Sprint 6: Multi-Tenancy & Lizenzierung – detaillierte Roadmap

Zeitraum: 15.–19. Sep 2025 (5 Arbeitstage)

Ziel: Die App muss Studios klar voneinander trennen (Multi-Tenancy) und Lizenzen überwachen, um nur autorisierten Gyms den vollen Funktionsumfang zu gewähren. Jede Schreiboperation soll nur erfolgen, wenn eine gültige Lizenz vorhanden ist. Ebenso wird ein internes Dashboard für den Betreiber geschaffen, um Lizenzen und Nutzerzahlen zu überblicken. Mit Abschluss dieses Sprints ist das System bereit für letzte Branding-Optimierungen (Sprint 7).

1. Voraussetzungen und Kontext

- **Vorherige Sprints:** Affiliate-Marktplatz (Sprint 5) ist implementiert; Dashboard (Sprint 4) läuft; Gamification (Sprint 3), Offline-Sync (Sprint 2) und Feedback (Sprint 1) sind stabil.
- **Aktuelle Architektur:** Firestore-Sammlung `gyms/{gymId}` mit Subcollections (devices, users, feedback, surveys, challenges, products, orders). Firestore-Regeln lassen momentan alle „admin"-Nutzer aus demselben Gym schreiben.
- **Notwendige Erweiterungen:** Lizenzdaten, Benutzertypen (Admin, Coach, User), globale *superadmin*-Rolle für den App-Betreiber, Prüfung der Lizenzen im Backend.

2. Aufgabenplan (Tag-für-Tag)

Tag 1 (Mo 15. Sep 2025) – Datenmodell & Lizenz-Definition

Lizenz-Dokumentmodell entwerfen (AM):

- Neue Collection: `gyms/{gymId}/license` (ein einziges Dokument) mit Feldern:
  - `plan`: String (`basic`, `pro`, `enterprise`).
  - `maxUsers`: Anzahl der maximal aktiven Nutzer.
  - `expiresAt`: Timestamp.
  - `status`: String (`active`, `expired`, `suspended`).
  - `createdAt, updatedAt.`
  - Optional: `featuresEnabled`: Array (z. B. [`'affiliate'`, `'3d_heatmap'`]) für feature-basiertes Licensing.

Benutzerrollen definieren (AM):

- Globale Rollen: `superadmin` (App-Betreiber), `studio_admin`, `coach`, `member`.
- Speichern der Rolle im JWT-Custom-Claim (`role`) sowie das aktuelle `gym_id`.
- `superadmin` darf alle Gyms lesen und Lizenzen verwalten; andere Rollen sind auf ihr Gym beschränkt.

Migration & Seed (PM):

- Script schreiben, das für bestehende Gyms ein Standard-Lizenz-Dokument erstellt (z. B. `plan=basic`, `maxUsers=50`, `expiresAt +30 Tage`).
- Cloud Function `onGymCreate` implementieren, die beim Anlegen eines neuen Gyms automatisch ein Lizenz-Dokument erzeugt.

Firestore-Regeln planen (PM):

- Entwurf der Regeln:
  - Schreibzugriffe in `gyms/{gymId}` nur erlaubt, wenn `request.auth.token.gym_id == gymId` und `request.auth.token.role` in `['studio_admin', 'coach']` und Lizenz `status == 'active'` und `expiresAt > now`.
  - Leserechte: `superadmin` kann alle Lizenzen und Gym-Daten lesen; andere Rollen nur die eigenen Daten.
- Zusätzlich: `maxUsers` soll durch Backend-Middleware geprüft werden (siehe Tag 2).

Tag 2 (Di 16. Sep 2025) – Backend-Middleware & Lizenzprüfung

Cloud Function `checkLicense` (AM):

- Als Callable oder Export für interne Nutzung:
  - Eingabe: `gymId`.
  - Lesen des Lizenz-Dokuments; Prüfung, ob `status == 'active'` und `expiresAt` in der Zukunft liegt.
  - Zählen der aktiven Nutzer (users-Subcollection mit Feld `active==true` bzw. Live-Check) und Vergleichen mit `maxUsers`.
  - Rückgabe: `ok: true` oder `false` plus Fehlercode (`expired`, `maxUsersExceeded`, `noLicense`).

Middleware in Firestore-Triggern (AM/PM):

- Trigger *beforeCreate/beforeWrite* existieren nicht direkt; daher:
  - Für kritische Schreiboperationen (z. B. neue Session, neues Device, neue Order) Cloud Functions einführen (`createLogWithLicenseCheck`, `createDeviceWithLicenseCheck`), die zunächst `checkLicense` aufrufen.
  - Anschließend die eigentliche Schreiboperation in Firestore ausführen, wenn `ok == true`.
  - Client-App darf Write-Zugriffe nur über diese Functions ausführen (oder Rules blockieren direkte Schreibzugriffe).
  - Alternative (bei geringerer Komplexität): Vor jedem Write im Client `checkLicense` aufrufen und Feedback anzeigen; Firestore-Regeln blockieren im Fehlerfall (verdoppelt Requests).

Scheduled Function zum Lizenz-Status (PM):

- `updateLicenses` (läuft täglich):
  - Prüft alle Lizenz-Dokumente; wenn `expiresAt < now`, setzt `status = 'expired'`.
  - Sendet E-Mail an Studio-Admins bei bevorstehendem Ablauf (7 Tage vorher) via Mail-Service.
  - Eventuell: Deaktiviert Features im Branding (`featuresEnabled`).

Aktive Nutzer zählen (PM):

- Definition „aktiv“: Nutzer, der sich in den letzten 30 Tagen eingeloggt hat.
- Cloud Function `countActiveUsers(gymId)` berechnet `activeUsers` und speichert sie in `gyms/{gymId}/licens`
- Diese Zahl wird im Admin-Dashboard angezeigt und beim Lizenz-Check verwendet.

Tag 3 (Mi 17. Sep 2025) – Auth-Flow & Rollenverwaltung

Anpassung des Registrierungs-/Login-Prozesses (AM):

- Beim Sign-up eines Nutzers: Studio-Admin wählt die Rolle aus (`member` vs. `coach`); Cloud Function `assignRole` setzt `customClaims` (`role, gym_id`).
- Beim Login: Client ruft `firebaseAuth.currentUser.getIdTokenResult()` und liest `role` und `gym_id` aus.
- `AuthProvider` im Flutter-Client speichert Rolle und verwendet sie, um Menüs/Funktionen ein- oder auszublenden (z. B. Admin-Panel nur für `studio_admin`).

Rollen-Änderung durch Superadmin:

- Separate Admin-Seite für den App-Betreiber (Superadmin), um Rollen anzupassen; Cloud Function `updateUserRole(userId, role, gymId)`.
- Änderungen an `customClaims` führen zum Forced-Refresh des Tokens (Client muss sich ab- und wieder anmelden).

UI-Einschränkungen basierend auf Rolle:

- Implementieren von `RoleGate` Widgets, die UI-Elemente ausblenden, wenn der Nutzer nicht über die erforderliche Rolle verfügt.
- Beispiele:
  - Nur `studio_admin` sieht den Plan-Editor, Challenges-Admin-Seite, Branding-Screen.
  - `member` darf keine Admin-Aktionen ausführen; `coach` kann Trainingspläne bearbeiten, aber keine Geräte anlegen.

Tag 4 (Do 18. Sep 2025) – Operator-Dashboard & Daten-Isolation

Operator-Dashboard für Lizenzen (AM):

- Neue Flutter-Web-/Flutter-App-Seite `LicenseManagementScreen` (nur `superadmin`).
- Anzeige aller Gyms in einer Tabelle: Spalten `GymName, Plan, maxUsers, activeUsers, expiresAt, status, monthlyRevenue`.
- Möglichkeit, Lizenzen zu verlängern oder Plan zu ändern (Formular zur Eingabe von `plan, maxUsers, newExpiresAt`).
- Link zur Umsatzübersicht (Daten aus Sprint 5-Umsatzbericht: `gyms/{gymId}/reports/{month}`), um zu sehen, wie viel Provision generiert wird.

Erweiterung der Branding-Seiten (PM):

- Im Branding-Screen (für Studio-Admins) Hinweismeldung „Ihre Lizenz läuft in X Tagen ab; jetzt verlängern“ mit Link zum Shop/Payment.
- Falls `status == 'expired'`, App in Read-Only versetzen: Kein Anlegen neuer Sessions, Feedback, Bestellungen; nur Lesen erlaubt.

Daten-Isolation sicherstellen:

- Code Review des gesamten Repos: Sicherstellen, dass alle Provider-Queries `gymId` filtern (z. B. `firestore.collection('gyms').doc(gymId)`).
- Falltests: Mitglied von Gym A sollte keinen Zugriff auf Geräte von Gym B erhalten; die UI darf solche Geräte/Pläne gar nicht anzeigen.
- Anpassungen vornehmen, falls irgendwo `collectionGroup`-Abfragen oder ungeschützte Queries verwendet werden.

Tag 5 (Fr 19. Sep 2025) – Tests, QA & Übergabe

Unit- und Integrationstests:

- Tests für `checkLicense` Cloud Function (verschiedene Szenarien: aktiv, abgelaufen, `maxUsers` überschritten).
- Test für Middleware/Proxy Functions, ob sie Write-Operationen blockieren, wenn Lizenz ungültig ist.

Firestore-Regeln-Tests mit Emulator:

- Unlizenzierter Nutzer versucht, Gerät anzulegen → abgelehnt.
- `superadmin` liest Lizenz-Dokumente anderer Gyms → erlaubt.

End-to-End-Tests:

- Kompletten Flow simulieren:
  - Gym A hat gültige Lizenz; Mitglied legt Session an → erfolgreich.
  - Lizenz läuft ab; Versuch, Session zu speichern → Fehlermeldung „Lizenz abgelaufen".
  - `superadmin` verlängert die Lizenz; erneuter Versuch → erfolgreich.
  - Überschreiten von `maxUsers`: Registriere zusätzliche Nutzer bis zum Limit; Test, dass neuer Nutzer nicht angelegt werden kann.

Dokumentation erweitern:

- `docs/licensing.md`: Beschreibung des Lizenz-Modells, Felder des Lizenz-Dokuments, Ablauf der Lizenzprüfung, Rollen & Rechte.
- `docs/auth_roles.md`: Übersicht der Rollen, zugehörige Berechtigungen und UI-Sichtbarkeit.
- Hinweise zur Lizenzverlängerung und zur Abrechnung (Verweis auf Sprint 5-Umsatzdaten).

Sprint-Review & Ausblick:

- Demo des Lizenz-Management-Systems: Lizenz abgelaufen → Funktionen deaktiviert; `superadmin` verlängert → Funktionen wieder aktiv.
- Darstellung des Operator-Dashboards mit aktiven Nutzern und Umsätzen.
- Diskussion: Welche Features sollen lizenziert werden (heatmap, affiliate, etc.) und wie wird der Preis berechnet?
- Sammeln offener Punkte für Sprint 7 (Branding-Erweiterungen & Polishing).

3. Definition of Done

- **Lizenz-Dokument pro Gym:** In Firestore existiert ein Lizenz-Objekt mit `plan`, user-Limit und Ablauf-Informationen; für alle bestehenden Gyms wurde ein Dokument angelegt.
- **Lizenzprüfung aktiv:** Middleware oder Proxy-Functions prüfen bei jeder Schreiboperation die Lizenz; bei ungültigen Lizenzen werden Schreibvorgänge blockiert.
- **Benutzerrollen implementiert:** Custom-Claims halten Rolle (`superadmin, studio_admin, coach, member`) und `gymId`; der Auth-Flow weist diese zu.

- **Operator-Dashboard:** Eine interne Oberfläche für den App-Betreiber zeigt alle Lizenzen, aktive Nutzerzahlen und Umsätze; Plan-Änderungen sind möglich.
- **UI-Änderungen:** Benutzer sehen nur Daten ihres Gyms; abgelaufene Lizenzen versetzen die App in den Read-Only-Modus; Hinweise im Branding-Screen informieren über Lizenzstatus.
- **Angepasste Firestore-Regeln:** Schreibzugriffe nur erlaubt, wenn Lizenz aktiv ist und Rolle stimmt; `superadmin` hat erweiterte Rechte.
- **Tests bestanden:** Unit-Tests, Integrationstests und Firestore-Regeln-Tests decken Szenarien (lizenzgültig, abgelaufen, `maxUsers` überschritten) ab.
- **Dokumentation aktuell:** Lizenzmodell, Rollen & Rechte sind dokumentiert; Hinweise zur Lizenzverlängerung und zu Limit-Überschreitungen vorhanden.
- **Sprint-Review abgeschlossen:** Stakeholder bestätigen, dass Daten-Isolation und Lizenzierung funktionieren und bereit für Polishing (Sprint 7) sind.

4. Ausblick auf Sprint 7

Mit implementierten Lizenzen und der Multi-Tenancy-Trennung kann sich das Team im Sprint 7 (22.–26. Sep 2025) der finalen Politur widmen: Erweiterung des Brandings (Schriftarten, App-Name, Icon-Set), UI-Feinschliff, End-to-End-Tests, Onboarding-Materialien und App-Store-Vorbereitung.