**GymTrackerX Softwareanforderungen**

**1. Einleitung**

**1.1 Übersicht**

GymTrackerX ist eine Fitness-Tracking-App, die Nutzern hilft, Workouts zu planen und Fortschritte zu verfolgen. Mit einer Übungsdatenbank, einem Punktesystem basierend auf dem Schwierigkeitsgrad der Übungen und Echtzeit-Synchronisation bietet die App eine umfassende Lösung für strukturiertes Training. Nutzer können vorgefertigte Workouts verwenden, eigene Übungen und Ziele erstellen und ihre Fortschritte visuell darstellen lassen, um langfristige Verbesserungen zu erkennen. GymTrackerX wird so zu einem persönlichen Assistenten für effektives und motivierendes Training.

**1.2 Geltungsbereich**

Das GymTrackerX-System wird mit dem Flutter-Framework entwickelt. Flutter ermöglicht die plattformübergreifende Entwicklung, sodass die App sowohl auf Android und IOS läuft.

Es deckt sowohl die Client-seitige Benutzeroberfläche als auch die serverseitigen Anforderungen ab.

**1.3 Definitionen, Akronyme und Abkürzungen**

|  |  |
| --- | --- |
| UML | Unified Modeling Language |
|  |  |
|  |  |
|  |  |

**1.4 Referenzen**

|  |  |  |
| --- | --- | --- |
| Titel | Datum | Link |
| Jira | 11.10.2024 | <https://gymtrackerx.atlassian.net/jira/software/projects/SCRUM/boards/1/timeline> |
| Blog | 15.10.2024 | <https://github.com/PatricReinsch/GymTrackerX/discussions> |
| GitHub | 11.10.2024 | <https://github.com/PatricReinsch/GymTrackerX> |
| MockUp | 21.10.2024 | <https://github.com/PatricReinsch/GymTrackerX/tree/main/doc/mockup> |
| User-Stories | 11.10.2024 | <https://gymtrackerx.atlassian.net/jira/software/projects/SCRUM/boards/1/timeline>  <https://github.com/PatricReinsch/GymTrackerX/tree/main/doc/project_management> |

**2. Funktionale Anforderungen**

**2.1 Übersicht**

GymTrackerX bietet Nutzern eine benutzerfreundliche Oberfläche, um Workouts zu erstellen, Übungen aus einem umfangreichen Katalog auszuwählen, Fortschritte zu verfolgen und Trainingsdaten auf mehreren Geräten in Echtzeit zu synchronisieren. Diese Anforderungen decken die wesentlichen Funktionen ab, um ein strukturiertes Training und eine motivierende Fortschrittsverfolgung zu ermöglichen.

**2.2 Login & Registrierung**

Beschreibung: Nutzer können sich in der App mit ihrer E-Mail und einem Passwort anmelden oder ein neues Konto erstellen. Die „Passwort vergessen“-Funktion ermöglicht es Nutzern, ihr Passwort zurückzusetzen.

• UI-Mockups: Login-Seite mit Eingabefeldern für E-Mail und Passwort, sowie Links zur Registrierung und Passwortwiederherstellung.

• Voraussetzungen: Der Nutzer darf nicht bereits eingeloggt sein.

• Nachbedingungen: Nach erfolgreicher Anmeldung wird der Nutzer auf die Hauptseite (Dashboard) weitergeleitet. Bei fehlerhaften Anmeldedaten wird eine Fehlermeldung angezeigt.

• Geschätzter Aufwand: Mittel

**2.3 Hauptseite**

Beschreibung: Nutzer können ein Workout direkt in der App starten und dabei Übungen aus einem vorkonfigurierten oder benutzerdefinierten Trainingsplan nutzen oder erstellen.

• User Stories: SCRUM-4

• UI-Mockups: Ein Button (+) auf der Hauptseite ein Workout hinzufügen.

• Voraussetzungen: Trainingsplan wurde erstellt oder wird erst erstellt.

• Nachbedingungen: Das Training wird aktiv gestartet, und der Fortschritt wird in Echtzeit gespeichert.

• Geschätzter Aufwand: Mittel

**2.4 Übungskatalog**

Beschreibung: Nutzer können Übungen aus einem großen Übungskatalog auswählen, um sie einem Trainingsplan zuzuordnen. Die Übungen werden nach verschiedenen Kategorien wie Oberkörper, Unterkörper, etc. sortiert.

• User Stories: SCRUM-3

• UI-Mockups: Eine Such- oder Filterfunktion zur Auswahl von Übungen aus dem Katalog.

• Voraussetzungen: Der Übungskatalog muss verfügbar sein.

• Nachbedingungen: Ausgewählte Übungen werden dem aktuellen Trainingsplan hinzugefügt.

• Geschätzter Aufwand: Mittel

**2.5 Training konfigurieren**

Beschreibung: Während der Erstellung eines Trainingsplans können Nutzer Wiederholungen, Gewicht und den Trainingsfokus (z.B. Oberkörper, Unterkörper) angeben.

• User Stories: SCRUM-5

• UI-Mockups: Eingabefelder für Wiederholungen und Gewicht.

• Voraussetzungen: Ein Trainingsplan muss ausgewählt sein, und der Übungskatalog muss bereitstehen.

• Nachbedingungen: Die Eingaben werden gespeichert und auf das aktuelle Workout angewendet.

• Geschätzter Aufwand: Hoch

**2.6 Timer starten und stoppen**

Beschreibung: Nutzer können während eines Workouts einen Timer starten, der das Training trackt. Nach Beendigung des Timers können Nutzer das Gewicht und die Wiederholungen eintragen.

• User Stories: SCRUM-7

• UI-Mockups: Timer-Anzeige während des Workouts.

• Voraussetzungen: Ein aktives Workout muss laufen.

• Nachbedingungen: Der Timer wird beendet, und der Fortschritt des Workouts wird aufgezeichnet.

• Geschätzter Aufwand: Niedrig

**2.7 Fortschrittsanzeige**

Beschreibung: Der Nutzer kann seinen bisherigen Fortschritt pro Übung grafisch dargestellt einsehen. Dies ermöglicht die Verfolgung des Trainingsfortschritts über einen längeren Zeitraum.

• User Stories: SCRUM-6

• Voraussetzungen: Es müssen abgeschlossene Workouts und aufgezeichnete Daten vorhanden sein.

• Nachbedingungen: Der Fortschritt wird grafisch dargestellt und aktualisiert.

• Geschätzter Aufwand: Mittel

**3. Nichtfunktionale Anforderungen**

Diese Anforderungen stellen sicher, dass das System qualitativ hochwertig, sicher, effizient und leicht erweiterbar ist.

**3.1 Benutzerfreundlichkeit**

• Die Benutzeroberfläche der App muss intuitiv und einfach navigierbar sein. Die wichtigsten Funktionen, wie das Erstellen eines Workouts oder das Starten eines Timers, sollten leicht erreichbar sein.

**3.2 Zuverlässigkeit**

• Alle Daten, insbesondere Trainingsfortschritte und Workouts, müssen zuverlässig in Echtzeit synchronisiert werden, um konsistent auf allen Geräten verfügbar zu sein.

**3.4 Sicherheit**

• Alle persönlichen Daten, insbesondere Login-Informationen und Trainingsdaten, müssen durch sichere Verschlüsselungstechnologien geschützt.

**4. Technische Einschränkungen**