ECE 445

Spring 2025

Final Report

BioSteady

Team 46

Alisha Chakraborty
Asmita Pramanik
Pranav Nagarajan

TA: Surya Vasanth

Date : May 7, 2025

Abstract

This report presents the development of a wearable device designed to differentiate between physiological responses induced by caffeine from those induced by stress. This device will use a pulse oximeter and heart-rate sensor as well as a galvanic skin response sensor to collect the user's physiological data that will then be processed by a machine learning model to analyze trends and provide real-time classification and feedback. This data is displayed on a custom web application that offers personalized recommendations for caffeine consumption. Our approach aims to promote a device that offers healthier choices for caffeine users to track their intake through physiological awareness.

Table of Contents

1. Introduction

1.1 Problem Statement

Coffee and caffeinated beverages have become amongst the most sought after stimulants consumed by individuals who experience high stress levels due to the fast-paced and demanding nature of their academic and professional lives. We are constantly in an environment where the presence of cafes and vending machines are all around which makes it difficult for individuals to resist when they are looking for a way to maintain productivity for long periods of time.

In such scenarios, we fail to acknowledge the negative physiological changes that affect our overall health in the long term.We built BioSteady to help users assess their physiological reactions to caffeine by measuring changes in heart rate and skin conductance. To influence a positive change in lifestyle, we wanted to use our device to also provide users with actionable feedback so they can practice making informed decisions about their daily caffeine intake levels. This makes it essential for us to find a reliable way to differentiate between the physiological changes triggered by caffeine and stress in real-time so that we are able to make informed health-related decisions in the long term.

1.2 Solution

Our proposed solution is to ideate and build a wearable device that collects physiological data through the sensors used to estimate and report to users whether their bodily changes are likely influenced by caffeine intake and whether consuming caffeine at a given moment is advisable or not. This report will be displayed through a custom built web application which will also provide recommendations to users regarding caffeine consumption and stress management techniques. A considerable amount of research led us to a study which graphically demonstrates a significant difference in sensor readings with respect to stress and caffeine.

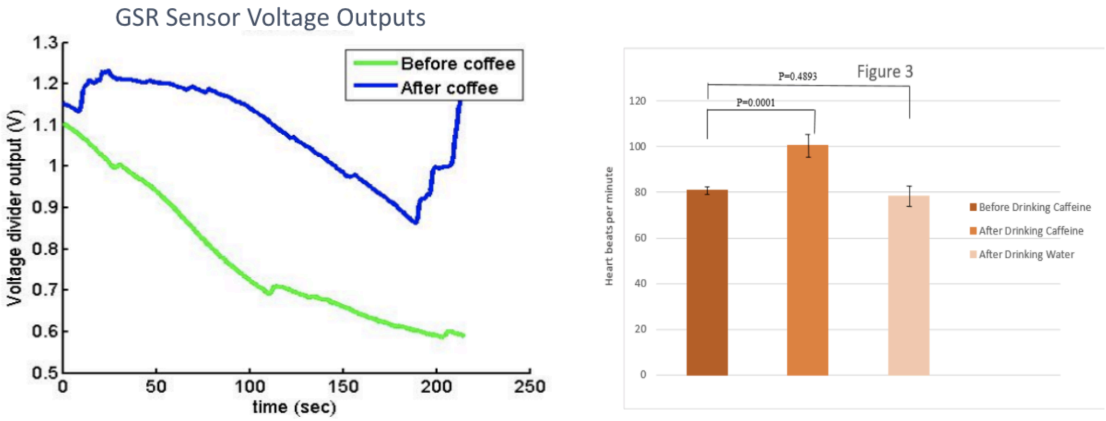


Figure 1: GSR and Heart Rate Changes Upon Consumption of Caffeine

Figure references : Villarejo, M. V., Zapirain, B. G., & Zorrilla, A. M. (2012). A stress sensor based on Galvanic Skin Response (GSR) controlled by ZigBee. *Sensors*, 12(5), 6075–6101.
Hovland, K. (n.d.). *The effects of caffeine on heart rate and blood pressure in college students*. Bethel University.
Guan, A., Hill, D., & Liang, L. (2022). *Can wearable sensors differentiate between caffeine and stress?* Journal of Emerging Investigators. https://emerginginvestigators.org/articles/22-001/pdf

Leveraging these trends and machine learning algorithms, we can effectively classify the physiological impact of stress and caffeine intake in one's body, which served as our overarching goal in mind while building BioSteady.

1.3 High Level Functionality Requirements

1) Data Collection and Processing : The real-time physiological data input collected from the heart rate sensor (MAX30102) and the Elecbee GSR sensor must be processed efficiently by the microcontroller(STM32L432KC) with minimum errors to classify between stressed and caffeinated individuals correctly. The microcontroller must efficiently process this data by performing filtering to improve the data classification accuracy.
2) Data Transmission and Communication Compatibility : The data from the microcontroller must be integrated into a web application with the right communication protocols (I2C for MAX30102 and ADC for GSR) and no significant latency upon data transmission. The data should then be processed by the microcontroller via UART and then transmitted to an external device, which will most likely be a computer.
3) User Interface: The web application must be able to correctly display the physiological state analysis output of the user along with effectively differentiating between stress-induced and caffeine-induced responses with actionable recommendations regarding the user's health.The UI must be at least reliable using a data classification algorithm. Users must be able to receive information in a user-friendly manner where they can understand the data collected by the

wearable device. The web application should include graphical visualizations of the data that is interpretable for the user.

2. Design

2.1 Block Diagram

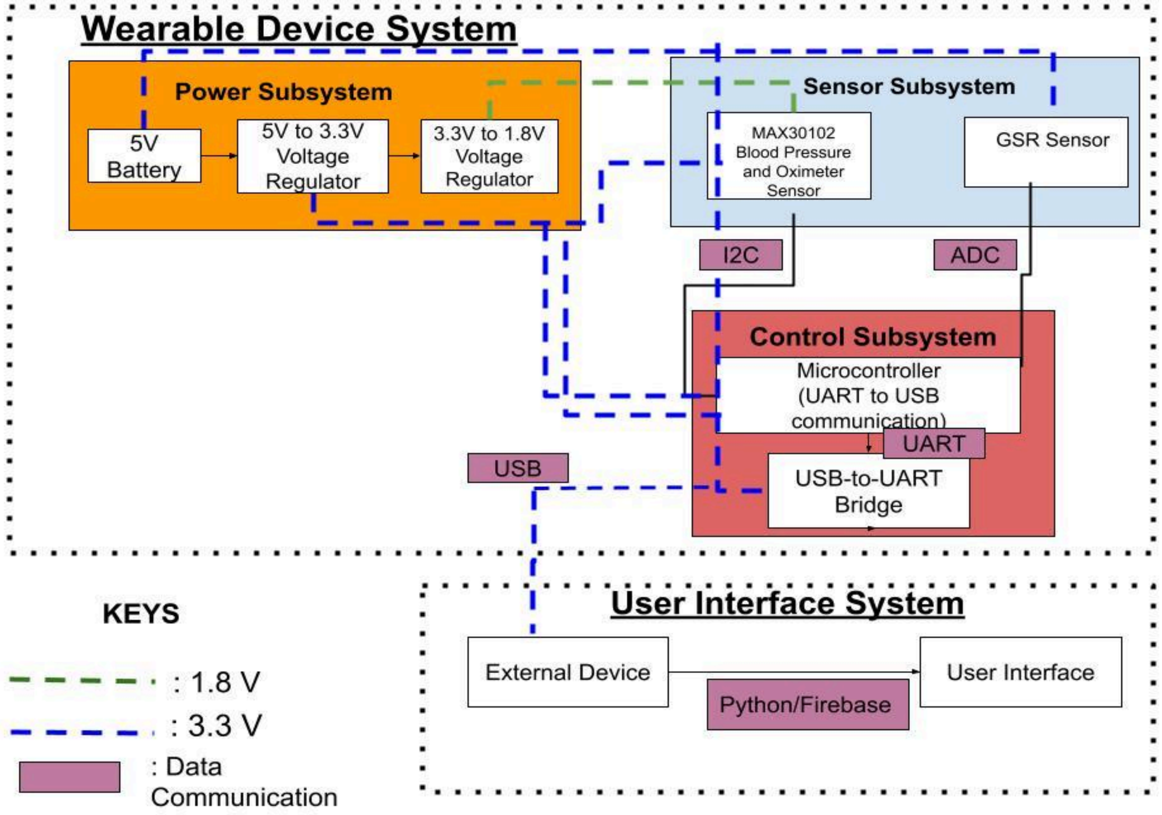


Figure 2: Block diagram for BioSteady Device

The above figure represents our block diagram for our BioSteady Device including both the wearable device and user interface systems. For the wearable device, we have three subsystems: power, sensor, and control.

2.2 Power Subsystem

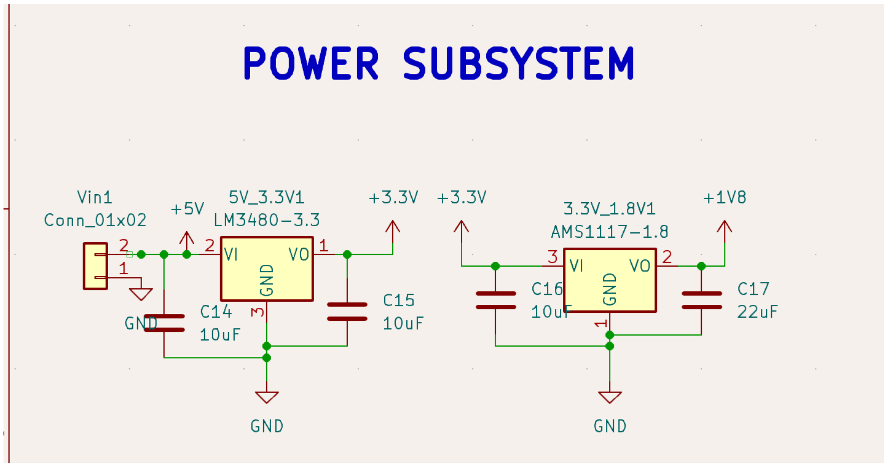


Figure 3: Schematic for Power Subsystem

2.2.1 Design Description

This subsystem provides a 5 V , 3.3 V , and 1.8 V power supply to our sensor subsystem and control subsystem. It requires a 5V battery input and two voltage regulators. The 5V battery input is used to power the GSR (Galvanic Skin Response) sensor in the sensor subsystem. The LM3480-3.3V [11] voltage regulator takes a 5 V input and provides a 3.3V power supply for the MAX30102 Heart rate and oximeter sensor and control subsystem. The AMS1117-1.8 [10] voltage regulator takes in the 3.3V power supply from the LM3480-3.3 voltage regulator to output 1.8 V to supply power for the LED in the MAX30102 heart rate sensor.

2.2.2 Design Justification and Alternatives

We decided to use the LM3480-3.3 voltage regulator because it provides a fixed 3.3V output with a low dropout voltage of around 0.5 V, allowing to regulate a 3.3 V power supply even if the input voltage drops to 3.8 V. It can also supply up to 100 mA which is sufficient for our low-power control subsystem. We chose the AMS117-1.8 V due to its higher current supply of up to 800 mA which is needed for more power-hungry components like the MAX30102. Our initial design used a 5V power supply from the USB-A connector in our control for our power subsystem. However, it was advised for each subsystem to be separate so we incorporated a 5 V battery input. This subsystem was tested using a power supply that supplies the 5 V input and a multimeter to test if there was a reliable 3.3 V and 1.8 V power output from the subsystem.

2.3 Sensor Subsystem

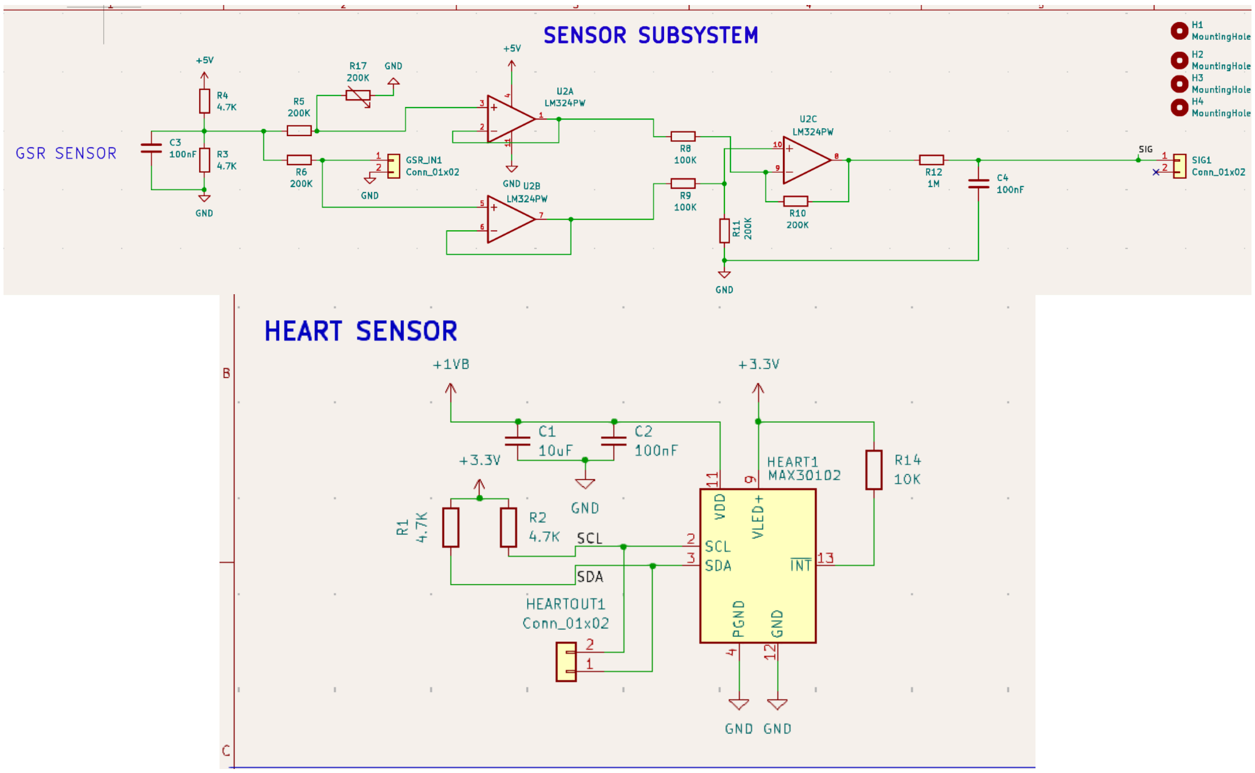


Figure 4: Schematic for Sensor Subsystem

2.3.1 Design Description

This subsystem has the MAX30102 Heart rate and oximeter sensor [12] and the GSR (Galvanic Skin Response) sensor [7] .

The MAX30102 heart rate and oximeter sensor is an integrated high-sensitivity pulse oximeter and heart rate sensor. It takes 1.8 V core power supply and a 3.3V LED power supply that is provided by the power subsystem. This sensor has a red LED at 660 nm, an infrared LED at 880 nm, a photodetector, and low-noise analog signal processing. The sensor will pulse red and infrared light that will get reflected back from the skin above the sensor. The photodetector will receive the reflected light signal and the analog front-end will filter the signal. The data is then stored in an internal First-in First-out (FIFO). It outputs two lines: SCL (clock) and SDA (data). These outputs will then get fetched by the microcontroller over I2C communication.

The GSR sensor will be powered by the 5V battery input from the power subsystem and detect skin resistance from input electrodes that are worn by a user. The sensor will apply a small, constant voltage across the two electrodes and measure the current through the circuit which is affected by the skin's resistance. This sensor will then convert the current into an analog voltage that is proportional to the skin conductance using the LM324 Op Amp [14] to stabilize the signal. The output is an analog voltage signal that represents the skin resistance that will get fetched by the microcontroller over ADC communication.

2.3.2 Design Justification/Alternatives

The MAX30102 heart rate and oximeter sensor is an essential component of our project because of its highly precise ability to detect and measure heart rate by emitting infrared light through the skin and detecting variations in the absorption of the light that is caused by changes in heart rate. This sensor was chosen because of its small and compact size and integrated infrared and red LEDs. It also is able to reduce noise due to its programmable sampling rate at 100 Hz. The GSR Skin Sensor Module detects variations in skin conductance by measuring the electrical resistance between two electrodes placed on the skin. These fluctuations are influenced by sweat gland activity, which tends to increase during emotional responses such as stress or excitement. There were many design inconsistencies for both sensors. Both sensors were tested and verified to have the correct current and voltage outputs using a power supply and multimeter, respectively. For the MAX30102 sensor, there were issues with a broken LED path within the sensor due to soldering issues of the sensor. The sensor's pads were directly underneath the chip which made them inaccessible to test once the chip was soldered to the PCB. To correct this problem, we hand-soldered each pad to ensure that each pad had solder on them and then used the PCB oven to solder the sensor onto the PCB board. With this change, the MAX30102 was able to have a stable LED path and emit both LED's reliably. For the GSR sensor, we replicated the GSR schematic onto the PCB board. A minor design issue we had with our GSR sensor was that one of the resistors from our GSR schematic was incorrectly traced during the PCB layout, but this was easily corrected by placing a through hole resistor onto the correct pads rather than using a SMD part.

2.4 Control Subsystem

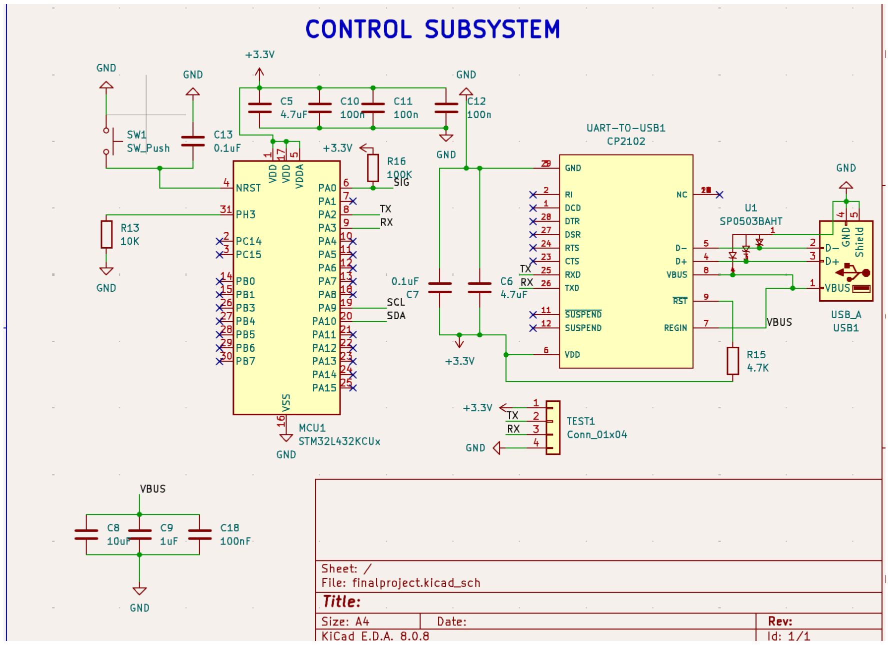


Figure 4: Schematic for Control Subsystem

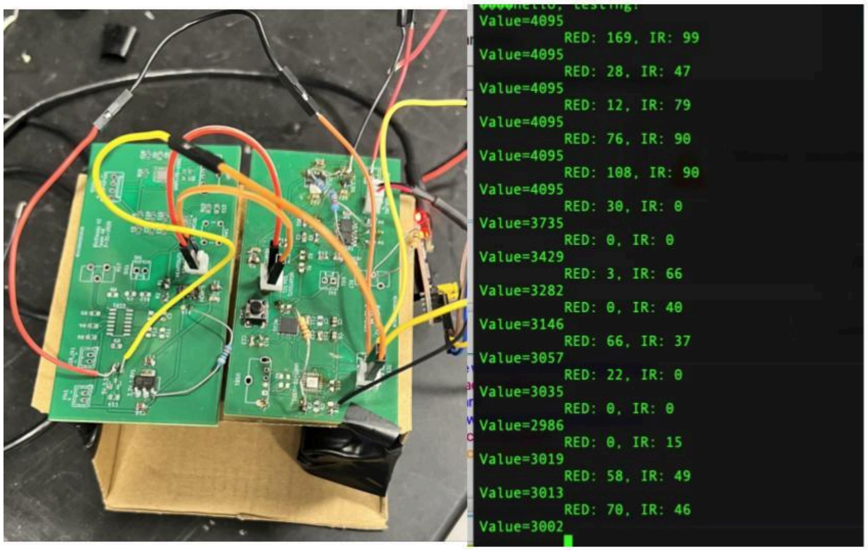
2.4.1 Design Description

This subsystem comprises both the STM32L432KCU6 [9] microcontroller and a CP2102 USB-to-UART bridge [13]. Our MCU will process the data inputs from the sensor subsystem using the appropriate protocols: I2C for the MAX30102 sensor and ADC for the GSR sensor. It then sends it to the USB-to-UART bridge through UART communication. Then, the USB-to-UART bridge will send the data from the MCU to an external device using USB-A serial protocol. Finally, our user interface subsystems will consist of the external device that receives the data from our wearable device using a python script to create a user interface that will allow the user to have access to the data collected by the wearable device and any personalized recommendation determined by the machine learning model.

2.4.2 Design Justification and Alternatives

We decided to use the STM32L432KCU6 microcontroller and CP2102 USB-to-UART bridge to prioritize a low-power, high-performance control subsystem. The STM32L432KCU6 is optimized for low-power embedded applications and operates at a current of around 0.006 A when programmed and active, which was done by testing the current through the lab bench power supply. This is ideal for our wearable device because we need to collect data from the user over a period of 15-20 minutes without needing to recharge the battery. The CP2102 allows seamless data transfer between the wearable device and the external device for data collection and processing. Alternative microcontrollers such as ESP32 were considered but we wanted to prioritize the space and power efficiency from the STM32L432KCU6. We had some design issues with the CP2102 UART-to-USB bridge because this device requires a specific VCP driver from Silicon Labs. However, this driver was not fully compatible with the newer macbooks due to new security protocols on the updated Macbooks. Due to time constraints, we decided to instead use a HiLetGo Uart-to-USB bridge to program our microcontroller and transfer data from the wearable device to our computers. Rather than using a USB connection, a better alternative would be to connect the wearable device to an external device via WiFI or bluetooth so that the device can be compatible with all computers rather than just ones with the CP2102 driver.

2.5 Results and Final Design



```
Hello, testing!
Value=4095
          RED: 169, IR: 99
Value=4095
          RED: 28, IR: 47
Value=4095
          RED: 12, IR: 79
Value=4095
          RED: 76, IR: 90
Value=4095
          RED: 108, IR: 90
Value=4095
          RED: 30, IR: 0
Value=3735
          RED: 0, IR: 0
Value=3429
          RED: 3, IR: 66
Value=3282
          RED: 0, IR: 40
Value=3146
          RED: 66, IR: 37
Value=3057
          RED: 22, IR: 0
Value=3035
          RED: 0, IR: 0
Value=2986
          RED: 0, IR: 15
Value=3019
          RED: 58, IR: 49
Value=3013
          RED: 70, IR: 46
Value=3002
```

Figure 5: Final Design for Wearable Device

This is our final design for our wearable device. On the right is our wearable device, and on the left is the data transmission between the wearable device and external device which will then be processed by our user interface system.

2.6 User Interface System

2.6.1 Firebase Cloud Database Implementation

To enable scalable and persistent storage of sensor readings, we set up a real-time database using **Google Firebase Firestore**. This allowed for continuous data upload from the custom PCB device via a Python script, using the firebase_admin Python SDK. Data was structured to log:

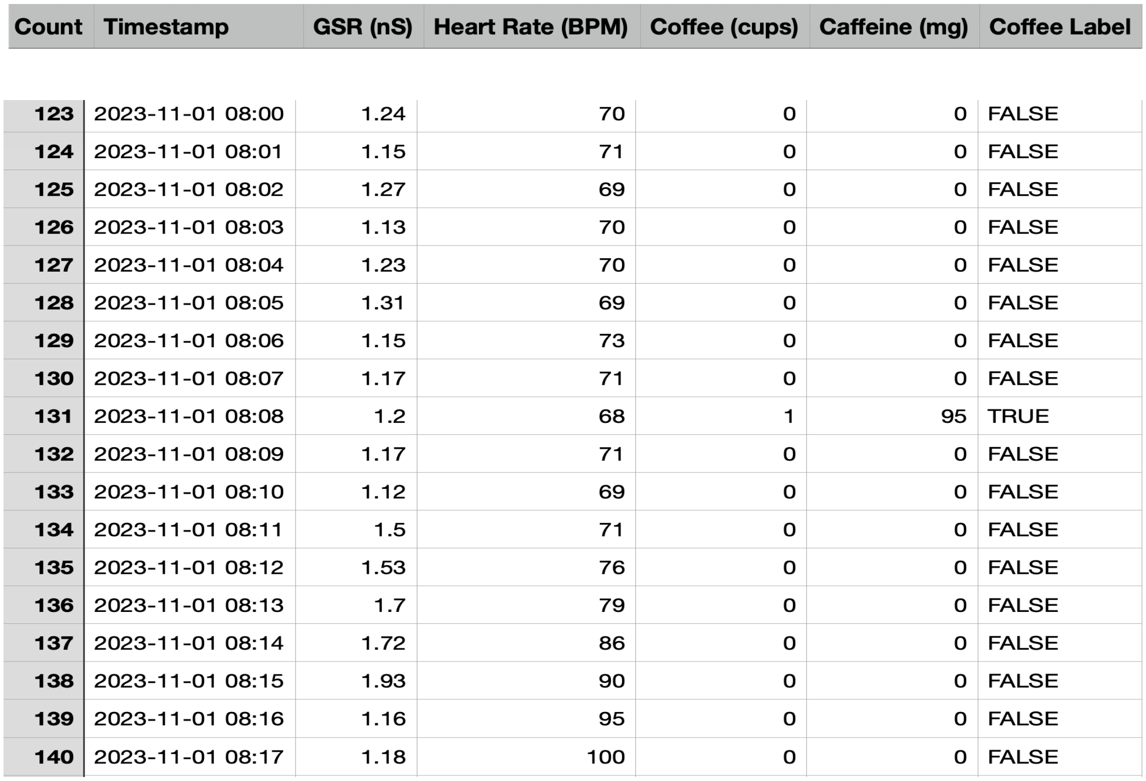
- gsr: Galvanic Skin Response value (in nanosiemens)
- heartRate: Beats per minute (BPM)
- timestamp: Formatted with timezone context (e.g., “April 24, 2025 at 01:02:31 PM UTC-0500”)

Data Path:
PCB (sensor data) → USB Serial → Python Script → Firebase Firestore

Polling occurred every ~5 seconds, and each reading was written as a new document under the sensorData collection. This setup supported time-aligned analysis with the ML pipeline, and also enabled front-end components (like a caffeine dashboard) to query live sensor data.

2.6.2 Caffeine Prediction ML Model

We designed a K-Nearest Neighbors (KNN)-based classifier to detect and interpret caffeine-induced physiological changes using sequential biometric data



Count	Timestamp	GSR (nS)	Heart Rate (BPM)	Coffee (cups)	Caffeine (mg)	Coffee Label
123	2023-11-01 08:00	1.24	70	0	0	FALSE
124	2023-11-01 08:01	1.15	71	0	0	FALSE
125	2023-11-01 08:02	1.27	69	0	0	FALSE
126	2023-11-01 08:03	1.13	70	0	0	FALSE
127	2023-11-01 08:04	1.23	70	0	0	FALSE
128	2023-11-01 08:05	1.31	69	0	0	FALSE
129	2023-11-01 08:06	1.15	73	0	0	FALSE
130	2023-11-01 08:07	1.17	71	0	0	FALSE
131	2023-11-01 08:08	1.2	68	1	95	TRUE
132	2023-11-01 08:09	1.17	71	0	0	FALSE
133	2023-11-01 08:10	1.12	69	0	0	FALSE
134	2023-11-01 08:11	1.5	71	0	0	FALSE
135	2023-11-01 08:12	1.53	76	0	0	FALSE
136	2023-11-01 08:13	1.7	79	0	0	FALSE
137	2023-11-01 08:14	1.72	86	0	0	FALSE
138	2023-11-01 08:15	1.93	90	0	0	FALSE
139	2023-11-01 08:16	1.16	95	0	0	FALSE
140	2023-11-01 08:17	1.18	100	0	0	FALSE

The training dataset came from anonymized **hospital records** containing GSR time-series data, heart rate time-series data and logged caffeine consumption (mg).

From this data, we extracted **spike patterns** which are changes in biometric signals following caffeine intake. Each subject's sequence was analyzed over 30–45 minutes post-ingestion, and the following features were computed:

1. ΔGSR (peak increase in GSR post-ingestion)
2. ΔHR (peak heart rate increase)
3. Time-to-Peak (minutes until max change)
4. Recovery Time (minutes until signal returned to ~baseline)
5. Caffeine Dosage (in mg from cups)

These were assigned a **caffeine impact weight** defined as:

$$\text{Weight} = \alpha 1 \cdot \Delta \text{GSR} + \alpha 2 \cdot \Delta \text{HR} + \alpha 3 \cdot \text{Time-to-Peak} - \alpha 4 \cdot \text{Recovery Time}$$

Where α1–α4 were tunable constants. For training purposes, we empirically set α1 = α2 = 1.0, α3 = 0.5, α4 = 0.7.

Each training example was then represented as a 5D feature vector:

[X1, X2, X3, X4, X5] = [ΔGSR, ΔHR, Time-to-Peak, Recovery Time, Caffeine Dosage]

2.6.3 Test Dataset and Manual Labeling

Our test dataset consisted of manually logged sessions. Each time one of us drank coffee, we recorded:

- The time and amount of caffeine consumed
- My biometric readings (via the PCB) at 5, 10, and 20 minutes post-ingestion

We calculated spike-related features the same way as in training:

- Found maximum changes
- Estimated time-to-peak and decay
- Derived the weight vector for the session

2.6.4 KNN Classification and Prediction

We implemented a K-Nearest Neighbors (KNN) classifier with k=3 to evaluate how closely a new, manually labeled test session resembled previously observed patterns in the hospital-derived training set. Each instance was represented as a 5-dimensional feature vector comprising:

- ΔGSR (maximum post-consumption increase)
- ΔHR (maximum heart rate increase)
- Time-to-Peak (in minutes)
- Recovery Time (time taken to return to baseline)
- Caffeine Dosage (in mg from cups)

For any new test input, the model computed the Euclidean Distance between its feature vector and every training vector given by:

$$Distance = \sqrt{\sum_{i=1\ to\ 5} (xi - yi) ** 2}$$

where xi and yi are the corresponding features from the test and training vectors.

The three closest training points (based on the smallest distances) were selected as the "neighbors." This was used to find the following -

- Estimated Duration of Caffeine Effect was calculated as the average recovery time of these neighbors.
- Next Safe Intake Time by adding the average duration of spike + recovery to the timestamp of caffeine ingestion, essentially modeling the safe physiological reset window.

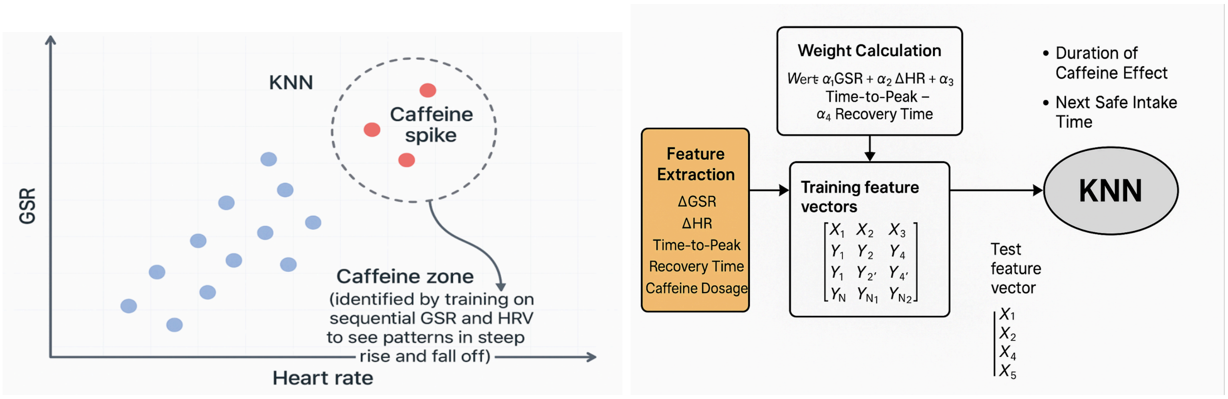

KNN
Caffeine spike
GSR
Caffeine zone
(identified by training on sequential GSR and HRV to see patterns in steep rise and fall off)
Heart rate
Weight Calculation
Wert α_1GSR + α_2 ΔHR + α_3 Time-to-Peak – α_4 Recovery Time
Feature Extraction
ΔGSR
ΔHR
Time-to-Peak
Recovery Time
Caffeine Dosage
Training feature vectors
$\begin{bmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_4 \\ Y_1 & Y_{2'} & Y_{4'} \\ Y_N & Y_{N_1} & Y_{N2} \end{bmatrix}$
KNN
• Duration of Caffeine Effect
• Next Safe Intake Time
Test feature vector
X_1
X_2
X_4
X_5

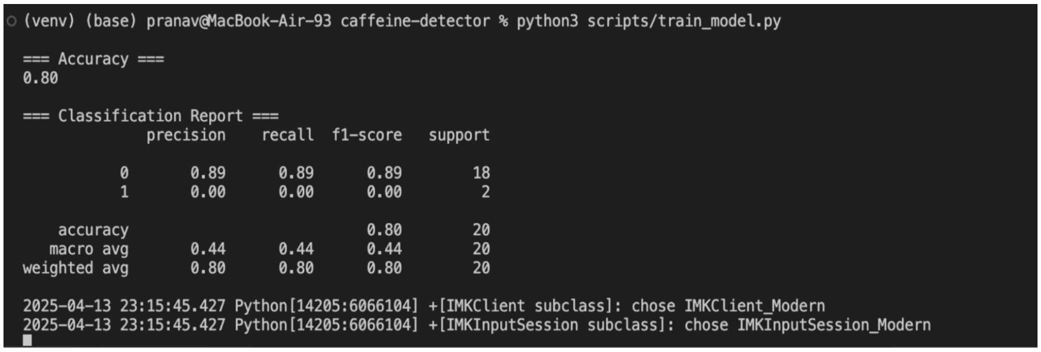
Figure 6: KNN Model Diagram

2.6.5 Model Accuracy & Personalization

The trained model achieved a final accuracy of 80% on the test dataset, as shown in the attached output screenshot. While some related research reports benchmarks around 83% for similar biometric classification tasks, this result is considered both satisfactory and practical given the personalized nature of the system.

Unlike generalized models trained purely on population-wide datasets, this model is designed to learn over time from an individual's own data. Because each user's physiological response to caffeine (in terms of GSR and heart rate variation) can vary, a profile-specific approach allows the classifier to become increasingly accurate with continued use.

As the system collects more labeled test sessions for a given user, the model will have access to a more representative set of feature vectors, making predictions more attuned to that person's unique physiological response patterns. This justifies using 80% as a robust final accuracy figure, with future improvement expected as the dataset deepens.



```
○ (venv) (base) pranav@MacBook-Air-93 caffeine-detector % python3 scripts/train_model.py

=== Accuracy ===
0.80

=== Classification Report ===
              precision    recall  f1-score   support

           0       0.89      0.89      0.89        18
           1       0.00      0.00      0.00         2

    accuracy                           0.80        20
   macro avg       0.44      0.44      0.44        20
weighted avg       0.80      0.80      0.80        20

2025-04-13 23:15:45.427 Python[14205:6066104] +[IMKClient subclass]: chose IMKClient_Modern
2025-04-13 23:15:45.427 Python[14205:6066104] +[IMKInputSession subclass]: chose IMKInputSession_Modern
█
```

Figure 7: Classification Report

2.6.6 User Interface & Experience

The BioSteady frontend is designed to provide an intuitive, real-time interface for users to monitor the physiological effects of caffeine and stress. The platform is profile-specific, meaning that each user's data is stored and used to **fine-tune predictions** over time based on prior readings and a clinically derived baseline.

1. Landing Page: Personalized Onboarding & Navigation

The homepage offers users quick access to the system's two core functionalities:

- Caffeine Analysis
- Heart Rate Monitor

It also presents three main cards:

- **Real-Time Monitoring**: Indicates that live biometric data is being recorded.
- **AI-Powered Insights**: Summarizes the backend ML system's ability to personalize recommendations over time by comparing the user's biometric signature with the hospital dataset.
- **Comprehensive Dashboard**: Provides users with centralized access to their metrics and visual history.

This page gives users a high-level understanding of their health status and facilitates engagement by offering actionable tools.

2. Heart Rate Page: Sequential Monitoring & Metric Extraction

This section is focused on capturing **real-time heart rate data** over a fixed period (e.g., 30 seconds), which is used both for display and backend processing. Key components include:

- **Live heart rate value**
- **Trend graph over time**
- **Activity summary**, including today's average, resting rate, peak, and variability

These readings are stored and used as time-sequenced vectors in the caffeine detection model. It allows the user to track cardiovascular responses even outside caffeine intake events, which helps refine baseline measurements.

3. Caffeine Detection Page: Model Outputs & Visual Feedback

This is the core functional page of BioSteady. It activates the full prediction pipeline and presents a holistic analysis of the user's current physiological state. Elements include:

- **Start Detection Button**: Triggers a 30-second scan that collects a burst of data for model inference.
- **Response Pattern Graph**: Plots both heart rate and skin conductance, clearly highlighting any spikes post-caffeine.
- **Caffeine Status Panel**: Displays the interpreted intensity, estimated effect duration, next safe intake window, and metabolic response category.
- **Recommendations Section**: Contextual suggestions based on model output (e.g., hydration, walking, reduced intake).

This section closes the loop between raw data and user action, delivering insights derived from the hospital-trained KNN model, tailored to the individual user's current and historical data.

3. COST AND SCHEDULE

3.1 Cost Analysis

We found that a reasonable average hourly rate of an ECE graduate from the University of Illinois Urbana-Champaign is about $44/hour. Based on this finding, we believe that the total labor cost will be as follows:

Team Member	**Hourly Rate**	**Multiplier**	**Hours to Complete**	**Total Individual Cost** **(Hourly Rate*Multiplier*Hours to Complete)**
Asmita Pramanik	$44	2.5	120	$13200
Alisha Chakraborty	$44	2.5	120	$13200
Pranav Nagarajan	$44	2.5	120	$13200
Total Labor Costs				**$39600**

Cost Analysis: Team

Description	Manufacturer	Part #	Link	Qty.	Cost
5V battery	Dantona Industries	3145-CMP-AAA-75-18-ND	LINK	1	$6.81
Heart Rate and Oximeter Sensor	Analog Devices Inc./Maxim Integrated (VA)	MAX30102EFD+TTR-ND-TR	LINK	1	$12.05
STM32L432KCU6	STMicroelectronics	497-16592-ND	LINK	1	$4.88
CP2102 USB to UART Bridge	Silicon Labs	336-1160-2-ND	LINK	1	$10.41
USB-A Connector	Molex	WM4078-ND	LINK	1	$1.52
LM324 Op AMP	Texas Instruments	595-LM324PW	LINK	1	$0.85
JST-PH connector for GSR sensor	JST Sales America	455-B2B-XH-A-ND	LINK	1	$0.10
4.7 K Ohm resistor	STMicroelectronics	497-16578-ND	LINK	5	$0.50
10K OHm resistor	Murata Electronics	13-RC1206FR-7W10KLTR-ND	LINK	2	$0.32
1M ohm resistor	Murata Electronics	490-GRM155C61E475ME15JTR-ND	LINK	1	$0.1
200K Ohm resistor	YAGEO	311-200KFRTR-ND	LINK	5	$0.50
100k Ohm resistor	YAGEO	13-RC1206FR-13100KLTR-ND	LINK	2	$0.20
0.1uF capacitor	Murata electronics	490-10931-2-ND	LINK	9	$3.24

4.7uF capacitor	Murata electronics	490-GRM155C61E475ME15JTR-ND	LINK	3	$0.60
0.47uF capacitor	Murata electronics	490-3266-2-ND	LINK	1	$0.10
33uF capacitor	TDK Corporation	445-5986-2-ND	LINK	1	$0.77
22uF capacitor	Samsung electro-mechanics	1276-1274-2-ND	LINK	1	$0.10
Switch Button	SCHURTER Inc.	486-4184-2-ND	LINK	1	$0.47
USB-A to USB-c cable	Qualtek	Q1238-ND	LINK	1	$5.91
Total Cost					**$49.33**

Cost Analysis: Project Parts

Total Cost of Project: $39600 + $49.33 = $39 649.33

3.2 Schedule

Weekly Schedule	**Team Objectives**
3/10	Alisha: Complete PCB design with integration of both sensors and submit a second round of PCB design Asmita: Start working on the breadboard connections and web application Pranav: Conduct research on heart rate datasets and work on web application All: Prepare for the breadboard demo session.
3/24	Alisha: Begin soldering PCB. Asmita: Write firmware code and debug/test data collection of sensors Pranav: Prepare layout of UI landing page
3/31	Alisha : Submit third round PCB design if necessary. Debug Add any necessary PCB changes for the fourth round of PCB design. Asmita, Pranav: Test/Debug data collection and processing portion of the software design, ensuring accurate collection and processing of both sensors.
4/7	Alisha : Submit fourth round of PCB design) Finalize PCB design to ensure that the PCB is fully functional for both sensors. Asmita: Test/debug data transmission from MCU to UART-to-USB bridge for reliable communication with external devices. Pranav: Analyze and display sensor data formatted properly on UI

4/21	Pranav: Finalize data collection and processing of sensors to external devices. Alisha : Ensure complete integration of hardware components and software components. Asmita: Further test software design and add any required modifications. All: Prepare for Mock Demo.
4/28	Asmita, Pranav: Finalize UI for software design for project. All : Complete Mock Demo and Mock Presentation. Complete optional EC video assignment
5/5	All: Complete final presentation. Submit final papers and lab notebook.

4. Conclusions

4.1 Accomplishments

Complete Integration of Sensors onto Hardware
We were able to integrate both sensors effectively onto our PCB and collect relevant data with minimum loss of data.

Reliable Data Collection and Transmission
We successfully transmitted processed data from microcontroller to web-based frontend. We built a Python backend script which successfully parsed physiological sensor data and sent these to our web application, offering clear classification and insights to users.

Effective Use of Cloud-based Storage for Future Scalability
We were able to implement Firebase as our backend storage solution which streamlined uploads of physiological data with timestamps, thereby ensuring reliable historic data logging for future debugging purposes.

4.2 Uncertainties

Variations in Sensor Outputs
IR and GSR readings may fluctuate due to differing environmental conditions. This can include temperature, difference in skin contact etc. This introduces the challenge of guaranteeing reliable data collected in different environments. More contextual data needs to be taken into account while processing these readings

Heart Rate Measurement Accuracy

Missed IR peaks can affect BPM calculations, which would hinder accurate interpretation of data. A possible solution to improve this would be to tune ***scipy.signal. find_peaks()*** parameters such as *prominence* to enhance accuracy.

4.3 Future Work

Wearability of Device
We believe, at the moment, that our device can be improved in terms of ease of wearability. Due to time and budget constraints, we were not able to implement an absolutely optimized design, which is a goal we want to achieve in the future.

User Testing
We want to expand our testing on more number of users with different physiological baselines and incorporate auxiliary sensors to account for external environmental conditions that may affect our sensor readings.

4.4 Ethical Considerations

Privacy and Data Security: Our project collected biometric data which can raise ethical concerns with the user's privacy and data security. We followed the IEEE Code of Ethics Section 1.5 [3] requires that collected physiological data must be handled securely to prevent misuse by doing. We will also follow the ACM Code of Ethics section 1.6-.17 [4] by ensuring that the data collected by the Biomedical sensor subsystem is handled with the utmost privacy. Users will have full transparency on when data is collected, how data is collected, and how it is used. User data will not be shared to anywhere other than the power subsystem and the USB-to-UART system, and users will have full access to the data collected by sensors through the software component.

Team Ethics: Our team followed the IEEE Code of Ethics Section II and III during this project to provide a safe environment for team members. We treated each other fairly and valued every person's work equally.

REFERENCES

[1] Smith, A. P., & Rogers, R. (2015). Caffeine consumption and self-assessed stress, anxiety, and depression in secondary school children. *Journal of Psychopharmacology, 29*(12), 1238–1245. https://doi.org/10.1177/0269881115612404

[2] Villarejo, M. V., Zapirain, B. G., & Zorrilla, A. M. (2012). A stress sensor based on Galvanic Skin Response (GSR) controlled by ZigBee. *Sensors, 12*(5), 6075-6101. https://doi.org/10.3390/s120506075

[3] *IEEE - IEEE Code of Ethics*, www.ieee.org/about/corporate/governance/p7-8.html. Accessed 13 Feb. 2025.

[4] "The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society." *Code of Ethics*, www.acm.org/code-of-ethics/. Accessed 13 Feb. 2025.

[5] Center for Devices and Radiological Health. "Overview of Device Regulation." *U.S. Food and Drug Administration*, FDA, www.fda.gov/medical-devices/device-advice-comprehensive-regulatory-assistance/overview-device-regulation. Accessed 13 Feb. 2025.

[6] **Ultra Librarian.** (2021, September 23). *MAX30102 datasheet: A complete solution for heart rate monitoring in wearable devices*. Ultra Librarian. Retrieved from https://www.ultralibrarian.com/2021/09/23/max30102-datasheet-a-complete-solution-for-heart-rate-monitoring-in-wearable-devices-ulc

[7] **Seeed Studio.** (n.d.). *Grove-GSR sensor*. Seeed Wiki. Retrieved from https://wiki.seeedstudio.com/Grove-GSR_Sensor/

[8] **Arm Mbed.** (n.d.). *ST Nucleo-L432KC platform*. Mbed OS. Retrieved from https://os.mbed.com/platforms/ST-Nucleo-L432KC/

[9] **Kynix.** (n.d.). *STM32L432KCU6 microcontrollers CAD models & datasheet application*. Kynix Electronics. Retrieved from https://www.kynix.com/components/STM32L432KCU6-MICROCONTROLLERS-CAD-MODELS-DATASHEET-APPLICATION.html

[10] **Semitech.** (2012, March 14). *AMS1117 voltage regulator datasheet*. AMS Semitech. Retrieved from http://www.ams-semitech.com/attachments/File/AMS1117_20120314.pdf

[11] **Texas Instruments**, *LM3480 100-mA Low-Dropout Linear Regulator With Shutdown*, Datasheet, Rev. S, Mar. 2024. [Online]. Retrieved from https://www.ti.com/lit/ds/symlink/lm3480.pdf

[12] **Analog Devices**, *MAX30102: High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*, Datasheet, Rev. 1, Oct. 2018. [Online]. Retrieved from https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf

[13] **Silicon Laboratories**, *CP2102/9: Single-Chip USB-to-UART Bridge*, Datasheet, Rev. 1.8, Jan. 2017. [Online]. Retrieved from https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

[14] **Texas Instruments,** *LMx24, LMx24x, LMx24xx, LM2902, LM2902x, LM2902xx, LM2902xxx Quadruple Operational Amplifiers*, Datasheet, Rev. AD, Oct. 2024. [Online]. Retrieved from https://www.ti.com/lit/ds/symlink/lm324.pdf

[15] **Guan, A., Hill, D., & Liang, L. (2022).** *Can wearable sensors differentiate between caffeine and stress? Journal of Emerging Investigators*. https://emerginginvestigators.org/articles/22-001/pdf

Appendix A - Requirements and Verification Table

1. Biomedical Sensing

REQUIREMENTS	VERIFICATION	SUCCESS CRITERION	OUTCOME (Yes/No)
The MAX30102 sensor must detect and measure heart rate in the range of 60-120 BPM with a precision of ±2 BPM.	Measure heart rate using a personal EKG/ECG heart rate monitor and compare these outcomes to the MAX30102 sensor's outcomes for 15 test subjects.	The readings of the sensor and the personal EKG/ECG device must match within ±2 BPM for majority (~85-90%) of the readings	No. We compared sensor outputs to heart rate data collected by the Apple Watch, which may not always be accurate
The Grove GSR Sensor must provide an analog voltage output that lies between 0 to 3.3V.	Use a calibrated variable resistor to prompt skin resistance change and measure output voltage using an oscilloscope	The analog output voltage must lie between 0 to 3.3V.	No. Our test involved noting GSR readings using our own skin conductance and checking whether they fell within expected range. We realize that this is an insufficient way to test a sensor and will test the sensor with more accurate methods (such as tape lift-off) in the future.
The response time of the sensor must range from 0.1 to 1 second if sudden change in skin conductance is detected.	Measure the sensor's response time using an oscilloscope	The response time in 90% of the trials must lie between 0.1 to 1 second.	Yes.

MCU and Power Management

REQUIREMENTS	VERIFICATION	SUCCESS CRITERION	OUTCOME (Yes/No)
I2C communication between	Use STM32 firmware to log No	I2C communication between the	Yes, but not over 500+ transactions. We

STM32L432KC micro-controller and MAX30102 sensor must have ≤2% error rate at 100kHz clock speed under regular operating voltage	Acknowledgement (NACK) errors from the I2C and record those errors over 500+ transactions and compute error rate	microcontroller and sensor must exhibit an error rate not greater than 2% for 500+ transactions	performed about 50-100 read/write transactions
Voltage regulators must provide stable outputs of 1.8V and 5V, with a tolerance of ±5% under load current variations of 10mA to 100mA.	Ensure that voltage outputs are within ±5% tolerance through measuring voltage regulator output via an oscilloscope for load currents between 10mA and 100mA.	Voltage regulators must provide stable voltage output within ±5% tolerance (i.e. (3.3V ±0.165V, 1.8V ±0.09V)	Yes
Input data from the sensors must be processed and must be transmitted via the USB to UART bridge within 50-80 milliseconds to ensure real-time updates	Note timestamps for the time taken to process sensor input and transmit output via UART	The time taken must lie between 50-80 milliseconds for 85-90% of the inputs.	Yes

3. Integration with Web Application

REQUIREMENTS	**VERIFICATION**	**SUCCESS CRITERION**	**OUTCOME (Yes/No)**
USB to UART Bridge must reliably transmit processed data at a rate of 115200 bps at minimum to ensure smooth relay of information	Set baud rate on UART firmware, then send known amount of data via the USB-UART bridge and measure rate of transmission to the receiving device	The data should be successfully transmitted at a rate of 115200 bps or higher without errors	Yes
There should be	Log errors in data	There should be very	Yes

minimum to no loss of data during data transmission from microcontroller to the web application	reception by transmitting known data	little to no data loss (≤1%) during data transmission	
Support real-time visualization and analysis by ensuring that the physiological data is displayed within 100ms of data reception to display on the web application	Transmit known sensor data over to the web application and analyze time taken to render on UI by using web inspection tools	Data must be rendered on UI in less than 100ms	Yes

Appendix B - Tolerance Analysis

Tolerance analysis ensures system reliability under component, operational, and environmental variations. The following sections detail the key tolerance considerations for biomedical sensing, MCU and power management, and web application integration.

1. Biomedical Sensing

The following table summarizes the tolerance related problems that may arise in our project:

Component	**Tolerance Range**	**Failure Impact**	**Mitigation Strategy**
MAX30102 Heart Rate Sensor	3.25V–3.35V (I2C voltage)	Communication errors, incorrect heart rate readings	Voltage regulation
MAX30102 LED Intensity	Temp-based variations	Incorrect heart rate estimations	Recalibrate with moving average filter
Grove GSR Sensor	Noise < 5%	Signal corruption in sweat measurement	Use moving average for smoothing
GSR Sensor Temperature Impact	±2°C variation	Resistance affecting GSR output	Recalibrate with variable resistor input

1) The MAX30102 Heart Rate Sensor requires a 3.3V stable voltage with 5% tolerance:

   Vmin = 3.25V,   Vmax= 3.35V

   *Allowed error rate = ((|Vmeasured - 3.3V|)/3.3V) * 100%*

2) Grove GSR Sensor (5% tolerance)

   *Voutput = I * Rskin*

   *Δ Rskin = 0.05 * Rskin*

   Where Rskin is skin resistance, Voutput is output voltage, I is current.

3) Voltage Regulators

   For 3.3 voltage regulation: 3.25 V ≤ Vadjust ≤ 3.35 V

   For 1.8 voltage regulation: 1.75 V ≤ Vadjust ≤ 1.85 V

4) USB to UART Display (2% tolerance)

   Transmission time t = (1/115200) = 8.68 us

   *Δt ≤ 8.68 * 0.02 = 0.174 us*

5) Web Application Display

   Real-time display delay =

   Data retrieval time + Backend processing time + Frontend processing time ≤ 100ms

2. MCU and Power Management

Component	**Tolerance Range**	**Failure Impact**	**Mitigation Strategy**
MCU (STM32L432KC)	Drift ±10%	Data processing delays	Temp-stable Oscillator
Voltage Regulators	±5% output variance	System instability, sensor failures	Switching Regulators
Power Supply Load Variations	Peak vs. normal operation	Voltage dips leading to sensor resets	Capacitor bank

3. Power Supply Feasibility

Subsystem	Voltage	Current (Avg.)	Power Consumption
MAX30102 Sensor	3.3V	0.6mA	1.98mW
Grove GSR Sensor	3.3V	1mA	3.3mW
STM32 MCU	3.3V	25mA	82.5mW
Voltage Regulator Losses	5V–3.3V	~5% loss	5mW (est.)

4. Integration with Web Application

Component	Tolerance Range	Failure Impact	Mitigation Strategy
USB to UART Bridge	Timing error < ±2%	Communication failures, data loss	Retry
Frontend Web Application	Acceptable delay < 100ms	Lag in real-time analysis	Data buffering, asynchronous processing
Packet Loss Handling	< 1% acceptable loss	Corrupt data output	Error-checking & retransmission