

Übungsblatt 6 – UML Use Case Diagramm

Luca M. Schmidt

1. Use Case Diagramm

Basierend auf den textuellen Anforderungen an die „Schrittzähler-App“.

a. Benutzer (Akteure) die mit dem System interagieren:

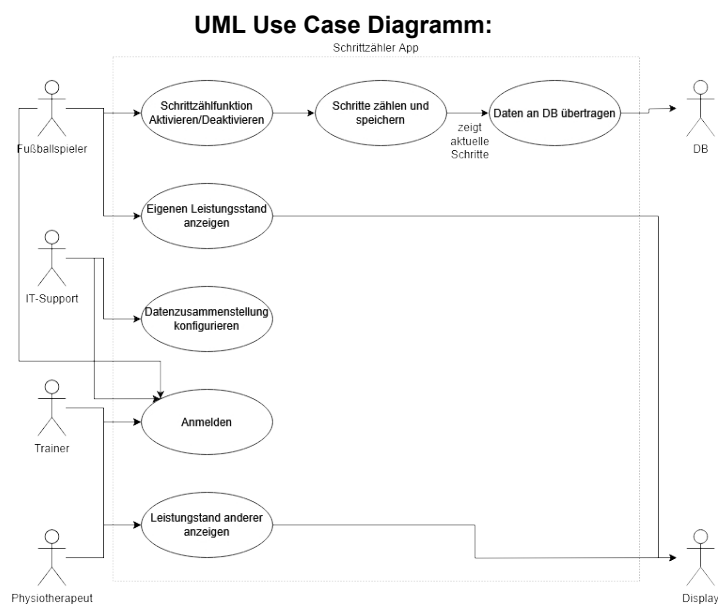
- **Fußballspieler:** Primärer Nutzer der Schrittzählfunktion.
- **Trainer:** Greift auf Spielerdaten zu.
- **Physiotherapeut:** Greift auf Spielerdaten zu.
- **IT-Support:** Konfiguriert Datenzusammenstellungen.
- **Display (Benutzeroberfläche):** Stellt Informationen dar (als externer Akteur, mit dem das System interagiert).
- **Zentrale Datenbank:** Empfängt Daten (als externes System/Akteur).

b. Wichtigste Funktionalitäten (Use Cases):

- Sich anmelden
- Schrittzählfunktion aktivieren/deaktivieren
- Eigenen Leistungsstand anzeigen
- Leistungsstand anderer Spieler anzeigen (für Trainer / Physiotherapeut)
- Datenzusammenstellung konfigurieren (für IT-Support)
- Schritte zählen und speichern (interne App-Funktion, ausgelöst durch Aktivierung)
- Daten an zentrale Datenbank übertragen

c. Benutzer mit unterschiedlichen Rollen als getrennte Akteure:

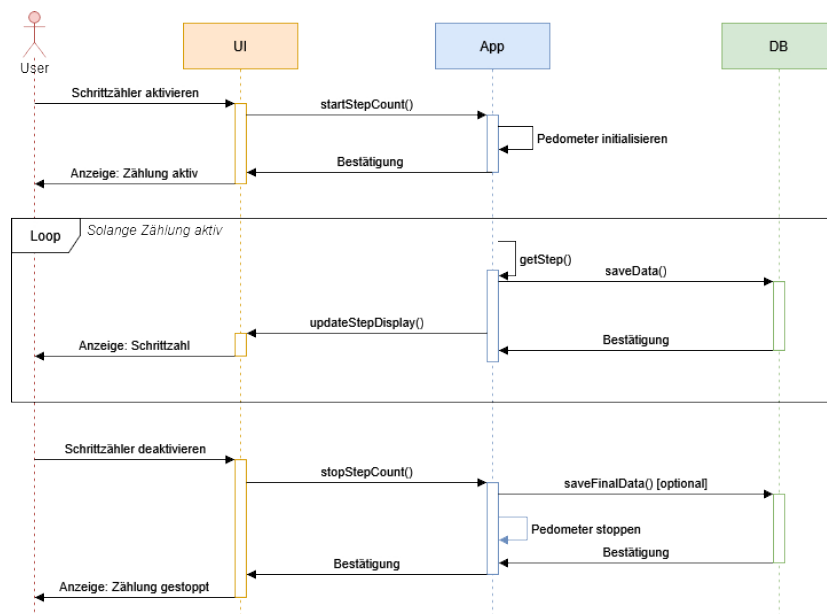
Dies ist durch die oben genannten Akteure (Fußballspieler, Trainer, Physiotherapeut, IT-Support) bereits abgedeckt.



Funktionalität der "Schrittzähler-App" als Ganzes: Die App ermöglicht es Fußballspielern, ihre Schritte zu zählen und ihren Leistungsstand einzusehen. Trainer und Physiotherapeuten können die Daten aller Spieler einsehen. Der IT-Support konfiguriert, wie Daten gesammelt und für die Übertragung an eine zentrale Datenbank vorbereitet werden, wo sie für Analysen bereitstehen. Die Anmeldung ist für alle personalisierten Funktionen notwendig.

2. Verfeinerte Darstellung mit einem Sequenzdiagramm

a. Sequenzdiagramm für Aktivieren/Deaktivieren der Schrittzähler-App:



b. Annahmen und Ideen, die zu dem Sequenzdiagramm führten:

- **Benutzerinteraktion:** Der User initiiert sowohl das Aktivieren als auch das Deaktivieren über das User Interface (Display).
- **Komponententrennung:** Klare Trennung zwischen User Interface, der eigentlichen App-Logik (Schrittzähler-App) und der internen Datenbank.
- **Kontinuierlicher Prozess:** Die Schrittzählung, Speicherung und Anzeige neuer Schritte erfolgt in einer Schleife (loop), nachdem die Funktion aktiviert wurde.
- **Pedometer-Integration:** Das Pedometer wird als integraler Bestandteil der "Schrittzähler-App" betrachtet und nicht als separate Lifeline dargestellt, um die Komplexität gering zu halten. Seine Aktivität (z.B. `erfasseSchritt()`) wird als Selbstaufwurf der App modelliert.

- **Interne vs. Zentrale DB:** Das Sequenzdiagramm fokussiert auf die "interne Datenbank" für die unmittelbare Speicherung während der Zählung. Die Synchronisation mit der "zentralen Datenbank" (aus dem Use Case Diagramm) ist ein separater Prozess, der hier nicht im Detail dargestellt wird, aber z.B. beim Stoppen oder periodisch erfolgen könnte.
- **Rückmeldungen:** Das System gibt dem User über das UI Rückmeldung über den Status (aktiviert, deaktiviert, aktualisierte Schrittzahl).
- **Synchrone Aufrufe:** Die meisten Interaktionen werden als synchrone Nachrichten dargestellt (Pfeil mit ausgefüllter Spitze), d.h. der Sender wartet auf eine Antwort bzw. den Abschluss der Operation.