HeartGuard

•••

Zwischenpräsentation

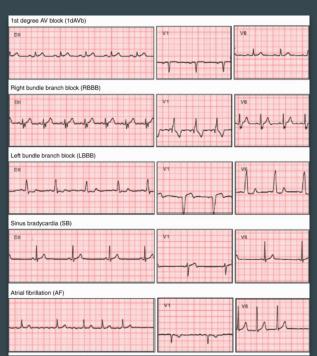
Recap: Ambient Assisted Living

Unser Lösungsvorschlag:

- Device mit EKG-Funktion oder Pulsmonitoring als Hardware
- Software: Erkennung von Herzstillstand (Asystolie)
- Evtl. Erkennung anderer lebensbedrohlicher Herzrhythmen
 - (Kammerflimmern, ventr. Tachykardie, AV-Block Grad III, torsades de pointes, etc.)
- Bedrohlicher Rhythmus erkannt, was nun?
 - Jede Minute zählt!
 - Daher: Automatisierte Verständigung des Rettungsdienstes



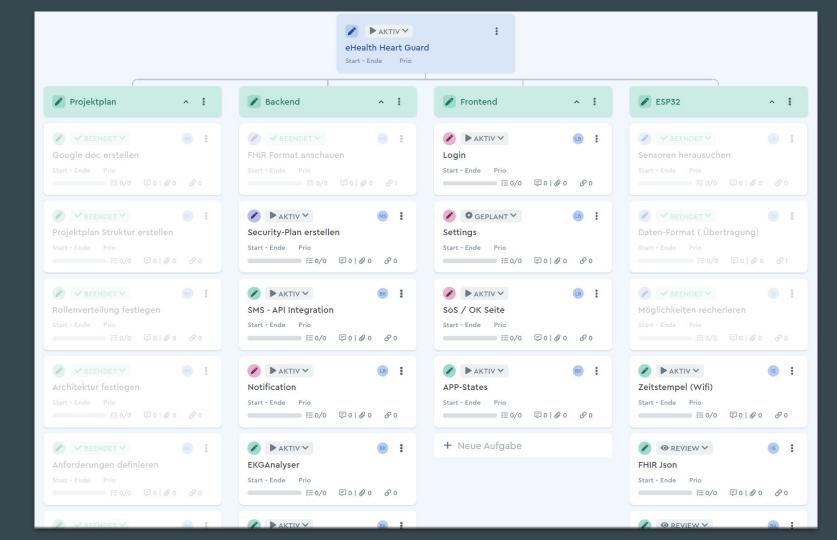




Herangehensweise

- Verteilung der Rollen/Erstellen der Arbeitspakete
- Wöchentliche Meetings über Discord
- Projekt: Agiler Ansatz mit Kanban
- Abstimmung über WhatsApp/Discord/GoogleDocs/GitHub
- Parallele Teamarbeit je nach Rolle an Hardware-Plattform/Backend/Frontend





Aktueller Stand

EKG Erkennung/Verarbeitung:

- ESP32 mit WLAN Modul
- EKG Modul (AD8232)
- Verarbeitung/Umwandlung EKG-Daten im Backend
- Erkennung von Asystolie mit eigenem Algorithmus
- Grundlegende Datenpersistenz im Backend

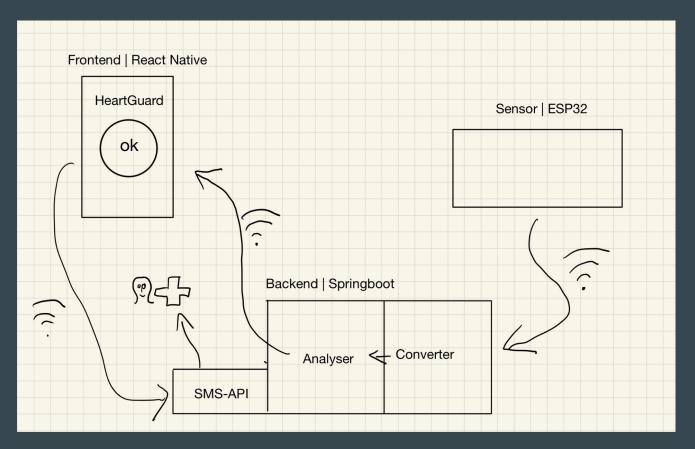


Sonstiges:

- Smartphone APP mit React Native
- UI konzipiert und grundsätzlich implementiert
- Messaging API angebunden

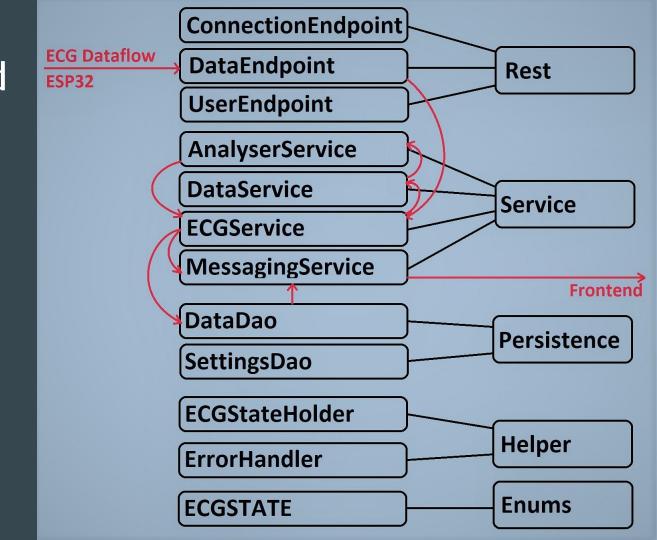


Gesamtarchitektur - Skizze



Architektur Backend

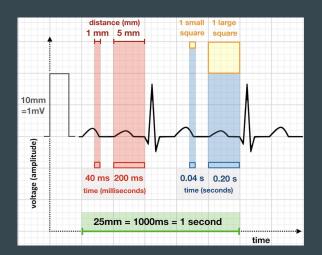
- Spring Boot 3.0.5
- H2 Database Engine
- HAPI FHIR

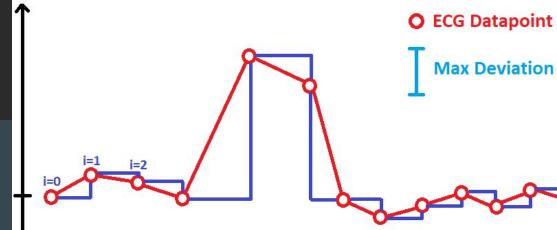


Algorithmus - Erkennung von Asystolie

```
LOGGER.info("Analyzing data:\n" + Arrays.toString(rawData.getData().split(regex: " ")) + "\n");
int largeDeviationCount = 0;
    if(Math.abs(data[i] - data[i+1]) > ecgAnalysisSettings.maxDeviation()) {
        largeDeviationCount++;
    if(largeDeviationCount > ecgAnalysisSettings.maxDeviationNum()) {
        return ECGSTATE.OK;
LOGGER.warn("Result WARNING");
```

return ECGSTATE. WARNING;





Architektur Frontend

Verwendete

Technologien:

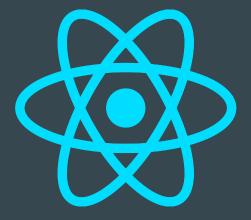
- React Native 0.71.4
- Expo 48.0.9
- Fontawesome 0.3.0

Seiten:

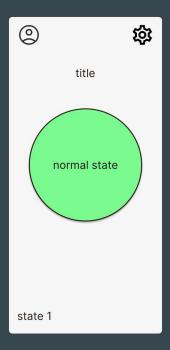
- Home Sc<u>reen</u>
- Login / Register Screen
- Einstellungen







Mockups/States







Innovationsaspekte

- Selbst entwickelter Algorithmus zur Erkennung einer Asystolie im EKG
- Senden einer Notruf-SMS
- Kontinuierliches Monitoring statt nur einmalige EKG-Messung
- Speichern und Abrufen abgeleiteter EKGs aus der Backend-Datenbank
- Eigene Hardware-Plattform zum Ableiten von EKGs auf Basis eines ESP32
- Zukunft:
 - Erkennung anderer Rhythmusstörungen?
 - Security-zentriertes system hardening (RSA-Krypto, Argon2-Hashing, kryptographisch sichere Anmeldung von Geräten ans Backend)

Use Case(s)

- Verarbeitung und Analyse abgeleiteter EKGs -> Erkennung von Asystolie ->
 Verständigung ausgewählter Nummern, z.B.:
 - +43 800 133 133 Hotline für Notrufe per SMS (Rettung, Polizei etc.)
 - Pflegepersonal
 - Verwandte
- Evtl. Früherkennung anderer Rhythmusstörungen/Herzprobleme

Aufgetretene Probleme

- Anfangs mangelnde Abstimmung (gleichzeitige Codeänderungen in verschiedene Richtungen)
- Entwicklung eines eigenen EKG-Erkennungsalgorithmus vs. Nutzung proprietärer Algorithmen (wenige für Java verfügbar, kostenpflichtig)
- Timing (wie oft Daten senden etc.)
- Welche Technologie für Datenermittlung



Danke für eure Aufmerksamkeit!