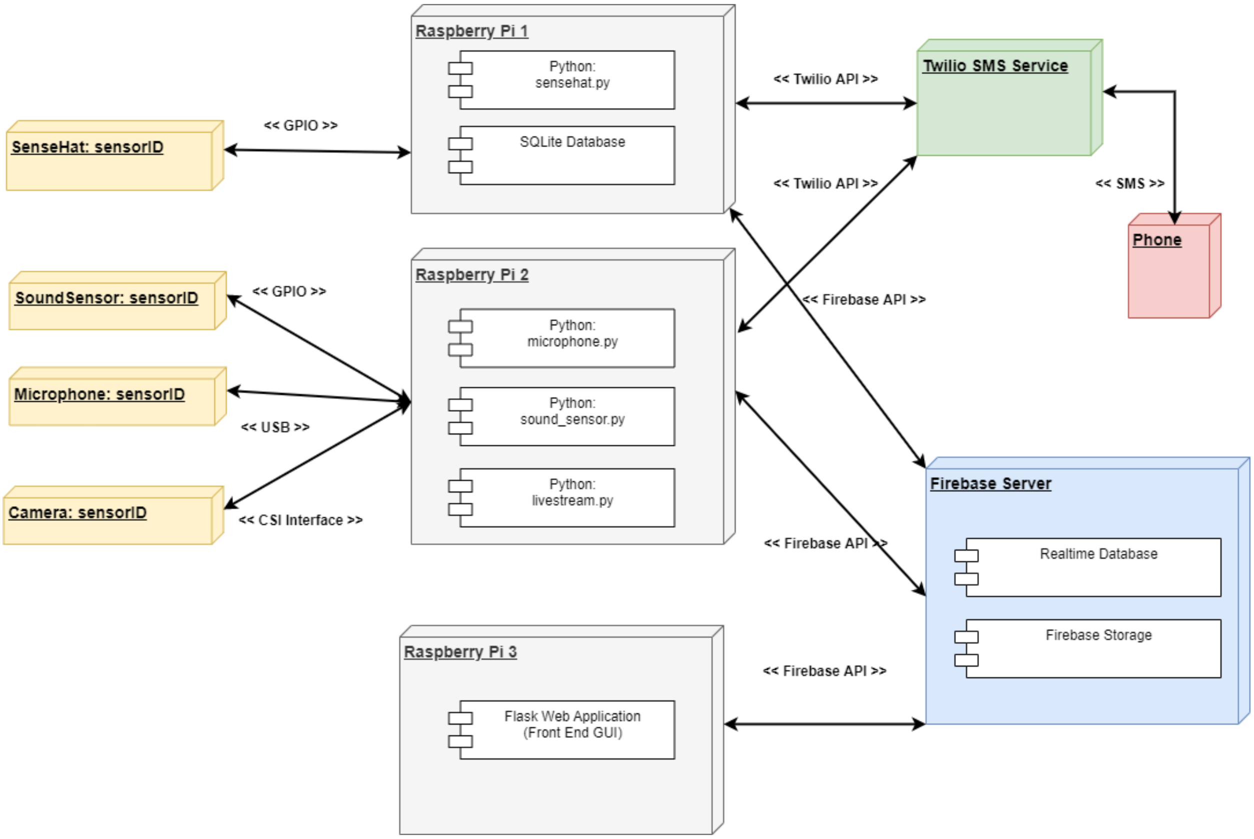
1 Deployment Diagram



2 Final Demo Outline

Introduction - Kenny

- Functionality:
    - View and monitor baby remotely from Flask app. Allows for passive-monitoring while doing daily activities inside house. Additional level of safety/security for baby while providing flexibility for parents.
- Quality:
    - Simple and easy to understand GUI, minimal setup, easy to add additional functions regarding detection (extend/add it onto Sound Sensor sound detection script), less than 150 lines of code makes for an efficient application. No additional overhead from using Web Development frameworks adds to performance (only HTML and CSS used).
- Utility:
    - Project is useful for parents. Applicable to real-life problems.

Goal of Project:

- Make a simple, reliable, and easy to use Smart Baby Monitoring system for the busy parent. To enable passive monitoring of babies and to enable parents flexibility and allow for more freedom in the household.

Part 1 - Kenny

- Demo device2_monitor.py
    - Sound Sensor detects high audio level
        - Notification sent to phone
        - Firebase database updated
        - Audio & video start recording concurrently on 2 processes for 10 seconds
            - .h264 converted into .mp4
            - .wav and .mp4 rendered into one .mp4 (audio and video channels)
        - .mp4 video file uploaded to Firebase database
            - Firebase database “recordings” entry updated

Part 2 - Chris

- Demo write to firebase database
    - Demo sensehat simulation for notification when temperature exceeds threshold
    - Demo sensehat button press
    - Demo write to local SQL database

Part 3 - Cristian

- “Get Data”: Show how we can get data from firebase, both temperature and sound. Point to the data Chris and Kenny entered in their demos
- “Set thresholds”: Explain what it does, will use it in chris’ part of the demo
- “Livestream”: show livestream hosted on the same pi
- “Recordings”: Show querying recordings, show latest recording from Kenny’s demo.

Project Milestones Completion - Cristian

Milestones

Technical Project Milestones	Description	Date (YYYY-MM-DD)	Completion
SenseHat -> Pi - > Firebase DB connection	Pi is able to query sensor data from SenseHat and insert it into the Firebase DB	2022-03-09	**Completed 2022-03-10**

established			
User can retrieve data through Front End GUI	Users are able to request sensor data through the Flask GUI. The Flask GUI retrieves this data from the Firebase DB and displays it.	2022-03-10	**Completed 2022-03-10**
User can set thresholds through Front End GUI	Users can set a temperature threshold in the Firebase DB through the Flask GUI. The Pi will see this change update the temperature detection threshold	2022-03-15 (Deadline advanced)	**Completed 2022-03-10**
Pi can store audio/video in Firebase storage.	The Pi can take audio and video recordings and upload them to Firebase storage.	*2022-03-25* (Moved up in priority)	**Completed 2022-03-10**
Pi can notify users.	The Pi can notify users when a) The temperature has surpassed pre-defined thresholds and b) The sound sensor detects high noise levels.	2022-03-18	**Completed**
Pi can host a live stream.	The Pi can host a video live stream that is remote viewable.	2022-03-22 (Deadline postponed)	**Completed**
User can view live stream through Front End GUI.	Users can view a live stream being hosted on a separate Pi, through the Front End GUI.	2022-04-01 (Deadline postponed)	**Completed**
User can view audio/video through Front End GUI.	Users can request to see audio or video recordings through the Flask GUI. The Flask GUI retrieves these files from Firebase Storage and displays them.	2022-04-02 (Deadline advanced)	**Completed**

Functional Requirements

Functional Requirement	Yes/No	Implementation in Project
6.1 Is there at least one computer per student in the group?	Yes	3 RPis.
6.2 Is at least one computer in headless mode?	Yes	3 RPis.

6.3 Is there at least one hardware device per student in the group with at least one actuator and one sensor?	Yes	Sound Sensor, Microphone, Pi Camera, SenseHAT
6.4 Is there an actuator? (Not every hardware device is a sensor)	Yes	SenseHAT button
6.5 Is there a feedback loop? (Interaction between input/output devices)	Yes	Pi 3 change min/max temp values. Pi 1 detects these changes. Pi 3 then displays updated values. Pi 3 is able to view video and audio recordings from Pi 2.
6.6 Is there a database with at least two tables and does the computer hosting the database have other responsibilities?	Yes	Pi 1 hosts database and also hosts SenseHAT night-light/button script.
6.7 Is there a periodic timing loop?	Yes	Pi 2 constantly monitors ambient environment (sound sensor can only trigger once every 60s).
6.8 Is there some processing or analysis of the IoT data read?	Yes	Pi 3 reads from database to display sensor data + video recording files.
6.9 Are notifications sent (SMS and/or email)?	Yes	Twilio. Pi 2 sends SMS when high sound level detected. Pi 1 sends SMS when temperature min/max values are changed.
6.10 Is there at least one GUI running on a computer?	Yes	Pi 3 is running Flask app with web GUI.

Reflection / Future Work

- Overall project went very well. Accomplished everything laid out/planned in the Milestones. Some hurdles and bumps we had to overcome (like any project).
    - Hurdles/Bumps:
        - Kenny: Video & Audio recording. Debugging issues regarding hardware/Pi configurations. Recording audio/video formats and merge/rendering into usable formats.

        - Chris: debugging how notification is being sent
        - Cristian: How to dynamically display data in the GUI.
- Future work:
    - Add motion detection (detect when baby is moving, detect if baby left the crib)
    - 2 way audio communication with baby (hear baby AND also talk to baby)
    - CRON: automate scripts
    - “Record Now” button allows user to take a video recording at the push of a button.