Heart rate continuing monitoring and analysis project

Participants:
Angel Duenas aad3773
Jose Perez

Problem and proposed solution

In today's world, many individuals suffer from heart-related ailments, necessitating consistent monitoring. While regular check-ups are beneficial, they don't provide real-time insights, which can be crucial in emergencies. Our project seeks to fill this gap by introducing a 24/7 IoT heart rate monitoring device tailored for those with heart diseases. This device not only tracks heart rates but also instantly uploads the data to the cloud. Algorithms analyze this data in real time, and if any irregularities are detected, immediate alerts are sent to emergency services or the patient's designated physician. Furthermore, all data is securely stored on a cloud database, providing medical professionals with an invaluable historical record of the patient's heart rate. Through a web portal, they can access and study these records. Our solution aims to help heart care, ensuring timely interventions and potentially saving lives.

Design of the Solution

Our project's proof of concept (PoC) leverages two M5StickC Plus devices, integrating a heart rate sensor for continuous monitoring and an alert system for immediate response. The heart rate monitor device will continuously transmit data to Google Cloud's IoT Core service via MQTT, a lightweight messaging protocol ideal for IoT applications due to its low bandwidth usage and efficient data delivery.

Design Sequence:

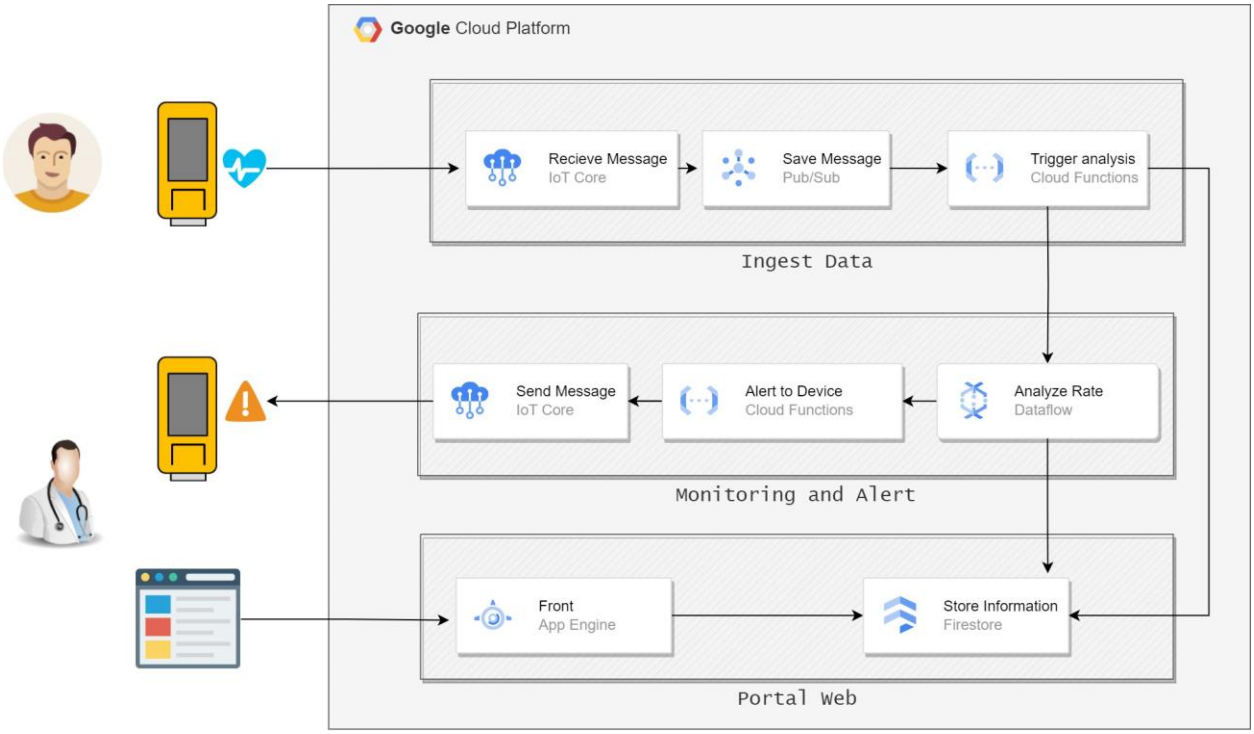
1. **Data Transmission**: The heart rate monitor captures the patient’s pulse and transmits the data to Google Cloud IoT Core using MQTT.
2. **Message Queueing**: Received data is published to a specific topic on Cloud Pub/Sub, effectively queuing the message for processing.
3. **Event Triggering**: A new entry in Cloud Pub/Sub triggers a Cloud Function, which initiates the data handling process.
4. **Data Handling**:
    4.1. **Data Storage**: Cloud Function stores the heart rate and corresponding timestamp in Firestore, a NoSQL document database.
    4.2. **Data Analysis**: Cloud Function invokes Cloud Dataflow to analyze the pulse data over the last 30 seconds.
5. **Anomaly Detection**: Cloud Dataflow examines heart rate patterns and, upon detecting rates exceeding predefined thresholds, triggers an alert via a separate Cloud Function.
6. **Alert Dispatch**: This Cloud Function communicates back with IoT Core to send an alert to the designated M5StickC Plus device.

7. **User Notification**: The receiving M5StickC Plus displays the alert, prompting immediate action from medical personnel or emergency services.
8. **Data Visualization**: Firestore data is used by a web portal hosted on Google App Engine to provide physicians with access to historical heart rate data. This portal serves as a tool for in-depth analysis and patient monitoring over time.

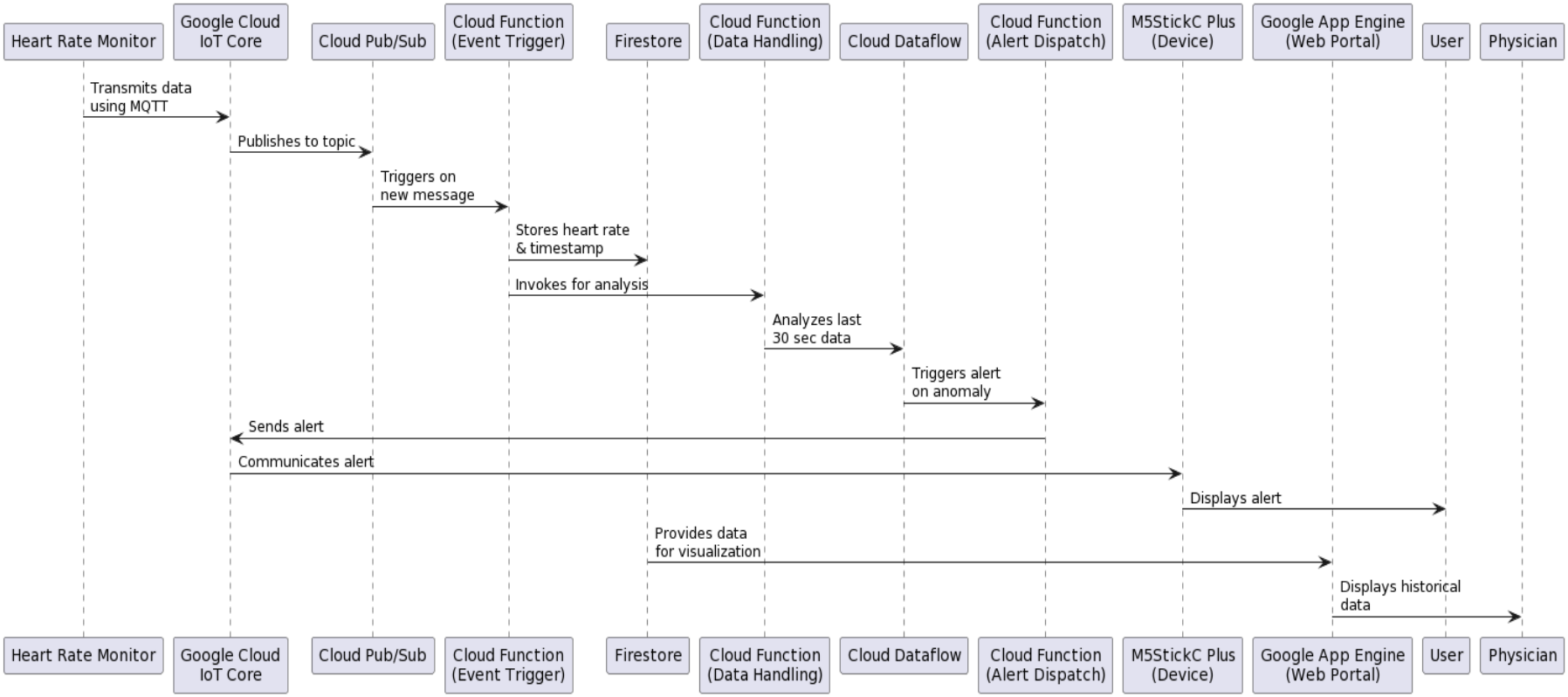
Technologies and Protocols:

- **MQTT Protocol**: Ensures reliable and efficient communication between devices and the cloud.
- **Google Cloud IoT Core**: Provides a managed service for device connection and management.
- **Cloud Pub/Sub**: Offers scalable event-driven messaging for data ingestion.
- **Cloud Functions**: Facilitates serverless execution of backend code in response to events.
- **Firestore**: Acts as a scalable database for storing health data.
- **Cloud Dataflow**: Enables real-time data processing and analytics.
- **App Engine**: Serves the web portal for historical data visualization.

Proposed architecture:



Sequence Diagram



By using this design, we aim to create a responsive, real-time monitoring system that can significantly improve patient outcomes by providing immediate attention during critical situations and enhancing long-term care through data analysis.

What do you need?

To execute this project successfully, we would need access to the following:

M5StickC Plus Devices:

- Need at least two M5StickC Plus devices.
- Heart rate sensor module (e.g., MAX30100) compatible with M5StickC Plus for real-time heart rate monitoring.

Cloud Services and Software:

- Access to Google Cloud Services including IoT Core, Pub/Sub, Cloud Functions, Firestore, Dataflow, and App Engine.
- Necessary software development tools and libraries for programming the M5StickC Plus (e.g., Arduino IDE, Python libraries for Google Cloud integration).

Web Development Tools:

- Tools and libraries for creating a web portal (e.g., HTML, CSS, JavaScript, React, etc.).

Testing and Simulation Tools:

- Software or tools to simulate heart rate data for initial testing.

Gaps

Here are some concerns and potential gaps:

Data Privacy and Security:

- Have we considered compliance with healthcare data protection regulations (e.g., HIPAA in the US)?

Device Reliability and Accuracy:

- How accurate is the heart rate sensor in detecting anomalies?
- What is the battery life of the M5StickC Plus, and how frequently would it need to be charged or replaced?

Network Dependability:

- What happens if the device loses connectivity? How can we ensure continuous monitoring?

Algorithm Robustness:

- Is the algorithm for anomaly detection reliable and does it minimize false positives/negatives?
- How can we ensure the algorithm adapts to different patients with varying heart conditions?

Scalability:

- Can this solution scale to support a large number of devices and data points?

These aspects would need to be taken into consideration and addressed during the development phase to ensure the successful implementation of the project.