

Pflichtenheft: Fitnessstudio App

Dieses Pflichtenheft beschreibt die technische Umsetzung der Anforderungen für eine mobile Fitnessstudio-App für Mitglieder.

1. Einleitung

Die Fitnessstudio-App soll Mitgliedern ermöglichen, sich sicher anzumelden und ihre persönlichen Trainingspläne einzusehen. Die Entwicklung konzentriert sich ausschließlich auf die Android-Plattform.

2. Technische Umsetzung

2.1. Plattform und Entwicklungsumgebung

- **Zielplattform:** Android (Native App)
- **Programmiersprache:** Kotlin
- **Entwicklungsumgebung:** Android Studio
- **Build-System:** Gradle

2.2. Authentifizierung und Login

- **Technologie:** Firebase Authentication
- **Implementierung:**
 - Mitglieder registrieren sich mit E-Mail-Adresse und Passwort.
 - Login-Maske mit Feldern für E-Mail und Passwort.
 - Passwort-Reset-Funktion über Firebase.
 - Sichere Speicherung der Nutzer-Tokens auf dem Gerät (z.B. Android Keystore).
 - Alle API-Anfragen an das Backend erfordern einen gültigen Authentifizierungstoken.

2.3. Datenhaltung (Trainingspläne)

- **Technologie:** Google Firebase Firestore (NoSQL-Cloud-Datenbank)
- **Datenmodell:**
 - **Sammlung users:** Speichert grundlegende Nutzerinformationen (z.B. `uid`, `email`, `name`).
 - **Sammlung training_plans:** Speichert die Trainingspläne. Jeder Plan ist einem `user_id` zugeordnet.
 - Ein Trainingsplan-Dokument könnte Felder enthalten wie: `plan_id`, `user_id`, `name`, `description`, `start_date`, `end_date`,

`exercises` (als Array von Objekten, die Übungsdetails wie `name`, `sets`, `reps`, `weight` enthalten).

- **Datenzugriff:**

- Die App greift über das Firebase SDK direkt auf Firestore zu.
- Sicherheitsregeln in Firestore stellen sicher, dass Nutzer nur auf ihre eigenen Trainingspläne zugreifen können.

2.4. Benutzeroberfläche (UI/UX)

- **Design-Prinzipien:** Material Design für Android, um eine konsistente und intuitive Nutzererfahrung zu gewährleisten.
- **Komponenten:**
 - Login-Bildschirm
 - Startbildschirm nach dem Login (z.B. mit Begrüßung und Navigation)
 - Trainingsplan-Ansicht: Liste der Trainingspläne, Detailansicht eines Plans mit allen Übungen.
 - Ladeindikatoren bei Datenabruf.
 - Fehlermeldungen bei fehlgeschlagenen Operationen (z.B. Login, Datenabruf).

2.5. API-Schnittstellen (falls Backend-Logik erforderlich wird)

- Für die reine Anzeige von Trainingsplänen ist kein separates Backend notwendig, da Firestore direkt genutzt wird.
- Sollten zukünftig komplexere Logiken (z.B. automatische Plan-Generierung, Studio-Ankündigungen) hinzukommen, würde ein Backend (z.B. mit Google Cloud Functions oder einem Node.js/Spring Boot Service) über REST-APIs angebunden.

2.6. Sicherheit

- **Authentifizierung:** Firebase Auth bietet robuste Sicherheitsmechanismen.
- **Datenbank-Sicherheit:** Firestore Security Rules für den Zugriffsschutz auf Dokumentenebene.
- **Datenübertragung:** Alle Firebase-Kommunikation ist standardmäßig TLS/SSL-verschlüsselt.
- **Lokale Daten:** Sensible Daten werden nicht unnötig lokal gespeichert.

2.7. Teststrategie

- **Unit Tests:** Für einzelne Funktionen und Logikkomponenten (Kotlin).
- **Integration Tests:** Für die Interaktion mit Firebase Authentication und Firestore.
- **UI Tests:** Mit Espresso für die Benutzeroberfläche auf Android-Geräten/Emulatoren.

- **Manuelle Tests:** Auf verschiedenen Android-Geräten und -Versionen.

3. Zukünftige Erweiterungen (Optional)

- Offline-Modus für Trainingspläne.
- Fortschrittsverfolgung und Diagramme.
- Push-Benachrichtigungen für Termine oder neue Pläne.
- Integration mit Wearables.