# Guidelines for Idea Pitches

|  |  |
| --- | --- |
| Problem - What? (30s) Describe the problem and share insights into why it exists.  What do your users feel, think, say, do when the problem occurs? | Cardiac conditions such as arrhythmias, bradycardia (slow heart rate), and a history of cardiac arrests affect millions globally. These conditions pose significant health risks, especially for patients with Chronic heart diseases, post-operative cardiac complications, Aging populations with weakening cardiovascular systems.  The three leading causes of death were heart disease (680,909 deaths), cancer (613,331), and unintentional injury (222,518) (according to CDC), most of those heart dieses where cardiac arrests due to weak pulses, the pulse slows down and eventually stops, a way to temporarily improve it until doctors' treatment is to use a defibrillator, which is a machine that create an electric shock that speeds and bring back a stronger pulse. During sleep a person cannot be aware of whether a cardiac arrest would occur or not and would have a slow to no reaction.  . |
| Context - Why? (30s) Define the context of the problem and the target group.  Provide data to show the scope of the issue.  Why is the world a better place if we tackle the issue? | Our target group includes high-risk cardiac patients who need continuous monitoring, particularly during sleep, to prevent cardiac arrests. This includes elderly individuals, those recovering from cardiac surgeries, and patients with chronic heart conditions.  If we tackle this issue, the world becomes a safer place for these patients by providing a device that continuously monitors their heart rate, detects anomalies, and intervenes instantly to stabilize their pulse. This not only reduces mortality rates but also alleviates the burden on caregivers and healthcare systems by enabling proactive and personalized care. By using AI to enhance precision and real-time communication with doctors, this solution empowers both patients and medical professionals to act faster and more effectively in critical moments. |
| Solution - How? (90s) Which features would a sensor-based digital product need to solve the problem?  How could this issue be tackled with other (non-technical) means? | To address this problem, we’re creating a personalized defibrillator that works while the patient is sleeping. It’s not just a regular machine—it’s smart and customized to each person.  This device monitors the patient’s pulse and sleep cycle in real-time. It’s designed to know the difference between normal changes during sleep and dangerous drops in pulse. If the pulse slows down too much or stops, the device automatically sends a gentle electric shock to bring the heart back to a safe rhythm.  Here’s where it gets even better: the device uses AI to learn the patient’s unique health patterns. It takes into account their age, medical history, and even their specific sleep habits to make the treatment as precise as possible. It’s like having a mini doctor by your side while you sleep. |
| Team & Implementation - Who? (120s) Who is conducting the project and what are your roles:   * Hardware & Data Acquisition: Which sensors are we using? Why exactly this sensor? * Data Analytics & AI: How do intend to process the data (descriptive or inductive statistics)? * Backend & Security: How is the data stored and transferred efficiently and securely? * Frontend & User Experience: What are the key features to attract and sustain users? | HW: we will use OpenSignals HRV, PPG-Sensor, motion sensor, Arduino, breadboard, heart rate module, SpO2, pulse sensor, ECG sensor, galvanic skin response (GSR) sensor, and sound sensor  AI: we will measure the AI: We need to measure heart-rate variability; hence we apply detection algorithm to the data and collect episodes during sleep. The AI will learn when and how to create a shock to stabilize the pulse, it will also contact the doctor and hospital informing them about the patient's situation.  BE: Python: For AI model training, backend logic, and integration.  C++/C: For Arduino and embedded device programming.  Node.js: For managing real-time API requests and notifications.  MongoDB: For storing structured patient profiles and medical histories.  Influx DB: For storing high-frequency time-series data like heart-rate measurements.  TensorFlow/PyTorch: To build and deploy AI models for anomaly detection and decision-making.  Scikit-learn: For preprocessing HRV data and early AI model prototyping.  NumPy and Pandas: For data analysis and feature extraction.  Kafka: For managing real-time data streams from the device.  Celery: For task scheduling (e.g., sending alerts to healthcare providers).  Twilio: For SMS and voice alerts to hospitals and doctors.  RESTful API: Built with FastAPI or Flask for secure data communication between the device, app, and medical systems.  OAuth 2.0: For secure API access by doctors and healthcare providers.  SSL/TLS: To encrypt communication between the device, cloud, and frontend. FE: Patient Interface The mobile app helps patients track their health and monitor the device. Key features include real-time heart rate monitoring, shock notifications, and device diagnostics. It’s built using Flutter or React Native for cross-platform compatibility, with Firebase for push notifications and Chart.js for visualizations. The app includes biometric authentication for security.Doctor’s Dashboard The web platform allows doctors to access patient profiles, view emergency alerts, and analyze health trends. It includes AI-generated insights and customization options for device settings. Built with React.js or Angular, it uses Highcharts for data visualizations and ensures seamless communication with the backend via RESTful APIs.Integration Both interfaces connect securely with the backend via HTTPS and WebSockets for real-time updates, ensuring a smooth and secure user experience for patients and doctors. |
| What next? (30s) What do you need to achieve your objectives?  What is the plan of action to get it done? | To develop the prototype, we will start by assembling accessible hardware components like Arduino, PPG sensors, and motion sensors, using alternatives such as vibration motors or LEDs to simulate shocks during testing. AI will be trained using simulated heart-rate variability data to detect anomalies and simulate shock timing without delivering actual shocks. We will consult professors for technical feasibility advice and cardiologists to ensure the idea aligns with medical needs. A survey will be conducted to gather feedback from doctors on the device’s usefulness and practical features. The focus will be on perfecting the monitoring and AI detection systems first, postponing the implementation of the shock-administering function until more resources and expertise are available. This iterative process will provide a functional prototype, feedback from medical experts, and a roadmap for future improvements. |