

INCID-ARCH DIE SMARTE LÖSUNG DER UNFALLBERICHT PROTOKOLLIERUNG

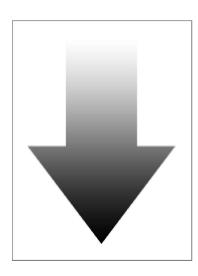
Created by Nick Obreiter, Felix Blank, Alexander Geier and Cristiano Gomes

VISION

Befreien Sie sich von der Last handschriftlich ausgefüllter Unfallberichte und dem Chaos von Papierstapeln mit schwer entzifferbaren Vorfällen. **Digitale Unfallprotokolle** sind nicht nur auf dem Vormarsch – sie sind bereits hier und erleichtern Ihren Alltag erheblich. Erfüllen Sie Ihre **Dokumentationspflicht nach DGUV § 24 Abs.** 6 einfach, sicher und digital. Alle Daten werden nach DSGVO ausschließlich in Europa verarbeitet und gespeichert. Schluss mit dem lästigen Papierkram und herzlich willkommen zu einer effizienteren, modernen und rechtlich konformen Lösung

APP-ANSPRÜCHE

Was soll die App mitbringen?



Ziele	Status
Konto (Organisations) Erstellung	✓
Einloggen	✓
Ausloggen	✓
Erstellen eines neuen Vorfalls	OK
Bearbeiten eines Vorfalls	X

Ziele	Status
User Dashboard für Vorfälle	✓
Generieren von Berichten	X
Datensicherheit	✓
Verwalten von Benutzerrollen und Rechten	X
Benachrichtigungen	X
Benutzerfreundlichkeit	✓

WIE STARTET MAN SO EINE REISE?

PLANUNG VIELLEICHT?

Aber was brauchen wir nun alles? Ein Keks vielleicht?.

Projektmanagement

Front-End



Geduld

MIT NER MENGE FANTASIE



PROJEKTMANAGEMENT



PROJEKTPHASEN

- Initiierung
- Planung
- Durchführung
- Überwachung und Steuerung
- Abschluss

PROJEKTZIELE

- SMART-Prinzip
- Spezifisch
- Messbar
- **E**reichbar
- Realistisch
- **T**erminiert

ROLLEN IM PROJEKT

Projektteam:

• Felix: Backend, Frontend

Nick: Frontend, Backend

Alex: Qualitätssicherung, Projektplanung

Cristiano: Qualitätssicherung, Frontend

WERKZEUGE

- Jira
- Github
- Discord

TASK BOARD

- Anfangs: GitHub
- Später: Jira
- Aufgabenplanung
- Statusverfolgung
- Teamkoordination

KOMMUNIKATION

- Regelmeetings (Discord)
- IRL Gespräche
- Feedback

MEETINGS

- Wöchentliches Teammeeting:
- Wochenends (Samstags, 20:00 ~ 21:00 Uhr)
- Planungsmeeting:
- Jeden Mittwoch, 15:00 17:00 Uhr

ERFOLGSFAKTOREN

- Strukturierte & informierte Planung
- Sprints definieren, diskutieren, abgrenzen.
- Pointierte & Atomare Arbeitspakete
- Transparente Kommunikation
- Flexible Arbeitszeiten, Pensum gerecht.
- Kontinuierliche Wertschätzung der Arbeit.



ARCHITEKTUR

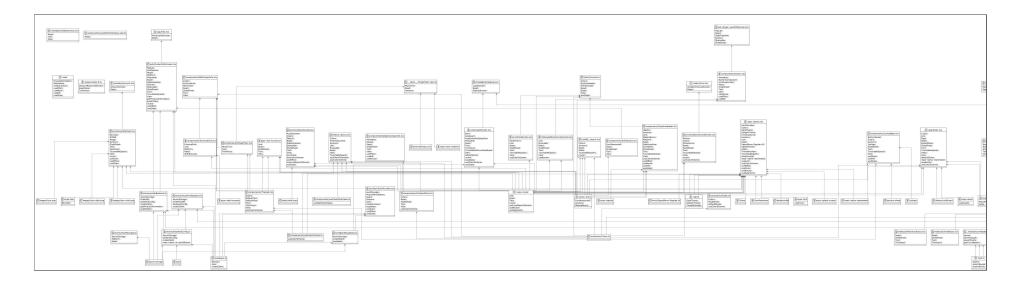
CODE STYLE PATTERNS

Entscheidung	Begründung
ATOMIC	Übersichtliche DX und Best Practices
MVP	Lose Kopplung des Backends
SOLID	Lesbarkeit des Codes

BEST PRACTICE IMPLEMENTIERUNG

Entscheidung	Begründung
Stateless Components & Stores	Wiederverwendung durch Cache und unifizierte Datenbankanbindung
Row Level Secruity Policies	Blacklist Basis

UML DIAGRAMM



TECHNOLOGIE-STACK

Name Art
React Native Framework
EXPO Framework
Type-Script Programmiersprache (JS)
PostgreSQL DBMS
DENO Supabase-Edgefunctions

Name	Art
SonarQube	Statische Codeanalyse
Jest	Unit-Test
Bash	Scriptsprache
EXPO-Go	App-Test Umgebung
Supabase	Backend as a Service

QUALITÄTSSICHERUNG



ZIELE DER QUALITÄTSSICHERUNG

- Erfüllung von Anforderungen
- Einhaltung von Qualitätsstandards
- Gewährleistung der Zuverlässigkeit und Sicherheit

QUALITÄTSSTANDARDS

- Leistung
- Zuverlässigkeit
- Sicherheit
- Benutzerfreundlichkeit

TESTSTRATEGIE

- Unit Tests
- ▶ End-to-End-Tests
- Performance-Tests X
- Security-Tests or

VERWENDUNG VON JEST UND MOCKS

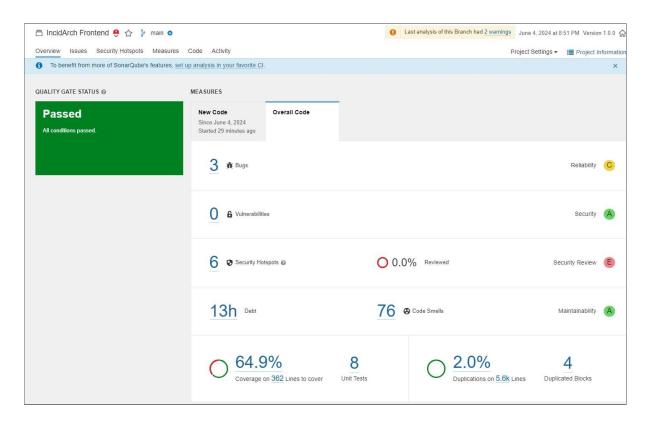
- Jest für JavaScript/Type-Script -Tests
- Umfassende Tests für Funktionen oder Komponenten
- Verwendung von Mocks für externe Abhängigkeiten
- Isolierte Tests ohne tatsächlichen Zugriff auf Abhängigkeiten

KONTINUIERLICHE INTEGRATION

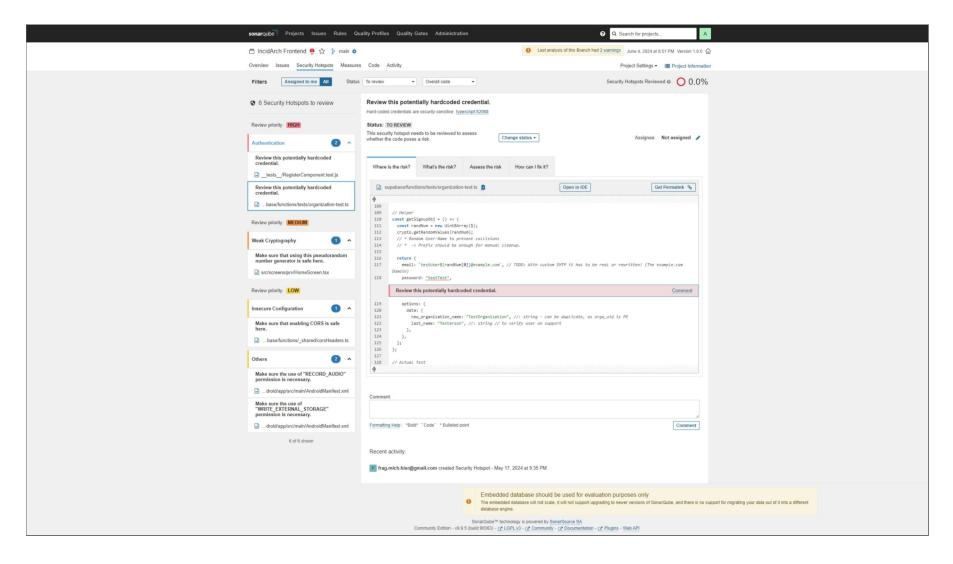
- Automatisierte Tests X
- ▶ Build und Deployment
- Code Reviews

QUALITÄTSKONTROLLE

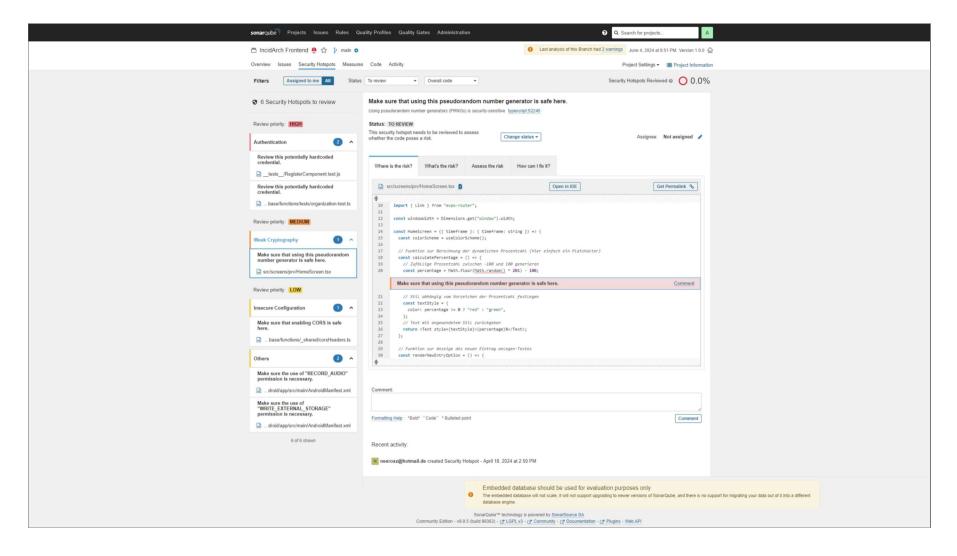
SONARQUBE



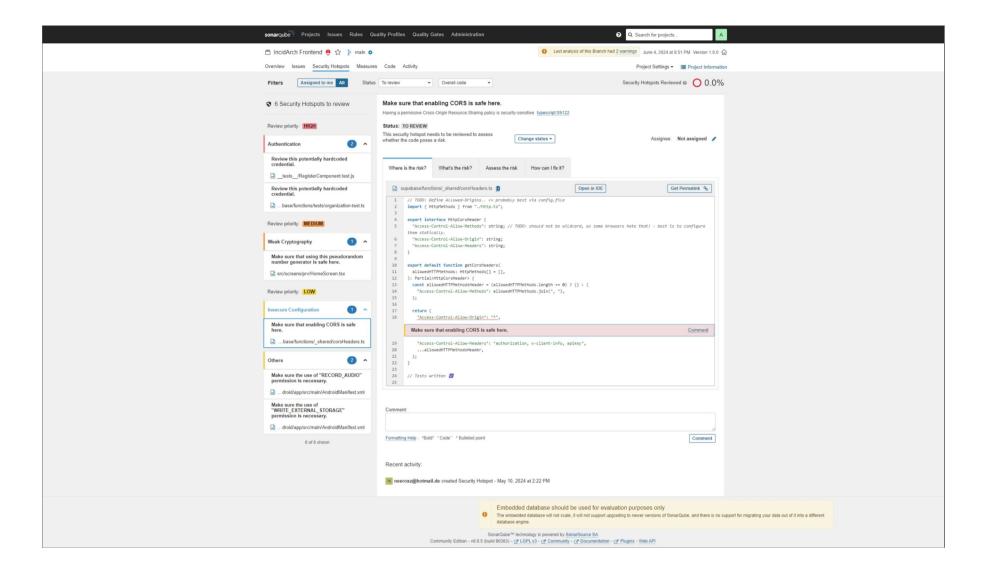
SONARQUBE: HIGH SECURITY **5**



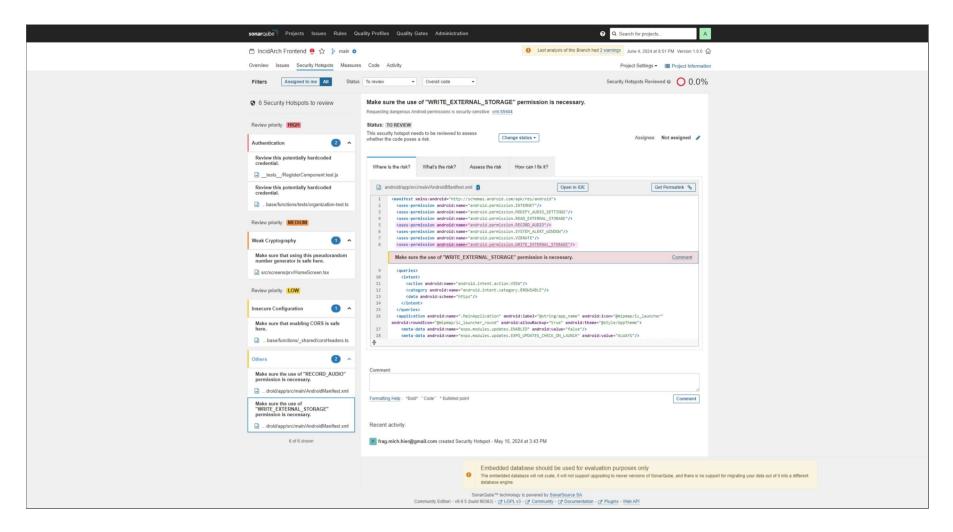
SONARQUBE: MEDIUM SECURITY



SONARQUBE: LOW SECURITY OK



SONARQUBE: 2. LOW SECURITY OK



STATISTIKEN



STUNDENVERTEILUNG

Monthly Report

01/01/2024 - 31/01/2024

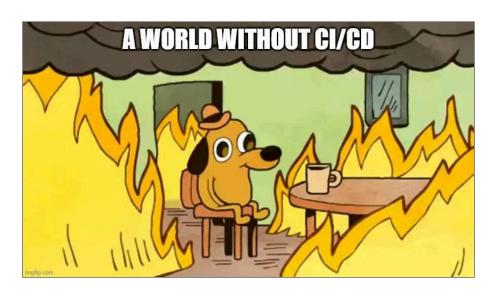
Total: 245:15

eams/Issue	Key	Tot
Freshlix		75
Clean Code Refactoring	IA-62	4
Wochenaufgabe 4 (S4)	IA-59	10
Firebase/Superbase	IA-13	61
Cristiano		40
Aufgabe Woche 6 (S4)	IA-77	2
Refactoring und Styling-Anpassungen der Formanzeige	IA-76	10
Wochenaufgabe 4 (S4)	IA-59	4
AUFGABEN WOCHE 7 (S4) CI/CD	IA-63	14
Pipeline erstellen/automatisieren	IA-49	10
Alexander Geier		50:3
Einträge auf Dashboard	IA-57	4
Dokumentation	IA-46	8:3
Aufgaben Woche 10 (S4)	IA-78	2
Unit Tests	IA-60	18
Dokumentation des Projektes	IA-45	3:3
Neue Aufgaben in Scrumboard erstellen	IA-17	3
Backlog auf neusten Stand bringen	IA-23	9:3
Wechsel zu Jira	IA-20	2
NeeroxZ		79:4
Formular- Klickstrecke	IA-53	8
Hinzufügen dynamischer Dark/white Modus	IA-75	2
Hinzufügen der Stack Ansicht der Account Einstellungen	IA-74	3
Datenschutz Screen.	IA-73	1
Home Button	IA-40	2
Account in Header	IA-56	3:1
Upgrade zu SDK 51	IA-61	12
Log In Seite	IA-25	1
Code Refactor	IA-44	8
Formular manuell eintragbar	IA-12	5:3
HomePage Design	IA-39	8
LoginScreen, SignUp Screen Modal	IA-72	1:5

BUILDS

EAS Build EAS Build #10: Manually run by cristiano605	main	昔 5 days ago Ö 21m 55s	
EAS Build EAS Build #9: Manually run by cristiano605		🗎 3 weeks ago 💍 18m 29s	
EAS Build EAS Build #8: Manually run by cristiano605		🗎 3 weeks ago 💍 22m 14s	
EAS Build EAS Build #7: Manually run by cristiano605		🗎 3 weeks ago ♂ 5m 41s	•••
EAS Build EAS Build #6: Manually run by cristiano605		🗎 3 weeks ago 💆 6m 10s	
EAS Build EAS Build #5: Manually run by cristiano605		🗎 3 weeks ago 💍 43s	
EAS Build EAS Build #4: Manually run by cristiano605	main	苗 3 weeks ago 💍 3m 3s	

CI/CD



VORTEILE VON CI/CD

- ◆ Schnellere Iterationen
- * Frühe Fehlererkennung
- Kontinuierliche Auslieferung

GITHUB ACTIONS

- Automatisierung
- Tests und Builds
- Integriert in GitHub



EXPO APPLICATION SERVICES (EAS)

- Cloud-basierte Builds
- ✓ Veröffentlichung
- Plattformübergreifend



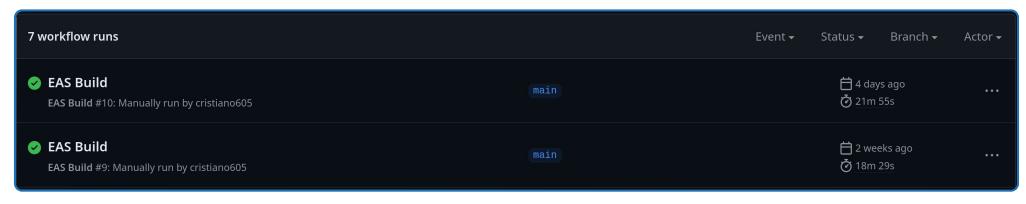
WORKFLOW ERSTELLUNG

```
name: EAS Build
 workflow_dispatch:
     version:
       description: 'Version number'
       required: true
jobs:
   name: Install and build
   permissions: write-all
   runs-on: ubuntu-latest
   steps:
     uses: actions/checkout@v3
    - uses: actions/setup-node@v3
         node-version: 20.x
         cache: npm
     - name: Setup Expo and EAS
       uses: expo/expo-github-action@v8
         eas-version: latest
         token: ${{ secrets.EXPO_TOKEN }}
     - name: Install dependencies
       run: npm i
     - name: Set up Java
       uses: actions/setup-java@v2
         java-version: 17
         distribution: "temurin"
```

```
- name: Build on EAS
 run: eas build --platform android --local --profile preview --non-interactive --no-wait --output=./Incidarch.apk
- name: Create GitHub Release
 id: create release
 uses: actions/create-release@v1
   GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
   tag_name: ${{ github.event.inputs.version }}
   release_name: 'Release ${{ github.event.inputs.version }}'
   draft: false
   prerelease: false
- name: Upload APK to Release
 uses: actions/upload-release-asset@v1
   GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
 with:
   upload_url: ${{ steps.create_release.outputs.upload_url }}
   asset_path: ./Incidarch.apk
   asset_name: Incidarch-${{ github.event.inputs.version }}.apk
   asset_content_type: application/vnd.android.package-archive
```

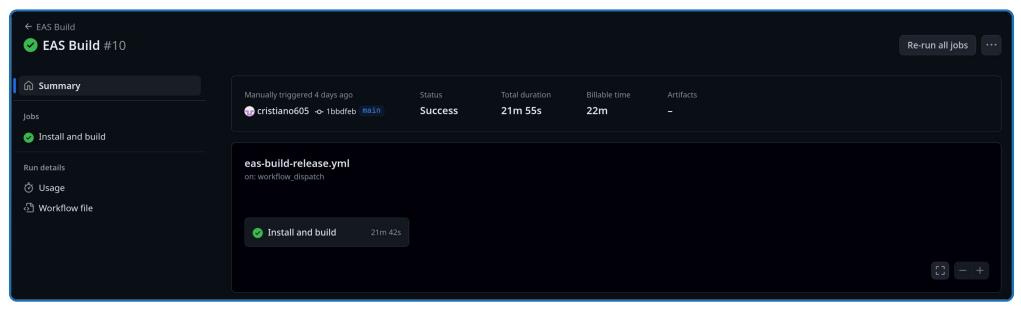
WORKFLOW-ÜBERSICHT

Verlauf und Status aller Builds



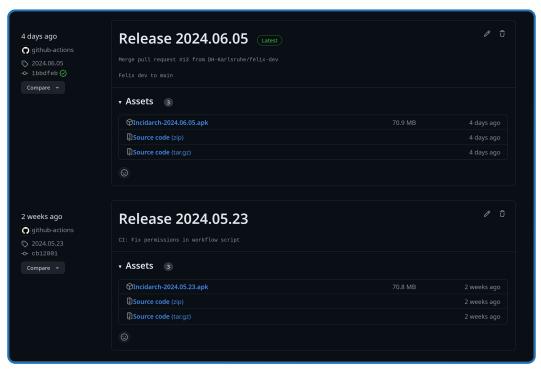
RUN-ZUSAMMENFASSUNG

Details zu einem Run



RELEASES

Gebaute Artefakte und Versionen



LIVE DEMO

WAS WIR GELERNT HABEN?



Arbeitspakete sind bestmöglich auf atomare Teile/ Sub-Issues herunterzubrechen.	

Planung & Organisation, bestmöglich durch eine oder wenige Personen im Kern. Verpflichtende jour-fixe.

Dokumentierte Kommunikation
(Protokollführung).