologies, and existing monitors, this section shows how Nappi plans to tackle the comprehensive sleep monitoring challenges discussed previously, while establishing a clear roadmap for advanced AI-driven and hardware-enhanced features.

2.6.1 Caregiver Survey Insights

To better understand parents current sleep tracking habits, identified gaps, comfort with technology, and informational needs regarding infant sleep, we conducted this survey as a foundation for designing Nappi's solution. Ten different caregivers took part in our survey, raw results available in our git repository.

Survey Questions Presented to Caregivers:

1. What tools or methods do you currently use to track your baby's sleep patterns, and why?
2. On a scale of 1-10 (1 = very low, 10 = very high), how strong is your need for continuous tracking of your baby's sleep hours and sleep quality?
3. What do you feel is missing in your current sleep tracking to know that your baby is okay throughout the night?
4. Have you ever documented or tracked your baby's sleep patterns over days or weeks?
5. Have you noticed if your baby's awakenings are related to noise or room temperature conditions?
6. On a scale of 1-10 (1 = not aware, 10 = very aware), How aware are you of temperature and humidity changes in the room during the night?
7. On a scale of 1-10 (1 = not uncomfortable, 10 = very comfortable), how comfortable are you with placing digital/internet-connected devices in your baby's environment?
8. What kind of information would you like to receive to know that your baby is sleeping safely?
9. On a scale of 1-10 (1 = very low, 10 = very high), How much would you like to receive weekly automatic reports on your baby's sleep quality?

Key Takeaways:

The survey responses highlight the individuality of each infant's sleep and each caregiver's approach. While some parents experience no major sleep issues, others either sense something is off but lack the tools to understand it, or struggle to notice meaningful patterns. A smaller group reports significant sleep challenges and actively seeks support. Common concerns include understanding sleep quality, identifying causes for awakenings, and monitoring environmental factors like temperature and noise.

2.6.2 Sleep Trend Analytics and Visualization Tools

Nappi's digital twin platform synthesizes multi-sensor data streams to create a comprehensive analytical framework that reveals meaningful correlations within the infant's sleep ecosystem, the system would include:

Multi-Source Data Integration

The platform will aggregate real-time information from various sensors **(**Heart-Rate, Noise, Temperature & Humidity, camera) throughout the sleep environment, creating a clear view of factors that may impact infant rest. By monitoring multiple parameters simultaneously, the system will detect correlations between environmental changes and sleep disturbances.

Pattern Recognition and Correlation Analysis

Through continuous data collection, Nappi will identify recurring patterns and establishes connections between different variables. This analytical approach helps caregivers understand which environmental factors most significantly influence their infant's sleep patterns.

Adaptive Insights Dashboard

The system will present these correlations through an intuitive interface that evolves with the data. Rather than overwhelming parents with raw metrics, the platform focuses on meaningful relationships and trends that can inform better sleep management decisions.

Night Sleep Overview

The system will provide the caregivers with sleep insights including periodically reports, recommendations, recent sleep session data and infant idle room metrics and will present all of those in a user-friendly UI. All in the goal to help caregivers improve their infant sleep over time.

2.6.3 Machine-Learning Research Path

As improvement for the future implementation of Nappi, we thought about integrating ML for data analysis, mainly three ML features that fit our data and hardware and can enrich our system:

Wake-Up Prediction (Next 5 Minutes)

Model: XGBoost (gradient-boosted trees) trained on features like heart-rate, recent noise bursts, and recent temperature shifts. XGBoost handles mixed numeric inputs efficiently and runs quickly on small CPUs, making it ideal for on-device forecasts.

Purpose: By continuously updating this model with our regularly collected sensor data, Nappi can forecast an infant's imminent awakening up to five minutes in advance. This early warning helps link specific triggers, like a sudden noise or a change in room climate - to the predicted wake-up, enabling caregivers to intervene proactively rather than reactively.

Advance Sleep-Stage Tagging (REM vs. NREM)

Model: A Bidirectional LSTM network trained on short windows of heart-rate variability and eye-closure flags. By processing each segment both forward and backward, the BiLSTM learns the rapid, irregular heart-rate and brief eye movements classify REM, versus the steady heart-rate and sustained eye closure of NREM.

Purpose: Once trained on our sleep recordings, this model tags each time segment as REM or NREM in near real time. Caregivers then get a clear hypnogram showing how long their baby spent in deep versus active sleep, making it easier to assess sleep quality and spot any unusual stage shifts.

Apnea-Style Pause Detection

Model: A One-Class SVM trained on normal heartbeat waveforms captured by the heart rate sensor (and optionally refined with camera-based heart rate detection). By learning the typical beat-to-beat pattern, the SVM defines a boundary around healthy rhythms.

Purpose: Once deployed, this detector flags any unusually long pause in the heartbeat as a potential apnea-like event. Caregivers receive an immediate alert if the pause exceeds the learned normal range, enabling quick intervention. Similar approaches in ECG-based apnea studies have demonstrated over 95 % precision in identifying true pauses.

2.6.4 Future Possibilities for Predictive Sleep Improvement

- **Light Monitoring**: Integrate a dedicated light sensor to continuously measure room light level and trigger an alert whenever nighttime light levels exceed the optimal threshold, helping caregivers maintain the dark environment that supports healthy infant sleep.
- **Air-Quality Monitoring**: CO_2 and VOC sensor to continuously track room air quality and trigger alerts whenever levels exceed healthy thresholds helping caregivers maintain fresh, clean air and insuring the baby's health.
- **ChatGPT-Powered Q&A Assistant:** Integrate ChatGPT API with our backend so that all data the system collected, parsed, and stored sensor data can be queried in natural language. Caregivers could ask questions like "What time should I start the bedtime routine?" or "Why did my baby wake up last night?" and receive context-aware, step-by-step recommendations drawn from their own infant's history.

3 Expected Achievements

In this chapter, we outline Nappi's expected achievements, highlighting its core innovations and measures of success. Nappi is a fully non-contact infant sleep monitor that combines under-mattress sensor, camera, audio sensors and temperature and humidity sensor. All raw data are pre-processed on a Raspberry Pi, then analyzed by our backend to deliver at-most accurate sleep\awake detection, real-time alerts for environmental dangers or heart-rate events, and environmental context correlation. Results and recommendations are presented in an intuitive Mobile app. Our design builds on the latest advances in contactless vital-sign monitoring, infant sleep classification, and scalable IoT architectures.

3.1 Outcomes

By combining data from multiple sensors as described, Nappi will continuously track sleeping patterns, pulse signals, and infant room environment (temperature, humidity, noise level). Caregivers will receive accurate sleep-awake timelines, as well as real-time alerts for potential heart-rate anomalies, and critical environment changes.

To power these features, we will implement efficient, optimized algorithms on our backend side, pulling data from sensors through Raspberry Pi controller.

Our algorithms will then classify sleep stages using AI and infer likely causes of awakenings, whether from noise, temperature shifts, or discomfort - while maintaining low latency and personal privacy by keeping raw data on the device.

This signal-processing pipeline is designed to perform reliably even under challenging conditions like poor-light conditions or infant movement.

Finally, Nappi's cross-platform Mobile app will present every finding in a user-friendly format. Frame-by-frame analysis ensures each sleep cycle sequence is annotated independently, and caregivers can explore periodic reports, view interactive charts, and receive individualize suggestions. This combination of precise sensing, real-time alerting, and intuitive presentation will empower parents with actionable guidance to improve their baby's sleep health.

3.2 Features

3.2.1 Contact-Free Multi-Modal Monitoring

Nappi uniquely combining under-mattress heart bit sensing, video capture, and room temperature/humidity/noise into one seamless system. No wearable tags required to build a complete, non-invasive picture of an infant's sleep.

3.2.2 Edge Processing & Privacy

All raw data streams are filtered and summarized on a bedside Raspberry Pi, preserving privacy, slashing bandwidth use, and delivering insights with minimal delay.

3.2.3 Real-Time Event Alerts

Nappi pushing instant notifications for heart-rate deviations, crying bouts, or environmental spikes as soon as they occur, empowering caregivers to respond without hesitation.

3.2.4 Environmental Context Correlation

What sets Nappi apart is its ability to use researches optimal conditions and AI analysis to seamlessly link sleep disturbances, like wake-ups or restlessness, to specific room conditions (temperature rise, humidity change, noise), helping parents pinpoint and eliminate environmental disruptors.

3.2.5 Intuitive Cross-Platform App

Our Mobile app delivers live dashboards, interactive periodically summaries, and push alerts in a smooth, user-friendly interface on both iOS and Android, ensuring parents would have an easy access to their infant sleep progress over time.

3.3 Criteria for Success

3.3.1 Awake-Sleep Identification

The system would distinguish between Awake and Sleep, providing a clear timeline of each phase. Success is measured by prototype tests, with a goal of correctly identifying 70% of them.

3.3.2 Critical Event Alerts

Nappi delivers timely Critical notifications while minimal missing events or false alarms. Alerts arrive on caregiver’s devices as soon as abnormal sensors data is detected, ensuring caregiver’s awareness and response.

3.3.3 Environmental Context Correlation

The system links sleep disturbances to notable shifts in temperature, humidity or noise that occur around the time the infant wakes or becomes restless, and then highlights the most likely environmental trigger in the report.

3.3.4 User Experience and Usability

The mobile app provides a clean dashboard with intuitive navigation. Caregivers can review reports and recommendation. Installation of Nappi would be easy and intuitive.

3.3.5 Cross-Platform Operation

Nappi works identically on both iOS and Android devices, all core functions are accessible across platforms.

3.3.6 Data Transparency and Trust

Every report includes easy-to-understand labels and brief explanations of detected events. Alerts are accompanied by context notes (e.g., “Brief breathing pause detected - room temperature rose”) to build confidence in the system’s insights.

3.3.7 System Stability and Reliability

Our system will run all through all sleeping period with minimal fuss. The system will handle device restarts or network connection cut off to prevent data lost.

4 Research/Engineering Process

We have split Nappi’s work into two clear phases: the first phase focused on studying infant sleep patterns, surveying parents, and designing our non-contact sensing framework. The second phase will be all about implementing our plans and designs building and integrating the Raspberry Pi, Sensors, FastAPI backend and Flutter app, and finally validating everything in real-world tests. This step-by-step approach lets us confirm our ideas, requirements and invest a lot of time and thoughts on our plan and architecture before moving into hands-on development.

4.1 Engineering research - Sensor Selection

In Nappi, we rely entirely on **external, non-contact sensors** that sit in the crib area to monitor an infant’s heart rate, eye state, sound environment, and room climate. Each sensor was chosen for its simplicity, reliability, and seamless integration with a Raspberry Pi edge device. At this stage of development, all selected sensors are based on our research and may be adjusted according to future needs that arise during more advanced development phases.

4.1.1 Under-mattress Piezo Vibration Sensor

We plan to use the SparkFun Piezo Vibration Sensor (ID 9196) a thin film strip that converts tiny mattress vibrations from the baby’s heartbeat into electrical pulses. Since the Raspberry Pi cannot read analog signals directly, we include an MCP3008 analog-to-digital converter plus a single pull-down resistor to bring those pulses into the Raspberry pi. Compared to complex under-mattress radar modules or pressure mats that drift over time, the piezo strip costs under five dollars and delivers a clear, continuous waveform covering each heartbeat.

We considered commercial systems like the BabySense 7 pressure pads, but those only alarm on no motion and do not provide any raw signal for true heartbeat extraction. In contrast, our piezo solution gives us a reliable, contact-free signal that we can filter in Python to detect every beat accurately.

4.1.2 Camera + IR Illumination

We decided to use the Raspberry Pi Camera Module NoIR paired with an 850 nm IR LED ring as our camera sensor due to its main features - works on the Raspberry Pi's CSI port, delivering 1080p, 30 FPS video in total darkness without any extra drivers or USB bandwidth issues. Unlike the Arducam 5 MP OV5647 NoIR board which, despite offering built-in IR LEDs and a motorized IR-cut filter, often needs custom drivers and capture frames at a lower rate. The official NoIR module natively supported by OpenCV, gives consistent low-light performance, and frees up USB ports for other tasks. This setup guarantees clear, head coverage for our Dlib eye-aspect-ratio.

4.1.3 Microphone Module (uxcell Voice Detection)

We chose uxcell voice-detection mic board because it gives us two handy outputs in one tiny package:

Instant Alerts: Its digital pin goes HIGH as soon as sound crosses a set level, so the Pi's GPIO (General-Purpose Input/Output pins) can fire an immediate high noise/infant cry alert.

Noise Logging: Its analog pin feeds into an MCP3008 ADC, letting us record exact volume levels over time.

At under $3 it's far simpler and cheaper than I²S MEMS microphones (which need special drivers) or USB sound cards (which add bulk and setup headaches), and it works straight away with standard Python GPIO and SPI libraries.

4.1.4 Temperature & Humidity Sensor (DHT11)

For monitoring the nursery's climate, we opted for the **DHT11** sensor. It reports temperature within ±1 °C and humidity within ±1 %, which is good enough to detect a deviation from the limits we set and provide clear data that we can present in case of awakenings.

The DHT11 uses a single data wire to talk to the Pi and runs on a tiny Python library, requiring no extra components or calibration. We considered the more precise DHT22 and feature-rich BME280, but for our needs the DHT11's low cost, low power use, and plug-and-play simplicity made it a better choice.

4.2 Technology research - Eye Detection Framework: Why Dlib?

Below is our detailed research for comparing available eye detection frameworks:

MTCNN - (Multi-Task Cascaded Convolutional Networks)

MTCNN is a face-detection model composed of the three small neural networks, P-Net, R-Net, and O-Net, which perform the detection process by first detecting the coarse face regions, followed by refining the detected regions and lastly marking five key points (eyes, nose, mouth corners) as well as the exact face boxes in each face.

Performing both detection and alignment in one process, MTCNN handles different face sizes, angles, and lighting better than simpler detector. The down side is that when you run all three deep nets on a CPU it will be very slow, usually only few frames per second so to get real time speed you will need a GPU. **In contrast**, Dlib processes frames at near-real-time on a Raspberry Pi with no special hardware, its built-in 68-point landmark predictor output's the exact same facial/eye coordinates that MTCNN does, but at a fraction of the CPU cost.

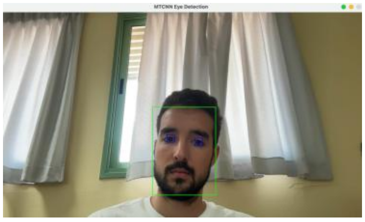


Figure 2. MTCNN Eye Detection

Viola-Jones - uses very simple black-and-white pattern checks (called Haar features) to slide across the image and instantly reject non-face areas, so it runs extremely fast even on a basic CPU. However, because it is only trained to spot clear, front-facing faces in good light, it often misses tilted heads, shadowed faces, or closed eyes. **In contrast**, Dlib's approach still runs near real-time on a Raspberry Pi, handles different angles and lighting, and with its landmark predictor reliably finds eyes whether they are open or close, making it a much better fit for our baby-monitoring needs.

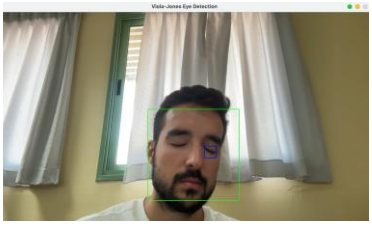


Figure 3. Viola-Jones Eye Detection

Dlib - provides two face detectors, as well as a landmark predictor all in a single package. The default HOG+SVM (**Histograms of Oriented Gradients + Support Vector Machine**) detector to find face-shaped patterns very quickly, on a small frame it takes just 0.2 seconds on a budget dual-core CPU, which is sufficiently fast for near-real-time operation on hardware like Raspberry Pi.

Dlib also includes a second, CNN-based detector called MMOD. MMOD uses a small neural network to find faces and figure out their exact angle even in bad lighting or when several faces appear together. It’s slower around 3 seconds per frame on a basic CPU. If you need that extra precision and can spare the extra time MMOD is the better choice. After being able to detect a face, Dlib’s 68-point shape predictor affixed key facial landmarks such as the corners of the eyes, tip of the nose and edges of the mouth instantly in one unified process.

Using Dlib's library alone we can detect faces and their landmarks without the aid of any other software libraries.

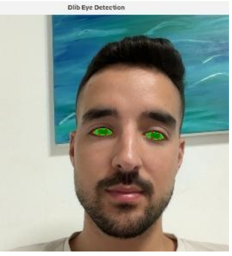


Figure 4. Dlib Eye Detection

Conclusion

Across benchmarks and real-world tests, dlib consistently balances speed, accuracy, and robustness better than MTCNN and Viola-Jones: its HOG+SVM detector achieves near real-time performance on CPU with minimal false positives, and its MMOD CNN variant and landmark predictor enhance detection quality and simplify downstream processing, securing dlib as the superior face and eye detection framework.

4.3 Design & application research - Three-Tier Architecture

In Nappi, the Raspberry Pi acts as an **edge device**: it collects and preprocesses sensor data, then streams clean JSON over HTTP to our backend. From there, we maintain a clear **three-tier** design that isolates concerns, maximizes performance, and simplifies development:

1. **Frontend Tier** - Flutter App

   Motivation**:** Flutter was selected as the development framework for the frontend tier due to our existing familiarity with the technology, allowing for rapid and efficient development.

   Key Advantages:

   Cross-platform: One codebase for iOS and Android.

   Hot reload: Instant UI updates accelerate design and bug-fix cycles.

   Rich widgets: Built-in charts and dashboards let us craft a polished interface quickly.

   Responsibilities: Fetch live and historical data via secure REST APIs.

   Display real-time alerts, sleep status, daily/weekly charts, suggestions, and analytics.

2. **Business Logic Tier** - Python & FastAPI Backend

   Motivation: Python's ecosystem covers our every need: vision, signal processing, web services, and database ORM while FastAPI delivers high-performance async endpoints with built-in validation and auto-generated docs.

   Libraries We Rely On:

   Dlib & OpenCV for face/eye detection.

   NumPy & SciPy for numerical and signal-processing routines.

   FastAPI for RESTful route definitions, async I/O, and Swagger UI.

   SQLAlchemy to map Python objects to PostgreSQL tables cleanly.

   Responsibilities:

   Receive JSON data sent by the Raspberry Pi. Use rules to detect sleep or wake, send alerts when needed, and match room changes (like noise or temperature spikes) to the baby waking up. prepare data to the Flutter app and insert it to the DB, handle authentication and error handling.

3. **Data Tier** - PostgreSQL Database

   Motivation: It's a mature, open-source database with strong ACID guarantees, ideal for time-series and session data.

   Key Advantages:

   Completely free and open-source: PostgreSQL has no licensing fees, so you can use and modify it without any cost.

   Highly stable and reliable: It's known for robustness and strong ACID compliance, keeping your data safe and consistent even under stress.

   Cross-platform support: Runs smoothly on Windows, Linux, macOS, and more.

   Responsibilities:

   - Save every sensor data coming from the backend and add precise timestamps to events.
   - Offer built-in tools to export raw or summary data in CSV/PDF formats and perform regular backups to keep caregiver's records safe.

Why This Architecture?

Separation of Concerns: The Raspberry Pi handles only sensor I/O and real-time filtering, the Python backend focuses on business logic and APIs, and the Flutter app concentrates on user interaction.

Independent Development: Each layer can be built, tested, and deployed on its own minimizing cross-work bottlenecks.

Resource Optimization: Low-level hardware code never burdens the server, compute-heavy vision and signal tasks run where they belong - and the UI remains lightweight.

Scalability & Maintainability: FastAPI's async model and PostgreSQL's robustness let us scale to many devices, while Flutter's single codebase and hot reload speed up feature delivery.

This three-layer approach delivers a **responsive, reliable**, and **easy-to-extend** baby-monitoring system that meets both caregiver needs and developer workflows.

4.4 Technology & application research - API & Communication

We chose FastAPI for our backend because it combines speed, simplicity, and rich features in pure Python. FastAPI key features:

Native Python & async I/O: FastAPI is built on Python's "async/await" coroutines giving a non blocking endpoints that can handle many concurrent requests without extra threads or processes.

Built-in validation & documentation: By using Pydantic models, FastAPI automatically checks incoming JSON against the schemas and generates interactive Swagger docs and data type check.

Low overhead: Benchmarks show FastAPI outperforms traditional frameworks like Django or Flask in raw HTTP request, yet remains far lighter-weight than socket-level or MQTT solutions.

Simple syntax: Defining routes, models, and security takes just a few lines of code much less than comparable setups in other frameworks.

Security & robustness: FastAPI comes with ready-made tools to keep your API safe. You can easily require each request to carry a valid token before granting access, force all traffic over encrypted HTTPS, and control which clients or browsers are allowed to call your endpoints. These built-in protections mean you don't have to write your own login or encryption and just use FastAPI abilities.

Alternatives considered:

MQTT - would require the integration of an additional broker component and comprehensive security validation. Moreover, due to its message queuing mechanism, increased latency can lead to delayed message delivery, reducing the system's real-time responsiveness.

GraphQL - adds query complexity and is overkill for our simple REST CRUD needs.

As described above, FastAPI have many benefits, it lets us define clean, async REST endpoints with automatic input checks and documentation, integrate security, and write minimal, maintainable code all while achieving high request performance and low latency. Therefore, we choose to use it for our communication protocol

4.5 Methodology and Development Process

We follow an **Agile** approach, organizing our work into focused sprints around each major component of Nappi's architecture: the edge device & sensors, the backend, the data layer, and the frontend.

Edge Device & Sensors:

Hardware Prototyping: wire the sensors (piezo strip, NoIR camera, IR LEDs, uxcell mic & DHT11) to the Raspberry Pi.

Calibration Sprints: Tune resistors and ADC settings for heartbeat, set IR exposure for eye detection, adjust mic and DHT11 thresholds.

Preprocessing Pipeline: Develop and test noise-filtering, Data extraction (eye-state, noise events, temp/humidity), JSON packaging, offline buffering, and self-health checks on the Raspberry Pi.

Backend Layer (Python + FastAPI):

Algorithms: implement rules and algorithms to detect sleep/awake, create sleep insights, generate suggestions and real-time alerts

API Development: Build FastAPI endpoints to ingest sensor JSON, run sleep/awake rules, trigger alerts, and correlate environment changes.

Library Integration: Wire in Dlib/OpenCV for eye detection and NumPy/SciPy for signal processing, adding validation and Swagger docs in each sprint.

Database integration: create routes and functions to efficiently insert and pull data from the database. make sure the backend is the only component that communicate with the database and it expose api's to all the others.

Data Layer (PostgreSQL)

Schema & ORM Setup: Define PostgreSQL tables and create ORM models for them, using toolkit such as SQLAlchemy.

Design the schema and relationships: Plan out each table, normalize to avoid duplication, and define primary/foreign keys so records link correctly.

Optimization: Write an efficient query and add indexes on commonly queried columns so dashboards and reports load instantly.

Frontend Layer (Flutter App)

UI: Build screens for live sleep status, daily/weekly charts, alerts, and sleep insights using Flutter’s widget library.

API Integration: Connect the app to FastAPI routes for live data streams, authentication flows, and report fetching.

Usability Iterations: Leverage hot reload to refine layouts, optimize animations, and incorporate caregiver feedback in rapid design cycles.

By aligning sprints to each architectural layer, we ensure focused progress, easy integration testing, and the ability to adapt based on early test and mentor feedback. delivering a robust, maintainable, and user-centered baby-monitoring system.

5 Product

The system architecture follows a classic **three-tier structure**: presentation, logic, and data.
This type of an architecture ensures a clean separation, supporting modularity, scalability, and maintainability. It consists of a mobile frontend (presentation tier), a backend server (logic tier), and a database layer (data tier), with clearly defined communication between each layer.
In addition to these core tiers, the system includes a **Raspberry Pi** functioning as an **IoT edge device**. This device interfaces with multiple environmental and physiological sensors, performs initial data preprocessing from varies sensors, and transmits the structured data to the backend for further analysis.

- **Presentation Tier (Frontend):** The mobile application enables parents to interact with the system, displaying personalized sleep insights, environmental data, system recommendations, all through an intuitive and user-friendly interface.
- **Logic Tier (Backend):** Serving as the core of the system, the backend receives and processes the incoming sensor data from the Raspberry Pi, performs validation and analysis, and provides real-time or historical insights to the mobile app via secure RESTful APIs.
- **Data Tier (Database):** All collected sensor data, sleep patterns and parameters, and user profiles are securely stored in a structured database. This layer supports analytics, trend tracking, and historical review.

This architecture ensures the system is both modular and extensible. It allows for seamless future integration of additional sensor types, advanced analytics, or AI-driven features, while maintaining clear separation between components to support independent development and maintenance.

5.1 IoT edge device

The Raspberry Pi serves as the IoT edge device, interfacing with multiple sensors to gather environmental and physiological data, perform initial filtering and video to frame to eye open\closed classification, and securely transmit structured data to the backend for deeper analysis. This functionality will be demonstrated in a POC on the next page.

5.1.1 Raspberry Pi Operations

Face & Eye Detection: First, frames are extracted from video input from camera using Python's cv2.VideoCapture (OpenCV) for precise frame capture. Then we leverage DLib to detect and crop the infant's face and eye regions from these frames. DLib was chosen for its robust accuracy and efficient performance, making it well-suited for real-time sleep monitoring.
Eyes open\closed State Classification: Calculate the Eye Aspect Ratio (EAR) from the six key eye landmarks on each cropped eye image to accurately determine whether the eyes are open or closed.

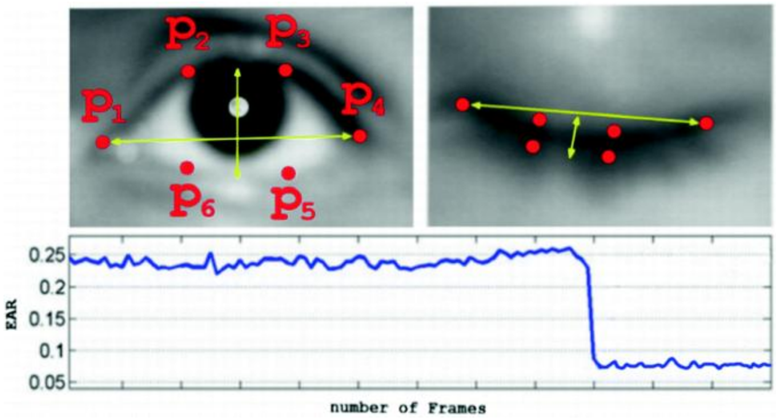
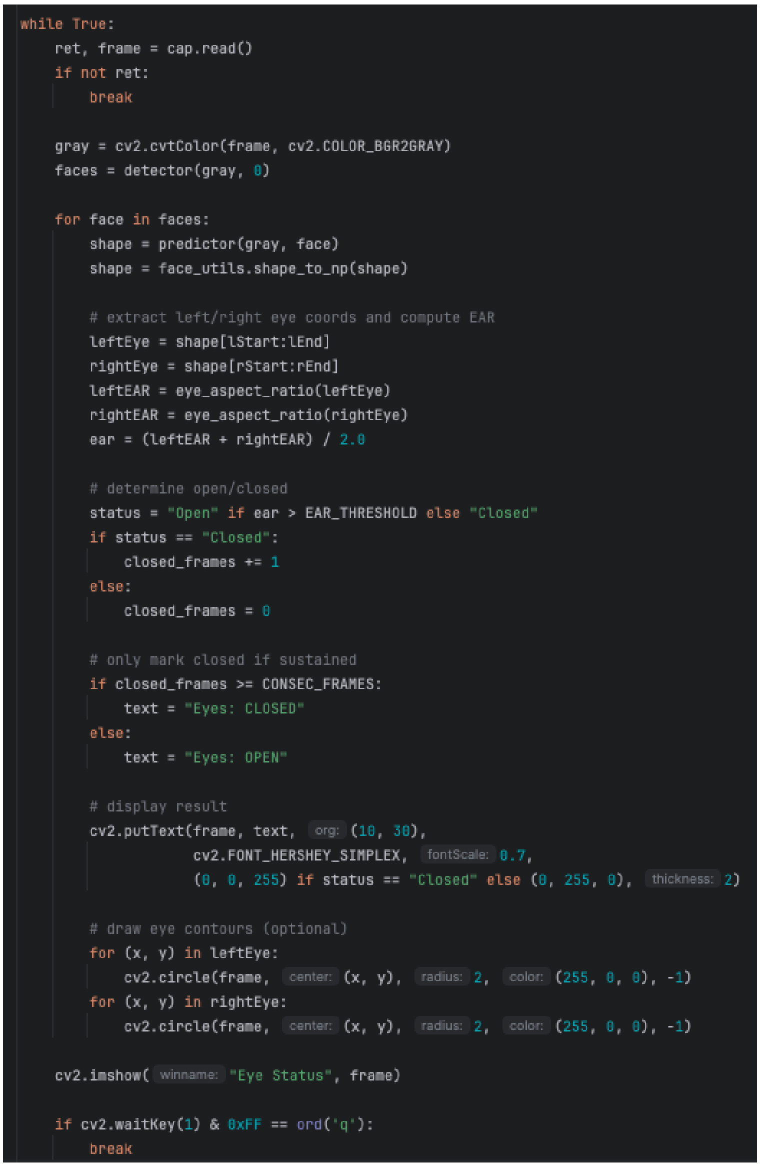


Figure 5. EAR remains consistently high while the eye is open, then sharply decreases when the eye closes.



```
while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray, 0)

    for face in faces:
        shape = predictor(gray, face)
        shape = face_utils.shape_to_np(shape)

        # extract left/right eye coords and compute EAR
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)
        ear = (leftEAR + rightEAR) / 2.0

        # determine open/closed
        status = "Open" if ear > EAR_THRESHOLD else "Closed"
        if status == "Closed":
            closed_frames += 1
        else:
            closed_frames = 0

        # only mark closed if sustained
        if closed_frames >= CONSEC_FRAMES:
            text = "Eyes: CLOSED"
        else:
            text = "Eyes: OPEN"

        # display result
        cv2.putText(frame, text,  org: (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX,  fontScale: 0.7,
                    (0, 0, 255) if status == "Closed" else (0, 255, 0),  thickness: 2)

        # draw eye contours (optional)
        for (x, y) in leftEye:
            cv2.circle(frame,  center: (x, y),  radius: 2,  color: (255, 0, 0), -1)
        for (x, y) in rightEye:
            cv2.circle(frame,  center: (x, y),  radius: 2,  color: (255, 0, 0), -1)

    cv2.imshow( winname: "Eye Status", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Figure 6. POC of our initial Eyes open\closed State Classification code

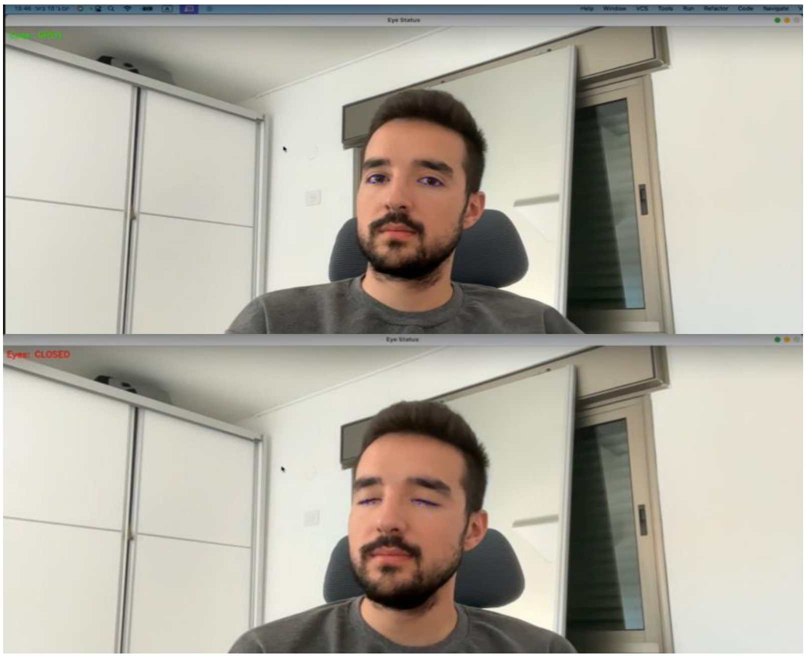


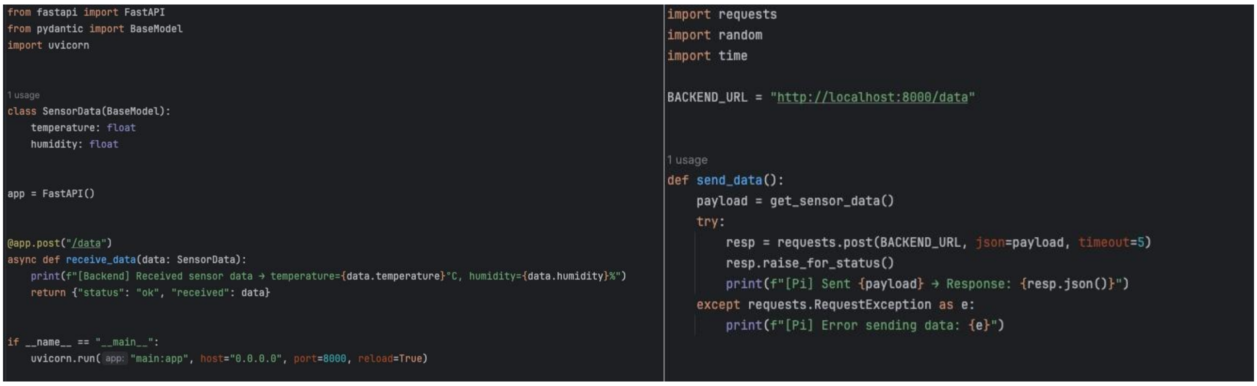
Figure 7. POC of our initial Eyes open\closed State Classification results.

5.2 Backend

The backend is our platform main mediator between frontend\Raspberry pi and our data database. It collects sensor data from the Raspberry Pi, processes and analyzes sleep-related and environmental metrics in real time, and exposes secure RESTful APIs for the mobile app to retrieve insights, trends, and alerts.

5.2.1 Backend Operations

Data Ingestion & Preprocessing: Using FastAPI, the backend defines secure API endpoints to receive real-time sensor feeds from the Raspberry Pi. Incoming payloads are validated against predefined schemas, then cleaned, timestamped, and normalized. Converting raw sleep and environmental readings into standardized data for downstream analysis.



```
from fastapi import FastAPI
from pydantic import BaseModel
import uvicorn


1 usage
class SensorData(BaseModel):
    temperature: float
    humidity: float


app = FastAPI()


@app.post("/data")
async def receive_data(data: SensorData):
    print(f"[Backend] Received sensor data → temperature={data.temperature}°C, humidity={data.humidity}%")
    return {"status": "ok", "received": data}


if __name__ == "__main__":
    uvicorn.run( app: "main:app", host="0.0.0.0", port=8000, reload=True)
```

```
import requests
import random
import time

BACKEND_URL = "http://localhost:8000/data"


1 usage
def send_data():
    payload = get_sensor_data()
    try:
        resp = requests.post(BACKEND_URL, json=payload, timeout=5)
        resp.raise_for_status()
        print(f"[Pi] Sent {payload} → Response: {resp.json()}")
    except requests.RequestException as e:
        print(f"[Pi] Error sending data: {e}")
```

Figure 8. Simple implementation demonstrating the use of fastAPI for communication between frontend\Raspberry pi (right image) & backend (left image)

Sleep & Environmental Insights: Identify the start and end of sleep sessions, detect wake-up events, correlate environmental changes with sleep disturbances, and track patterns in sleep quality and environmental conditions over user-defined timeframes.

Insight Generation: By combining real-time and historical data on sleep patterns and environmental conditions, the system identifies individualized trends and behaviors. It then generates relevant insights, such as adjusting room settings (temperature, noise level, light etc.)

Alerting: Monitor data against preset limits and send notifications for critical health issues, such as irregular pulse or unsafe room conditions, as well as system problems, such as a disconnected Raspberry Pi or malfunctioning sensors.

5.2.2 Backend Technologies

Python Libraries:

- **FastAPI**: Used as the asynchronous REST framework to receive sensor data from the Raspberry Pi.
- **OpenCV (cv2)** Handles night compatible camera frames: motion detection, segmentation, and preprocessing for downstream algorithms.
- **NumPy & SciPy** Provide core numerical routines and signal-processing tools (e.g. filters, spectral analysis) for various sensors data (audio, pulse, temperature etc.).
- **DLib** used for face and eye detection.

Database & Object Relational Mapping (ORM):

- **PostgreSQL** Stores relational data: user accounts, device metadata, alert logs, and the “digital twin” of each baby’s time stamped data.
- **SQLAlchemy** Provides an ORM layer (with optional raw-SQL support) for schema definitions, migrations, and clean interactions with PostgreSQL.

5.3 Frontend

The frontend offers a responsive, intuitive mobile app using **Flutter,** retrieves data from the backend and presents sleep insights, environmental metrics, and personalized recommendations in a clear, user-friendly layout.

5.3.1 Frontend Operations

Communication with Backend: The Flutter app interacts with the backend via secure RESTful APIs to retrieve sensor data, computed insights, alert statuses, and personalized recommendations in real time.

Insight Visualization and Trend Analysis: The app displays intuitive visualizations of sleep summaries, wake-up events, and environmental trends. Personalized recommendations are shown alongside historical data filters (daily, weekly, monthly) to help parents track and optimize their infant's sleep over time.

Alerts and Notifications: Push and in-app notifications inform users when abnormal readings are detected, such as out-of-range vitals, unsafe room conditions, or sensor/network issues, thus enabling timely parental response.

User Management and Settings: Users can complete registration, follow a simple installation guide, create and manage profiles for multiple infants using separate Nappi kits, set custom alert thresholds, and monitor connectivity status of their Raspberry Pi devices and sensor.

5.4 Requirements

5.4.1 FR Requirements

Table 1. Functional Requirements

No.	Requirement
1.	The system shall be cross-platform
2.	The system shall provide a user interface
3.	The system shall authenticate users
4.	The system shall support multiple infant's profiles under the same account
5.	The system shall receive real-time sensor data through Raspberry pi
6.	The system shall validate and timestamp all incoming sensor data
7.	The system shall preform preprocessing of data on raspberry pi
8.	The system shall measure heart rate
9.	The system shall capture frames from live video
10.	The system shall capture room audio level
11.	The system shall measure room temperature and humidity
12.	The system shall detect eyes position
13.	The system shall classify eyes as open/close
14.	The system shall detect the start and end of infant sleep session

No.	Requirement
15.	The system shall correlate environmental changes with detected sleep disturbances
16.	The system shall generate real-time alerts
17.	The system shall detect and notify on technical issues
18.	The system shall operate locally when network is unavailable
19.	The system shall expose secure RESTful API's
20.	The system shall provide analytics dashboard
21.	The system shall generate and display reports
22.	The system shall display personalized infant recommendations

5.4.2 NFR Requirements

Table 2. Non-Functional Requirements

No.	Requirement	Type
1.	Generate technical issues notifications for low battery, power loss, sensor failure, and network dropout.	Reliability
2.	Generate health-environment issues notifications for "no pulse found" Excessive room environment: noise (> 90 dB), temperature (< 20 °C or > 25 °C), humidity (< 30% or > 55%).	Safety
3.	Cache all sensor data locally when connectivity is lost and sync it automatically upon reconnection.	Reliability
4.	Allow users add, switch, and manage multiple infant profiles in one app session without data mix-ups.	Scalability
5.	Recommendations based on rules that continuously learn from each infant's historical sleep patterns.	Reliability
6.	Display sleep trends and events with clear, mobile-friendly visual components that load in under 2 s.	Performance
7.	Document RESTful API using Swagger, including rate limits, and response standards.	Testability
8.	Alert notifications reach the mobile app within 3 seconds from trigger moment.	Performance
9.	Sensor or Raspberry Pi disconnects are detected and logged within 30 s.	Maintainability
10.	Data is transmitted over secure, encrypted channels (HTTPS)	Safety
11.	Retain raw sensor data 30 days and aggregated metrics 1 year.	Maintainability
12.	Log all alerts, errors, and configuration changes with timestamps and IDs.	Maintainability
13.	Multiple caregivers can access the same account concurrently without data conflicts.	Reliability

No.	Requirement	Type
14.	Run seamlessly on both ios and Android devices	Scalability
15.	Store only processed states (awake / asleep / moving / still). never save raw video or frames.	Safety
16.	Maintain eye-state detection accuracy of ≥ 80 % for open vs. closed classification.	Reliability
17.	Provide time-range filters in all reports and analytics views.	Usability
18.	Log every awake-to-sleep or sleep-to-awake transition and notify the user immediately.	Reliability
19.	Present sleep-improvement advice strictly as non-prescriptive suggestions.	Reliability
20.	Support authentication via email + password, SMS one-time code, or a combination of both.	Safety

5.5 Architecture Overview

The process begins at the edge, where four sensor streams: under-mattress motion, camera, microphone, and temperature/humidity. All sensors continuously capture raw data (heart-rate, video frames, audio levels, and environmental readings). The system remains in an **idle** state when no infant is detected in the bed and switches to a **work** state as soon as the motion sensor registers the baby's presence. Each stream is then preprocessed on the Raspberry Pi:

Video frames are sampled at short intervals and analyzed with OpenCV and DLib to detect and crop eye regions, those crops are then evaluated using the Eye Aspect Ratio (EAR) to classify eyes as open or closed, triggering events on each state change.

Audio is scanned for crying, triggering caregiver alerts when detected, and continuously sampled to monitor room noise levels.

Temperature and humidity are also sampled continuously, and alerts are sent whenever either value falls outside the normal range.

Heart-rate data from the under-mattress sensor is sampled and monitored continuously to detect abnormally low or high rates, sending alerts whenever readings fall outside the normal range.

At fixed intervals, these extracted features are bundled into JSON messages and sent via FastAPI to the backend. Dedicated microservices then store the raw data in the database. Additionally, the backend waits for an awake-sleep trigger to mark the start of a sleep session. Upon the next trigger, it computes and saves key metrics such as - total sleep duration, wake-up counts, and associated environmental conditions. Then it correlates each awakening or sleep onset with environmental changes. The backend also issues real-time alerts for critical events.

On the user side, a Flutter mobile app is used to monitor infant sleep patterns, analytics, and more. When a user requests their infant's data, the app fetches it via secure RESTful APIs and presents caregivers with interactive dashboards, customizable time-range charts, and personalized sleep-improvement suggestions. Backend-generated triggers also appear in real time on the app.

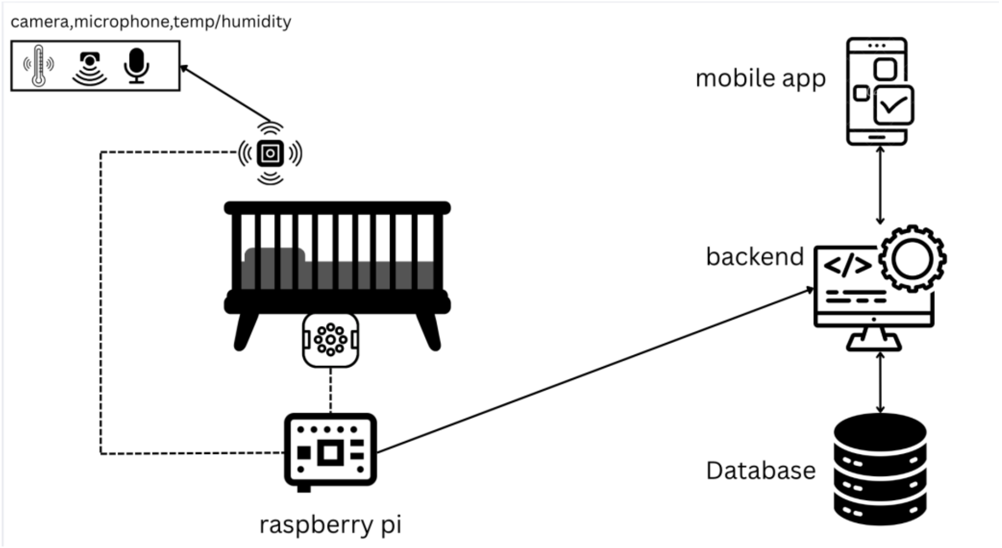


Figure 9. Architecture of Nappi

5.6 System Flow Chart

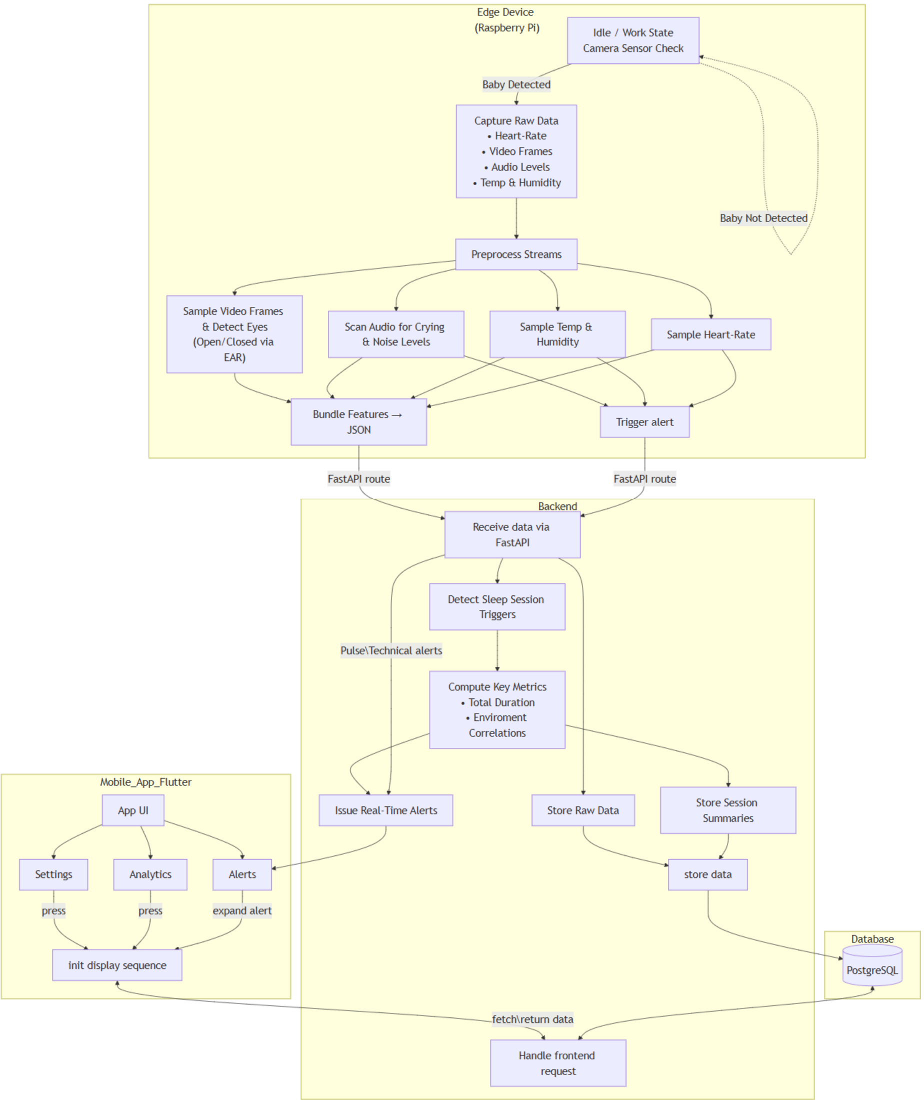


Figure 10. System's flow chart

5.7 System Use Case

The use case diagram details the system in general, the actors who take part in the processes in the system, and the actions the user can perform.

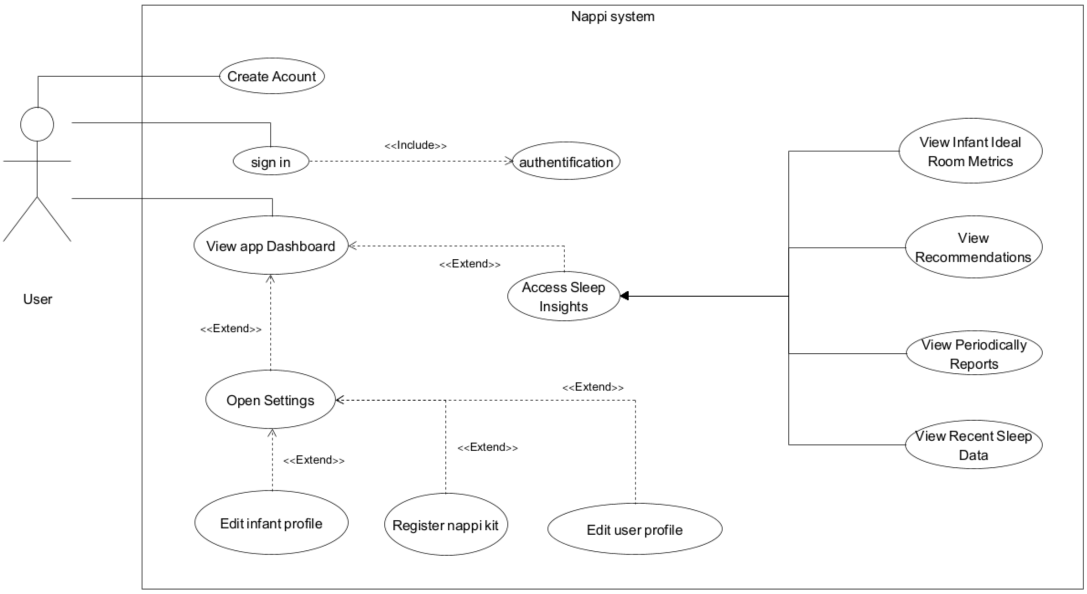


Figure 11. System's use case

5.8 System Activity Diagram

Our activity diagram illustrates the system’s main internal processes, assigns each action to its responsible actor, and sequences the steps from start to finish.

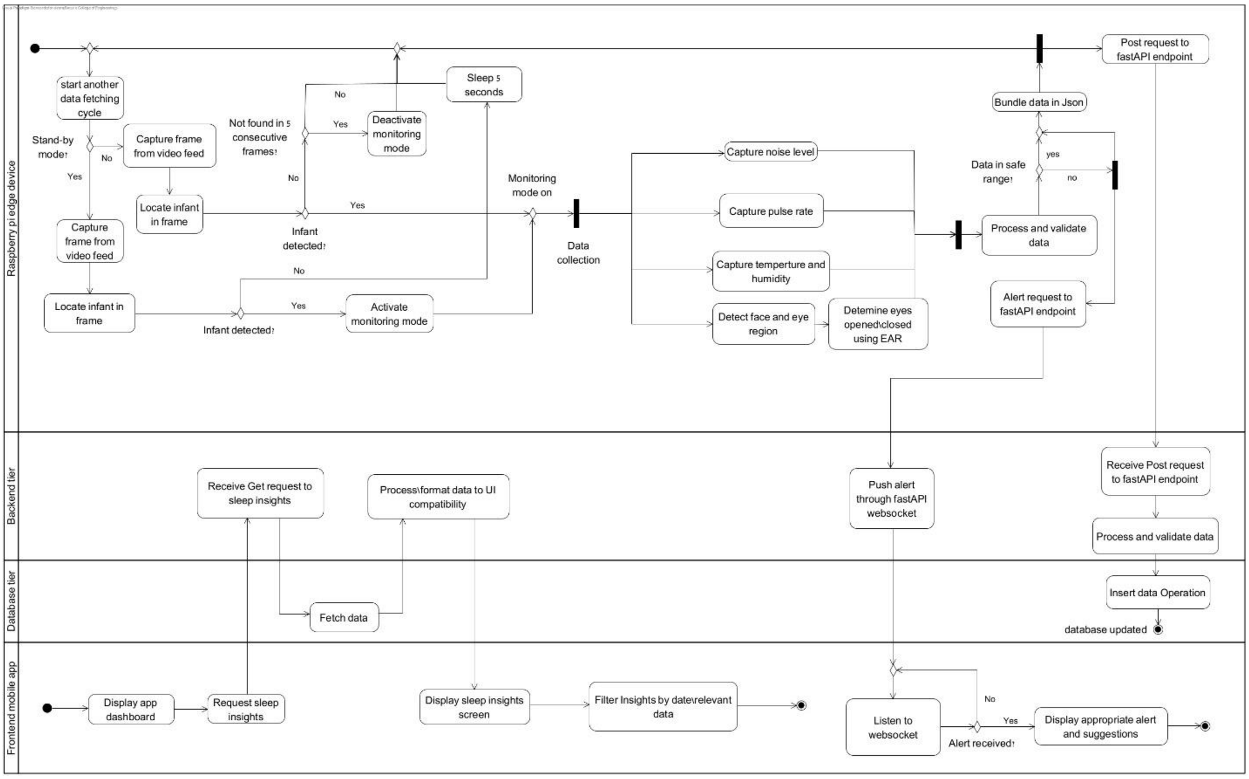

Raspberry pi edge device
start another data fetching cycle
Stand-by mode?
No
Yes
Capture frame from video feed
Locate infant in frame
Capture frame from video feed
Locate infant in frame
Infant detected?
Yes
No
Activate monitoring mode
Not found in 5 consecutive frames?
Yes
No
Deactivate monitoring mode
Sleep 5 seconds
Infant detected?
Yes
No
Monitoring mode on
Data collection
Capture noise level
Capture pulse rate
Capture temperture and humidity
Detect face and eye region
Detemine eyes opened\closed using EAR
Process and validate data
Data in safe range?
yes
no
Bundle data in Json
Post request to fastAPI endpoint
Alert request to fastAPI endpoint
Backend tier
Receive Get request to sleep insights
Process\format data to UI compatibility
Push alert through fastAPI websocket
Receive Post request to fastAPI endpoint
Process and validate data
Database tier
Fetch data
Insert data Operation
database updated
Frontend mobile app
Display app dashboard
Request sleep insights
Display sleep insights screen
Filter Insights by date\relevant data
Listen to websocket
Alert received?
No
Yes
Display appropriate alert and suggestions

Figure 12. System’s activity diagram

5.9 System User Interface

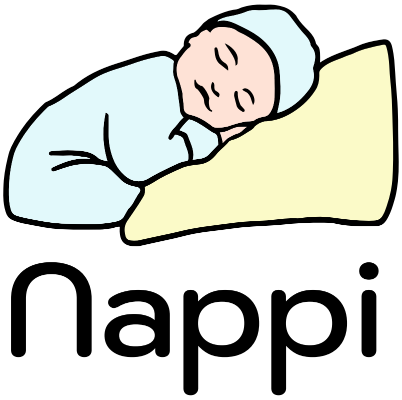
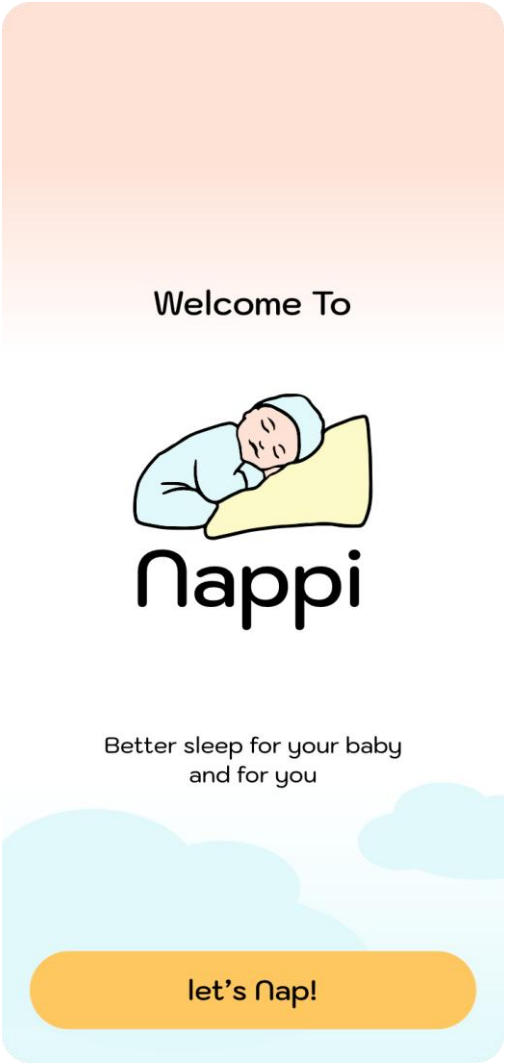


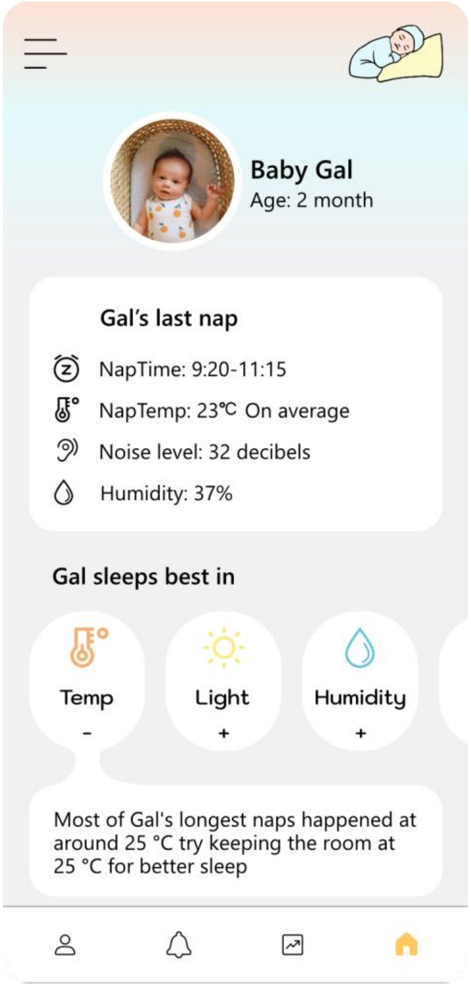
Figure 13. Nappi’s Logo



Welcome To

Better sleep for your baby
and for you

Figure 14. Nappi’s ‘On boarding’ UI



Baby Gal
Age: 2 month

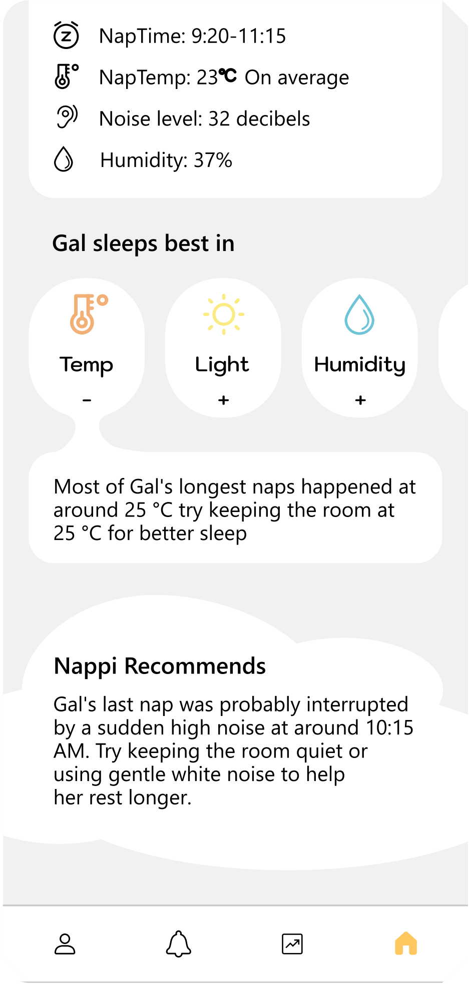
Gal’s last nap

NapTime: 9:20-11:15
NapTemp: 23℃ On average
Noise level: 32 decibels
Humidity: 37%

Gal sleeps best in

Most of Gal's longest naps happened at around 25 °C try keeping the room at 25 °C for better sleep

Figure 15. Nappi’s scrollable homepage UI



NapTime: 9:20-11:15
NapTemp: 23℃ On average
Noise level: 32 decibels
Humidity: 37%

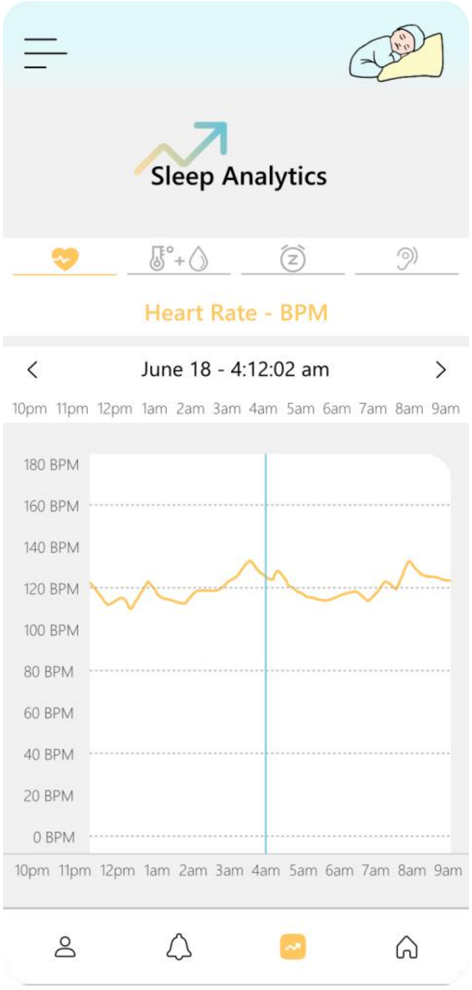
Gal sleeps best in

Most of Gal's longest naps happened at around 25 °C try keeping the room at 25 °C for better sleep

Nappi Recommends

Gal's last nap was probably interrupted by a sudden high noise at around 10:15 AM. Try keeping the room quiet or using gentle white noise to help her rest longer.

Figure 16. Nappi’s scrollable homepage



Sleep Analytics

Heart Rate - BPM

June 18 - 4:12:02 am

10pm 11pm 12pm 1am 2am 3am 4am 5am 6am 7am 8am 9am
180 BPM
160 BPM
140 BPM
120 BPM
100 BPM
80 BPM
60 BPM
40 BPM
20 BPM
0 BPM
10pm 11pm 12pm 1am 2am 3am 4am 5am 6am 7am 8am 9am

Figure 17. Nappi’s Sleep Analytics UI

6 Verification and Evaluation

6.1 Evaluation

The primary goal of Nappi is to build a non-contact infant monitoring system that parents can trust. We will evaluate success across two main dimensions: **functionality** and **user experience**.

Functionally, we will assess how well Nappi identifies sleep versus wake periods, detects heart-rate events, recognizes environmental changes in temp/humidity and noise, and links these changes to disturbances to awaking events all while delivering real-time alerts.

For user experience, we will provide a clear and east to use mobile interface, and the usefulness of its recommendations. We will also test robustness under real-world challenges low light, baby movement, brief network or power interruptions and verify that Nappi handles sensor errors or missing data with clear, actionable messages. Our evaluation combines lab bench tests, and in-home pilots, with all findings documented to ensure Nappi meets its objectives and provides a dependable, practical solution for families.

6.2 Verification

Because we build Nappi in iterative sprints, we treat each major component as its own testable module:

- Edge Pipeline (sensors + Raspberry Pi).
- Business Logic (Python/FastAPI backend).
- Data Layer (PostgreSQL).
- Frontend (Flutter app).

We will first validate each piece on its own using unit tests and manual QA to confirm it meets its functional requirements. After that, we will perform integration and system-level testing checking end-to-end data flow, performance under load, and real-world usability to ensure the complete baby-monitoring workflow works reliably.

In the next page a table with our tests for each module:

Table 3. System's verification using a test table

No.	Module	Test	Test method	Expected Result
1	Sensors & Raspberry Pi	Sensor integrity	Turn on all sensors and check output	All sensors report readings every interval for 1 hour without gaps
2	Sensors & Raspberry Pi	Sensor Failure Handling	Plug out a sensor	Raspberry pi triggers alert about sensor failure
3	Sensors & Raspberry Pi + Business Logic -Backend	Out of safety range metrices detected	Increase sound/temp/humidity	Sensor detects the change and the backend generated real-time alert
4	Business Logic -Backend	Data Type Validation	POST a JSON payload where a known field is sent as a string instead of a number	API responds with an error indicating it must be int and no data is saved.
5	Business Logic - Backend	Sleep/Awake Decision	Use the camera on one of use and test the sleep detection according to our algorithm	The backend will decide correctly sleep/awake according to the cases
6	Sensors & Raspberry Pi	Record Persistence	Simulate network outage. on reconnect, verify all buffered data are passed to the backend	No data loss. all data passed when network reconnected.
7	Data Layer - PostgreSQL	Query Performance	Execute typical dashboard and report queries.	Queries complete in under 2 s.
8	Frontend - Flutter App	Authentication Flow	Attempt login with valid and invalid, observe UI responses.	Valid login succeeds, invalid show user-friendly errors.
9	Frontend - Flutter App	UI Navigation & Clarity	Ask test users to complete key tasks without guidance.	80% or more of user’s complete tasks successfully on first try.
10	Sensors & Raspberry Pi	Temperature Sensor Functionality	Increase the heat near the sensor	The sensor will detect this increase and alert will be generated
11	Sensors & Raspberry Pi	Camera Face Detection	Test on our face under low, medium, and bright lighting and verify face detection.	The system correctly detects the face in all tested lighting levels.
12	Sensors & Raspberry Pi	Presence Detection	Put a doll in the crib and remove it.	Presence flag toggles accurately when the doll is in the crib and missing flag toggles when the crib is empty.
13	Sensors & Raspberry Pi	Difficult positions presence detection	Put a doll in the crib on the stomach	The camera will not detect the face but will detect the baby's presence

No.	Module	Test	Test method	Expected Result
14	Business Logic -Backend	Schema Validation	POST a JSON payload that omits a required field or includes an extra field not in the schema	API responds with an error saying that there is a missing or unexpected field names and payload is rejected.
15	Frontend - Flutter App	UI Latency	Tap on a button in the dashboard and measure the time until new data is displayed.	Updated data appears within 2 seconds of the user action.
16	Data Layer - PostgreSQL	Multi-Kit Support	Register a second kit via the UI	Both kits appear correctly under the same account but return independent data sets.

7 References

[1] Jodi A. Mindell, Brett Kuhn, Daniel S. Lewin, Lisa J. Meltzer, Avi Sadeh, “Behavioral treatment of bedtime problems and night wakings in infants and young children: An American Academy of Sleep Medicine review.” Sleep, vol. 29, no. 10, 2006, pp. 1263-1276.

[2] Elaine KH Tham, Nora Schneider, Birit FP Broekman. “Infant sleep and its relation with cognition, growth: a narrative review.” National Library of Medicine, Nature and Science of Sleep, vol. 9, 2017, pp. 135-149.

[3] Helen L. Ball & Alice-Amber Keegan “Digital health tools to support parents with parent-infant sleep and mental well being”. npj Digital Medicine, vol. 5, 2022.

[4] Rancea, A., Anghel, I., & Cioara, T. (2024). “Edge computing in healthcare: Innovations, opportunities, and challenges”. 16(9), 329.

[5] Bhargava, S. (2011). Diagnosis and management of common sleep problems in children. Pediatrics in Review, 32(3), 91-101.

[6] “Optimizing Temperature in the Baby's Room”, Israel Ministry of Health.

[7] “Your health and humidity”, article was reviewed and supported by AAP, Asthma and Allergy foundation of America. Published at Boston Children’s Hospital.

[8] Sarah C Hugh, Nikolaus E Wolter, Evan J Propst, Karen A Gordon, Sharon L Cushing, Blake C Papsin. “Infant sleep machines and hazardous sound pressure levels” .

[9] Tarullo, A. R., Balsam, P. D., & Fifer, W. P. (2011). “Sleep and infant learning. Infant and Child Development”, 20(1), 35-46.

[10] dibi, S., Rajabifard, A., Shojaei, D., & Wickramasinghe, N. (2024). Enhancing healthcare through sensor-enabled digital twins in smart environments: A comprehensive analysis. Sensors, 24(9), 2793.