

Detaillierte Roadmap: Implementierungsphase (Phase 4)

Ihr Name

2. Februar 2025

Inhaltsverzeichnis

1	Einleitung	2
2	Setup und Infrastruktur	2
2.1	1. Entwicklungsumgebung einrichten	2
2.2	2. Projektstruktur und Kollaborationstools	2
3	Modulare Entwicklung in Sprints	2
3.1	Sprint 1: Projekt-Setup und Grundgerüst	3
3.2	Sprint 2: Nutzerregistrierung, Authentifizierung und QR-Code-Scan-Logik .	3
3.3	Sprint 3: Entwicklung des Nutzer-Dashboards und Dateneingabe	3
3.4	Sprint 4: Reporting-Dashboard für Studiobetreiber	4
3.5	Sprint 5: Integration von Erweiterungsfeatures	4
4	Code-Reviews, Dokumentation und Testing	4
4.1	Code-Reviews	4
4.2	Dokumentation	4
4.3	Testing	5
5	Zusammenfassung der Implementierungsphase	5

1. Einleitung

Diese Roadmap beschreibt detailliert den Entwicklungsprozess (Phase 4) für das Gym Progress Tracking System. Ziel ist es, Schritt für Schritt alle notwendigen Aufgaben, den Aufbau der Infrastruktur, die modulare Entwicklung in Sprints sowie die Qualitätssicherung zu dokumentieren. Dabei wird jeder Schritt in einfacher Sprache erklärt, um auch Entwicklern ohne bisherige Erfahrung einen strukturierten Einstieg zu ermöglichen.

2. Setup und Infrastruktur

Bevor der eigentliche Code geschrieben wird, müssen die Grundlagen geschaffen werden, die für eine stabile und effiziente Entwicklung notwendig sind.

2.1. 1. Entwicklungsumgebung einrichten

- **Code-Editor/IDE:** Wähle eine Entwicklungsumgebung wie Visual Studio Code, Atom oder IntelliJ IDEA.
- **Versionskontrolle:** Installiere und konfiguriere Git. Erstelle ein Repository (z.B. auf GitHub oder GitLab) zur gemeinsamen Codeverwaltung.
- **Lokale Serverumgebung:** Installiere Node.js, um den lokalen Entwicklungsserver für das Backend zu betreiben.

2.2. 2. Projektstruktur und Kollaborationstools

- **Ordnerstruktur:** Lege eine klare Struktur an, beispielsweise:
 - `/frontend` für die PWA,
 - `/backend` für die API,
 - `/docs` für Dokumentationen,
 - `/tests` für Testskripte.
- **CI/CD-Pipeline:** Richte eine automatisierte Test- und Deployment-Pipeline ein (z.B. GitHub Actions oder Jenkins), um sicherzustellen, dass bei jeder Änderung automatische Tests ausgeführt werden.
- **Cloud-Hosting:** Wähle einen Cloud-Provider (z.B. AWS, Google Cloud oder Azure) für die spätere Produktionsumgebung.

3. Modulare Entwicklung in Sprints

Die agile Entwicklung teilt den Entwicklungsprozess in kurze, überschaubare Sprints. Jeder Sprint hat spezifische Ziele, die sukzessive umgesetzt werden.

3.1. Sprint 1: Projekt-Setup und Grundgerüst

Ziel: Einrichtung der grundlegenden Infrastruktur und Erstellung des Basisgerüsts.

1. Initialisiere ein Git-Repository und lege die Ordnerstruktur an.
2. Richte das Backend mit Node.js und Express ein; erstelle ein Basisprojekt.
3. Setze das Frontend mit React als Progressive Web App (PWA) auf.
4. Verifiziere die Kommunikation zwischen Frontend und Backend in der lokalen Umgebung.

3.2. Sprint 2: Nutzerregistrierung, Authentifizierung und QR-Code-Scan-Logik

Ziel: Entwicklung der Kernfunktionen zur Nutzerverwaltung und QR-Code-Erkennung.

1. Registrierung und Login:

- Implementiere im Backend eine Registrierung und Login-Funktionalität unter Verwendung bewährter Authentifizierungsmethoden (z.B. JWT).
- Erstelle im Frontend entsprechende Formulare zur Eingabe von Nutzerinformationen.

2. QR-Code-Scan:

- Integriere eine Bibliothek zur QR-Code-Erkennung (z.B. `react-qr-reader`).
- Implementiere Logik, um nach dem Scan das korrekte Trainingsgerät zu identifizieren und die zugehörige Trainingshistorie anzuzeigen.

3.3. Sprint 3: Entwicklung des Nutzer-Dashboards und Dateneingabe

Ziel: Aufbau eines Dashboards, in dem Nutzer ihre Trainingshistorie einsehen und neue Trainingseinheiten eingeben können.

1. **Dashboard-Design:** Entwickle eine übersichtliche Benutzeroberfläche im Frontend, die die bisherigen Trainingsdaten anzeigt.
2. **Eingabefunktion:** Erstelle Formulare zur Eingabe von Trainingsdaten (Sätze, Gewicht, Datum).
3. **Backend-Integration:** Verbinde die Formulare mit der API, sodass die eingegebenen Daten in der Datenbank gespeichert werden.
4. **Validierung:** Implementiere Validierungslogik, um sicherzustellen, dass alle eingegebenen Daten korrekt und vollständig sind.

3.4. Sprint 4: Reporting-Dashboard für Studiobetreiber

Ziel: Entwicklung eines Dashboards für Studiobetreiber, das aggregierte Nutzungsdaten und Analysen anzeigt.

1. **Datenaggregation:** Erstelle Backend-Module, die relevante Daten (z.B. Gerätebelastung, Nutzungshäufigkeit) aus der Datenbank zusammenführen.
2. **Visualisierung:** Implementiere im Frontend Diagramme und Tabellen, um die Daten anschaulich darzustellen (z.B. mit Chart.js oder D3.js).
3. **Filterfunktionen:** Baue Filter- und Sortierfunktionen ein, damit Betreiber spezifische Analysen durchführen können.

3.5. Sprint 5: Integration von Erweiterungsfeatures

Ziel: Hinzufügen von Zusatzfunktionen zur Steigerung der Nutzerbindung.

1. **Ranglistenmodul:** Entwickle ein Modul, das die Leistungen der Nutzer vergleicht und Ranglisten erstellt.
2. **Streak-Tracking:** Implementiere eine Funktion, die regelmäßige Trainingseinheiten belohnt und Streaks aufzeichnet.
3. **Gamification:** Füge Elemente wie Abzeichen, Levels und Belohnungssysteme hinzu, um die Motivation zu erhöhen.
4. **Test und Feedback:** Integriere diese Features zunächst in der Pilotphase, um Feedback zu sammeln und ggf. Anpassungen vorzunehmen.

4. Code-Reviews, Dokumentation und Testing

4.1. Code-Reviews

- Regelmäßige Peer-Reviews sind notwendig, um sicherzustellen, dass der Code sauber, wartbar und fehlerfrei ist.
- Nutze GitHub Pull Requests und Review-Meetings, um den Code gemeinsam zu überprüfen.

4.2. Dokumentation

- Kommentiere den Code ausführlich und erstelle ein Entwicklerhandbuch, das den Aufbau und die Funktionsweise der Software erklärt.
- Dokumentiere die API mit Tools wie Swagger/OpenAPI, sodass auch andere Entwickler leicht verstehen, wie sie die API verwenden können.
- Erstelle Benutzerhandbücher für Endnutzer und Administratoren.

4.3. Testing

- **Unit-Tests:** Schreibe Tests für einzelne Funktionen und Module, um sicherzustellen, dass jede Komponente wie erwartet arbeitet.
- **Integrationstests:** Überprüfe, ob alle Komponenten (Frontend, Backend, Datenbank) reibungslos zusammenarbeiten.
- **Usability-Tests:** Führe Tests mit echten Nutzern durch, um die Benutzerfreundlichkeit des Dashboards und der Eingabefunktionen zu validieren.
- **Lasttests:** Simuliere hohe Nutzerzahlen, um sicherzustellen, dass das System auch unter hoher Last performant bleibt.

5. Zusammenfassung der Implementierungsphase

Die Implementierungsphase besteht aus dem Aufbau einer soliden Infrastruktur, der schrittweisen Entwicklung aller Kernfunktionen in agilen Sprints und der kontinuierlichen Qualitätssicherung. Durch diesen strukturierten, iterativen Ansatz können alle Module von Grund auf entwickelt und bei Bedarf angepasst werden. Jedes Modul wird einzeln implementiert, getestet und in die Gesamtarchitektur integriert, sodass das System am Ende robust und erweiterbar ist.

Diese detaillierte Roadmap ermöglicht es, jeden Schritt der Softwareentwicklung nachvollziehbar und systematisch abzuarbeiten – auch wenn du als Entwickler noch am Anfang stehst. So wird sichergestellt, dass das Gym Progress Tracking System nicht nur technisch einwandfrei, sondern auch benutzerfreundlich und skalierbar umgesetzt wird.